

PIC18 Peripheral Library Help Document

Table of Contents

PIC18 Peripheral Library	1
Introduction	2
SW Licence Agreement	4
Release Notes	5
Getting Started	9
Devices	11
PIC18F66K80 Family	21
PIC18F66K80	22
PIC18F65K80	23
PIC18F46K80	24
PIC18F45K80	25
PIC18F26K80	26
PIC18F25K80	27
PIC18LF66K80	28
PIC18LF65K80	29
PIC18LF46K80	30
PIC18LF45K80	31
PIC18LF26K80	32
PIC18LF25K80	33
PIC18F47Jx3 Family	33
PIC18F47J53	34
PIC18F46J53	35
PIC18F27J53	36
PIC18F26J53	37
PIC18F47J13	39
PIC18F46J13	40
PIC18F26J13	41
PIC18F27J13	42
PIC18F87K22 Family	44
PIC18F87K22	44

PIC18F86K22	45
PIC18F86K27	46
PIC18F85K22	47
PIC18F67K22	48
PIC18F66K22	49
PIC18F66K27	50
PIC18F65K22	51
PIC18F97J72 Family	52
PIC18F97J72	52
PIC18F96J72	53
PIC18F87J72	55
PIC18F86J72	56
PIC18F87K90 Family	58
PIC18F87K90	58
PIC18F86K95	59
PIC18F86K90	60
PIC18F85K90	61
PIC18F67K90	62
PIC18F66K95	63
PIC18F66K90	64
PIC18F65K90	65
PIC18F87J93 Family	66
PIC18F87J93	66
PIC18F86J93	67
PIC18F67J93	69
PIC18F66J93	70
PIC18F87J90 Family	72
PIC18F87J90	72
PIC18F86J90	73
PIC18F67J90	75
PIC18F66J90	76
PIC18F46J50 Family	78
PIC18F46J50	78
PIC18F45J50	79
PIC18F44J50	80
PIC18F26J50	82
PIC18F25J50	83
PIC18F24J50	84
PIC18LF46J50	85

PIC18LF45J50	87
PIC18LF44J50	88
PIC18LF26J50	89
PIC18LF25J50	90
PIC18LF24J50	92
PIC18F46J11 Family	93
PIC18F46J11	93
PIC18F45J11	94
PIC18F44J11	96
PIC18F26J11	97
PIC18F25J11	98
PIC18F24J11	99
PIC18LF46J11	101
PIC18LF45J11	102
PIC18LF44J11	103
PIC18LF26J11	104
PIC18LF25J11	106
PIC18LF24J11	107
PIC18F1xK50 Family	108
PIC18F14K50	108
PIC18F13K50	109
PIC18LF14K50	110
PIC18LF13K50	111
PIC18F1xK22 Family	111
PIC18F14K22	111
PIC18F13K22	112
PIC18LF14K22	113
PIC18LF13K22	114
PIC18F4553 Family	115
PIC18F4553	115
PIC18F4458	115
PIC18F2553	116
PIC18F2458	116
PIC18F8493 Family	117
PIC18F8493	117
PIC18F8393	117
PIC18F6493	118
PIC18F6393	119
PIC18F8723 Family	119

PIC18F8723	119
PIC18F8628	120
PIC18F6723	120
PIC18F6628	121
PIC18F84J15 Family	121
PIC18F84J15	121
PIC18F64J15	122
PIC18F87J50 Family	122
PIC18F87J50	122
PIC18F86J55	123
PIC18F86J50	124
PIC18F85J50	125
PIC18F67J50	125
PIC18F66J55	126
PIC18F66J50	127
PIC18F65J50	128
PIC18F87J11 Family	128
PIC18F87J11	128
PIC18F86J16	129
PIC18F86J11	130
PIC18F67J11	131
PIC18F66J16	131
PIC18F66J11	132
PIC18F46K20 Family	133
PIC18F46K20	133
PIC18F45K20	134
PIC18F44K20	135
PIC18F43K20	135
PIC18F26K20	136
PIC18F25K20	137
PIC18F24K20	138
PIC18F23K20	138
PIC18F4431 Family	139
PIC18F4431	139
PIC18F4331	140
PIC18F2431	141
PIC18F2331	142
PIC18F97J60 Family	143
PIC18F97J60	143

PIC18F96J65	144
PIC18F96J60	145
PIC18F87J60	146
PIC18F86J65	146
PIC18F86J60	147
PIC18F67J60	148
PIC18F66J65	149
PIC18F66J60	150
PIC18F87J10 Family	150
PIC18F87J10	151
PIC18F86J15	151
PIC18F86J10	152
PIC18F85J15	153
PIC18F85J10	154
PIC18F67J10	154
PIC18F66J15	155
PIC18F66J10	156
PIC18F65J15	157
PIC18F65J10	158
PIC18F85J90 Family	158
PIC18F85J90	158
PIC18F84J90	159
PIC18F83J90	160
PIC18F65J90	161
PIC18F64J90	162
PIC18F63J90	163
PIC18F85J11 Family	163
PIC18F85J11	164
PIC18F84J11	164
PIC18F83J11	165
PIC18F65J11	166
PIC18F64J11	167
PIC18F63J11	168
PIC18F45J10 Family	169
PIC18F45J10	169
PIC18F44J10	170
PIC18F25J10	170
PIC18F24J10	171
PIC18F8621 Family	172

PIC18F8621	172
PIC18F8525	173
PIC18F6621	174
PIC18F6525	175
PIC18F8680 Family	176
PIC18F8680	176
PIC18F8585	177
PIC18F6680	178
PIC18F6585	179
PIC18F8722 Family	180
PIC18F8722	180
PIC18F8627	181
PIC18F8622	182
PIC18F8527	182
PIC18F6722	183
PIC18F6627	184
PIC18F6622	184
PIC18F6527	185
PIC18F8490 Family	186
PIC18F8490	186
PIC18F8390	186
PIC18F6490	187
PIC18F6390	188
PIC18F8410 Family	188
PIC18F8410	188
PIC18F8310	189
PIC18F6410	190
PIC18F6310	190
PIC18F4321 Family	191
PIC18F4321	191
PIC18F4221	192
PIC18F2321	193
PIC18F2221	194
PIC18F4685 Family	194
PIC18F4685	195
PIC18F4682	195
PIC18F2685	196
PIC18F2682	197

PIC18F4680 Family	197
PIC18F4680	198
PIC18F4585	198
PIC18F2680	199
PIC18F2585	200
PIC18F4620 Family	200
PIC18F4620	201
PIC18F4525	201
PIC18F2620	202
PIC18F2525	203
PIC18F4610 Family	204
PIC18F4610	204
PIC18F4515	205
PIC18F4510	206
PIC18F4410	207
PIC18F2610	208
PIC18F2515	208
PIC18F2510	209
PIC18F2410	210
PIC18F4580 Family	211
PIC18F4580	211
PIC18F4480	212
PIC18F2580	213
PIC18F2480	213
PIC18F4550 Family	214
PIC18F4550	214
PIC18F4455	215
PIC18F2550	216
PIC18F2455	217
PIC18F4450 Family	218
PIC18F4450	218
PIC18F2450	219
PIC18F4523 Family	220
PIC18F4523	220
PIC18F4423	221
PIC18F2523	222
PIC18F2423	223
PIC18F4520 Family	224

PIC18F4520	224
PIC18F4420	225
PIC18F2520	226
PIC18F2420	227
PIC18F4320 Family	228
PIC18F4320	228
PIC18F4220	228
PIC18F2320	229
PIC18F2220	230
PIC18F1330 Family	230
PIC18F1330	230
PIC18F1230	231
PIC18F1320 Family	232
PIC18F1320	232
PIC18F1220	232
PIC18F8720 Family	233
PIC18F8720	233
PIC18F8620	234
PIC18F8520	235
PIC18F6720	236
PIC18F6620	237
PIC18F6520	238
PIC18C858 Family	239
PIC18C858	239
PIC18C658	240
PIC18C801 Family	241
PIC18C801	241
PIC18C601	242
PIC18F4539 Family	243
PIC18F4539	244
PIC18F4439	245
PIC18F2539	246
PIC18F2439	247
PIC18F458 Family	248
PIC18F458	248
PIC18F448	248
PIC18F258	249
PIC18F248	250

PIC18F452 Family	251
PIC18F452	251
PIC18F442	252
PIC18F252	253
PIC18F242	254
PIC18C452 Family	254
PIC18C452	255
PIC18C442	256
PIC18C252	257
PIC18C242	258
PIC18F45K22 Family	258
PIC18F45K22	259
PIC18LF45K22	260
PIC18F44K22	261
PIC18LF44K22	262
PIC18F43K22	263
PIC18LF43K22	264
PIC18F46K22	265
PIC18LF46K22	266
PIC18F26K22	267
PIC18LF26K22	268
PIC18F25K22	269
PIC18LF25K22	270
PIC18F24K22	271
PIC18LF24K22	272
PIC18F23K22	273
PIC18LF23K22	274

Device Modules 275

18F66K80 Family	275
ADC (66K80 Set)	275
Analog Comparator (66K80 Set)	276
CTMU (66K80 Set)	277
Input Capture (66K80 Set)	277
Output Compare (66K80 Set)	278
PWM (66K80 Set)	279
I2C (66K80 Family)	279
I2C EEPROM (66K80 Family)	280
EEP (66K80 Set)	281

IO Ports (66K80 Set)	281
MWIRE (66K80 Family)	282
SPI (66K80 Family)	282
SW_RTCC (66K80 Family)	283
Timers (66K80 Set)	283
Flash (66K80 Family)	284
USART (66K80 Set)	285
Deep Sleep (66K80 Family)	286
18F47J53 Family	286
PIC18F2xJ53 Set	287
ADC (2xJ53 Family)	287
Analog Comparator (2xJ53 Set)	287
CTMU (2xJ53 Family)	288
PPS (2xJ53 Family)	289
Deep Sleep (2xJ53 Family)	289
Input Capture (2xJ53 Set)	289
Output Compare (2xJ53 Set)	290
PWM (2xJ53 Set)	291
I2C EEPROM (2xJ53 Family)	292
SPI (2xJ53 Family)	293
IO Ports (2xJ53 Family)	294
RTCC (2xJ53 Family)	294
SW_RTCC (2xJ53 Family)	296
Timers (2xJ53 Set)	296
USART (2xJ53 Family)	297
MWIRE (2xJ53 Family)	298
Flash (2xJ53 Set)	299
PIC18F4xJ53 Set	299
ADC (4xJ53 Family)	300
Analog Comparator (4xJ53 Set)	300
CTMU (4xJ53 Family)	301
PPS (4xJ53 Family)	301
Deep Sleep (4xJ53 Family)	302
Input Capture (4xJ53 Set)	302
Output Compare (4xJ53 Set)	303
PWM (4xJ53 Set)	304
I2C (4xJ53 Family)	305
I2C EEPROM (4xJ53 Family)	306
SPI (4xJ53 Family)	307
IO Ports (4xJ53 Family)	308

RTCC (4xJ53 Family)	308
SW_RTCC (4xJ53 Family)	310
Timers (4xJ53 Set)	310
USART (4xJ53 Family)	311
MWIRE (4xJ53 Family)	312
Flash (4xJ53 Set)	313
PMP (4xJ53 Family)	314
18F97J72 Family	315
ADC (97J72 Family)	315
Analog Comparator (97J72 Family)	316
CTMU (97J72 Family)	316
Input Capture (97J72 Family)	317
Output Compare (97J72 Family)	317
PWM (97J72 Family)	318
I2C (97J72 Family)	318
I2C EEPROM (97J72 Family)	319
IO Ports (97J72 Family)	320
RTCC (97J72 Family)	320
SW_RTCC (97J72 Family)	322
SPI (97J72 Family)	322
Timers (97J72 Family)	323
USART (97J72 Family)	324
MWIRE (97J72 Family)	325
Flash (97J72 Family)	325
18F87K90/22 Family	326
PIC18F6xK90 Set	326
PIC18F6xK90_non_32	326
ADC (6xK90 Set)	326
Analog Comparator (6xK50 Set)	328
CTMU (6xK90 Set)	329
Input Capture (6xK90 Set)	330
Output Compare (6xK90 Set)	331
PWM (6xK90 Set)	331
I2C (6xK90 Set)	332
I2C EEPROM (6xK90 Set)	334
EEP (6xK90 Set)	334
IO Ports (6xK90 Set)	335
MWIRE (6xK90 Set)	335
RTCC (6xK90 Set)	336
SW_RTCC (6xK90 Set)	337

SPI (6xK90 Set)	338
USART (6xK90 Set)	339
Timers (6xK90 Set)	340
Flash (6xK90)	341
PIC18F6xK90_32	342
ADC (65K90 Set)	342
Analog Comparator (65K90 Set)	343
CTMU (65K90 Set)	344
Input Capture (65K90 Set)	345
Output Compare (65K90 Set)	345
PWM (65K90 Set)	346
I2C (65K90 Set)	347
I2C EEPROM (65K90 Set)	348
EEP (65K90 Set)	349
IO Ports (65K90 Set)	349
MWIRE (65K90 Set)	350
RTCC (65K90 Set)	351
SW_RTCC (65K90 Set)	352
SPI (65K90 Set)	352
USART (65K90 Set)	353
Timers (65K90 Set)	354
Flash (65K90 Set)	356
PIC18F8xK90 Set	356
PIC18F8xK90_non_32	356
ADC (8xK90 Set)	356
Analog Comparator (8xK90 Set)	357
CTMU (8xK90 Set)	357
Input Capture (8xK90 Set)	358
Output Compare (8xK90 Set)	359
PWM (8xK90 Set)	360
I2C (8xK90 Set)	361
I2C EEPROM (8xK90 Set)	362
EEP (8xK90 Set)	363
IO Ports (8xK90 Set)	363
MWIRE (8xK90 Set)	364
RTCC (8xK90 Set)	365
SW_RTCC (8xK90 Set)	366
SPI (8xK90 Set)	366
USART (8xK90 Set)	367
Timers (8xK90 Set)	368
Flash (8xK90 Set)	370

PIC18F8xK90_32	370
ADC (85K90 Set)	370
Analog Comparator (85K90 Set)	371
CTMU (85K90 Set)	371
Input Capture (85K90 Set)	372
Output Compare (85K90 Set)	373
PWM (85K90 Set)	374
I2C (85K90 Set)	374
I2C EEPROM (85K90 Set)	376
EEP (85K90 Set)	376
IO Ports (85K90 Set)	377
MWIRE (85K90 Set)	377
RTCC (85K90 Set)	378
SW_RTCC (85K90 Set)	379
SPI (853K90 Set)	380
USART (85K90 Set)	381
Timers (85K90 Set)	382
Flash (85K90 Set)	383
18F87J90 Family	384
ADC (87J90 Family)	384
Analog Comparator (87J90 Family)	384
CTMU (87J90 Family)	385
Input Capture (87J90 Family)	385
Output Compare (87J90 Family)	386
PWM (87J90 Family)	386
I2C (87J90 Family)	387
I2C EEPROM (87J90 Family)	388
IO Ports (87J90 Family)	388
RTCC (87J90 Family)	389
SW_RTCC (87J90 Family)	390
SPI (87J90 Family)	390
Timers (87J90 Family)	391
USART (87J90 Family)	392
MWIRE (87J90 Family)	393
Flash (87J90 Family)	394
18F87J93 Family	394
ADC (87J93 Family)	394
Analog Comparator (87J93 Family)	395
CTMU (87J93 Family)	395
Input Capture (87J93 Family)	396

Output Compare (87J93 Family)	397
PWM (87J93 Family)	397
I2C (87J93 Family)	397
I2C EEPROM (87J93 Family)	398
SPI (87J93 Family)	399
IO Ports (87J93 Family)	400
RTCC (87J93 Family)	400
SW_RTCC (87J93 Family)	402
Timers (87J93 Family)	402
USART (87J93 Family)	403
MWIRE (87J93 Family)	404
Flash (87J93 Family)	404
18F/LF46J50 Family	405
PIC18F/LF26J50 Set	405
ADC (2xJ50 Family)	405
Analog Comparator (2xJ50 Family)	406
CTMU (2xJ50 Family)	406
Deep Sleep (2xJ50 Family)	407
PPS (2xJ50 Family)	408
Input Capture (2xJ50 Family)	408
Output Compare (2xJ50 Family)	408
PWM (2xJ50 Family)	409
I2C (2xJ50 Family)	409
I2C EEPROM (2xJ50 Family)	410
SPI (2xJ50 Family)	410
IO Ports (2xJ50 Family)	412
RTCC (2xJ50 Family)	412
SW_RTCC (2xJ50 Family)	414
Timers (2xJ50 Family)	414
USART (2xJ50 Family)	415
MWIRE (2xJ50 Family)	416
Flash (2xJ50 Set)	416
PIC18F/LF46J50 Set	417
ADC (4xJ50 Family)	417
Analog Comparator (4xJ50 Family)	418
CTMU (4xJ50 Family)	418
Deep Sleep (4xJ50 Family)	419
PPS (4xJ50 Family)	419
Input Capture (4xJ50 Family)	420
Output Compare (4xJ50 Family)	420

PWM (4xJ50 Family)	420
I2C (4xJ50 Family)	421
I2C EEPROM (4xJ50 Family)	422
IO Ports (4xJ50 Family)	423
RTCC (4xJ50 Family)	424
SW_RTCC (4xJ50 Family)	425
SPI (4xJ50 Family)	425
Timers (4xJ50 Family)	426
USART (4xJ50 Family)	427
PMP (4xJ50 Family)	428
MWIRE (4xJ50 Family)	430
Flash (4xJ50 Set)	430
18F/LF46J11 Family	431
PIC18F/LF26J11 Set	431
ADC (2xJ11 Family)	431
Analog Comparator (2xJ11 Family)	432
CTMU (2xJ11 Family)	432
Deep Sleep (2xJ11 Family)	433
PPS (2xJ11 Family)	434
Input Capture (2xJ11 Family)	434
Output Compare (2xJ11 Family)	434
PWM (2xJ11 Family)	435
I2C (2xJ11 Family)	435
I2C EEPROM (2xJ11 Family)	436
IO Ports (2xJ11 Family)	436
RTCC (2xJ11 Family)	437
SW_RTCC (2xJ11 Family)	438
SPI (2xJ11 Family)	439
Timers (2xJ11 Family)	440
USART (2xJ11 Family)	441
MWIRE (2xJ11 Family)	442
Flash (2xJ11 Set)	442
PIC18F/LF46J11 Set	443
ADC (4xJ11 Family)	443
Analog Comparator (4xJ11 Family)	444
CTMU (4xJ11 Family)	444
Deep Sleep (4xJ11 Family)	445
PPS (4xJ11 Family)	445
Input Capture (4xJ11 Family)	446
Output Compare (4xJ11 Family)	446

PWM (4xJ11 Family)	446
I2C (4xJ11 Family)	447
I2C EEPROM (4xJ11 Family)	448
IO Ports (4xJ11 Family)	449
RTCC (4xJ11 Family)	449
SW_RTCC (4xJ11 Family)	451
SPI (4xJ11 Family)	451
Timers (4xJ11 Family)	452
USART (4xJ11 Family)	453
PMP (4xJ11 Family)	454
MWIRE (4xJ11 Family)	455
Flash (4xJ11 Set)	456
18F/LF14K50 Family	457
ADC (14K50 Family)	457
Analog Comparator (14K50 Family)	458
Input Capture (14K50 Family)	458
Output Compare (14K50 Family)	458
PWM (14K50 Family)	459
I2C (14K50 Family)	459
I2C EEPROM (14K50 Family)	460
EEP (14K50 Family)	461
IO Ports (14K50 Family)	461
SW_RTCC (14K50 Family)	462
SPI (14K50 Family)	462
Timers (14K50 Family)	463
USART (14K50 Family)	464
MWIRE (14K50 Family)	464
Flash (14K50 Family)	465
18F/LF14K22 Family	465
ADC (14K22 Family)	466
Analog Comparator (14K22 Family)	466
Input Capture (14K22 Family)	467
Output Compare (14K22 Family)	467
PWM (14K22 Family)	467
I2C (14K22 Family)	468
I2C EEPROM (14K22 Family)	469
EEP (14K22 Family)	469
IO Ports (14K22 Family)	469
SW_RTCC (14K22 Family)	470
SPI (14K22 Family)	470

Timers (14K22 Family)	471
USART (14K22 Family)	472
MWIRE (14K22 Family)	473
Flash (14K22 Family)	473
18F4553 Family	474
ADC (4553 Family)	474
Analog Comparator (4553 Family)	475
Input Capture (4553 Family)	475
Output Compare (4553 Family)	475
PWM (4553 Family)	476
I2C (4553 Family)	476
I2C EEPROM (4553 Family)	477
EEP (4553 Family)	477
IO Ports (4553 Family)	478
SW_RTCC (4553 Family)	478
SPI (4553 Family)	479
Timers (4553 Family)	480
USART (4553 Family)	480
MWIRE (4553 Family)	481
Flash (4553 Family)	482
18F8493 Family	482
ADC (8493 Family)	482
Analog Comparator (8493 Family)	483
Input Capture (8493 Family)	483
Output Compare (8493 Family)	484
PWM (8493 Family)	484
I2C (8493 Family)	485
I2C EEPROM (8493 Family)	485
IO Ports (8493 Family)	486
SW_RTCC (8493 Family)	487
SPI (8493 Family)	487
Timers (8493 Family)	488
USART (8493 Family)	489
MWIRE (8493 Family)	490
Flash (8493 Family)	490
18F8723 Family	491
ADC (8723 Family)	491
Analog Comparator (8723 Family)	491
Input Capture (8723 Family)	492
Output Compare (8723 Family)	492

PWM (8723 Family)	493
I2C (8723 Family)	493
I2C EEPROM (8723 Family)	495
EEP (8723 Family)	495
IO Ports (8723 Family)	496
SW_RTCC (8723 Family)	496
SPI (8723 Family)	497
Timers (8723 Family)	498
USART (8723 Family)	499
MWIRE (8723 Family)	500
Flash (8723 Family)	501
18F84J15 Family	501
ADC (84J15 Family)	501
Analog Comparator (84J15 Family)	502
Input Capture (84J15 Family)	502
Output Compare (84J15 Family)	503
PWM (84J15 Family)	503
I2C (84J15 Family)	504
I2C EEPROM (84J15 Family)	505
IO Ports (84J15 Family)	506
SW_RTCC (84J15 Family)	507
SPI (84J15 Family)	507
Timers (84J15 Family)	508
USART (84J15 Family)	509
MWIRE (84J15 Family)	510
Flash (84J15 Family)	511
18F87J50 Family	511
ADC (87J50 Family)	511
Analog Comparator (87J50 Family)	512
Input Capture (87J50 Family)	513
Output Compare (87J50 Family)	513
PWM (87J50 Family)	514
I2C (87J50 Family)	515
I2C EEPROM (87J50 Family)	516
IO Ports (87J50 Family)	517
SW_RTCC (87J50 Family)	517
SPI (87J50 Family)	518
Timers (87J50 Family)	519
USART (87J50 Family)	520
MWIRE (87J50 Family)	521

Flash (87J50 Family)	521
18F87J11 Family	522
ADC (87J11 Family)	522
Analog Comparator (87J11 Family)	523
Input Capture (87J11 Family)	523
Output Compare (87J11 Family)	524
PWM (87J11 Family)	524
I2C (87J11 Family)	525
I2C EEPROM (87J11 Family)	526
IO Ports (87J11 Family)	527
SW_RTCC (87J11 Family)	528
SPI (87J11 Family)	528
Timers (87J11 Family)	529
USART (87J11 Family)	530
MWIRE (87J11 Family)	531
Flash (87J11 Family)	532
18F46K20 Family	533
ADC (46K20 Family)	533
Analog Comparator (46K20 Family)	533
Input Capture (46K20 Family)	534
Output Compare (46K20 Family)	534
PWM (46K20 Family)	535
I2C (46K20 Family)	535
I2C EEPROM (46K20 Family)	536
EEP (46K20 Family)	536
IO Ports (46K20 Family)	537
SW_RTCC (46K20 Family)	537
SPI (46K20 Family)	538
Timers (46K20 Family)	539
USART (46K20 Family)	539
MWIRE (46K20 Family)	540
Flash (46K20 Family)	541
18F97J60 Family	541
PIC18F67J60 Set	542
ADC (67J60 Family)	542
Analog Comparator (67J60 Family)	542
Input Capture (67J60 Family)	543
Output Compare (67J60 Family)	543
PWM (67J60 Family)	544
I2C (67J60 Family)	544

I2C EEPROM (67J60 Family)	545
IO Ports (67J60 Family)	546
SW_RTCC (67J60 Family)	546
SPI (67J60 Family)	547
Timers (67J60 Family)	548
USART (67J60 Family)	549
MWIRE (67J60 Family)	549
Flash (67J60 Set)	550
PIC18F87J60 Set	551
ADC (87J60 Family)	551
Analog Comparator (87J60 Family)	551
Input Capture (87J60 Family)	552
Output Compare (87J60 Family)	552
PWM (87J60 Family)	553
I2C (87J60 Family)	553
I2C EEPROM (87J60 Family)	554
IO Ports (87J60 Family)	555
SW_RTCC (87J60 Family)	555
SPI (87J60 Family)	556
Timers (87J60 Family)	557
USART (87J60 Family)	558
MWIRE (87J60 Family)	559
Flash (87J60 Set)	559
PIC18F97J60 Set	560
ADC (97J60 Family)	560
Analog Comparator (97J60 Family)	561
Input Capture (97J60 Family)	561
Output Compare (97J60 Family)	562
PWM (97J60 Family)	562
I2C (97J60 Family)	563
I2C EEPROM (97J60 Family)	564
IO Ports (97J60 Family)	565
SW_RTCC (97J60 Family)	565
SPI (97J60 Family)	565
Timers (97J60 Family)	567
USART (97J60 Family)	567
MWIRE (97J60 Family)	568
Flash (97J60 Set)	569
18F87J10 Family	570
ADC (87J10 Family)	570

Analog Comparator (87J10 Family)	571
Input Capture (87J10 Family)	571
Output Compare (87J10 Family)	572
PWM (87J10 Family)	572
I2C (87J10 Family)	573
I2C EEPROM (87J10 Family)	574
IO Ports (87J10 Family)	575
SW_RTCC (87J10 Family)	576
SPI (87J10 Family)	576
Timers (87J10 Family)	577
USART (87J10 Family)	578
MWIRE (87J10 Family)	579
Flash (87J10 Family)	580
18F85J90/11 Family	580
ADC (85J90/11 Family)	581
Analog Comparator (85J90/11 Family)	581
Input Capture (85J90/11 Family)	582
Output Compare (85J90/11 Family)	582
PWM (85J90/11 Family)	582
I2C (85J90/11 Family)	583
I2C EEPROM (85J90/11 Family)	584
IO Ports (85J90/11 Family)	584
SW_RTCC (85J90/11 Family)	585
SPI (85J90/11 Family)	585
Timers (85J90/11 Family)	586
USART (85J90/11 Family)	587
MWIRE (85J90/11 Family)	588
Flash (85J90/11 Family)	588
18F8621 Family	589
ADC (8621 Family)	589
Analog Comparator (8621 Family)	590
Input Capture (8621 Family)	590
Output Compare (8621 Family)	591
PWM (8621 Family)	591
I2C (8621 Family)	592
I2C EEPROM (8621 Family)	593
EEP (8621 Family)	593
IO Ports (8621 Family)	594
SW_RTCC (8621 Family)	594
SPI (8621 Family)	595

Timers (8621 Family)	596
USART (8621 Family)	597
MWIRE (8621 Family)	598
Flash (8621 Family)	598
18F8680 Family	599
ADC (8680 Family)	599
Analog Comparator (8680 Family)	600
Input Capture (8680 Family)	600
Output Compare (8680 Family)	600
PWM (8680 Family)	601
I2C (8680 Family)	601
I2C EEPROM (8680 Family)	602
SPI (8680 Family)	602
EEP (8680 Family)	603
SW_RTCC (8680 Family)	604
IO Ports (8680 Family)	604
Timers (8680 Family)	605
USART (8680 Family)	605
MWIRE (8680 Family)	606
Flash (8680 Family)	607
18F8722 Family	607
ADC (8722 Family)	607
Analog Comparator (8722 Family)	608
Input Capture (8722 Family)	608
Output Compare (8722 Family)	609
PWM (8722 Family)	610
I2C (8722 Family)	610
I2C EEPROM (8722 Family)	612
SPI (8722 Family)	612
EEP (8722 Family)	613
IO Ports (8722 Family)	614
SW_RTCC (8722 Family)	614
Timers (8722 Family)	615
USART (8722 Family)	616
MWIRE (8722 Family)	617
Flash (8722 Family)	617
18F8490 Family	618
ADC (8490 Family)	618
Analog Comparator (8490 Family)	619
Input Capture (8490 Family)	619

Output Compare (8490 Family)	619
PWM (8490 Family)	620
I2C (8490 Family)	620
I2C EEPROM (8490 Family)	621
SPI (8490 Family)	622
SW_RTCC (8490 Family)	622
IO Ports (8490 Family)	623
Timers (8490 Family)	623
USART (8490 Family)	624
MWIRE (8490 Family)	625
Flash (8490 Family)	626
18F8410 Family	626
ADC (8410 Family)	626
Analog Comparator (8410 Family)	627
Input Capture (8410 Family)	627
Output Compare (8410 Family)	628
PWM (8410 Family)	628
I2C (8410 Family)	629
I2C EEPROM (8410 Family)	630
SPI (8410 Family)	630
IO Ports (8410 Family)	631
SW_RTCC (8410 Family)	632
Timers (8410 Family)	632
USART (8410 Family)	633
MWIRE (8410 Family)	634
Flash (8410 Family)	634
18F4321 Family	635
PIC18F4x21 Set	635
ADC (4x21 Family)	635
Analog Comparator (4x21 Family)	636
Input Capture (4x21 Family)	636
Output Compare (4x21 Family)	636
PWM (4x21 Family)	637
I2C (4x21 Family)	637
I2C EEPROM (4x21 Family)	638
SPI (4x21 Family)	638
IO Ports (4x21 Family)	639
SW_RTCC (4x21 Family)	640
EEP (4x21 Family)	640
Timers (4x21 Family)	640

USART (4x21 Family)	641
MWIRE (4x21 Family)	642
Flash (4x21 Family)	642
PIC18F2x21 Set	643
ADC (2x21 Family)	643
Analog Comparator (2x21 Family)	644
Input Capture (2x21 Family)	644
Output Compare (2x21 Family)	644
PWM (2x21 Family)	645
I2C (2x21 Family)	645
I2C EEPROM (2x21 Family)	646
SPI (2x21 Family)	646
IO Ports (2x21 Family)	647
EEP (2x21 Family)	648
SW_RTCC (2x21 Family)	648
Timers (2x21 Family)	648
USART (2x21 Family)	649
MWIRE (2x21 Family)	650
Flash (2x21 Family)	650
18F4x80/5 Family	651
PIC18F4x80/5 Set	652
ADC (4x8x Family)	652
Analog Comparator (4x8x Family)	652
Input Capture (4x8x Family)	653
Output Compare (4x8x Family)	653
PWM (4x8x Family)	654
I2C (4x8x Family)	654
I2C EEPROM (4x8x Family)	655
SPI (4x8x Family)	655
EEP (4x8x Family)	656
SW_RTCC (4x8x Family)	656
IO Ports (4x8x Family)	657
Timers (4x8x Family)	657
USART (4x8x Family)	658
MWIRE (4x8x Family)	659
Flash (4x80/5 Family)	659
PIC18F2x80/5 Set	660
ADC (2x8x Family)	660
Input Capture (2x8x Family)	661
Output Compare (2x8x Family)	661

PWM (2x8x Family)	661
I2C (2x8x Family)	662
I2C EEPROM (2x8x Family)	663
SPI (2x8x Family)	663
EEP (2x8x Family)	664
IO Ports (2x8x Family)	664
SW_RTCC (2x8x Family)	665
Timers (2x8x Family)	665
USART (2x8x Family)	666
MWIRE (2x8x Family)	667
Flash (2x80/5 Family)	667
18F4610 Family	668
PIC18F4610 Set	668
ADC (4610 Family)	668
Analog Comparator (4610 Family)	669
Input Capture (4610 Family)	669
Output Compare (4610 Family)	669
PWM (4610 Family)	670
I2C (4610 Family)	670
I2C EEPROM (4610 Family)	671
SPI (4610 Family)	671
IO Ports (4610 Family)	672
SW_RTCC (4610 Family)	673
Timers (4610 Family)	673
USART (4610 Family)	674
MWIRE (4610 Family)	674
Flash (4610Family)	675
PIC18F2610 Set	675
ADC (2610 Family)	676
Analog Comparator (2610 Family)	676
Input Capture (2610 Family)	676
Output Compare (2610 Family)	677
PWM (2610 Family)	677
I2C (2610 Family)	678
I2C EEPROM (2610 Family)	678
SPI (2610 Family)	679
IO Ports (2610 Family)	680
SW_RTCC (2610 Family)	680
Timers (2610 Family)	681
USART (2610 Family)	681

MWIRE (2610 Family)	682
Flash (2610 Family)	683
18F4620 Family	683
PIC18F4620 Set	683
ADC (4620 Family)	683
Analog Comparator (4620 Family)	684
Input Capture (4620 Family)	684
Output Compare (4620 Family)	685
PWM (4620 Family)	685
I2C (4620 Family)	685
I2C EEPROM (4620 Family)	686
SPI (4620 Family)	687
EEP (4620 Family)	687
IO Ports (4620 Family)	688
SW_RTCC (4620 Family)	688
Timers (4620 Family)	689
USART (4620 Family)	689
MWIRE (4620 Family)	690
Flash (4620Family)	691
PIC18F2620 Set	691
ADC (2620 Family)	691
Analog Comparator (2620 Family)	692
Input Capture (2620 Family)	692
Output Compare (2620 Family)	693
I2C (2620 Family)	693
I2C EEPROM (2620 Family)	694
SPI (2620 Family)	694
PWM (2620 Family)	695
EEP (2620 Family)	695
IO Ports (2620 Family)	696
SW_RTCC (2620 Family)	696
Timers (2620 Family)	697
USART (2620 Family)	698
MWIRE (2620 Family)	698
Flash (2620 Family)	699
18F4550/23/20 Family	699
ADC (45xx Family)	700
Analog Comparator (45xx Family)	700
Input Capture (45xx Family)	701
Output Compare (45xx Family)	701

PWM (45xx Family)	702
I2C (45xx Family)	702
I2C EEPROM (45xx Family)	703
SPI (45xx Family)	703
IO Ports (45xx Family)	704
EEP (45xx Family)	705
SW_RTCC (45xx Family)	705
Timers (45xx Family)	706
USART (45xx Family)	706
MWIRE (45xx Family)	707
Flash (45xx Family)	708
18F4450 Family	708
ADC (4450 Family)	708
Input Capture (4450 Family)	709
Output Compare (4450 Family)	709
PWM (4450 Family)	710
IO Ports (4450 Family)	710
SW_RTCC (4450 Family)	711
Timers (4450 Family)	711
USART (4450 Family)	712
Flash (4450 Family)	712
18F4320 Family	713
ADC (4320 Family)	713
Analog Comparator (4320 Family)	714
Input Capture (4320 Family)	714
Output Compare (4320 Family)	715
PWM (4320 Family)	715
I2C (4320 Family)	715
I2C EEPROM (4320 Family)	716
SPI (4320 Family)	717
IO Ports (4320 Family)	718
EEP (4320 Family)	718
SW_RTCC (4320 Family)	719
Timers (4320 Family)	719
USART (4320 Family)	720
MWIRE (4320 Family)	720
Flash (4320 Family)	721
18F8720 Family	722
ADC (8720 Family)	722
Analog Comparator (8720 Family)	722

Input Capture (8720 Family)	723
Output Compare (8720 Family)	723
PWM (8720 Family)	724
I2C (8720 Family)	725
I2C EEPROM (8720 Family)	725
SPI (8720 Family)	726
IO Ports (8720 Family)	727
EEP (8720 Family)	727
SW_RTCC (8720 Family)	728
Timers (8720 Family)	728
USART (8720 Family)	729
MWIRE (8720 Family)	730
Flash (8720 Family)	730
18C858 Family	731
ADC (858 Family)	731
Analog Comparator (858 Family)	732
Input Capture (858 Family)	732
Output Compare (858 Family)	732
PWM (858 Family)	733
I2C (858 Family)	733
I2C EEPROM (858 Family)	734
SPI (858 Family)	735
IO Ports (858 Family)	735
SW_RTCC (858 Family)	736
Timers (858 Family)	736
USART (858 Family)	737
MWIRE (858 Family)	738
18C801 Family	738
ADC (801 Family)	739
Input Capture (801 Family)	739
Output Compare (801 Family)	740
PWM (801 Family)	740
I2C (801 Family)	740
I2C EEPROM (801 Family)	741
SPI (801 Family)	742
IO Ports (801 Family)	743
SW_RTCC (801 Family)	743
Timers (801 Family)	744
USART (801 Family)	744
MWIRE (801 Family)	745

18F4539 Family	746
ADC (4539 Family)	746
I2C (4539 Family)	746
I2C EEPROM (4539 Family)	747
SPI (4539 Family)	748
IO Ports (4539 Family)	749
EEP (4539 Family)	749
SW_RTCC (4539 Family)	750
Timers (4539 Family)	750
USART (4539 Family)	751
MWIRE (4539 Family)	751
Flash (4539 Family)	752
18F452 Family	753
ADC (F452 Family)	753
Input Capture (F452 Family)	753
Output Compare (F452 Family)	754
PWM (F452 Family)	754
I2C (F452 Family)	755
I2C EEPROM (F452 Family)	755
SPI (F452 Family)	756
IO Ports (F452 Family)	757
EEP (F452 Family)	757
SW_RTCC (F452 Family)	758
Timers (F452 Family)	758
USART (F452 Family)	759
MWIRE (F452 Family)	760
Flash (F452 Family)	760
18C452 Family	761
ADC (C452 Family)	761
Input Capture (C452 Family)	762
Output Compare (C452 Family)	762
PWM (C452 Family)	762
I2C (C452 Family)	763
I2C EEPROM (C452 Family)	764
SPI (C452 Family)	764
IO Ports (C452 Family)	765
SW_RTCC (C452 Family)	766
Timers (C452 Family)	766
USART (C452 Family)	767
MWIRE (C452 Family)	767

18F458 Family	768
PIC18F258 Set	768
ADC (258 Family)	768
Input Capture (258 Family)	769
Output Compare (258 Family)	769
PWM (258 Family)	770
I2C (258 Family)	770
I2C EEPROM (258 Family)	771
SPI (258 Family)	771
IO Ports (258 Family)	772
EEP (258 Family)	773
SW_RTCC (258 Family)	773
Timers (258 Family)	773
USART (258 Family)	774
MWIRE (258 Family)	775
Flash (258 Family)	775
PIC18F458 Set	776
ADC (458 Family)	776
Analog Comparator (258 Family)	777
Input Capture (458 Family)	777
Output Compare (458 Family)	777
PWM (458 Family)	778
I2C (458 Family)	778
I2C EEPROM (458 Family)	779
SPI (458 Family)	779
IO Ports (458 Family)	780
EEP (458 Family)	781
SW_RTCC (458 Family)	781
Timers (458 Family)	781
USART (458 Family)	782
MWIRE (458 Family)	783
Flash (458 Family)	783
18F1320 Family	784
ADC (1320 Family)	784
Input Capture (1320 Family)	785
Output Compare (1320 Family)	785
PWM (1320 Family)	785
IO Ports (1320 Family)	786
EEP (1320 Family)	786
SW_RTCC (1320 Family)	787

Timers (1320 Family)	787
USART (1320 Family)	788
Flash (1320 Family)	789
18F45J10 Family	789
PIC18F45J10 Set	789
ADC (45J10 Family)	789
Analog Comparator (45J10 Family)	790
Input Capture (45J10 Family)	790
Output Compare (45J10 Family)	791
PWM (45J10 Family)	791
I2C (45J10 Family)	791
I2C EEPROM (45J10 Family)	793
SPI (45J10 Family)	793
IO Ports (45J10 Family)	795
SW_RTCC (45J10 Family)	795
Timers (45J10 Family)	795
USART (45J10 Family)	796
MWIRE (45J10 Family)	797
Flash (45J10 Family)	798
PIC18F25J10 Set	798
ADC (25J10 Family)	798
Analog Comparator (25J10 Family)	799
Input Capture (25J10 Family)	799
Output Compare (25J10 Family)	799
PWM (25J10 Family)	800
I2C (25J10 Family)	800
I2C EEPROM (25J10 Family)	801
SPI (25J10 Family)	802
IO Ports (25J10 Family)	803
SW_RTCC (25J10 Family)	803
Timers (25J10 Family)	804
USART (25J10 Family)	804
MWIRE (25J10 Family)	805
Flash (25J10 Family)	806
18F1330 Family	806
ADC (1330 Family)	806
PCPWM (1330 Family)	807
EEP (1330 Family)	808
IO Ports (1330 Family)	808
SW_RTCC (1330 Family)	809

Timers (1330 Family)	809
USART (1330 Family)	810
Flash (1330 Family)	810
18F4431 Family	811
PIC18F4431 Set	811
ADC (4431 Family)	811
Input Capture (4431 Family)	812
Output Compare (4431 Family)	812
PWM (4431 Family)	813
PCPWM (4431 Family)	813
I2C (4431 Family)	814
SPI (4431 Family)	815
IO Ports (4431 Family)	815
EEP (4431 Family)	816
Timers (4431 Family)	816
SW_RTCC (4431 Family)	817
USART (4431 Family)	817
MWIRE (4431 Family)	818
Flash (4431 Family)	819
PIC18F2431 Set	819
ADC (2431 Family)	819
Input Capture (2431 Family)	820
Output Compare (2431 Family)	820
PWM (2431 Family)	821
PCPWM (2431 Family)	821
I2C (2431 Family)	822
SPI (2431 Family)	822
IO Ports (2431 Family)	823
EEP (2431 Family)	824
SW_RTCC (2431 Family)	824
Timers (2431 Family)	824
USART (2431 Family)	825
MWIRE (2431 Family)	826
Flash (2431 Family)	826
18F45K22 Family	827
ADC (45K22 Set)	827
Analog Comparator (45K22 Family)	828
CTMU (45K22 Set)	829
Input Capture (45K22 Set)	830
Output Compare (45K22 Set)	830

PWM (45K22 Set)	831
I2C (45K22 Family)	832
I2C EEPROM (44K22 Family)	833
SPI (45K22 Family)	834
IO Ports (45K22 Family)	835
EEP (45K22 Set)	835
MWIRE (45K22 Family)	836
SW_RTCC (45K22 Family)	837
Timers (45K22 Set)	837
Flash (45K22 Family)	838
USART (45K22 Set)	839

Modules 840

A/D Converter (ADC)	840
ADC Functions	840
Open_ADC	840
OpenADC_Page1	840
OpenADC_Page2	842
OpenADC_Page3	844
OpenADC_Page4	846
OpenADC_Page5	848
OpenADC_Page6	851
OpenADC_Page7	854
OpenADC_Page8	855
OpenADC_Page9	858
OpenADC_Page10	860
OpenADC_Page11	862
OpenADC_Page12	865
OpenADC_Page13	867
OpenADC_Page14	870
OpenADC_Page15	873
OpenADC_Page16	876
OpenADC_Page17	879
SetChan_ADC	882
SetChanADC_Page1	882
SetChanADC_Page2	883
SetChanADC_Page3	884
SetChanADC_Page4	885
SetChanADC_Page5	886
SetChanADC_Page6	887

SetChanADC_Page7	887
SetChanADC_Page8	889
SelChanConv_ADC	889
SelChanConvADC_Page1	889
SelChanConvADC_Page2	890
SelChanConvADC_Page3	891
SelChanConvADC_Page4	892
SelChanConvADC_Page5	893
SelChanConvADC_Page6	894
SelChanConvADC_Page7	895
SelChanConvADC_Page8	896
Convert_ADC	896
Busy_ADC	897
Read_ADC	897
Close_ADC	897
ADC Structs,Records,Enums	898
ADC Macros	898
ADC Examples	898
ADC Example 1	898
Analog Comparator (ANCOMP)	899
Comparator Functions	899
Open_ANCOMP	899
Open_ancomp_Page1	899
Open_ancomp1_Page1	900
Open_ancomp1_Page2	902
Open_ancomp1_Page3	904
Open_ancomp1_Page4	906
Open_ancomp1_Page5	907
Open_amcomp2_Page1	909
Open_ancomp2_Page2	911
Open_ancomp2_Page3	912
Open_ancomp2_Page4	914
Open_ancomp2_Page5	915
Open_ancomp3_Page1	917
Open_ancomp1_Page6	918
Open_ancomp2_Page6	920
Open_ancomp3_Page2	921
Close_ANCOMP	922
Close_ancomp_Page1	923
Close_ancomp1_Page1	923

Close_ancomp2_Page1	923
Close_ancomp3_Page1	923
Comparator Macros	924
Comparator Examples	924
Comparator Example1	924
Charge Time Measurement Unit (CTMU)	925
CTMU Functions	925
Open_CTMU	925
CurrentControl_CTMU	927
Close_CTMU	927
CTMU Macros	928
CTMU Examples	928
CTMU Example1	928
Deep Sleep (DPSLP)	929
Deep Sleep Functions	929
Goto_DeepSleep	930
DeepSleep_WakeUpSource	930
IsResetFrom_DeepSleep	931
Read_DSGPR	931
ULPWakeUp_Enable	931
DeepSleep Structs,Records,Enums	932
_BOOL	932
Deep Sleep Macros	932
Deep Sleep Examples	932
Deep Sleep Example1	932
Peripheral Pin Select (PPS)	934
PPS Macros	935
PPS_Input_Page1/2	935
iPPS_Input_Page1/2	936
iPPS_Input_Page1/2_1	938
PPS_Input_Page1/2_1	939
PPS_Output_Page1	941
PPS_Output_Page2	942
iPPS_Output_Page1	944
iPPS_Output_Page2	945
iPPS_Input_Page3	947
PPS_Input_Page3	948
iPPS_Output_Page3	950
PPS_Output_Page3	951
PPS Examples	953

PPS Example1	953
Input Capture (INCAP)	954
Input Capture Function	954
Open_Capture	954
Open_Capture1	954
Open_Capture2	955
Open_Capture2_Page2	956
Open_Capture3	957
Open_Capture3_Page2	958
Open_Capture4	959
Open_Capture4_Page2	960
Open_Capture4_Page3	962
Open_Capture5	963
Open_Capture5_Page2	964
Open_Capture5_Page3	965
Open_ECapture1	966
Open_ECapture1_Page2	967
Open_ECapture1_Page3	968
Open_ECapture1_Page4	969
Open_ECapture2_Page1	970
Open_ECapture2_Page2	971
Open_ECapture3_Page1	972
Open_Capture6_Page1	973
Open_Capture7_Page1	974
Open_Capture8_Page1	975
Open_Capture8_Page2	976
Open_Capture9_Page1	977
Open_Capture10_Page1	978
Open_ECapture1_Page5	979
Open_ECapture2_Page4	980
Open_ECapture3_Page4	981
Open_Capture5_Page4	982
Read_Capture	983
Read_Capture1	983
Read_Capture2	984
Read_Capture3	984
Read_Capture4	984
Read_Capture5	984
Read_Capture6	985
Read_Capture7	985

Read_Capture8	985
Read_Capture9	986
Read_Capture10	986
Read_ECapture1	986
Read_ECapture2	986
Read_ECapture3	987
Close_Capture	987
Close_Capture1	987
Close_Capture2	987
Close_Capture3	988
Close_Capture4	988
Close_Capture5	988
Close_Capture6	989
Close_Capture7	989
Close_Capture8	989
Close_Capture9	989
Close_Capture10	990
Close_ECapture1	990
Close_ECapture2	990
Close_ECapture3	991
Input Capture Structs, Records, Enums	991
Input Capture Macros	991
Input Capture Examples	991
Input Capture Example1	991
Output Compare (OCMP)	992
Output Compare Functions	992
Open_Compare	992
Open_Compare1	992
Open_Compare2	993
Open_Compare2_Page2	994
Open_Compare3	995
Open_Compare3_Page2	996
Open_Compare4	997
Open_Compare4_Page2	998
Open_Compare4_Page3	999
Open_Compare5	1000
Open_Compare5_Page2	1001
Open_Compare5_Page3	1002
Open_ECompare1	1003
Open_ECompare1_Page2	1004

Open_ECompare1_Page3	1005
Open_ECompare1_Page4	1006
Open_ECompare2_Page1	1007
Open_ECompare2_Page2	1008
Open_ECompare3_Page1	1009
Open_Compare6_Page1	1010
Open_Compare7_Page1	1011
Open_Compare8_Page1	1012
Open_Compare8_Page2	1013
Open_Compare9_Page1	1014
Open_Compare10_Page1	1015
Open_ECompare1_Page5	1016
Open_ECompare2_Page4	1017
Open_ECompare3_Page4	1018
Open_Compare4_Page4	1019
Open_Compare5_Page4	1020
Close_Compare	1021
Close_Compare1	1021
Close_Compare2	1022
Close_Compare3	1022
Close_Compare4	1022
Close_Compare5	1022
Close_Compare6	1023
Close_Compare7	1023
Close_Compare8	1023
Close_Compare9	1024
Close_Compare10	1024
Close_ECompare1	1024
Close_ECompare2	1024
Close_ECompare3	1025
Output Compare Macros	1025
Output Compare Examples	1025
Output Compare Example1	1025
Pulse Width Modulation (PWM)	1026
PWM Functions	1026
Open_PWM	1026
Open_PWM1	1026
Open_PWM2	1027
Open_PWM3	1027
Open_PWM3_Page2	1028

Open_PWM4	1028
Open_PWM4_Page2	1029
Open_PWM4_Page3	1029
Open_PWM2_Page2	1030
Open_PWM5	1030
Open_PWM5_Page2	1031
Open_PWM5_Page3	1032
Open_PWM6_Page1	1032
Open_PWM7_Page1	1033
Open_PWM8_Page1	1033
Open_PWM8_Page2	1034
Open_PWM9_Page1	1035
Open_PWM10_Page1	1035
Open_EPWM1	1036
Open_EPWM1_Page2	1036
Open_EPWM1_Page3	1037
Open_EPWM2_Page1	1038
Open_EPWM2_Page2	1038
Open_EPWM3_Page1	1039
Open_EPWM1_Page4	1040
Open_EPWM2_Page3	1040
Open_EPWM3_Page3	1041
Open_PWM4_Page4	1041
Open_PWM5_Page4	1042
SetOutput_PWM	1043
SetOutput_PWM1	1043
SetOutput_PWM2	1043
SetOutput_PWM3	1044
SetOutput_EPWM1	1045
SetOutput_EPWM2_Page1	1046
SetOutput_EPWM3_Page1	1046
SetDC_PWM	1047
SetDC_PWM1	1047
SetDC_PWM2	1048
SetDC_PWM3	1048
SetDC_PWM4	1049
SetDC_PWM5	1050
SetDC_PWM6_Page1	1050
SetDC_PWM7_Page1	1051
SetDC_PWM8_Page1	1052
SetDC_PWM9_Page1	1052

SetDC_PWM10_Page1	1053
SetDC_EPWM1	1054
SetDC_EPWM2_Page1	1054
SetDC_EPWM3_Page1	1055
Close_PWM	1056
Close_PWM1	1056
Close_PWM2	1056
Close_PWM3	1056
Close_PWM4	1057
Close_PWM5	1057
Close_PWM6_Page1	1057
Close_PWM7_Page1	1057
Close_PWM8_Page1	1058
Close_PWM9_Page1	1058
Close_PWM10_Page1	1058
Close_EPWM1	1059
Close_EPWM3_Page1	1059
Close_EPWM2_Page1	1059
PWM Structs,Records,Enums	1059
PWM Macros	1059
PWM Examples	1060
PWM Example1	1060
EEPROM (EEP)	1060
EEP Functions	1061
Write_Byte_EEP	1061
Read_Byte_EEP	1061
Busy__EEP	1062
EEP Examples	1062
EEP Example1	1062
Inter Integrated Circuit Communication (I2C)	1063
I2C Functions	1063
I2C_Open	1063
Open_I2C	1063
Open_I2C1	1064
Open_I2C2	1065
I2C_Write	1065
Write_I2C	1066
Write_I2C1	1066
Write_I2C2	1067
I2C_Read	1067

Read_I2C	1067
Read_I2C1	1067
Read_I2C2	1068
I2C_puts	1068
puts_I2C	1068
puts_I2C1	1069
puts_I2C2	1069
I2C_gets	1070
gets_I2C	1070
gets_I2C1	1070
gets_I2C2	1071
Close_I2C	1071
Idle_I2C	1072
I2C Macros	1072
I2C Examples	1072
I2C Example1_Master	1072
I2C Example1_Slave	1074
I2C EEPROM (I2C_EEP)	1075
I2C_EEP Functions	1075
EEAckPolling_I2C	1075
I2C_EEAckPolling	1075
I2C_EEAckPolling1	1076
I2C_EEAckPolling2	1076
EECurrentAddRead_I2C	1077
I2C_EECurrentAddRead	1077
I2C_EECurrentAddRead1	1078
I2C_EECurrentAddRead2	1078
EEByteWrite_I2C	1079
I2C_EEByteWrite	1079
I2C_EEByteWrite1	1080
I2C_EEByteWrite2	1080
EEPPageWrite_I2C	1081
I2C_EEPPageWrite	1081
I2C_EEPPageWrite1	1082
I2C_EEPPageWrite2	1083
EESequentialRead_I2C	1083
I2C_EESequentialRead	1083
I2C_EESequentialRead1	1084
I2C_EESequentialRead2	1085
EERandomRead_I2C	1086

I2C_EERandomRead	1086
I2C_EERandomRead1	1086
I2C_EERandomRead2	1087
I2C_EEP Examples	1088
I2C_EEP Example1	1088
IO Ports	1088
Ports Functions	1088
Open_PORTB	1089
Open_RB0INT	1089
Open_RB1INT	1090
Open_RB2INT	1091
Open_RB3INT	1092
Ports Macros	1093
Ports Examples	1093
Ports Example1	1093
Real Time Clock & Calender (RTCC)	1095
RTCC Functions	1095
RTCC_InitClock	1095
RTCC_ReadAlrmDate	1095
RTCC_ReadAlrmTime	1096
RTCC_ReadAlrmTimeDate	1096
RTCC_ReadDate	1096
RTCC_ReadTime	1097
RTCC_ReadTimeDate	1097
RTCC_SetAlarmRpt	1098
RTCC_WrOn	1098
RTCC_WriteTimeDate	1099
RTCC_WriteTime	1099
RTCC_WriteDate	1100
RTCC_WriteAlrmTimeDate	1101
RTCC_WriteAlrmTime	1101
RTCC_WriteAlrmDate	1102
RTCC_SetChimeEnable	1103
RTCC_SetCalibration	1103
RTCC_SetAlarmRptCount	1104
RTCC Structs,Records,Enums	1104
RTCC Macros	1104
RTCC Examples	1104
RTCC Example1	1104
Software RTCC (SW_RTCC)	1106

SW_RTCC Functions	1106
Open_SW_RTCC	1106
update_SW_RTCC	1106
Close_SW_RTCC	1106
SW_RTCC Examples	1107
SW_RTCC Example1	1107
Serial Peripheral Interface (SPI)	1108
SPI Functions	1108
SPI_Open	1108
Open_SPI	1108
Open_SPI1	1109
Open_SPI2	1110
SPI_Write	1111
Write_SPI	1111
Write_SPI1	1112
Write_SPI2	1112
SPI_Read	1113
Read_SPI	1113
Read_SPI1	1113
Read_SPI2	1113
SPI_puts	1113
puts_SPI	1114
puts_SPI1	1114
puts_SPI2	1114
SPI_gets	1115
gets_SPI	1115
gets_SPI1	1115
gets_SPI2	1116
SPI Macros	1116
SPI Examples	1116
SPI Example1_Master	1116
SPI Example1_Slave	1117
TIMERS	1118
TIMERS Functions	1119
Open_Timer	1119
Open_Timer0	1119
Open_Timer1_Page1	1120
Open_Timer1_Page2	1121
Open_Timer2	1123
Open_Timer3_Page1	1124

Open_Timer3_Page2	1125
Open_Timer5	1127
Open_Timer5_Page2	1129
Open_Timer4	1130
Open_Timer6_Page1	1131
Open_Timer7_Page1	1132
Open_Timer8_Page1	1134
Open_Timer10_Page1	1135
Open_Timer12_Page1	1136
Read_Timer	1137
Read_Timer0	1137
Read_Timer1	1138
Read_Timer3	1138
Read_Timer5	1139
Read_Timer7_Page1	1139
Write_Timer	1139
Write_Timer0	1139
Write_Timer1	1140
Write_Timer3	1140
Write_Timer5	1140
Write_Timer7_Page1	1141
Set_TmrCCPSrc	1141
Set_TmrCCPSrc_Page1	1141
Set_TmrCCPSrc_Page2	1142
Set_TmrCCPSrc_Page3	1142
Close_Timer	1143
Close_Timer0	1143
Close_Timer1	1143
Close_Timer2	1144
Close_Timer3	1144
Close_Timer4	1144
Close_Timer5	1144
Close_Timer6_Page1	1145
Close_Timer7_Page1	1145
Close_Timer8_Page1	1145
Close_Timer10_Page1	1146
Close_Timer12_Page1	1146
TIMERS Structs,Records,Enums	1146
TIMERS Macros	1146
TIMERS Examples	1146
TIMERS Example1	1146

Universal Asynchronous/Synchronous Receiver Transmitter (USART)	1148
USART Functions	1148
USART_Open	1148
Open_1USART	1148
Open_2USART	1150
Open_USART	1152
USART_Write	1154
Write_1USART	1154
Write_2USART	1155
Write_USART	1155
USART_baud	1155
baud_1USART	1155
baud_2USART	1157
baud_USART	1158
USART_gets	1159
gets_1USART	1159
gets_2USART	1159
gets_USART	1160
USART_putrs	1160
putrs_1USART	1160
putrs_2USART	1161
putrs_USART	1161
USART_puts	1162
puts_1USART	1162
puts_2USART	1162
puts_USART	1163
USART_Read	1163
Read_1USART	1163
Read_2USART	1163
Read_USART	1164
USART Structs,Records,Enums	1164
USART Macros	1164
USART Examples	1164
USART Example1	1164
MicroWire (MWIRE)	1165
MWIRE Functions	1165
Mwire_Open	1165
Open_Mwire1	1166
Open_Mwire2	1166
Open_Mwire	1167

Mwire_gets	1167
gets_Mwire1	1167
gets_Mwire2	1168
gets_Mwire	1168
Mwire_Write	1169
Write_Mwire1	1169
Write_Mwire2	1169
Write_Mwire	1170
Mwire_Read	1170
Read_Mwire1	1170
Read_Mwire2	1171
Read_Mwire	1171
MWIRE Macros	1171
MWIRE Examples	1172
MWIRE Example1	1172
Power Control PWM (PCPWM)	1173
PCPWM Functions	1174
Open_pcpwm_Page1	1174
Open_pcpwm_Page2	1178
Setdc0_pcpwm	1182
Setdc1_pcpwm	1182
Setdc2_pcpwm	1183
Setdc3_pcpwm	1183
OVD_CTRL_pcpwm	1184
OVD_IO_STA_pcpwm	1184
dt_clk_source_pcpwm	1185
dt_assignment_pcpwm	1185
Close_pcpwm	1186
PCPWM Macros	1186
Parallel Master Port (PMP)	1186
PMP Functions	1187
Open_PMP	1187
SetAddress_PMP	1187
MasterRead_PMP	1188
MasterWrite_PMP	1189
IsBufferNEmpty_PMP	1189
IsBufferNFull_PMP	1189
SlaveReadBufferN_PMP	1190
SlaveReadBuffers_PMP	1190
SlaveWriteBufferN_PMP	1191

SlaveWriteBuffers_PMP	1191
Close_PMP	1192
PMP Structs,Records,Enums	1192
PMP Macros	1192
PMP Examples	1192
PMP Example1	1192
Flash	1196
Flash Functions	1196
Erase_Flash_Page1	1196
Erase_Flash_Page2	1196
Erase_Flash_Page3	1197
Read_Flash_Page1	1198
WriteBlock_Flash_Page1	1198
WriteBlock_Flash_Page2	1199
WriteBlock_Flash_Page3	1199
WriteBlock_Flash_Page4	1200
WriteBlock_Flash_Page5	1200
WriteBlock_Flash_Page6	1201
WriteWord_Flash_Page1	1202
WriteBytes_Flash_Page1	1202
WriteBytes_Flash_Page2	1203
WriteBytes_Flash_Page3	1203
Flash Macros	1204
Flash Examples	1204
Flash Example1	1204
Prototype_Page	1206
Symbol Reference	1207
Files	1207
adc.h	1207
ancomp.h	1207
capture.h	1207
compare.h	1208
ctmu.h	1208
dpslp.h	1208
EEP.h	1208
flash.h	1208
i2c.h	1208
mwire.h	1208

pcpwm.h	1208
pmp.h	1208
portb.h	1208
pps.h	1208
ProMPT.h	1209
pwm.h	1209
reset.h	1209
rtcc.h	1209
spi.h	1209
timers.h	1209
usart.h	1209

Index

a

1 PIC18 Peripheral Library



Peripheral Library Support

for the

PIC18 Microcontroller Family

2 Introduction

PIC18 MCU Peripheral Library

Introduction

The PIC18 MCU Peripheral Library provides a set of functions for setting up and controlling the operation of all the peripheral modules available in the PIC18 devices, as well as functions for interfacing with an external LCD, Software implemented communication modules. The Peripheral Library serves as a convenient layer of abstraction over the specific details of the peripherals and their associated control and status registers. The Peripheral Library supports the following hardware peripheral modules:

1. 10-bit A/D Converter
2. 12-bit A/D Converter
3. Analog Comparator
4. timers
5. Input Capture
6. Output Compare
7. Charge Time Measurement Unit (CTMU)
8. Deep Sleep
9. Internal EEPROM
10. Power Control PWM (PCPWM)
11. PWM
12. Programmable Motor Control Technology (ProMPT™)
13. Real Time Clock & Calendar (RTCC)
14. Software RTCC
15. I/O Ports and External Interrupts
16. Reset
17. Usart
18. SPI
19. I2C™
20. MWIRE
21. CAN2510
22. Parallel Master Port (PMP)
23. Functions for controlling an external LCD through configurable I/O Port pins are also provided
24. Software I2C
25. Software SPI
26. Software UART
27. FLASH

Features

Key features of the PIC18 MCU Peripheral Library include:

- Free library included with MCC18 compiler release
- The Peripheral Library contains a library file for each individual device from the PIC218 MCU family. The library file for each device includes functions corresponding to peripherals present in that particular device.
- The Peripheral Library also contains C include files that enable a user to take advantage of pre-defined constants for passing parameters to various library functions. There is an include file for each peripheral module.
- Since the functions are in the form of pre-compiled libraries, they may be called from a user application program written in either MCC18 or 8-bit PIC18 assembly language.
- The C source code is also included, so users can customize the functions to suit their specific application requirements if needed.
- The pre-defined constants in the C include files eliminates the need to refer to the details and structure of every Special Function Register while initializing peripherals or checking status bits.
- Library is developed considering the aspect of optimization, which upholds the ambit for the users effectively develop the application
- Library design facilitates the easy integration of all peripheral module support in a single application
- Library cogitates the migration possibility among the PIC MCU with least possible or no modifications to the application layer.
- Comprehensive code examples making use of peripheral library support in application layer, to facilitate the use of peripheral library

Resource Requirements

Program Memory: The Peripheral Library functions have been optimized for reduced Program Memory usage. Since the functions are in the form of libraries, the actual Program Memory requirements depend on the functions being called by the application, as well as on the specific PIC18 device being used.

Data Memory: The vast majority of the functions do not use RAM at all. Each of the remaining functions uses less than 10 bytes of RAM.

Peripheral modules: Makes use of any other peripherals that are required to work in conjunction with the peripheral module in use as depicted in module architecture available in device data sheet.

3 SW Licence Agreement

Copyright © 2009 released Microchip Technology Inc. All rights reserved.

Microchip licenses to you the right to use, modify, copy and distribute Software only when embedded on a Microchip microcontroller or digital signal controller that is integrated into your product or third party product (pursuant to the sublicense terms in the accompanying license agreement).

You should refer to the license agreement accompanying this Software for additional information regarding your rights and obligations.

SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.

4 Release Notes

Introduction (see page 2)

This chapter gives an overview of the MCC18 library files and pre compiled object files that can be included in an application.

Overview

A library is a collection of functions grouped for reference and ease of linking. The MCC18 libraries are included in the lib subdirectory of the installation. These can be linked directly into an application using the MPLINK linker.

These files were pre compiled in the ..\MCC18\lib directory at Microchip. The directory ..\MCC18\src\pmc_common\.. contains the library files.

Devices (see page 11) Supported

The following PIC18 devices are supported:

PIC18C242 (see page 258) PIC18C252 (see page 257) PIC18C442 (see page 256) PIC18C452 (see page 255)
PIC18F242 (see page 254) PIC18F252 (see page 253) PIC18F442 (see page 252) PIC18F452 (see page 251)
PIC18F248 (see page 250) PIC18F258 (see page 249) PIC18F448 (see page 248) PIC18F458 (see page 248)
PIC18F2439 (see page 247) PIC18F2539 (see page 246) PIC18F4439 (see page 245) PIC18F4539 (see page 244)
PIC18C601 (see page 242) PIC18C801 (see page 241) PIC18C658 (see page 240) PIC18C858 (see page 239)
PIC18F6620 (see page 237) PIC18F6720 (see page 236) PIC18F8620 (see page 234) PIC18F8720 (see page 233)
PIC18F6520 (see page 238) PIC18F8520 (see page 235)
PIC18F1220 (see page 232) PIC18F1320 (see page 232) PIC18F1230 (see page 231) PIC18F1330 (see page 230)
PIC18F2220 (see page 230) PIC18F2320 (see page 229) PIC18F4220 (see page 228) PIC18F4320 (see page 228)
PIC18F2420 (see page 227) PIC18F2520 (see page 226) PIC18F4420 (see page 225) PIC18F4520 (see page 224)
PIC18F2423 (see page 223) PIC18F2523 (see page 222) PIC18F4423 (see page 221) PIC18F4523 (see page 220)
PIC18F2450 (see page 219) PIC18F4450 (see page 218) PIC18F2455 (see page 217) PIC18F2550 (see page 216)
PIC18F4455 (see page 215) PIC18F4550 (see page 214)
PIC18F2480 (see page 213) PIC18F2580 (see page 213) PIC18F4480 (see page 212) PIC18F4580 (see page 211)
PIC18F2410 (see page 210) PIC18F2510 (see page 209) PIC18F2515 (see page 208) PIC18F2610 (see page 208)
PIC18F4410 (see page 207) PIC18F4510 (see page 206) PIC18F4515 (see page 205) PIC18F4610 (see page 204)
PIC18F2525 (see page 203) PIC18F2620 (see page 202) PIC18F4525 (see page 201) PIC18F4620 (see page 201)
PIC18F2585 (see page 200) PIC18F2680 (see page 199) PIC18F4585 (see page 198) PIC18F4680 (see page 198)
PIC18F2682 (see page 197) PIC18F2685 (see page 196) PIC18F4682 (see page 195) PIC18F4685 (see page 195)
PIC18F2221 (see page 194) PIC18F2321 (see page 193) PIC18F4221 (see page 192) PIC18F4321 (see page 191)
PIC18F6310 (see page 190) PIC18F6410 (see page 190) PIC18F8310 (see page 189) PIC18F8410 (see page 188)
PIC18F6390 (see page 188) PIC18F6490 (see page 187) PIC18F8390 (see page 186) PIC18F8490 (see page 186)

PIC18F6527 (see page 185) PIC18F6622 (see page 184) PIC18F6627 (see page 184) PIC18F6722 (see page 183)
PIC18F8527 (see page 182) PIC18F8622 (see page 182) PIC18F8627 (see page 181) PIC18F8722 (see page 180)
PIC18F6585 (see page 179) PIC18F6680 (see page 178) PIC18F8585 (see page 177) PIC18F8680 (see page 176)
PIC18F6525 (see page 175) PIC18F6621 (see page 174) PIC18F8525 (see page 173) PIC18F8621 (see page 172)
PIC18F24J10 (see page 171) PIC18F25J10 (see page 170) PIC18F44J10 (see page 170) PIC18F45J10 (see page 169)
PIC18F65J10 (see page 158) PIC18F66J10 (see page 156) PIC18F67J10 (see page 154) PIC18F85J10 (see page 154)
PIC18F86J10 (see page 152) PIC18F87J10 (see page 151)
PIC18F63J11 (see page 168) PIC18F64J11 (see page 167) PIC18F65J11 (see page 166) PIC18F83J11 (see page 165)
PIC18F84J11 (see page 164) PIC18F85J11 (see page 164)
PIC18F85J15 (see page 153) PIC18F65J15 (see page 157) PIC18F66J15 (see page 155) PIC18F86J15 (see page 151)
PIC18F63J90 (see page 163) PIC18F64J90 (see page 162) PIC18F65J90 (see page 161) PIC18F83J90 (see page 160)
PIC18F84J90 (see page 159) PIC18F85J90 (see page 158)
PIC18F66J60 (see page 150) PIC18F67J60 (see page 148) PIC18F86J60 (see page 147) PIC18F87J60 (see page 146)
PIC18F96J60 (see page 145)
PIC18F66J65 (see page 149) PIC18F86J65 (see page 146) PIC18F96J65 (see page 144) PIC18F97J60 (see page 143)
PIC18F2331 (see page 142) PIC18F2431 (see page 141) PIC18F4331 (see page 140) PIC18F4431 (see page 139)
PIC18F24K20 (see page 138) PIC18F25K20 (see page 137) PIC18F26K20 (see page 136)
PIC18F43K20 (see page 135) PIC18F44K20 (see page 135) PIC18F45K20 (see page 134) PIC18F46K20 (see page 133)
PIC18F66J11 (see page 132) PIC18F67J11 (see page 131) PIC18F86J11 (see page 130) PIC18F87J11 (see page 128)
PIC18F66J16 (see page 131) PIC18F86J16 (see page 129)
PIC18F65J50 (see page 128) PIC18F66J50 (see page 127) PIC18F67J50 (see page 125)
PIC18F85J50 (see page 125) PIC18F86J50 (see page 124) PIC18F87J50 (see page 122)
PIC18F66J55 (see page 126) PIC18F86J55 (see page 123)
PIC18F64J15 (see page 122) PIC18F84J15 (see page 121)
PIC18F8723 (see page 119) PIC18F6723 (see page 120)
PIC18F8493 (see page 117) PIC18F8393 (see page 117) PIC18F6493 (see page 118) PIC18F6393 (see page 119)
PIC18F8628 (see page 120) PIC18F6628 (see page 121)
PIC18F2458 (see page 116) PIC18F2553 (see page 116) PIC18F4458 (see page 115) PIC18F4553 (see page 115)
PIC18F13K50 (see page 109) PIC18F14K50 (see page 108) PIC18LF13K50 (see page 111) PIC18LF14K50 (see page 110)
PIC18F13K22 (see page 112) PIC18F14K22 (see page 111) PIC18LF13K22 (see page 114) PIC18LF14K22 (see page 113)
PIC18F24J50 (see page 84) PIC18F25J50 (see page 83) PIC18F26J50 (see page 82) PIC18F44J50 (see page 80)
PIC18F45J50 (see page 79) PIC18F46J50 (see page 78)
PIC18LF24J50 (see page 92) PIC18LF25J50 (see page 90) PIC18LF26J50 (see page 89) PIC18LF44J50 (see page 88)
PIC18LF45J50 (see page 87) PIC18LF46J50 (see page 85)

PIC18F25J11 ([see page 98](#)) PIC18F24J11 ([see page 99](#)) PIC18F26J11 ([see page 97](#)) PIC18F45J11 ([see page 94](#))
 PIC18F44J11 ([see page 96](#)) PIC18F46J11 ([see page 93](#))

PIC18LF25J11 ([see page 106](#)) PIC18LF24J11 ([see page 107](#)) PIC18LF26J11 ([see page 104](#)) PIC18LF45J11 ([see page 102](#))
 PIC18LF44J11 ([see page 103](#)) PIC18LF46J11 ([see page 101](#))

PIC18F66J90 ([see page 76](#)) PIC18F67J90 ([see page 75](#)) PIC18F86J90 ([see page 73](#)) PIC18F87J90 ([see page 72](#))

PIC18F66J93 ([see page 70](#)) PIC18F67J93 ([see page 69](#)) PIC18F86J93 ([see page 67](#)) PIC18F87J93 ([see page 66](#))

PIC18F87K90 ([see page 58](#)) PIC18F86K95 ([see page 59](#)) PIC18F86K90 ([see page 60](#)) PIC18F85K90 ([see page 61](#))

PIC18F67K90 ([see page 62](#)) PIC18F66K95 ([see page 63](#)) PIC18F66K90 ([see page 64](#)) PIC18F65K90 ([see page 65](#))

PIC18F87K22 ([see page 44](#)) PIC18F86K27 ([see page 46](#)) PIC18F86K22 ([see page 45](#)) PIC18F85K22 ([see page 47](#))

PIC18F67K22 ([see page 48](#)) PIC18F66K27 ([see page 50](#)) PIC18F66K22 ([see page 49](#)) PIC18F65K22 ([see page 51](#))

PIC18F86J72 ([see page 56](#)) PIC18F87J72 ([see page 55](#)) PIC18F96J72 ([see page 53](#)) PIC18F97J72 ([see page 52](#))

PIC18F26J53 ([see page 37](#)) PIC18F27J53 ([see page 36](#)) PIC18F46J53 ([see page 35](#)) PIC18F47J53 ([see page 34](#))

PIC18F66K80 ([see page 22](#)) PIC18F65K80 ([see page 23](#)) PIC18F46K80 ([see page 24](#)) PIC18F45K80 ([see page 25](#))
 PIC18F26K80 ([see page 26](#)) PIC18F25K80 ([see page 27](#))

PIC18LF66K80 ([see page 28](#)) PIC18LF65K80 ([see page 29](#)) PIC18LF46K80 ([see page 30](#)) PIC18LF45K80 ([see page 31](#))
 PIC18LF26K80 ([see page 32](#)) PIC18LF25K80 ([see page 33](#))

Installation

Peripheral Library comes along with MCC18 Compiler. No separate installation required for library.

What's New in this Release.

This support update to version 2.0.0 includes 12 new devices.

PIC18F26K22 ([see page 267](#)) PIC18F25K22 ([see page 269](#)) PIC18F24K22 ([see page 271](#)) PIC18F23K22 ([see page 273](#))
 PIC18LF26K22 ([see page 268](#)) PIC18LF25K22 ([see page 270](#)) PIC18LF24K22 ([see page 272](#))
 PIC18LF23K22 ([see page 274](#))

PIC18F46K22 ([see page 265](#)) PIC18F45K22 ([see page 259](#)) PIC18F44K22 ([see page 261](#)) PIC18F43K22 ([see page 263](#))
 PIC18LF46K22 ([see page 266](#)) PIC18LF45K22 ([see page 260](#)) PIC18LF44K22 ([see page 262](#))
 PIC18LF43K22 ([see page 264](#))

Migrating to Version 2.0.0

Existing projects can be migrated without modification.

Fixed Issues

- Flash ([see page 1196](#)) buffer window length in library

- PIC18F14K22 (see page 111)/50 Family ADC channel select macros

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. Technical support is available through the web site at: <http://support.microchip.com>

Microchip Technology Inc.

www.microchip.com

Voice: (480) 792-7200

Fax: (480) 792-7277

Microchip Technical Support : **<http://support.microchip.com>**

5 Getting Started

Using the Peripheral Libraries - PIC18 Family

These libraries provide a set of common interface functions to initialize and use peripheral modules on Microchip 8-bit microcontrollers. The libraries have been optimized for code size efficiency.

Building a Project with the Peripheral Library

Building an application which utilizes the 8-bit peripheral library support, requires the library file ".lib" file specific to device under use and a header file for each peripheral module. The archived library file contains all the individual object files for each library function. For each peripheral, the corresponding header file provides all the function prototypes, macros and typedefs used in the library.

The library files are of the form p18wxyz.lib or p18wxyz_e.lib (depending on executable/object file format selected during the compile operation). The library supports all the variants of PIC18 devices as given under the "Devices (see page 11)" heading.

The header files are of the form peripheral.h, where peripheral is the name of the particular peripheral being used (e.g., adc.h (see page 1207) for ADC Module). When linking an application, the library file (p18wxyz.lib or p18wxyz_e.lib) must be provided as an input to the linker (using the --library or -l linker switch) such that the functions used by the application may be linked into the application. The .lib file to be added to the Project window under Library files. When compiling an application, each header file must be referenced (using #include) by all source files which call a function in the library or use its symbols or typedefs.

Library Function Calling Convention

The peripheral library contains some functions that do not require any input arguments, some that require single values or variables as arguments, and some whose arguments can be expressed as a logical AND/AND-OR of several constants that are pre-defined in the header file.

Each constant included in any of these AND-ed expressions corresponds to the 8-bit mask for a possible selection for a certain bit or bit-field in a Special Function Register (SFR). Each argument, in turn, typically corresponds to the value written to an entire SFR. Thus, by logically combining chosen values for every bit-field in the SFR, the complete initialization value of the SFR is obtained.

To use the OR mask setting, you will need to define the macro (#define USE_OR_MASKS) at the beginning of application code prior to inclusion of respective peripheral header files using #include. In addition, the peripheral.h file provides macros xxx_MASK for each bitfield in an SFR. During runtime these macros can be used to clear bitfields. To clear a bitfield, bitwise AND the xxx_MASK with the current contents of the respective register.

Example: Use of xxx_MASK

```
AD1CON &= ADC_TAD_MASK; /* To Clear the bit field */
```

```
AD1CON |= ADC_48_TAD; /* To Set the bit field to new value */
```

Note:

- To avoid inadvertently setting a register bit to an undesired state, please select one (and only one) constant from each group in the logical expression that generates the argument.
- While using the AND_MASK to pass the value to argument of a function, it is necessary to add the macros of all selection/settings with necessary configuration associated with that parameter

Rebuilding the Peripheral Library

The batch file makeall.bat may be used to remake the libraries. The default behavior is to build peripheral libraries for all supported target processors; however, you may select a particular processor to build by adding the device names under the list in the batch file.

For example:

1. At DOS prompt, go to the src directory (by default, it is ..\MCC18\src)
2. Edit the processor list for which the library has to be rebuilt.
3. Type makeall.bat
4. After execution, process ends with the text Build Successful.
5. After successful build it generates p18____.lib or p18____e.lib files in the ..\MCC18\lib folder.

6 Devices

Devices with the Peripheral Library Support

PIC18F45K22_Family
PIC18F46K22 (see page 258)
PIC18LF46K22 (see page 258)
PIC18F45K22 (see page 258)
PIC18LF45K22 (see page 258)
PIC18F44K22 (see page 258)
PIC18LF44K22 (see page 258)
PIC18F43K22 (see page 258)
PIC18LF43K22 (see page 258)
PIC18F26K22 (see page 258)
PIC18LF26K22 (see page 258)
PIC18F25K22 (see page 258)
PIC18LF25K22 (see page 258)
PIC18F24K22 (see page 258)
PIC18LF24K22 (see page 258)
PIC18F23K22 (see page 258)
PIC18LF23K22 (see page 258)

PIC18F66K80_Family
PIC18F66k80 (see page 21)
PIC18F65k80 (see page 21)
PIC18F46k80 (see page 21)
PIC18F45k80 (see page 21)
PIC18F26k80 (see page 21)
PIC18F25k80 (see page 21)
PIC18LF66k80 (see page 21)
PIC18LF65k80 (see page 21)
PIC18LF46k80 (see page 21)
PIC18LF45k80 (see page 21)
PIC18LF26k80 (see page 21)
PIC18LF25k80 (see page 21)

PIC18LF47J53_FamilyPIC18LF26J53 ([↗](#) see page 287)PIC18LF27J53 ([↗](#) see page 287)PIC18LF46J53 ([↗](#) see page 299)PIC18LF47J53 ([↗](#) see page 299)**PIC18F47J13_Family**PIC18F26J13 ([↗](#) see page 287)PIC18F27J13 ([↗](#) see page 287)PIC18F46J13 ([↗](#) see page 299)PIC18F47J13 ([↗](#) see page 299)**PIC18F47J53_Family**PIC18F26J53 ([↗](#) see page 287)PIC18F27J53 ([↗](#) see page 287)PIC18F46J53 ([↗](#) see page 299)PIC18F47J53 ([↗](#) see page 299)**PIC18F87K22_Family**PIC18F87K22 ([↗](#) see page 44)PIC18F86K22 ([↗](#) see page 44)PIC18F86K27 ([↗](#) see page 44)PIC18F85K22 ([↗](#) see page 44)PIC18F67K22 ([↗](#) see page 44)PIC18F66K22 ([↗](#) see page 44)PIC18F66K27 ([↗](#) see page 44)PIC18F65K22 ([↗](#) see page 44)**PIC18F97J72_Family**PIC18F86J72 ([↗](#) see page 52)PIC18F87J72 ([↗](#) see page 52)PIC18F96J72 ([↗](#) see page 52)PIC18F97J72 ([↗](#) see page 52)**PIC18F87K90_Family**PIC18F87K90 ([↗](#) see page 58)

PIC18F86K90 (↗ see page 58)
--

PIC18F86K95 (↗ see page 58)
--

PIC18F85K90 (↗ see page 58)
--

PIC18F67K90 (↗ see page 58)
--

PIC18F66K90 (↗ see page 58)
--

PIC18F66K95 (↗ see page 58)
--

PIC18F65K90 (↗ see page 58)
--

PIC18F87J90_Family

PIC18F66J90 (↗ see page 72)
--

PIC18F67J90 (↗ see page 72)
--

PIC18F86J90 (↗ see page 72)
--

PIC18F87J90 (↗ see page 72)
--

PIC18F87J93_Family

PIC18F66J93 (↗ see page 66)
--

PIC18F67J93 (↗ see page 66)
--

PIC18F86J93 (↗ see page 66)
--

PIC18F87J93 (↗ see page 66)
--

PIC18F46J50_Family

PIC18F24J50 (↗ see page 405)

PIC18F25J50 (↗ see page 405)

PIC18F26J50 (↗ see page 405)

PIC18F44J50 (↗ see page 417)

PIC18F45J50 (↗ see page 417)

PIC18F46J50 (↗ see page 417)

PIC18LF46J50_Family

PIC18LF24J50 (↗ see page 405)
--

PIC18LF25J50 (↗ see page 405)
--

PIC18LF26J50 (↗ see page 405)
--

PIC18LF44J50 (↗ see page 417)
--

PIC18LF45J50 (↗ see page 417)
--

PIC18LF46J50 (↗ see page 417)
--

PIC18F46J11_FamilyPIC18F24J11 ([↗](#) see page 431)PIC18F25J11 ([↗](#) see page 431)PIC18F26J11 ([↗](#) see page 431)PIC18F44J11 ([↗](#) see page 443)PIC18F45J11 ([↗](#) see page 443)PIC18F46J11 ([↗](#) see page 443)**PIC18LF46J11_Family**PIC18LF24J11 ([↗](#) see page 431)PIC18LF25J11 ([↗](#) see page 431)PIC18LF26J11 ([↗](#) see page 431)PIC18LF44J11 ([↗](#) see page 443)PIC18LF45J11 ([↗](#) see page 443)PIC18LF46J11 ([↗](#) see page 443)**PIC18LF14K22_Family**PIC18LF13K22 ([↗](#) see page 114)PIC18LF14K22 ([↗](#) see page 113)**PIC18F14K22_Family**PIC18F13K22 ([↗](#) see page 112)PIC18F14K22 ([↗](#) see page 111)**PIC18LF14K50_Family**PIC18LF13K50 ([↗](#) see page 111)PIC18LF14K50 ([↗](#) see page 110)**PIC18F14K50_Family**PIC18F13K50 ([↗](#) see page 109)PIC18F14K50 ([↗](#) see page 108)**PIC18F4553_Family**PIC18F2458 ([↗](#) see page 115)PIC18F2553 ([↗](#) see page 115)

PIC18F4458 (↗ see page 115)
--

PIC18F4553 (↗ see page 115)
--

PIC18F8493_Family

PIC18F6393 (↗ see page 117)
--

PIC18F6493 (↗ see page 117)
--

PIC18F8393 (↗ see page 117)
--

PIC18F8493 (↗ see page 117)
--

PIC18F8723_Family

PIC18F6628 (↗ see page 119)
--

PIC18F6723 (↗ see page 119)
--

PIC18F8628 (↗ see page 119)
--

PIC18F8723 (↗ see page 119)
--

PIC18F84J15_Family

PIC18F64J15 (↗ see page 121)

PIC18F84J15 (↗ see page 121)

PIC18F87J50_Family

PIC18F65J50 (↗ see page 122)

PIC18F66J50 (↗ see page 122)

PIC18F66J55 (↗ see page 122)

PIC18F67J50 (↗ see page 122)

PIC18F85J50 (↗ see page 122)

PIC18F86J50 (↗ see page 122)

PIC18F86J55 (↗ see page 122)

PIC18F87J50 (↗ see page 122)

PIC18F87J11_Family

PIC18F66J11 (↗ see page 128)

PIC18F66J16 (↗ see page 128)

PIC18F67J11 (↗ see page 128)

PIC18F86J11 (↗ see page 128)

PIC18F86J16 (↗ see page 128)

PIC18F87J11 (↗ see page 128)

PIC18F46K20_Family

PIC18F23K20 ([see page 133](#))

PIC18F24K20 ([see page 133](#))

PIC18F25K20 ([see page 133](#))

PIC18F26K20 ([see page 133](#))

PIC18F43K20 ([see page 133](#))

PIC18F44K20 ([see page 133](#))

PIC18F45K20 ([see page 133](#))

PIC18F46K20 ([see page 133](#))

PIC18F4431_Family

PIC18F2331 ([see page 819](#))

PIC18F2431 ([see page 819](#))

PIC18F4331 ([see page 811](#))

PIC18F4431 ([see page 811](#))

PIC18F97J60_Family

PIC18F66J60 ([see page 143](#))

PIC18F66J65 ([see page 143](#))

PIC18F67J60 ([see page 143](#))

PIC18F86J60 ([see page 143](#))

PIC18F86J65 ([see page 143](#))

PIC18F87J60 ([see page 143](#))

PIC18F96J60 ([see page 143](#))

PIC18F96J65 ([see page 143](#))

PIC18F97J60 ([see page 143](#))

PIC18F87J10_Family

PIC18F65J10 ([see page 150](#))

PIC18F65J15 ([see page 150](#))

PIC18F66J10 ([see page 150](#))

PIC18F66J15 ([see page 150](#))

PIC18F67J10 ([see page 150](#))

PIC18F85J10 ([see page 150](#))

PIC18F85J15 ([see page 150](#))

PIC18F86J10 ([see page 150](#))

PIC18F86J15 ([see page 150](#))

PIC18F87J10 ([↗](#) see page 150)

PIC18F85J90_Family

PIC18F63J90 ([↗](#) see page 163)

PIC18F64J90 ([↗](#) see page 162)

PIC18F65J90 ([↗](#) see page 161)

PIC18F83J90 ([↗](#) see page 160)

PIC18F84J90 ([↗](#) see page 159)

PIC18F85J90 ([↗](#) see page 158)

PIC18F85J11_Family

PIC18F63J11 ([↗](#) see page 168)

PIC18F64J11 ([↗](#) see page 167)

PIC18F65J11 ([↗](#) see page 166)

PIC18F83J11 ([↗](#) see page 165)

PIC18F84J11 ([↗](#) see page 164)

PIC18F85J11 ([↗](#) see page 164)

PIC18F45J10_Family

PIC18F24J10 ([↗](#) see page 798)

PIC18F25J10 ([↗](#) see page 798)

PIC18F44J10 ([↗](#) see page 789)

PIC18F45J10 ([↗](#) see page 789)

PIC18F8621_Family

PIC18F6525 ([↗](#) see page 172)

PIC18F6621 ([↗](#) see page 172)

PIC18F8525 ([↗](#) see page 172)

PIC18F8621 ([↗](#) see page 172)

PIC18F8680_Family

PIC18F6585 ([↗](#) see page 176)

PIC18F6680 ([↗](#) see page 176)

PIC18F8585 ([↗](#) see page 176)

PIC18F8680 ([↗](#) see page 176)

PIC18F8722_Family

PIC18F6527 (see page 180)

PIC18F6622 (see page 180)

PIC18F6627 (see page 180)

PIC18F6722 (see page 180)

PIC18F8527 (see page 180)

PIC18F8622 (see page 180)

PIC18F8627 (see page 180)

PIC18F8722 (see page 180)

PIC18F8490_Family

PIC18F6390 (see page 186)

PIC18F6490 (see page 186)

PIC18F8390 (see page 186)

PIC18F8490 (see page 186)

PIC18F8410_Family

PIC18F6310 (see page 188)

PIC18F6410 (see page 188)

PIC18F8310 (see page 188)

PIC18F8410 (see page 188)

PIC18F4321_Family

PIC18F2221 (see page 643)

PIC18F2321 (see page 643)

PIC18F4221 (see page 635)

PIC18F4321 (see page 635)

PIC18F4685_Family

PIC18F2682 (see page 660)

PIC18F2685 (see page 660)

PIC18F4682 (see page 652)

PIC18F4685 (see page 652)

PIC18F4680_Family

PIC18F2585 (see page 660)

PIC18F2680 (see page 660)

PIC18F4585 (see page 652)

PIC18F4680 (see page 652)

PIC18F4620_Family

PIC18F2525 (see page 691)

PIC18F2620 (see page 691)

PIC18F4525 (see page 683)

PIC18F4620 (see page 683)

PIC18F4610_Family

PIC18F2410 (see page 675)

PIC18F2510 (see page 675)

PIC18F2515 (see page 675)

PIC18F2610 (see page 675)

PIC18F4410 (see page 668)

PIC18F4510 (see page 668)

PIC18F4515 (see page 668)

PIC18F4610 (see page 668)

PIC18F4580_Family

PIC18F2480 (see page 660)

PIC18F2580 (see page 660)

PIC18F4480 (see page 652)

PIC18F4580 (see page 652)

PIC18F4550_Family

PIC18F2455 (see page 217)

PIC18F2550 (see page 216)

PIC18F4455 (see page 215)

PIC18F4550 (see page 214)

PIC18F4450_Family

PIC18F2450 (see page 218)

PIC18F4450 (↗ see page 218)
--

PIC18F4523_Family

PIC18F2423 (↗ see page 223)
--

PIC18F2523 (↗ see page 222)
--

PIC18F4423 (↗ see page 221)
--

PIC18F4523 (↗ see page 220)
--

PIC18F4520_Family

PIC18F2420 (↗ see page 227)
--

PIC18F2520 (↗ see page 226)
--

PIC18F4420 (↗ see page 225)
--

PIC18F4520 (↗ see page 224)
--

PIC18F4320_Family

PIC18F2220 (↗ see page 228)
--

PIC18F2320 (↗ see page 228)
--

PIC18F4220 (↗ see page 228)
--

PIC18F4320 (↗ see page 228)
--

PIC18F1330_Family

PIC18F1230 (↗ see page 230)
--

PIC18F1330 (↗ see page 230)
--

PIC18F1320_Family

PIC18F1220 (↗ see page 232)
--

PIC18F1320 (↗ see page 232)
--

PIC18F8720_Family

PIC18F6520 (↗ see page 233)
--

PIC18F6620 (↗ see page 233)
--

PIC18F6720 (↗ see page 233)
--

PIC18F8520 (↗ see page 233)
--

PIC18F8620 (↗ see page 233)
--

PIC18F8720 (↗ see page 233)
--

PIC18C858_FamilyPIC18C658 ([see page 239](#))PIC18C858 ([see page 239](#))**PIC18C801_Family**PIC18C601 ([see page 241](#))PIC18C801 ([see page 241](#))**PIC18F4539_Family**PIC18F2439 ([see page 243](#))PIC18F2539 ([see page 243](#))PIC18F4439 ([see page 243](#))PIC18F4539 ([see page 243](#))**PIC18F458_Family**PIC18F248 ([see page 768](#))PIC18F258 ([see page 768](#))PIC18F448 ([see page 776](#))PIC18F458 ([see page 776](#))**PIC18F452_Family**PIC18F242 ([see page 251](#))PIC18F252 ([see page 251](#))PIC18F442 ([see page 251](#))PIC18F452 ([see page 251](#))**PIC18C452_Family**PIC18C242 ([see page 254](#))PIC18C252 ([see page 254](#))PIC18C442 ([see page 254](#))PIC18C452 ([see page 254](#))

6.1 PIC18F66K80 Family

6.1.1 PIC18F66K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.2 PIC18F65K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.3 PIC18F46K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.4 PIC18F45K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.5 PIC18F26K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.6 PIC18F25K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.7 PIC18LF66K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.8 PIC18LF65K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.9 PIC18LF46K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.10 PIC18LF45K80

CLICK HERE ( see page 275) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.11 PIC18LF26K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.1.12 PIC18LF25K80

CLICK HERE ([see page 275](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.2 PIC18F47Jx3 Family

6.2.1 PIC18F47J53

CLICK HERE ([see page 299](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
 - programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.2 PIC18F46J53

CLICK HERE ([see page 299](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.3 PIC18F27J53

CLICK HERE ([see page 287](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control

- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.4 PIC18F26J53

CLICK HERE (see page 287) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,

- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.5 PIC18F47J13

CLICK HERE ([see page 299](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
 - programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.6 PIC18F46J13

CLICK HERE ([see page 299](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.7 PIC18F26J13

CLICK HERE ([see page 287](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control

- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.2.8 PIC18F27J13

CLICK HERE (see page 287) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,

- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write
Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)
modules:
- Two Master Synchronous Serial Port (MSSP)
modules Supporting Three-Wire SPI (all four
modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel
Slave Port
- Two-Rail – Rail Analog Comparators with Input
Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)
Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.3 PIC18F87K22 Family

6.3.1 PIC18F87K22

CLICK HERE ([see page 356](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support

- Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
- Auto-acquisition and Sleep operation

6.3.2 PIC18F86K22

CLICK HERE ([see page 356](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
- Seven Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)

- 10-bit A/D converter with up to 24 channels:
- Auto-acquisition and Sleep operation

6.3.3 PIC18F86K27

CLICK HERE ([see page 356](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:

- Auto-acquisition and Sleep operation

6.3.4 PIC18F85K22

CLICK HERE ([see page 370](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.3.5 PIC18F67K22

CLICK HERE ([see page 326](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.3.6 PIC18F66K22

CLICK HERE ([see page 326](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.3.7 PIC18F66K27

CLICK HERE ([see page 326](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.3.8 PIC18F65K22

CLICK HERE ([see page 342](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.4 PIC18F97J72 Family

6.4.1 PIC18F97J72

CLICK HERE ([see page 315](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode

- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC)
with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.4.2 PIC18F96J72

CLICK HERE (see page 315) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Low-Power Features:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz

- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical

- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.4.3 PIC18F87J72

CLICK HERE (see page 315) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode

- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC)
with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.4.4 PIC18F86J72

CLICK HERE (see page 315) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Low-Power Features:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz

- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical

- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.5 PIC18F87K90 Family

6.5.1 PIC18F87K90

CLICK HERE (see page 356) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:

- 3/4-wire SPI (supports all four SPI modes)
- I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.5.2 PIC18F86K95

CLICK HERE ([see page 356](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)

- I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.5.3 PIC18F86K90

CLICK HERE ([see page 356](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode

- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.5.4 PIC18F85K90

CLICK HERE ([see page 370](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:

- LIN/J2602 support
- Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
- Auto-acquisition and Sleep operation

6.5.5 PIC18F67K90

CLICK HERE ([see page 326](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support

- Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
- Auto-acquisition and Sleep operation

6.5.6 PIC18F66K95

CLICK HERE ([see page 326](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
- Seven Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)

- 10-bit A/D converter with up to 24 channels:
- Auto-acquisition and Sleep operation

6.5.7 PIC18F66K90

CLICK HERE ( see page 326) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:

- Auto-acquisition and Sleep operation

6.5.8 PIC18F65K90

CLICK HERE ([see page 342](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- Ten CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- Eleven 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3,5,7 – 16-bit timer/counter
 - Timer2,4,6,8,10,12 – 8-bit timer/counter
- Three analog comparators
- Configurable reference clock output
- Hardware Real-Time Clock and Calendar (RTCC) module with clock, calendar and alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 10-bit A/D converter with up to 24 channels:
 - Auto-acquisition and Sleep operation

6.6 PIC18F87J93 Family

6.6.1 PIC18F87J93

CLICK HERE ([see page 394](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode

- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC)
with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.6.2 PIC18F86J93

CLICK HERE (see page 394) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Low-Power Features:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz

- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical

- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.6.3 PIC18F67J93

CLICK HERE (see page 394) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode

- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC)
with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.6.4 PIC18F66J93

CLICK HERE (see page 394) for the Peripheral Library Support Details for this Device

Micro Controller Features:-**Low-Power Features:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz

- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical

- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.7 PIC18F87J90 Family

6.7.1 PIC18F87J90

CLICK HERE (see page 384) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules

- Master Synchronous Serial Port (MSSP) module

with two modes of operation:

- 3-wire/4-wire SPI (supports all four SPI modes)

- I2C™ Master and Slave mode

- One addressable USART module

- One enhanced addressable USART module:

- LIN/J2602 support

- Auto-wake-up on Start bit and Break character

- Auto-Baud Detect (ABD)

- 10-Bit, up to 12-Channel A/D Converter:

- Auto-acquisition

- Conversion available during Sleep

- Two analog comparators

- Programmable Reference Voltage for Comparators

- Hardware Real-Time Clock and Calendar (RTCC)

with Clock, Calendar and Alarm functions

- Charge Time Measurement Unit (CTMU):

- Capacitance measurement

- Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program

memory, typical

- Flash (see page 1196) retention 20 years, minimum

- Self-programmable under software control

- Word write capability for Flash (see page 1196) program memory

for data EEPROM emulators

6.7.2 PIC18F86J90

CLICK HERE (see page 384) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on

- Idle: CPU off, peripherals on

- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement

- Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.7.3 PIC18F67J90

CLICK HERE (see page 384) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Low-Power Features:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules

- Master Synchronous Serial Port (MSSP) module

with two modes of operation:

- 3-wire/4-wire SPI (supports all four SPI modes)
- I2C™ Master and Slave mode

- One addressable USART module

- One enhanced addressable USART module:

- LIN/J2602 support
- Auto-wake-up on Start bit and Break character
- Auto-Baud Detect (ABD)

- 10-Bit, up to 12-Channel A/D Converter:

- Auto-acquisition
- Conversion available during Sleep

- Two analog comparators

- Programmable Reference Voltage for Comparators

- Hardware Real-Time Clock and Calendar (RTCC)

with Clock, Calendar and Alarm functions

- Charge Time Measurement Unit (CTMU):

- Capacitance measurement
- Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.7.4 PIC18F66J90

CLICK HERE (see page 384) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Low-Power Features:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on

- Sleep: CPU off, peripherals off
- Two-Speed Oscillator Start-up
- Two Crystal modes, 4-25 MHz
- Two External Clock modes, Up to 48 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block with PLL:
 - Eight user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 at 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock fails

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- One addressable USART module
- One enhanced addressable USART module:
 - LIN/J2602 support
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two analog comparators
- Programmable Reference Voltage for Comparators
- Hardware Real-Time Clock and Calendar (RTCC) with Clock, Calendar and Alarm functions
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement

- Time measurement with 1 ns typical resolution

Special Microcontroller Features:

- 10,000 erase/write cycle Flash (see page 1196) program memory, typical
- Flash (see page 1196) retention 20 years, minimum
- Self-programmable under software control
- Word write capability for Flash (see page 1196) program memory for data EEPROM emulators

6.8 PIC18F46J50 Family

6.8.1 PIC18F46J50

CLICK HERE (see page 417) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:

- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)
modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four
modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine

- 8-Bit Parallel Master Port/Enhanced Parallel
Slave Port

- Two-Rail – Rail Analog Comparators with Input
Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.8.2 PIC18F45J50


CLICK HERE ( see page 417) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ( see page 1196) Technology

- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.3 PIC18F44J50

CLICK HERE (see page 417) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.4 PIC18F26J50

CLICK HERE ([see page 405](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
 - programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four

modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.8.5 PIC18F25J50

CLICK HERE ([see page 405](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:

- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)
modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four
modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine

- 8-Bit Parallel Master Port/Enhanced Parallel
Slave Port

- Two-Rail – Rail Analog Comparators with Input
Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)
Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.8.6 PIC18F24J50

CLICK HERE ([see page 405](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology

- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.7 PIC18LF46J50

CLICK HERE (see page 417) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.8 PIC18LF45J50

CLICK HERE ([see page 417](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four

modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.8.9 PIC18LF44J50

CLICK HERE ([see page 417](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:

- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine

- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.8.10 PIC18LF26J50

CLICK HERE ([see page 405](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology

- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.11 PIC18LF25J50

CLICK HERE (see page 405) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.8.12 PIC18LF24J50

CLICK HERE ([see page 405](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
 - programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four

modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine

- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module

- Charge Time Measurement Unit (CTMU):

- Two Enhanced USART modules:

- Auto-Baud Detect

6.9 PIC18F46J11 Family

6.9.1 PIC18F46J11

CLICK HERE ([see page 443](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):

- Flash (see page 1196) Program Memory of 10,000 Erase/Write

Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA

(PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.2 PIC18F45J11

CLICK HERE (see page 443) for the *Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers,
- programmable WDT or RTCC alarm

- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.3 PIC18F44J11

CLICK HERE ([see page 443](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers, programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.4 PIC18F26J11

CLICK HERE ([see page 431](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.5 PIC18F25J11

CLICK HERE ([see page 431](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control

- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.6 PIC18F24J11

CLICK HERE (see page 431) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,

- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.7 PIC18LF46J11

CLICK HERE ([see page 443](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers, programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.8 PIC18LF45J11

CLICK HERE ([see page 443](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.9 PIC18LF44J11

CLICK HERE ([see page 443](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers, programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control

- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.10 PIC18LF26J11

CLICK HERE (see page 431) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**nanoWatt XLP™ for Extreme Low-Power:**

- Deep Sleep mode: CPU off, Peripherals off,

- Able to wake-up on external triggers,
programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash (see page 1196) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash (see page 1196) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA
(PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Two-Rail – Rail Analog Comparators with Input Multiplexing
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module
- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.11 PIC18LF25J11

CLICK HERE ([see page 431](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
 - Able to wake-up on external triggers, programmable WDT or RTCC alarm
 - Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
- Hardware Real-Time Clock and Calendar (RTCC):
- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Master Synchronous Serial Port (MSSP) modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes
- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input

Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.9.12 PIC18LF24J11

CLICK HERE ([see page 431](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

nanoWatt XLP™ for Extreme Low-Power:

- Deep Sleep mode: CPU off, Peripherals off,
- Able to wake-up on external triggers,
- programmable WDT or RTCC alarm
- Ultra Low-Power Wake-up (ULPWU)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital only pins)
- Low-Power, High-Speed CMOS Flash ([see page 1196](#)) Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
- Flash ([see page 1196](#)) Program Memory of 10,000 Erase/Write

Cycles Minimum and 20-Year Data Retention

Peripheral Highlights:

- Peripheral Pin Select:
 - Hardware Real-Time Clock and Calendar (RTCC):
 - High-Current Sink/Source 25 mA/25 mA
- (PORTB and PORTC)

- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Enhanced Capture/Compare/PWM (ECCP)

modules:

- Two Master Synchronous Serial Port (MSSP)

modules Supporting Three-Wire SPI (all four modes) and I2C™ Master and Slave modes

- Full-Duplex Master/Slave SPI DMA Engine
- 8-Bit Parallel Master Port/Enhanced Parallel

Slave Port

- Two-Rail – Rail Analog Comparators with Input Multiplexing

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D)

Converter module

- High/Low-Voltage Detect module
- Charge Time Measurement Unit (CTMU):
- Two Enhanced USART modules:
- Auto-Baud Detect

6.10 PIC18F1xK50 Family

6.10.1 PIC18F14K50

CLICK HERE ([see page 457](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module

- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.10.2 PIC18F13K50

CLICK HERE ( *see page 457*) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)

- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.10.3 PIC18LF14K50

CLICK HERE ([see page 457](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.10.4 PIC18LF13K50

CLICK HERE ([see page 457](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.11 PIC18F1xK22 Family

6.11.1 PIC18F14K22

CLICK HERE ([see page 465](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
 - Enhanced Addressable USART module
 - 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
 - Dual analog comparators
 - Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
 - On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.11.2 PIC18F13K22

CLICK HERE ([see page 465](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups

- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.11.3 PIC18LF14K22

CLICK HERE ([see page 465](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module

- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.11.4 PIC18LF13K22

CLICK HERE ([see page 465](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- 7 independent input-change interrupts
- 7 independent weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes with address mask
- Enhanced Addressable USART module
- 10-bit, up to 9-channel Analog-to-Digital Converter module (ADC)
- Dual analog comparators
- Programmable On-Chip Voltage Reference (CVREF) module (% of VDD) or 1.024V Fixed Voltage Reference (FVR)
- On-chip 3.2V LDO Regulator (PIC18F1XK50 devices only)

6.12 PIC18F4553 Family

6.12.1 PIC18F4553

CLICK HERE ([see page 474](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
- Enhanced Capture/Compare/PWM (ECCP) module:
- Enhanced USART module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C?
- Master and Slave modes
- 12-Bit, up to 13-Channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.12.2 PIC18F4458

CLICK HERE ([see page 474](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
- Enhanced Capture/Compare/PWM (ECCP) module:

- Enhanced USART module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C?
- Master and Slave modes
- 12-Bit, up to 13-Channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.12.3 PIC18F2553

CLICK HERE ([see page 474](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
- Enhanced Capture/Compare/PWM (ECCP) module:
- Enhanced USART module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C?
- Master and Slave modes
- 12-Bit, up to 13-Channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.12.4 PIC18F2458

CLICK HERE ([see page 474](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)

- Up to 2 Capture/Compare/PWM (CCP) modules:
- Enhanced Capture/Compare/PWM (ECCP) module:
- Enhanced USART module:
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C?
- Master and Slave modes
- 12-Bit, up to 13-Channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.13 PIC18F8493 Family

6.13.1 PIC18F8493

CLICK HERE ([see page 482](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 12-Channel Analog-to-Digital (A/D) Converter module:
- High-Current Sink/Source 25 mA/25 mA
- Four External Interrupts
- Four Input Change Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Real-Time Clock (RTC) Software module:
- Up to Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module Supporting Three-Wire SPI (all four modes) and I2C? Master and Slave modes
- Addressable USART module:
- Enhanced Addressable USART module:
- Dual Analog Comparators with Input Multiplexing

6.13.2 PIC18F8393

CLICK HERE (see page 482) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 12-Channel Analog-to-Digital (A/D) Converter module:
- High-Current Sink/Source 25 mA/25 mA
- Four External Interrupts
- Four Input Change Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Real-Time Clock (RTC) Software module:
- Up to Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module

Supporting Three-Wire SPI (all four modes) and

I²C? Master and Slave modes

- Addressable USART module:
- Enhanced Addressable USART module:
- Dual Analog Comparators with Input Multiplexing

6.13.3 PIC18F6493

CLICK HERE (see page 482) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 12-Channel Analog-to-Digital (A/D) Converter module:
- High-Current Sink/Source 25 mA/25 mA
- Four External Interrupts
- Four Input Change Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Real-Time Clock (RTC) Software module:
- Up to Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module

Supporting Three-Wire SPI (all four modes) and

I²C? Master and Slave modes

- Addressable USART module:
- Enhanced Addressable USART module:
- Dual Analog Comparators with Input Multiplexing

6.13.4 PIC18F6393

CLICK HERE ([see page 482](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 12-Channel Analog-to-Digital (A/D) Converter module:
- High-Current Sink/Source 25 mA/25 mA
- Four External Interrupts
- Four Input Change Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Real-Time Clock (RTC) Software module:
- Up to Two Capture/Compare/PWM (CCP) modules
- Master Synchronous Serial Port (MSSP) module

Supporting Three-Wire SPI (all four modes) and

I2C? Master and Slave modes

- Addressable USART module:
- Enhanced Addressable USART module:
- Dual Analog Comparators with Input Multiplexing

6.14 PIC18F8723 Family

6.14.1 PIC18F8723

CLICK HERE ([see page 491](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 16-Channel Analog-to-Digital
- Converter module (A/D):
- Two Master Synchronous Serial Port (MSSP) modules supporting 2/3/4-Wire SPI (all four modes) and I2C Master and Slave modes
- Two Capture/Compare/PWM (CCP) modules

- Three Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Enhanced Addressable USART modules:
- Dual Analog Comparators with Input Multiplexing
- High-Current Sink/Source 25 mA/25 mA
- Four Programmable External Interrupts
- Four Input Change Interrupts

6.14.2 PIC18F8628

CLICK HERE ([see page 491](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 16-Channel Analog-to-Digital
- Converter module (A/D):
- Two Master Synchronous Serial Port (MSSP) modules supporting 2/3/4-Wire SPI (all four modes) and I2C Master and Slave modes
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Enhanced Addressable USART modules:
- Dual Analog Comparators with Input Multiplexing
- High-Current Sink/Source 25 mA/25 mA
- Four Programmable External Interrupts
- Four Input Change Interrupts

6.14.3 PIC18F6723

CLICK HERE ([see page 491](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 16-Channel Analog-to-Digital
- Converter module (A/D):
- Two Master Synchronous Serial Port (MSSP) modules supporting 2/3/4-Wire SPI (all four modes) and I2C Master and Slave modes
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:

- Two Enhanced Addressable USART modules:
- Dual Analog Comparators with Input Multiplexing
- High-Current Sink/Source 25 mA/25 mA
- Four Programmable External Interrupts
- Four Input Change Interrupts

6.14.4 PIC18F6628

CLICK HERE ([see page 491](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- 12-Bit, Up to 16-Channel Analog-to-Digital
- Converter module (A/D):
- Two Master Synchronous Serial Port (MSSP) modules supporting 2/3/4-Wire SPI (all four modes) and I2C Master and Slave modes
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
- Two Enhanced Addressable USART modules:
- Dual Analog Comparators with Input Multiplexing
- High-Current Sink/Source 25 mA/25 mA
- Four Programmable External Interrupts
- Four Input Change Interrupts

6.15 PIC18F84J15 Family

6.15.1 PIC18F84J15

CLICK HERE ([see page 501](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
(PORTB and PORTC)

- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
- Dual analog comparators with input multiplexing

6.15.2 PIC18F64J15

CLICK HERE (🔗 see page 501) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
(PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
- Dual analog comparators with input multiplexing

6.16 PIC18F87J50 Family

6.16.1 PIC18F87J50

CLICK HERE (see page 511) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.16.2 PIC18F86J55

CLICK HERE (see page 511) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs

- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect

6.16.3 PIC18F86J50

CLICK HERE ([see page 511](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:

- Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect

6.16.4 PIC18F85J50

CLICK HERE ([see page 511](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.16.5 PIC18F67J50

CLICK HERE ([see page 511](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.16.6 PIC18F66J55

CLICK HERE ([see page 511](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time

- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect

6.16.7 PIC18F66J50

CLICK HERE ( see page 511) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit

- Auto-Baud Detect

6.16.8 PIC18F65J50

CLICK HERE ([see page 511](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.17 PIC18F87J11 Family

6.17.1 PIC18F87J11

CLICK HERE (see page 522) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.17.2 PIC18F86J16

CLICK HERE (see page 522) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs

- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.17.3 PIC18F86J11

CLICK HERE ([see page 522](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:

- Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect

6.17.4 PIC18F67J11

CLICK HERE ([see page 522](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.17.5 PIC18F66J16

CLICK HERE ([see page 522](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

6.17.6 PIC18F66J11

CLICK HERE (see page 522) for the Peripheral Library Support Details for this Device

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25mA (PORTB and PORTC)
- Four Programmable External Interrupts
- Four Input Change Interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time

- Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I2C? Master and Slave modes
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing Peripheral Highlights (continued):
- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect

6.18 PIC18F46K20 Family

6.18.1 PIC18F46K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)

- I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.18.2 PIC18F45K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)
 - I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.18.3 PIC18F44K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)
 - I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.18.4 PIC18F43K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)
 - I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.18.5 PIC18F26K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs

- Selectable polarity
- Programmable dead time
- Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
- 3-wire SPI (supports all 4 modes)
- I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
- Supports RS-485, RS-232 and LIN
- RS-232 operation using internal oscillator
- Auto-Wake-up on Break
- Auto-Baud Detect

6.18.6 PIC18F25K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
- High-Current Sink/Source 25 mA/25 mA
- Three programmable external interrupts
- Four programmable interrupt-on-change
- Eight programmable weak pull-ups
- Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
- One, two or four PWM outputs
- Selectable polarity
- Programmable dead time
- Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
- 3-wire SPI (supports all 4 modes)
- I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
- Supports RS-485, RS-232 and LIN
- RS-232 operation using internal oscillator

- Auto-Wake-up on Break
- Auto-Baud Detect

6.18.7 PIC18F24K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)
 - I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.18.8 PIC18F23K20

CLICK HERE ([see page 533](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- Up to 35 I/O pins plus 1 input-only pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Eight programmable weak pull-ups
 - Programmable slew rate
- Capture/Compare/PWM (CCP) module
- Enhanced CCP (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module
 - 3-wire SPI (supports all 4 modes)
 - I2C™ Master and Slave modes with address mask
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module:
 - Supports RS-485, RS-232 and LIN
 - RS-232 operation using internal oscillator
 - Auto-Wake-up on Break
 - Auto-Baud Detect

6.19 PIC18F4431 Family

6.19.1 PIC18F4431

CLICK HERE ([see page 811](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Two Capture/Compare/PWM (CCP) modules:

- Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: PWM resolution is 1 to 10 bits
- Enhanced USART module:
- Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect
- RS-232 Operation using Internal Oscillator Block
(no external crystal required)

High-Speed, 200 ksps 10-Bit A/D Converter:

- Up to 9 Channels
- Simultaneous, Two-Channel Sampling
- Sequential Sampling: 1, 2 or 4 Selected Channels
- Auto-Conversion Capability
- 4-Word FIFO with Selectable Interrupt Frequency
- Selectable External Conversion Triggers
- Programmable Acquisition Time

14-Bit Power Control PWM Module:

- Up to 4 Channels with Complementary Outputs
- Edge or Center-Aligned Operation
- Flexible Dead-Band Generator
- Hardware Fault Protection Inputs
- Simultaneous Update of Duty Cycle and Period:
- Flexible Special Event Trigger output

6.19.2 PIC18F4331

CLICK HERE ([see page 811](#)) ***for the Peripheral Library Support Details for this Device***

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Two Capture/Compare/PWM (CCP) modules:
- Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max. resolution 100 ns (TCY)

- PWM output: PWM resolution is 1 to 10 bits
- Enhanced USART module:
- Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect
- RS-232 Operation using Internal Oscillator Block
(no external crystal required)

High-Speed, 200 ksps 10-Bit A/D Converter:

- Up to 9 Channels
- Simultaneous, Two-Channel Sampling
- Sequential Sampling: 1, 2 or 4 Selected Channels
- Auto-Conversion Capability
- 4-Word FIFO with Selectable Interrupt Frequency
- Selectable External Conversion Triggers
- Programmable Acquisition Time

14-Bit Power Control PWM Module:

- Up to 4 Channels with Complementary Outputs
- Edge or Center-Aligned Operation
- Flexible Dead-Band Generator
- Hardware Fault Protection Inputs
- Simultaneous Update of Duty Cycle and Period:
- Flexible Special Event Trigger output

6.19.3 PIC18F2431

CLICK HERE ([see page 819](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: PWM resolution is 1 to 10 bits
- Enhanced USART module:

- Supports RS-485, RS-232 and LIN 1.2
- Auto-wake-up on Start bit
- Auto-Baud Detect
- RS-232 Operation using Internal Oscillator Block
(no external crystal required)

High-Speed, 200 ksps 10-Bit A/D Converter:

- Up to 9 Channels
- Simultaneous, Two-Channel Sampling
- Sequential Sampling: 1, 2 or 4 Selected Channels
- Auto-Conversion Capability
- 4-Word FIFO with Selectable Interrupt Frequency
- Selectable External Conversion Triggers
- Programmable Acquisition Time

14-Bit Power Control PWM Module:

- Up to 4 Channels with Complementary Outputs
- Edge or Center-Aligned Operation
- Flexible Dead-Band Generator
- Hardware Fault Protection Inputs
- Simultaneous Update of Duty Cycle and Period:
- Flexible Special Event Trigger output

6.19.4 PIC18F2331

CLICK HERE ([see page 819](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10 bits
- Enhanced USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit

- Auto-Baud Detect
- RS-232 Operation using Internal Oscillator Block
(no external crystal required)

High-Speed, 200 ksps 10-Bit A/D Converter:

- Up to 9 Channels
- Simultaneous, Two-Channel Sampling
- Sequential Sampling: 1, 2 or 4 Selected Channels
- Auto-Conversion Capability
- 4-Word FIFO with Selectable Interrupt Frequency
- Selectable External Conversion Triggers
- Programmable Acquisition Time

14-Bit Power Control PWM Module:

- Up to 4 Channels with Complementary Outputs
- Edge or Center-Aligned Operation
- Flexible Dead-Band Generator
- Hardware Fault Protection Inputs
- Simultaneous Update of Duty Cycle and Period:
- Flexible Special Event Trigger output

6.20 PIC18F97J60 Family

6.20.1 PIC18F97J60

CLICK HERE ([see page 560](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
- Five Timer modules (Timer0 to Timer4)
- Four External Interrupt pins
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity

- Programmable dead time
- Auto-shutdown and auto-restart
- Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
- Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
- 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Parallel Slave Port (PSP) module
(100-pin devices only)

6.20.2 PIC18F96J65

CLICK HERE ( [see page 560](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
- Five Timer modules (Timer0 to Timer4)
- Four External Interrupt pins
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
- Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)

- 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.3 PIC18F96J60

CLICK HERE ([see page 560](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
 - Five Timer modules (Timer0 to Timer4)
 - Four External Interrupt pins
 - Two Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
 - Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.4 PIC18F87J60

CLICK HERE ([see page 551](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
 - Five Timer modules (Timer0 to Timer4)
 - Four External Interrupt pins
 - Two Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
 - Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.5 PIC18F86J65

CLICK HERE ([see page 551](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
 - Five Timer modules (Timer0 to Timer4)
 - Four External Interrupt pins
 - Two Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
 - Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.6 PIC18F86J60

CLICK HERE ([see page 551](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
- Five Timer modules (Timer0 to Timer4)
- Four External Interrupt pins
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs

- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
- Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
- 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Parallel Slave Port (PSP) module
(100-pin devices only)

6.20.7 PIC18F67J60

CLICK HERE ([see page 542](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
- Five Timer modules (Timer0 to Timer4)
- Four External Interrupt pins
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
- Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit

- Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.8 PIC18F66J65

CLICK HERE ([see page 542](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
 - Five Timer modules (Timer0 to Timer4)
 - Four External Interrupt pins
 - Two Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
 - Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.20.9 PIC18F66J60

CLICK HERE ([see page 542](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
 - Five Timer modules (Timer0 to Timer4)
 - Four External Interrupt pins
 - Two Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I2C™ Master and Slave modes
 - Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
 - 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Dual Analog Comparators with Input Multiplexing
 - Parallel Slave Port (PSP) module
- (100-pin devices only)

6.21 PIC18F87J10 Family

6.21.1 PIC18F87J10

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.2 PIC18F86J15

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.3 PIC18F86J10

CLICK HERE ( see page 570) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time

- Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-Wake-up on Start bit
- Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.4 PIC18F85J15

CLICK HERE (📄 see page 570) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
- Auto-Wake-up on Start bit
- Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability

- Conversion available during Sleep
- Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.5 PIC18F85J10

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.6 PIC18F67J10

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.7 PIC18F66J15

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:

- One, two or four PWM outputs
- Selectable polarity
- Programmable dead time
- Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.8 PIC18F66J10

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit

- Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.9 PIC18F65J15

CLICK HERE ([see page 570](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.21.10 PIC18F65J10

CLICK HERE ( see page 570) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Four programmable external interrupts
- Four input change interrupts
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Two Enhanced Addressable USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 15-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Self-calibration feature
- Dual analog comparators with input multiplexing

6.22 PIC18F85J90 Family

6.22.1 PIC18F85J90

CLICK HERE ([see page 580](#)) ***for the Peripheral Library Support Details for this Device***

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.22.2 PIC18F84J90

CLICK HERE ([see page 580](#)) ***for the Peripheral Library Support Details for this Device***

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA

- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.22.3 PIC18F83J90

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)

- PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.22.4 PIC18F65J90

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:

- Supports LIN 1.2
- Auto-wake-up on Start bit and Break character
- Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.22.5 PIC18F64J90

CLICK HERE ( see page 580) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep

- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.22.6 PIC18F63J90

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23 PIC18F85J11 Family

6.23.1 PIC18F85J11

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23.2 PIC18F84J11

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23.3 PIC18F83J11

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules

- Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23.4 PIC18F65J11

CLICK HERE ( [see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:

- 3-wire/4-wire SPI (supports all 4 SPI modes)
- I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23.5 PIC18F64J11

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character

- Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
- Auto-acquisition
- Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.23.6 PIC18F63J11

CLICK HERE ([see page 580](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
- Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
- 3-wire/4-wire SPI (supports all 4 SPI modes)
- I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
- Supports LIN 1.2
- Auto-wake-up on Start bit and Break character
- Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
- Auto-acquisition
- Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.24 PIC18F45J10 Family

6.24.1 PIC18F45J10

CLICK HERE ([see page 789](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.24.2 PIC18F44J10

CLICK HERE ([see page 789](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.24.3 PIC18F25J10

CLICK HERE ([see page 798](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
 - Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.24.4 PIC18F24J10

CLICK HERE ([see page 798](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA (PORTB and PORTC)
- Sleep current as low as 100nA
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules

- Uses Timer1
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of Operation:
 - 3-wire/4-wire SPI (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- One Addressable USART module
- One Enhanced USART module:
 - Supports LIN 1.2
 - Auto-wake-up on Start bit and Break character
 - Auto-Baud Detect
- 10-Bit, up to 12-Channel A/D Converter:
 - Auto-acquisition
 - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

6.25 PIC18F8621 Family

6.25.1 PIC18F8621

CLICK HERE ([see page 589](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter

- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.25.2 PIC18F8525

CLICK HERE ([see page 589](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter

- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.25.3 PIC18F6621

CLICK HERE ([see page 589](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3

- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.25.4 PIC18F6525

CLICK HERE ( see page 589) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:

- Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.26 PIC18F8680 Family

6.26.1 PIC18F8680

CLICK HERE ( see page 599) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter

- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.26.2 PIC18F8585

6

CLICK HERE ([see page 599](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter

- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.26.3 PIC18F6680

6

CLICK HERE ([see page 599](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter

- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.26.4 PIC18F6585

CLICK HERE ([see page 599](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter

- Secondary oscillator clock option – Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: 1 to 10-bit PWM resolution
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - Same Capture/Compare features as CCP
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown on external event
 - Auto-Restart
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 2/3/4-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Rate Detect
- Parallel Slave Port (PSP) module

6.27 PIC18F8722 Family

6.27.1 PIC18F8722

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):

- One, two or four PWM outputs
- Programmable dead time
- Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.2 PIC18F8627

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.3 PIC18F8622

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.4 PIC18F8527

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):

- One, two or four PWM outputs
- Programmable dead time
- Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.5 PIC18F6722

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.6 PIC18F6627

CLICK HERE ( see page 607) for the *Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.7 PIC18F6622

CLICK HERE ( see page 607) for the *Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):

- One, two or four PWM outputs
- Programmable dead time
- Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.27.8 PIC18F6527

CLICK HERE ([see page 607](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module

6.28 PIC18F8490 Family

6.28.1 PIC18F8490

CLICK HERE ([see page 618](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.28.2 PIC18F8390

CLICK HERE ([see page 618](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA

- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.28.3 PIC18F6490

CLICK HERE ( see page 618) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep

- Dual Analog Comparators with Input Multiplexing

6.28.4 PIC18F6390

CLICK HERE ([see page 618](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.29 PIC18F8410 Family

6.29.1 PIC18F8410

CLICK HERE ([see page 626](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.29.2 PIC18F8310

CLICK HERE ( see page 626) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:

- Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.29.3 PIC18F6410

CLICK HERE ([see page 626](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.29.4 PIC18F6310

CLICK HERE ([see page 626](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts

- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.30 PIC18F4321 Family

6.30.1 PIC18F4321

CLICK HERE ([see page 635](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2

- RS-232 operation using internal oscillator block (no external crystal required)
- Auto-Wake-up on Start bit
- Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual analog comparators with input multiplexing
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.30.2 PIC18F4221

CLICK HERE ([see page 635](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep

- Dual analog comparators with input multiplexing
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.30.3 PIC18F2321

CLICK HERE ([see page 643](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual analog comparators with input multiplexing
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.30.4 PIC18F2221

CLICK HERE ([see page 643](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual analog comparators with input multiplexing
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.31 PIC18F4685 Family

6.31.1 PIC18F4685

CLICK HERE ([see page 652](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksp/s:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.31.2 PIC18F4682

CLICK HERE ([see page 652](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts

- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksp/s:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.31.3 PIC18F2685

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)

- Auto-wake-up on Start bit
- Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
- Auto-acquisition capability
- Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.31.4 PIC18F2682

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.32 PIC18F4680 Family

6.32.1 PIC18F4680

CLICK HERE ([see page 652](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksp/s:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.32.2 PIC18F4585

CLICK HERE ([see page 652](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts

- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.32.3 PIC18F2680

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)

- Auto-wake-up on Start bit
- Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
- Auto-acquisition capability
- Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.32.4 PIC18F2585

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.33 PIC18F4620 Family

6.33.1 PIC18F4620

CLICK HERE ([see page 683](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.33.2 PIC18F4525

CLICK HERE ([see page 683](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.33.3 PIC18F2620

CLICK HERE ([see page 691](#)) ***for the Peripheral Library Support Details for this Device***

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts

- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.33.4 PIC18F2525

CLICK HERE ([see page 691](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity

- Programmable dead time
- Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34 PIC18F4610 Family

6.34.1 PIC18F4610

CLICK HERE ( [see page 668](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time

- Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.2 PIC18F4515

CLICK HERE ( see page 668) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602

- RS-232 operation using internal oscillator block (no external crystal required)
- Auto-wake-up on Start bit
- Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.3 PIC18F4510

CLICK HERE ([see page 668](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability

- Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.4 PIC18F4410

CLICK HERE ([see page 668](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.5 PIC18F2610

CLICK HERE ([see page 675](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.6 PIC18F2515

CLICK HERE ([see page 675](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.7 PIC18F2510

CLICK HERE ([see page 675](#)) ***for the Peripheral Library Support Details for this Device***

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts

- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.34.8 PIC18F2410

CLICK HERE ([see page 675](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity

- Programmable dead time
- Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

6.35 PIC18F4580 Family

6.35.1 PIC18F4580

CLICK HERE ( see page 652) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.35.2 PIC18F4480

CLICK HERE ([see page 652](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksps:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.35.3 PIC18F2580

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksp/s:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.35.4 PIC18F2480

CLICK HERE ([see page 660](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/source 25 mA/25 mA
- Three External Interrupts

- One Capture/Compare/PWM (CCP1) module
- Enhanced Capture/Compare/PWM (ECCP1) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter module (A/D), up to 100 ksp/s:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.36 PIC18F4550 Family

6.36.1 PIC18F4550

CLICK HERE ( see page 699) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:

- Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.36.2 PIC18F4455

CLICK HERE ( [see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA

- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.36.3 PIC18F2550

CLICK HERE ( see page 699) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)

- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.36.4 PIC18F2455

CLICK HERE ( see page 699) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)

- Compare is 16-bit, max. resolution 83.3 ns (TCY)
- PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.37 PIC18F4450 Family

6.37.1 PIC18F4450

CLICK HERE ([see page 708](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)

- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.37.2 PIC18F2450

CLICK HERE ( see page 708) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)

- Compare is 16-bit, max. resolution 83.3 ns (TCY)
- PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.38 PIC18F4523 Family

6.38.1 PIC18F4523

CLICK HERE ( see page 699) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)

- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.38.2 PIC18F4423

CLICK HERE ( see page 699) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)

- Compare is 16-bit, max. resolution 83.3 ns (TCY)
- PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.38.3 PIC18F2523

CLICK HERE ( [see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit

- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.38.4 PIC18F2423

CLICK HERE ([see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes

- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Enhanced USART module:
- LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.39 PIC18F4520 Family

6.39.1 PIC18F4520

CLICK HERE ([see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit

- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.39.2 PIC18F4420

CLICK HERE ([see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes

- Selectable polarity
- Programmable dead time
- Auto-shutdown and auto-restart
- Enhanced USART module:
- LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.39.3 PIC18F2520

CLICK HERE ([see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time

- Auto-shutdown and auto-restart
- Enhanced USART module:
- LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.39.4 PIC18F2420

CLICK HERE ([see page 699](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:

- LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

6.40 PIC18F4320 Family

6.40.1 PIC18F4320

CLICK HERE ([see page 713](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution is 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution is 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Compatible 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators
- Addressable USART module:
 - RS-232 operation using internal oscillator block (no external crystal required)

6.40.2 PIC18F4220

CLICK HERE ([see page 713](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution is 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution is 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Compatible 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators
- Addressable USART module:
 - RS-232 operation using internal oscillator block (no external crystal required)

6.40.3 PIC18F2320

CLICK HERE ([see page 713](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution is 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution is 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Compatible 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time

- Dual Analog Comparators
- Addressable USART module:
 - RS-232 operation using internal oscillator block (no external crystal required)

6.40.4 PIC18F2220

CLICK HERE ( *see page 713*) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution is 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution is 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Compatible 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators
- Addressable USART module:
 - RS-232 operation using internal oscillator block (no external crystal required)

6.41 PIC18F1330 Family

6.41.1 PIC18F1330

CLICK HERE ( *see page 806*) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Up to 4 Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 4-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Up to 3 Analog Comparators
- Programmable Reference Voltage for Comparators
- Programmable, 15-Level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection

6.41.2 PIC18F1230

CLICK HERE ( *see page 806*) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Up to 4 Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 4-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep

- Up to 3 Analog Comparators
- Programmable Reference Voltage for Comparators
- Programmable, 15-Level Low-Voltage Detection

(LVD) module:

- Supports interrupt on Low-Voltage Detection

6.42 PIC18F1320 Family

6.42.1 PIC18F1320

CLICK HERE (see page 784) for the Peripheral Library Support Details for this Device

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupts
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
 - Capture is 16-bit, max resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max resolution 100 ns (TCY)
- Compatible 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Enhanced USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect

6.42.2 PIC18F1220

CLICK HERE (see page 784) for the Peripheral Library Support Details for this Device

Micro Controller Features:-**Peripheral Highlights:**

- High current sink/source 25 mA/25 mA
- Three external interrupts
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Capture is 16-bit, max resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max resolution 100 ns (TCY)
- Compatible 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Enhanced USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect

6.43 PIC18F8720 Family

6.43.1 PIC18F8720

CLICK HERE ( see page 722) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter

- Secondary oscillator clock option – Timer1/Timer3
- Five Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.43.2 PIC18F8620

CLICK HERE ([see page 722](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3

- Five Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.43.3 PIC18F8520

CLICK HERE ( *see page 722*) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Five Capture/Compare/PWM (CCP) modules:

- Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
- Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: PWM resolution is 1 to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.43.4 PIC18F6720

CLICK HERE (🔗 [see page 722](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Five Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)

- Compare is 16-bit, max. resolution 100 ns (TCY)
- PWM output: PWM resolution is 1 to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.43.5 PIC18F6620

CLICK HERE ( [see page 722](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Five Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)

- PWM output: PWM resolution is 1 to 10-bit
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection

(LVD) module:

- Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.43.6 PIC18F6520

CLICK HERE ([see page 722](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Timer4 module: 8-bit timer/counter
- Secondary oscillator clock option – Timer1/Timer3
- Five Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit

- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Two Addressable USART modules:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- 10-bit, up to 16-channel Analog-to-Digital Converter (A/D):
 - Conversion available during Sleep
- Programmable 16-level Low-Voltage Detection (LVD) module:
 - Supports interrupt on Low-Voltage Detection
- Programmable Brown-out Reset (PBOR)
- Dual analog comparators:
 - Programmable input/output configuration

6.44 PIC18C858 Family

6.44.1 PIC18C858

CLICK HERE ( [see page 731](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Up to 76 I/O with individual direction control
- Four external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules CCP pins can be configured as:

- Capture input: 16-bit, max resolution 6.25 ns
- Compare is 16-bit, max resolution 100 ns (TCY)
- PWM output: PWM resolution is 1- to 10-bit.

Max. PWM freq. @:8-bit resolution = 156 kHz

10-bit resolution = 39 kHz

- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI™ (Supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module: Supports Interrupt on Address bit Advanced Analog Features:
- 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
 - Up to 16 channels available
- Analog Comparator Module:
 - 2 Comparators
 - Programmable input and output multiplexing
- Comparator Voltage Reference Module
- Programmable Low Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
- Programmable Brown-out Reset (BOR)

6.44.2 PIC18C658

CLICK HERE ([see page 731](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Up to 76 I/O with individual direction control
- Four external interrupt pins
- Timer0module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1module: 16-bit timer/counter
- Timer2module: 8-bit timer/counter with 8-bit period register (time base for PWM)
- Timer3module: 16-bit timer/counter

- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules CCP pins can be configured as:
 - Capture input: 16-bit, max resolution 6.25 ns
 - Compare is 16-bit, max resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit.Max. PWM freq. @:8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI™ (Supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module: Supports Interrupt on Address bit Advanced Analog Features:
- 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
 - Up to 16 channels available
- Analog Comparator Module:
 - 2 Comparators
 - Programmable input and output multiplexing
- Comparator Voltage Reference Module
- Programmable Low Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
- Programmable Brown-out Reset (BOR)

6.45 PIC18C801 Family

6.45.1 PIC18C801

CLICK HERE ( see page 738) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA

- Up to 76 I/O with individual direction control
 - Four external interrupt pins
 - Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1 module: 16-bit timer/counter
 - Timer2 module: 8-bit timer/counter with 8-bit period register (time base for PWM)
 - Timer3 module: 16-bit timer/counter
 - Secondary oscillator clock option - Timer1/Timer3
 - Two Capture/Compare/PWM (CCP) modules CCP pins can be configured as:
 - Capture input: 16-bit, max resolution 6.25 ns
 - Compare is 16-bit, max resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit.
- Max. PWM freq. @:8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI™ (Supports all 4 SPI modes)
 - I2C™ Master and Slave mode
 - Addressable USART module: Supports Interrupt on Address bit Advanced Analog Features:
 - 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
 - Up to 16 channels available
 - Analog Comparator Module:
 - 2 Comparators
 - Programmable input and output multiplexing
 - Comparator Voltage Reference Module
 - Programmable Low Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
 - Programmable Brown-out Reset (BOR)

6.45.2 PIC18C601

CLICK HERE ([see page 738](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
 - Up to 76 I/O with individual direction control
 - Four external interrupt pins
 - Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1 module: 16-bit timer/counter
 - Timer2 module: 8-bit timer/counter with 8-bit period register (time base for PWM)
 - Timer3 module: 16-bit timer/counter
 - Secondary oscillator clock option - Timer1/Timer3
 - Two Capture/Compare/PWM (CCP) modules CCP pins can be configured as:
 - Capture input: 16-bit, max resolution 6.25 ns
 - Compare is 16-bit, max resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit.
- Max. PWM freq. @:8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI™ (Supports all 4 SPI modes)
 - I2C™ Master and Slave mode
 - Addressable USART module: Supports Interrupt on Address bit Advanced Analog Features:
 - 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
 - Up to 16 channels available
 - Analog Comparator Module:
 - 2 Comparators
 - Programmable input and output multiplexing
 - Comparator Voltage Reference Module
 - Programmable Low Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
 - Programmable Brown-out Reset (BOR)

6.46 PIC18F4539 Family

6.46.1 PIC18F4539

CLICK HERE ([see page 746](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two PWM modules:
 - Resolution is 1- to 10-bit,

Max. PWM freq. @ 8-bit resolution = 156 kHz

10-bit resolution = 39 kHz
- Single Phase Induction Motor Control kernel
 - Programmable Motor Control Technology (ProMPT™) provides open loop Variable Frequency (VF) control
 - User programmable Voltage vs. Frequency curve
 - Most suitable for shaded pole and permanent split capacitor type motors
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

6.46.2 PIC18F4439

CLICK HERE ([see page 746](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two PWM modules:
 - Resolution is 1- to 10-bit,
Max. PWM freq. @ 8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Single Phase Induction Motor Control kernel
 - Programmable Motor Control Technology (ProMPT™) provides open loop Variable Frequency (VF) control
 - User programmable Voltage vs. Frequency curve
 - Most suitable for shaded pole and permanent split capacitor type motors
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

6.46.3 PIC18F2539

CLICK HERE ([see page 746](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two PWM modules:
 - Resolution is 1- to 10-bit,
Max. PWM freq. @ 8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Single Phase Induction Motor Control kernel
 - Programmable Motor Control Technology (ProMPT™) provides open loop Variable Frequency (VF) control
 - User programmable Voltage vs. Frequency curve
 - Most suitable for shaded pole and permanent split capacitor type motors
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

6.46.4 PIC18F2439

CLICK HERE ([see page 746](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - DNL = ± 1 LSb, INL = ± 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two PWM modules:
 - Resolution is 1- to 10-bit,

Max. PWM freq. @ 8-bit resolution = 156 kHz

10-bit resolution = 39 kHz
- Single Phase Induction Motor Control kernel
 - Programmable Motor Control Technology (ProMPT™) provides open loop Variable Frequency (VF) control
 - User programmable Voltage vs. Frequency curve
 - Most suitable for shaded pole and permanent split capacitor type motors
- Master Synchronous Serial Port (MSSP) module with two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

6.47 PIC18F458 Family

6.47.1 PIC18F458

CLICK HERE ([see page 776](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter (A/D) module, up to 100 ksp/s
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.47.2 PIC18F448

CLICK HERE ([see page 776](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter (A/D) module, up to 100 ksps
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.47.3 PIC18F258

CLICK HERE ([see page 768](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-**Peripheral Highlights:**

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter (A/D) module, up to 100 ksps
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.47.4 PIC18F248

CLICK HERE ([see page 768](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- One Capture/Compare/PWM (CCP) module
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I2C™ Master and Slave modes
- Enhanced Addressable USART module
 - Supports RS-485, RS-232 and LIN 1.3
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 11-Channel Analog-to-Digital Converter (A/D) module, up to 100 ksps
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing

6.48 PIC18F452 Family

6.48.1 PIC18F452

CLICK HERE ([see page 753](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSB
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection

- Programmable Brown-out Reset (BOR)

6.48.2 PIC18F442

CLICK HERE ([see page 753](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.48.3 PIC18F252

CLICK HERE ([see page 753](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.48.4 PIC18F242

CLICK HERE ([see page 753](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.49 PIC18C452 Family

6.49.1 PIC18C452

CLICK HERE ([see page 761](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.49.2 PIC18C442

CLICK HERE ([see page 761](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.49.3 PIC18C252

CLICK HERE ([see page 761](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.49.4 PIC18C242

CLICK HERE ([see page 761](#)) *for the Peripheral Library Support Details for this Device*

Micro Controller Features:-

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16)
 - Compare is 16-bit, max. resolution 100 ns (TCY)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I2C™ Master and Slave mode
- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity = 1 LSb
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

6.50 PIC18F45K22 Family

6.50.1 PIC18F45K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.2 PIC18LF45K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.3 PIC18F44K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.4 PIC18LF44K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.5 PIC18F43K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.6 PIC18LF43K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.7 PIC18F46K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.8 PIC18LF46K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.9 PIC18F26K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.10 PIC18LF26K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.11 PIC18F25K22

CLICK HERE ( see page 827) for the *Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.12 PIC18LF25K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.13 PIC18F24K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.14 PIC18LF24K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.15 PIC18F23K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

6.50.16 PIC18LF23K22

CLICK HERE ([see page 827](#)) *for the Peripheral Library Support Details for this Device*

Peripheral Highlights:

- 1/4 CCP/ECCP modules
 - Seven Capture/Compare/PWM (CCP) modules
 - Three Enhanced Capture/Compare/PWM (ECCP) modules
- 5 8/16-bit timer/counter modules:
 - Timer0 – 8/16-bit timer/counter with 8-bit programmable prescaler
 - Timer1,3 – 16-bit timer/counter
 - Timer2,4 – 8-bit timer/counter
- Two analog comparators
- Configurable reference clock output
- Charge Time Measurement Unit (CTMU):
 - Capacitance measurement
 - Time measurement with 1-ns typical resolution
- High-current sink/source 25 mA/25 mA (PORTB and PORTC)
- Up to four external interrupts
- One Master Synchronous Serial Port (MSSP) modules:
 - 3/4-wire SPI (supports all four SPI modes)
 - I2C™ Master and Slave mode
- Two Enhanced Addressable USART modules:
 - LIN/J2602 support
 - Auto-Baud Detect (ABD)
- 12-bit A/D converter with up to 14 channels:
 - Auto-acquisition and Sleep operation

7 Device Modules

Device Families with Peripheral Library support

7.1 18F66K80 Family

PIC18F25K80 ([see page 27](#)) \ PIC18LF25K80 ([see page 33](#)) \ PIC18F26K80 ([see page 26](#)) \ PIC18LF26K80 ([see page 32](#))

PIC18F45K80 ([see page 25](#)) \ PIC18LF45K80 ([see page 31](#)) \ PIC18F46K80 ([see page 24](#)) \ PIC18LF46K80 ([see page 30](#))

PIC18F65K80 ([see page 23](#)) \ PIC18LF65K80 ([see page 29](#)) \ PIC18F66K80 ([see page 22](#)) \ PIC18LF66K80 ([see page 28](#))

7.1.1 ADC (66K80 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 876)
SetChanADC (see page 889)
SelChanConvADC (see page 896)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

```

* ENABLE_AN0_DIG()
* ENABLE_AN0_ANA()
* ENABLE_AN1_DIG()
* ENABLE_AN1_ANA()
* ENABLE_AN2_DIG()
* ENABLE_AN2_ANA()
* ENABLE_AN3_DIG()
* ENABLE_AN3_ANA()
* ENABLE_AN4_DIG()
* ENABLE_AN4_ANA()
* ENABLE_AN5_DIG()
* ENABLE_AN5_ANA()
* ENABLE_AN6_DIG()
* ENABLE_AN6_ANA()
* ENABLE_AN7_DIG()
* ENABLE_AN7_ANA()
* ENABLE_ALL_ANA_0_7()
* ENABLE_ALL_DIG_0_7()

* ENABLE_AN8_DIG()
* ENABLE_AN8_ANA()
* ENABLE_AN9_DIG()
* ENABLE_AN9_ANA()
* ENABLE_AN10_DIG()
* ENABLE_AN10_ANA()

```

ADC Examples (see page 898)

7.1.2 Analog Comparator (66K80 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 907)

Open_ancomp2 (see page 915)

Close_ancomp1 (see page 923)

Close_ancomp2 (see page 923)

Macros

Config_CVREF

Comparator Examples (see page 924)

7.1.3 CTMU (66K80 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.1.4 Input Capture (66K80 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECapture1 (see page 969)
OpenCapture2 (see page 956)
OpenCapture3 (see page 958)
OpenCapture4 (see page 962)
OpenCapture5 (see page 965)
ReadECapture1 (see page 986)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseECapture1 (see page 990)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.1.5 Output Compare (66K80 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1006)
OpenCompare2 (see page 994)
OpenCompare3 (see page 996)
OpenCompare4 (see page 999)
OpenCompare5 (see page 1002)
CloseECompare1 (see page 1024)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)

CloseCompare5 ([see page 1022](#))

Output Compare Examples ([see page 1025](#))

7.1.6 PWM (66K80 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1036)
OpenPWM2 (see page 1030)
OpenPWM3 (see page 1028)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
SetDCEPWM1 (see page 1054)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputEPWM1 (see page 1045)
CloseEPWM1 (see page 1059)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.1.7 I2C (66K80 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.1.8 I2C EEPROM (66K80 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddRead (see page 1077)
--

EEPageWrite (see page 1081)

EERandomRead (see page 1086)
--

EESequentialRead (see page 1083)
--

I2C EEPROM Examples ([see page 1088](#))

7.1.9 EEP (66K80 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)
--

Busy_eep (see page 1062)
--

EEP Examples ([see page 1062](#))

7.1.10 IO Ports (66K80 Set)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)
--

OpenRB1INT (see page 1090)
--

OpenRB2INT (see page 1091)
--

OpenRB3INT (see page 1092)
--

Macros

EnablePullups

DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.1.11 MWIRE (66K80 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.1.12 SPI (66K80 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.1.13 SW_RTCC (66K80 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.1.14 Timers (66K80 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers

- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1121)

ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)

CloseTimer2 (see page 1144)

OpenTimer3 (see page 1125)

ReadTimer3 (see page 1138)

WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

OpenTimer4 (see page 1130)

CloseTimer4 (see page 1144)

Macros

WriteTimer2

ReadTimer2

WriteTimer4

ReadTimer4

Timers Examples (see page 1146)

7.1.15 Flash (66K80 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash

- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.1.16 USART (66K80 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.1.17 Deep Sleep (66K80 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions
ULPWakeUpEnable (see page 931)

7.2 18F47J53 Family

PIC18F26J53 / PIC18F27J53 / PIC18F46J53 / PIC18F47J53

PIC18LF26J53 / PIC18LF27J53 / PIC18LF46J53 / PIC18LF47J53

PIC18F26J13 / PIC18F27J13 / PIC18F46J13 / PIC18F47J13

PIC18LF26J13 / PIC18LF27J13 / PIC18LF46J13 / PIC18LF47J13

7.2.1 PIC18F2xJ53 Set

PIC18F26J53 (see page 37) / PIC18F27J53 (see page 36)

PIC18LF26J53 / PIC18LF27J53

PIC18F26J13 (see page 41) / PIC18F27J13 (see page 42)

PIC18LF26J13 / PIC18LF27J13

7.2.1.1 ADC (2xJ53 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 873)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CALIB
ADC_NO_CALIB

ADC Examples (see page 898)

7.2.1.2 Analog Comparator (2xJ53 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

FunctionsOpen_ancomp1 ([see page 918](#))Open_ancomp2 ([see page 920](#))Open_ancomp3 ([see page 921](#))Close_ancomp1 ([see page 923](#))Close_ancomp2 ([see page 923](#))Close_ancomp3 ([see page 923](#))**Macros**

Config_CVREF

Comparator Examples ([see page 924](#))

7.2.1.3 CTMU (2xJ53 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

FunctionsOpenCTMU ([see page 925](#))CurrentControlCTMU ([see page 927](#))CloseCTMU ([see page 927](#))**Macros**

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples ([see page 928](#))

7.2.1.4 PPS (2xJ53 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros
iPPSInput (see page 947)
PPSInput (see page 948)
iPPSOutput (see page 950)
PPSOutput (see page 951)

PPS Examples (see page 953)

7.2.1.5 Deep Sleep (2xJ53 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions
DeepSleepWakeUpSource (see page 930)
GotoDeepSleep (see page 930)
IsResetFromDeepSleep (see page 931)
ReadDSGPR (see page 931)
ULPWakeUpEnable (see page 931)

Macros
ReleaseDeepSleep
Write_DSGPR

Deep Sleep Examples (see page 932)

7.2.1.6 Input Capture (2xJ53 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECapture1 (see page 968)
OpenECapture2 (see page 971)
OpenECapture3 (see page 972)
OpenCapture4 (see page 960)
OpenCapture5 (see page 964)
OpenCapture6 (see page 973)
OpenCapture7 (see page 974)
OpenCapture8 (see page 976)
ReadECapture1 (see page 986)
ReadECapture2 (see page 986)
ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)

Input Capture Examples (see page 991)

7.2.1.7 Output Compare (2xJ53 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1005)
OpenECompare2 (see page 1008)

OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)
OpenCompare8 (see page 1013)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)

Output Compare Examples ([see page 1025](#))

7.2.1.8 PWM (2xJ53 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1037)
OpenEPWM2 (see page 1038)
OpenEPWM3 (see page 1039)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
OpenPWM6 (see page 1032)
OpenPWM7 (see page 1033)
OpenPWM8 (see page 1034)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetDCPWM6 (see page 1050)
SetDCPWM7 (see page 1051)
SetDCPWM8 (see page 1052)

SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)
SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)
CloseEPWM3 (see page 1059)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)
ClosePWM6 (see page 1057)
ClosePWM7 (see page 1057)
ClosePWM8 (see page 1058)

PWM Examples ([see page 1060](#))

7.2.1.9 I2C EEPROM (2xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples ([see page 1088](#))

7.2.1.10 SPI (2xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.2.1.11 IO Ports (2xJ53 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.2.1.12 RTCC (2xJ53 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccIs2ndHalfSecond
mRtccIsAlrmEnabled
mRtccIsOn
mRtccIsSync
mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync

mRtccWrOff

RTCC Examples ([see page 1104](#))

7.2.1.13 SW_RTCC (2xJ53 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.2.1.14 Timers (2xJ53 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)

OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)
OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)
OpenTimer8 (see page 1134)
CloseTimer8 (see page 1145)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6
WriteTimer8
ReadTimer8

Timers Examples (see page 1146)

7.2.1.15 USART (2xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)

Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.2.1.16 MWIRE (2xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMWire1
getcMWire1
putcMWire1
DataRdyMWire1
CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples (see page 1172)

7.2.1.17 Flash (2xJ53 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.2.2 PIC18F4xJ53 Set

PIC18F46J53 (see page 35) / PIC18F47J53 (see page 34)

PIC18LF46J53 / PIC18LF47J53

PIC18F46J13 (see page 40) / PIC18F47J13 (see page 39)

PIC18LF46J13 / PIC18LF47J13

7.2.2.1 ADC (4xJ53 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC ([see page 873](#))

SetChanADC ([see page 882](#))

SelChanConvADC ([see page 889](#))

ConvertADC ([see page 896](#))

BusyADC ([see page 897](#))

ReadADC ([see page 897](#))

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples ([see page 898](#))

7.2.2.2 Analog Comparator (4xJ53 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 ([see page 918](#))

Open_ancomp2 ([see page 920](#))

Open_ancomp3 ([see page 921](#))

Close_ancomp1 ([see page 923](#))

Close_ancomp2 ([see page 923](#))

Close_ancomp3 (see page 923)

Macros

Config_CVREF

Comparator Examples (see page 924)

7.2.2.3 CTMU (4xJ53 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.2.2.4 PPS (4xJ53 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros

iPPSInput (see page 947)

PPSInput (see page 948)

iPPSOutput (see page 950)

PPSOutput (see page 951)

PPS Examples (see page 953)

7.2.2.5 Deep Sleep (4xJ53 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions

DeepSleepWakeUpSource (see page 930)

GotoDeepSleep (see page 930)

IsResetFromDeepSleep (see page 931)

ReadDSGPR (see page 931)

ULPWakeUpEnable (see page 931)

Macros

ReleaseDeepSleep

Write_DSGPR

Deep Sleep Examples (see page 932)

7.2.2.6 Input Capture (4xJ53 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenECapture1 (see page 968)

OpenECapture2 (see page 971)

OpenECapture3 (see page 972)

OpenCapture4 (see page 960)
OpenCapture5 (see page 964)
OpenCapture6 (see page 973)
OpenCapture7 (see page 974)
OpenCapture8 (see page 976)
ReadECapture1 (see page 986)
ReadECapture2 (see page 986)
ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)

Input Capture Examples (see page 991)

7.2.2.7 Output Compare (4xJ53 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1005)
OpenECompare2 (see page 1008)
OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)

OpenCompare8 (see page 1013)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)

Output Compare Examples (see page 1025)

7.2.2.8 PWM (4xJ53 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1037)
OpenEPWM2 (see page 1038)
OpenEPWM3 (see page 1039)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
OpenPWM6 (see page 1032)
OpenPWM7 (see page 1033)
OpenPWM8 (see page 1034)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetDCPWM6 (see page 1050)
SetDCPWM7 (see page 1051)
SetDCPWM8 (see page 1052)
SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)
SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)

CloseEPWM3 (see page 1059)

ClosePWM4 (see page 1057)

ClosePWM5 (see page 1057)

ClosePWM6 (see page 1057)

ClosePWM7 (see page 1057)

ClosePWM8 (see page 1058)

PWM Examples (see page 1060)

7.2.2.9 I2C (4xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 (see page 1064)

WriteI2C1 (see page 1066)

ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

OpenI2C2 (see page 1065)

WriteI2C2 (see page 1067)

ReadI2C2 (see page 1068)

putsI2C2 (see page 1069)

getsI2C2 (see page 1071)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1
IdleI2C1
CloseI2C1
putI2C1
getI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.2.2.10 I2C EEPROM (4xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)

EEAckPolling2 (see page 1076)

EEByteWrite2 (see page 1080)

EECurrentAddRead2 (see page 1078)

EEPageWrite2 (see page 1083)

EERandomRead2 (see page 1087)

EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.2.2.11 SPI (4xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2
DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.2.2.12 IO Ports (4xJ53 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.2.2.13 RTCC (4xJ53 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)

RtccWriteAlrmTimeDate (see page 1101)

RtccWriteAlrmTime (see page 1101)

RtccWriteAlrmDate (see page 1102)

RtccSetChimeEnable (see page 1103)

RtccSetCalibration (see page 1103)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRpt (see page 1098)

RtccReadTimeDate (see page 1097)

RtccReadTime (see page 1097)

RtccReadDate (see page 1096)

RtccReadAlrmTimeDate (see page 1096)

RtccReadAlrmTime (see page 1096)

RtccReadAlrmDate (see page 1095)

RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit

mRtccAlrmDisable

mRtccAlrmEnable

mRtccClearAlrmPtr

mRtccClearRtcPtr

mRtccGetAlarmRpt

mRtccGetAlarmRptCount

mRtccGetCalibration

mRtccGetChimeEnable

mRtccGetClockOe

mRtccIs2ndHalfSecond

mRtccIsAlrmEnabled

mRtccIsOn

mRtccIsSync

mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.2.2.14 SW_RTCC (4xJ53 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.2.2.15 Timers (4xJ53 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)
OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)
OpenTimer8 (see page 1134)
CloseTimer8 (see page 1145)

Macros

WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6
WriteTimer8
ReadTimer8

Timers Examples (see page 1146)

7.2.2.16 USART (4xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)

Write1USART (see page 1154)

baud1USART (see page 1155)

gets1USART (see page 1159)

putrs1USART (see page 1160)

puts1USART (see page 1162)

Read1USART (see page 1163)

Open2USART (see page 1150)

Write2USART (see page 1155)

baud2USART (see page 1157)

gets2USART (see page 1159)

putrs2USART (see page 1161)

puts2USART (see page 1162)

Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART

Busy2USART

putc2USART

getc2USART

USART Examples (see page 1164)

7.2.2.17 MWIRE (4xJ53 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)

ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.2.2.18 Flash (4xJ53 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.2.2.19 PMP (4xJ53 Family)

This peripheral library module:

- Is a parallel 8-bit I/O module. PMP stands for Parallel Master Port module.
- Is specifically designed to communicate with a wide variety of parallel devices, such as communications peripherals, LCDs, external memory devices and microcontrollers.
- Is highly configurable, since the interface to parallel peripherals varies significantly.

Functions

PMPOpen (see page 1187)
PMPIsBufferNEmpty (see page 1189)
PMPIsBufferNFull (see page 1189)
PMPMasterRead (see page 1188)
PMPMasterWrite (see page 1189)
PMPSetAddress (see page 1187)
PMPSlaveReadBufferN (see page 1190)
PMPSlaveReadBuffers (see page 1190)
PMPSlaveWriteBufferN (see page 1191)
PMPSlaveWriteBuffers (see page 1191)
PMPClose (see page 1192)

Macros

mPMPIsBufferFull
mPMPSetWriteStrobePolarity
mPMPSetWriteStrobeEnable
mPMPSetWaitEnd
mPMPSetWaitMiddle
mPMPIsBufferOverflow
mPMPSetWaitBegin
mPMPSetReadStrobePolarity
mPMPSetReadStrobeEnable
mPMPSetPortPins
mPMPSetPortMode
mPMPSetPortEnable
mPMPSetAddrLatchPolarity
mPMPSetInterruptMode
mPMPClearBufferOverflow
mPMPSetIdle

mPMP_Clear_Intr_Status_Bit
mPMPSetInterruptPriority
mPMPSetInterruptEnable
mPMPSetDataMode
mPMPClearBufferUnderflow
mPMPSetChipSelect2Polarity
mPMPSetChipSelect2Enable
mPMPSetChipSelect1Polarity
mPMPSetChipSelect1Enable
mPMPSetChipSelectMode
mPMPSetByteEnable
mPMPSetByteEnablePolarity
mPMPIsBufferEmpty
mPMPIsBufferUnderflow
mPMPSetAddrIncMode
mPMPSetAddrMux

PMP Examples (see page 1192)

7.3 18F97J72 Family

PIC18F86J72 / PIC18F87J72 / PIC18F96J72 / PIC18F97J72

7.3.1 ADC (97J72 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 862)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples (see page 898)

7.3.2 Analog Comparator (97J72 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.3.3 CTMU (97J72 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2
Disbl_CTMUEdge1
Disbl_CTMUEdge2
CTMUEdge1_Status
CTMUEdge2_Status
CTMU_INT_ENABLE
CTMU_INT_DISABLE
CTMU_Clear_Intr_Status_Bit
CTMU_Intr_Status

CTMU Examples (see page 928)

7.3.4 Input Capture (97J72 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples (see page 991)

7.3.5 Output Compare (97J72 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)

CloseCompare1 (see page 1021)

CloseCompare2 (see page 1022)

Output Compare Examples (see page 1025)

7.3.6 PWM (97J72 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

OpenPWM2 (see page 1027)

SetDCPWM1 (see page 1047)

SetDCPWM2 (see page 1048)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.3.7 I2C (97J72 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.3.8 I2C EEPROM (97J72 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.3.9 IO Ports (97J72 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.3.10 RTCC (97J72 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)

RtccWriteAlarmTimeDate (see page 1101)

RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccls2ndHalfSecond
mRtcclsAlrmEnabled
mRtcclsOn
mRtcclsSync
mRtcclsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples ([↗](#) see page 1104)

7.3.11 SW_RTCC (97J72 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([↗](#) see page 1106)

update_RTCC ([↗](#) see page 1106)

Close_RTCC ([↗](#) see page 1106)

SW_RTCC Examples ([↗](#) see page 1107)

7.3.12 SPI (97J72 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI ([↗](#) see page 1108)

WriteSPI ([↗](#) see page 1111)

ReadSPI ([↗](#) see page 1113)

putsSPI ([↗](#) see page 1114)

getsSPI ([↗](#) see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.3.13 Timers (97J72 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1120)

ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)

CloseTimer2 (see page 1144)

OpenTimer3 (see page 1124)

ReadTimer3 (see page 1138)

WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.3.14 USART (97J72 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.3.15 MWIRE (97J72 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.3.16 Flash (97J72 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))

EraseFlash ([see page 1196](#))

WriteBlockFlash ([see page 1198](#))

WriteWordFlash ([see page 1202](#))

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.4 18F87K90/22 Family

PIC18F87K90 (see page 58) \ PIC18F86K95 (see page 59) \ PIC18F86K90 (see page 60) \ PIC18F85K90 (see page 61)

PIC18F67K90 (see page 62) \ PIC18F66K95 (see page 63) \ PIC18F66K90 (see page 64) \ PIC18F65K90 (see page 65)

PIC18F87K22 (see page 44) \ PIC18F86K27 (see page 46) \ PIC18F86K22 (see page 45) \ PIC18F85K22 (see page 47)

PIC18F67K22 (see page 48) \ PIC18F66K27 (see page 50) \ PIC18F66K22 (see page 49) \ PIC18F65K22 (see page 51)

7.4.1 PIC18F6xK90 Set

PIC18F67K90 (see page 62) \ PIC18F66K95 (see page 63) \ PIC18F66K90 (see page 64) \ PIC18F65K90 (see page 65)

7.4.1.1 PIC18F6xK90_non_32

PIC18F67K90 (see page 62) \ PIC18F66K95 (see page 63) \ PIC18F66K90 (see page 64)

7.4.1.1.1 ADC (6xK90 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 867)
SetChanADC (see page 887)
SelChanConvADC (see page 894)

ConvertADC (🔗 see page 896)

BusyADC (🔗 see page 897)

ReadADC (🔗 see page 897)

CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

```
* ENABLE_AN0_DIG()
* ENABLE_AN0_ANA()
* ENABLE_AN1_DIG()
* ENABLE_AN1_ANA()
* ENABLE_AN2_DIG()
* ENABLE_AN2_ANA()
* ENABLE_AN3_DIG()
* ENABLE_AN3_ANA()
* ENABLE_AN4_DIG()
* ENABLE_AN4_ANA()
* ENABLE_AN5_DIG()
* ENABLE_AN5_ANA()
* ENABLE_AN6_DIG()
* ENABLE_AN6_ANA()
* ENABLE_AN7_DIG()
* ENABLE_AN7_ANA()
* ENABLE_ALL_ANA_0_7()
* ENABLE_ALL_DIG_0_7()

* ENABLE_AN8_DIG()
* ENABLE_AN8_ANA()
* ENABLE_AN9_DIG()
* ENABLE_AN9_ANA()
* ENABLE_AN10_DIG()
* ENABLE_AN10_ANA()
* ENABLE_AN11_DIG()
* ENABLE_AN11_ANA()
* ENABLE_ALL_ANA_8_15()
* ENABLE_ALL_DIG_8_15()

* ENABLE_AN16_DIG()
* ENABLE_AN16_ANA()
* ENABLE_AN17_DIG()
* ENABLE_AN17_ANA()
* ENABLE_AN18_DIG()
* ENABLE_AN18_ANA()
* ENABLE_AN19_DIG()
* ENABLE_AN19_ANA()
* ENABLE_ALL_ANA_16_23()
* ENABLE_ALL_DIG_16_23()
```

ADC Examples (see page 898)

7.4.1.1.2 Analog Comparator (6xK50 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 907)

Open_ancomp2 (see page 915)

Open_ancomp3 (see page 917)

Close_ancomp1 (see page 923)

Close_ancomp2 (see page 923)

Close_ancomp3 (see page 923)

Macros

Config_CVREF

Comparator Examples (see page 924)

7.4.1.1.3 CTMU (6xK90 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.4.1.1.4 Input Capture (6xK90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECapture1 (see page 967)
OpenECapture2 (see page 970)
OpenECapture3 (see page 972)
OpenCapture4 (see page 960)
OpenCapture5 (see page 964)
OpenCapture6 (see page 973)
OpenCapture7 (see page 974)
OpenCapture8 (see page 975)
OpenCapture9 (see page 977)
OpenCapture10 (see page 978)
ReadECapture1 (see page 986)
ReadECapture2 (see page 986)
ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
ReadCapture9 (see page 986)
ReadCapture10 (see page 986)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)
CloseCapture9 (see page 989)
CloseCapture10 (see page 990)

Input Capture Examples ([see page 991](#))

7.4.1.1.5 Output Compare (6xK90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1004)
OpenECompare2 (see page 1007)
OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)
OpenCompare8 (see page 1012)
OpenCompare9 (see page 1014)
OpenCompare10 (see page 1015)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)
CloseCompare9 (see page 1024)
CloseCompare10 (see page 1024)

Output Compare Examples ([see page 1025](#))

7.4.1.1.6 PWM (6xK90 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1036)
OpenEPWM2 (see page 1038)

OpenEPWM3 (see page 1039)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
OpenPWM6 (see page 1032)
OpenPWM7 (see page 1033)
OpenPWM8 (see page 1033)
OpenPWM9 (see page 1035)
OpenPWM10 (see page 1035)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetDCPWM6 (see page 1050)
SetDCPWM7 (see page 1051)
SetDCPWM8 (see page 1052)
SetDCPWM9 (see page 1052)
SetDCPWM10 (see page 1053)
SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)
SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)
CloseEPWM3 (see page 1059)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)
ClosePWM6 (see page 1057)
ClosePWM7 (see page 1057)
ClosePWM8 (see page 1058)
ClosePWM9 (see page 1058)
ClosePWM10 (see page 1058)

PWM Examples (see page 1060)

7.4.1.1.7 I2C (6xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 (see page 1064)

Writel2C1 (see page 1066)

ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

OpenI2C2 (see page 1065)

Writel2C2 (see page 1067)

ReadI2C2 (see page 1068)

putsI2C2 (see page 1069)

getsI2C2 (see page 1071)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit

I2C2_Intr_Status

StopI2C2

StartI2C2

RestartI2C2

NotAckI2C2

AckI2C2

DataRdyI2C2

IdleI2C2

Closel2C2
putcl2C2
getcl2C2

I2C Examples (see page 1072)

7.4.1.1.8 I2C EEPROM (6xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.4.1.1.9 EEP (6xK90 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)

Read_b_eep (see page 1061)
--

Busy_eep (see page 1062)
--

EEP Examples ([see page 1062](#))

7.4.1.1.10 IO Ports (6xK90 Set)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)
--

OpenRB1INT (see page 1090)
--

OpenRB2INT (see page 1091)
--

OpenRB3INT (see page 1092)
--

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.4.1.1.11 MWIRE (6xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)
--

ReadMwire1 (see page 1170)
--

WriteMWire1 (see page 1169)
getsMWire1 (see page 1167)
OpenMWire2 (see page 1166)
ReadMWire2 (see page 1171)
WriteMWire2 (see page 1169)
getsMWire2 (see page 1168)

Macros
CloseMWire1
getcMWire1
putcMWire1
DataRdyMWire1
CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples (see page 1172)

7.4.1.1.12 RTCC (6xK90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)

RtccReadDate (see page 1096)

RtccReadAlrmTimeDate (see page 1096)

RtccReadAlrmTime (see page 1096)

RtccReadAlrmDate (see page 1095)

RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit

mRtccAlrmDisable

mRtccAlrmEnable

mRtccClearAlrmPtr

mRtccClearRtcPtr

mRtccGetAlarmRpt

mRtccGetAlarmRptCount

mRtccGetCalibration

mRtccGetChimeEnable

mRtccGetClockOe

mRtccls2ndHalfSecond

mRtcclsAlrmEnabled

mRtcclsOn

mRtcclsSync

mRtcclsWrEn

mRtccOff

mRtccOn

mRtccSetAlrmPtr

mRtccSetClockOe

mRtccSetInt

mRtccSetRtcPtr

mRtccWaitSync

mRtccWrOff

RTCC Examples (see page 1104)

7.4.1.1.13 SW_RTCC (6xK90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.4.1.1.14 SPI (6xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.4.1.1.15 USART (6xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART

Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.4.1.1.16 Timers (6xK90 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5/7 are 16 bit timers/counters
- Timer2/4/6/8/10/12 are 8 bit timers

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)
OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)
OpenTimer7 (see page 1132)

ReadTimer7 (see page 1139)
WriteTimer7 (see page 1141)
CloseTimer7 (see page 1145)
OpenTimer8 (see page 1134)
CloseTimer8 (see page 1145)
OpenTimer10 (see page 1135)
CloseTimer10 (see page 1146)
OpenTimer12 (see page 1136)
CloseTimer12 (see page 1146)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6
WriteTimer8
ReadTimer8
WriteTimer10
ReadTimer10
WriteTimer12
ReadTimer12

Timers Examples (see page 1146)

7.4.1.1.17 Flash (6xK90)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.4.1.2 PIC18F6xK90_32

PIC18F65K90 (see page 65)

7.4.1.2.1 ADC (65K90 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 867)
SetChanADC (see page 887)
SelChanConvADC (see page 894)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

```
* ENABLE_AN0_DIG()
* ENABLE_AN0_ANA()
* ENABLE_AN1_DIG()
* ENABLE_AN1_ANA()
* ENABLE_AN2_DIG()
* ENABLE_AN2_ANA()
* ENABLE_AN3_DIG()
* ENABLE_AN3_ANA()
* ENABLE_AN4_DIG()
* ENABLE_AN4_ANA()
* ENABLE_AN5_DIG()
* ENABLE_AN5_ANA()
* ENABLE_AN6_DIG()
* ENABLE_AN6_ANA()
* ENABLE_AN7_DIG()
* ENABLE_AN7_ANA()
* ENABLE_ALL_ANA_0_7()
* ENABLE_ALL_DIG_0_7()

* ENABLE_AN8_DIG()
* ENABLE_AN8_ANA()
* ENABLE_AN9_DIG()
* ENABLE_AN9_ANA()
* ENABLE_AN10_DIG()
* ENABLE_AN10_ANA()
* ENABLE_AN11_DIG()
* ENABLE_AN11_ANA()
* ENABLE_ALL_ANA_8_15()
* ENABLE_ALL_DIG_8_15()

* ENABLE_AN16_DIG()
* ENABLE_AN16_ANA()
* ENABLE_AN17_DIG()
* ENABLE_AN17_ANA()
* ENABLE_AN18_DIG()
* ENABLE_AN18_ANA()
* ENABLE_AN19_DIG()
* ENABLE_AN19_ANA()
* ENABLE_ALL_ANA_16_23()
* ENABLE_ALL_DIG_16_23()
```

ADC Examples (see page 898)

7.4.1.2.2 Analog Comparator (65K90 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 907)

Open_ancomp2 (see page 915)

Open_ancomp3 (see page 917)

Close_ancomp1 (see page 923)

Close_ancomp2 (see page 923)

Close_ancomp3 (see page 923)

Macros

Config_CVREF

Comparator Examples (see page 924)

7.4.1.2.3 CTMU (65K90 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.4.1.2.4 Input Capture (65K90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECapture1 (see page 968)
OpenECapture2 (see page 971)
OpenECapture3 (see page 972)
OpenCapture4 (see page 960)
OpenCapture5 (see page 964)
OpenCapture6 (see page 973)
OpenCapture7 (see page 974)
OpenCapture8 (see page 976)
ReadECapture1 (see page 986)
ReadECapture2 (see page 986)
ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)

Input Capture Examples ([see page 991](#))

7.4.1.2.5 Output Compare (65K90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available

- Configurable interrupt generation

Functions
OpenECompare1 (see page 1005)
OpenECompare2 (see page 1008)
OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)
OpenCompare8 (see page 1013)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)

Output Compare Examples ([see page 1025](#))

7.4.1.2.6 PWM (65K90 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1037)
OpenEPWM2 (see page 1038)
OpenEPWM3 (see page 1039)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
OpenPWM6 (see page 1032)
OpenPWM7 (see page 1033)
OpenPWM8 (see page 1034)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)

SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetDCPWM6 (see page 1050)
SetDCPWM7 (see page 1051)
SetDCPWM8 (see page 1052)
SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)
SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)
CloseEPWM3 (see page 1059)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)
ClosePWM6 (see page 1057)
ClosePWM7 (see page 1057)
ClosePWM8 (see page 1058)

PWM Examples (see page 1060)

7.4.1.2.7 I2C (65K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putSI2C1 (see page 1069)
getSI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putSI2C2 (see page 1069)
getSI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.4.1.2.8 I2C EEPROM (65K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.4.1.2.9 EEP (65K90 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.4.1.2.10 IO Ports (65K90 Set)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)
--

OpenRB1INT (see page 1090)
--

OpenRB2INT (see page 1091)
--

OpenRB3INT (see page 1092)
--

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.4.1.2.11 MWIRE (65K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)
--

ReadMwire1 (see page 1170)
--

WriteMwire1 (see page 1169)

getsMwire1 (see page 1167)
--

OpenMwire2 (see page 1166)
--

ReadMwire2 (see page 1171)
--

WriteMwire2 (see page 1169)

getsMwire2 (see page 1168)
--

Macros

CloseMwire1

getcMwire1

putcMwire1

DataRdyMwire1

CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples (see page 1172)

7.4.1.2.12 RTCC (65K90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr

mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccIs2ndHalfSecond
mRtccIsAlrmEnabled
mRtccIsOn
mRtccIsSync
mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples ([see page 1104](#))

7.4.1.2.13 SW_RTCC (65K90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.4.1.2.14 SPI (65K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.4.1.2.15 USART (65K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.4.1.2.16 Timers (65K90 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)
OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)
OpenTimer8 (see page 1134)
CloseTimer8 (see page 1145)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6
WriteTimer8
ReadTimer8

Timers Examples (see page 1146)

7.4.1.2.17 Flash (65K90 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.4.2 PIC18F8xK90 Set

PIC18F87K90 (see page 58) \ PIC18F86K95 (see page 59) \ PIC18F86K90 (see page 60) \ PIC18F85K90 (see page 61)

7.4.2.1 PIC18F8xK90_non_32

PIC18F87K90 (see page 58) \ PIC18F86K95 (see page 59) \ PIC18F86K90 (see page 60)

7.4.2.1.1 ADC (8xK90 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 870)

SetChanADC (see page 887)
SelChanConvADC (see page 895)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

7.4.2.1.2 Analog Comparator (8xK90 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 907)
Open_ancomp2 (see page 915)
Open_ancomp3 (see page 917)
Close_ancomp1 (see page 923)
Close_ancomp2 (see page 923)
Close_ancomp3 (see page 923)

Macros
Config_CVREF

Comparator Examples ([see page 924](#))

7.4.2.1.3 CTMU (8xK90 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU ([see page 925](#))
 CurrentControlCTMU ([see page 927](#))
 CloseCTMU ([see page 927](#))

Macros

Enbl_CTMUEdge1
 Enbl_CTMUEdge2
 Disbl_CTMUEdge1
 Disbl_CTMUEdge2
 CTMUEdge1_Status
 CTMUEdge2_Status
 CTMU_INT_ENABLE
 CTMU_INT_DISABLE
 CTMU_Clear_Intr_Status_Bit
 CTMU_Intr_Status

CTMU Examples ([see page 928](#))

7.4.2.1.4 Input Capture (8xK90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenECapture1 ([see page 967](#))
 OpenECapture2 ([see page 970](#))
 OpenECapture3 ([see page 972](#))
 OpenCapture4 ([see page 960](#))
 OpenCapture5 ([see page 964](#))
 OpenCapture6 ([see page 973](#))
 OpenCapture7 ([see page 974](#))
 OpenCapture8 ([see page 975](#))
 OpenCapture9 ([see page 977](#))
 OpenCapture10 ([see page 978](#))
 ReadECapture1 ([see page 986](#))
 ReadECapture2 ([see page 986](#))

ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
ReadCapture9 (see page 986)
ReadCapture10 (see page 986)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)
CloseCapture9 (see page 989)
CloseCapture10 (see page 990)

Input Capture Examples ([see page 991](#))

7.4.2.1.5 Output Compare (8xK90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1004)
OpenECompare2 (see page 1007)
OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)
OpenCompare8 (see page 1012)
OpenCompare9 (see page 1014)
OpenCompare10 (see page 1015)
CloseECompare1 (see page 1024)

CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)
CloseCompare9 (see page 1024)
CloseCompare10 (see page 1024)

Output Compare Examples (see page 1025)

7.4.2.1.6 PWM (8xK90 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1036)
OpenEPWM2 (see page 1038)
OpenEPWM3 (see page 1039)
OpenPWM4 (see page 1029)
OpenPWM5 (see page 1029)
OpenPWM6 (see page 1032)
OpenPWM7 (see page 1033)
OpenPWM8 (see page 1033)
OpenPWM9 (see page 1035)
OpenPWM10 (see page 1035)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetDCPWM6 (see page 1050)
SetDCPWM7 (see page 1051)
SetDCPWM8 (see page 1052)
SetDCPWM9 (see page 1052)
SetDCPWM10 (see page 1053)
SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)

SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)
CloseEPWM3 (see page 1059)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)
ClosePWM6 (see page 1057)
ClosePWM7 (see page 1057)
ClosePWM8 (see page 1058)
ClosePWM9 (see page 1058)
ClosePWM10 (see page 1058)

PWM Examples (see page 1060)

7.4.2.1.7 I2C (8xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
Writel2C1 (see page 1066)
ReadI2C1 (see page 1067)
putsI2C1 (see page 1069)
getsI2C1 (see page 1070)
OpenI2C2 (see page 1065)
Writel2C2 (see page 1067)
ReadI2C2 (see page 1068)
putsI2C2 (see page 1069)
getsI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status

StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putI2C1
getI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.4.2.1.8 I2C EEPROM (8xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)

EEAckPolling1 (see page 1076)

EEByteWrite1 (see page 1080)

EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.4.2.1.9 EEP (8xK90 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.4.2.1.10 IO Ports (8xK90 Set)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.4.2.1.11 MWIRE (8xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.4.2.1.12 RTCC (8xK90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)

RtccWriteAlrmTimeDate (see page 1101)

RtccWriteAlrmTime (see page 1101)

RtccWriteAlrmDate (see page 1102)

RtccSetChimeEnable (see page 1103)

RtccSetCalibration (see page 1103)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRpt (see page 1098)

RtccReadTimeDate (see page 1097)

RtccReadTime (see page 1097)

RtccReadDate (see page 1096)

RtccReadAlrmTimeDate (see page 1096)

RtccReadAlrmTime (see page 1096)

RtccReadAlrmDate (see page 1095)

RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit

mRtccAlrmDisable

mRtccAlrmEnable

mRtccClearAlrmPtr

mRtccClearRtcPtr

mRtccGetAlarmRpt

mRtccGetAlarmRptCount

mRtccGetCalibration

mRtccGetChimeEnable

mRtccGetClockOe

mRtccls2ndHalfSecond
mRtcclsAlrmEnabled
mRtcclsOn
mRtcclsSync
mRtcclsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.4.2.1.13 SW_RTCC (8xK90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.4.2.1.14 SPI (8xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI1 (see page 1109)
WriteSPI1 (see page 1112)
ReadSPI1 (see page 1113)
putsSPI1 (see page 1114)
getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)
WriteSPI2 (see page 1112)
ReadSPI2 (see page 1113)
putsSPI2 (see page 1114)
getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.4.2.1.15 USART (8xK90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.4.2.1.16 Timers (8xK90 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5/7 are 16 bit timers/counters
- Timer2/4/6/8/10/12 are 8 bit timers

FunctionsOpenTimer0 ([see page 1119](#))ReadTimer0 ([see page 1137](#))WriteTimer0 ([see page 1139](#))CloseTimer0 ([see page 1143](#))OpenTimer1 ([see page 1121](#))ReadTimer1 ([see page 1138](#))WriteTimer1 ([see page 1140](#))CloseTimer1 ([see page 1143](#))OpenTimer2 ([see page 1123](#))CloseTimer2 ([see page 1144](#))OpenTimer3 ([see page 1125](#))ReadTimer3 ([see page 1138](#))WriteTimer3 ([see page 1140](#))CloseTimer3 ([see page 1144](#))OpenTimer4 ([see page 1130](#))CloseTimer4 ([see page 1144](#))OpenTimer5 ([see page 1129](#))ReadTimer5 ([see page 1139](#))WriteTimer5 ([see page 1140](#))CloseTimer5 ([see page 1144](#))OpenTimer6 ([see page 1131](#))CloseTimer6 ([see page 1145](#))OpenTimer7 ([see page 1132](#))ReadTimer7 ([see page 1139](#))WriteTimer7 ([see page 1141](#))CloseTimer7 ([see page 1145](#))OpenTimer8 ([see page 1134](#))CloseTimer8 ([see page 1145](#))OpenTimer10 ([see page 1135](#))CloseTimer10 ([see page 1146](#))OpenTimer12 ([see page 1136](#))CloseTimer12 ([see page 1146](#))**Macros**

WriteTimer2

ReadTimer2

WriteTimer4

ReadTimer4

WriteTimer6

ReadTimer6
WriteTimer8
ReadTimer8
WriteTimer10
ReadTimer10
WriteTimer12
ReadTimer12

Timers Examples (see page 1146)

7.4.2.1.17 Flash (8xK90 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.4.2.2 PIC18F8xK90_32

PIC18F85K90 (see page 61)

7.4.2.2.1 ADC (85K90 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

FunctionsOpenADC ([see page 870](#))SetChanADC ([see page 887](#))SelChanConvADC ([see page 895](#))ConvertADC ([see page 896](#))BusyADC ([see page 897](#))ReadADC ([see page 897](#))CloseADC ([see page 897](#))**Macros**

ADC_INT_ENABLE

ADC_INT_DISABLE

7.4.2.2.2 Analog Comparator (85K90 Set)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

FunctionsOpen_ancomp1 ([see page 907](#))Open_ancomp2 ([see page 915](#))Open_ancomp3 ([see page 917](#))Close_ancomp1 ([see page 923](#))Close_ancomp2 ([see page 923](#))Close_ancomp3 ([see page 923](#))**Macros**

Config_CVREF

Comparator Examples ([see page 924](#))

7.4.2.2.3 CTMU (85K90 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU ([see page 925](#))

CurrentControlCTMU ([see page 927](#))

CloseCTMU ([see page 927](#))

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples ([see page 928](#))

7.4.2.2.4 Input Capture (85K90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenECapture1 ([see page 968](#))

OpenECapture2 ([see page 971](#))

OpenECapture3 ([see page 972](#))

OpenCapture4 ([see page 960](#))

OpenCapture5 ([see page 964](#))

OpenCapture6 ([see page 973](#))

OpenCapture7 ([see page 974](#))

OpenCapture8 ([see page 976](#))

ReadECapture1 ([see page 986](#))

ReadECapture2 ([see page 986](#))

ReadECapture3 ([see page 987](#))

ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
ReadCapture6 (see page 985)
ReadCapture7 (see page 985)
ReadCapture8 (see page 985)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)
CloseCapture6 (see page 989)
CloseCapture7 (see page 989)
CloseCapture8 (see page 989)

Input Capture Examples (see page 991)

7.4.2.2.5 Output Compare (85K90 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1005)
OpenECompare2 (see page 1008)
OpenECompare3 (see page 1009)
OpenCompare4 (see page 998)
OpenCompare5 (see page 1001)
OpenCompare6 (see page 1010)
OpenCompare7 (see page 1011)
OpenCompare8 (see page 1013)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)
CloseCompare6 (see page 1023)
CloseCompare7 (see page 1023)
CloseCompare8 (see page 1023)

Output Compare Examples ([↗](#) see page 1025)

7.4.2.2.6 PWM (85K90 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (↗ see page 1037)
OpenEPWM2 (↗ see page 1038)
OpenEPWM3 (↗ see page 1039)
OpenPWM4 (↗ see page 1029)
OpenPWM5 (↗ see page 1029)
OpenPWM6 (↗ see page 1032)
OpenPWM7 (↗ see page 1033)
OpenPWM8 (↗ see page 1034)
SetDCEPWM1 (↗ see page 1054)
SetDCEPWM2 (↗ see page 1054)
SetDCEPWM3 (↗ see page 1055)
SetDCPWM4 (↗ see page 1049)
SetDCPWM5 (↗ see page 1050)
SetDCPWM6 (↗ see page 1050)
SetDCPWM7 (↗ see page 1051)
SetDCPWM8 (↗ see page 1052)
SetOutputEPWM1 (↗ see page 1045)
SetOutputEPWM2 (↗ see page 1046)
SetOutputEPWM3 (↗ see page 1046)
CloseEPWM1 (↗ see page 1059)
CloseEPWM2 (↗ see page 1059)
CloseEPWM3 (↗ see page 1059)
ClosePWM4 (↗ see page 1057)
ClosePWM5 (↗ see page 1057)
ClosePWM6 (↗ see page 1057)
ClosePWM7 (↗ see page 1057)
ClosePWM8 (↗ see page 1058)

PWM Examples ([↗](#) see page 1060)

7.4.2.2.7 I2C (85K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 (see page 1064)

WriteI2C1 (see page 1066)

ReadI2C1 (see page 1067)

putSI2C1 (see page 1069)

getSI2C1 (see page 1070)

OpenI2C2 (see page 1065)

WriteI2C2 (see page 1067)

ReadI2C2 (see page 1068)

putSI2C2 (see page 1069)

getSI2C2 (see page 1071)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putCI2C1

getCI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit

I2C2_Intr_Status

StopI2C2

StartI2C2

RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.4.2.2.8 I2C EEPROM (85K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.4.2.2.9 EEP (85K90 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM

- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.4.2.2.10 IO Ports (85K90 Set)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.4.2.2.11 MWIRE (85K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMWire1 (see page 1166)
ReadMWire1 (see page 1170)
WriteMWire1 (see page 1169)
getsMWire1 (see page 1167)
OpenMWire2 (see page 1166)
ReadMWire2 (see page 1171)
WriteMWire2 (see page 1169)
getsMWire2 (see page 1168)

Macros
CloseMWire1
getcMWire1
putcMWire1
DataRdyMWire1
CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples (see page 1172)

7.4.2.2.12 RTCC (85K90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccls2ndHalfSecond
mRtcclsAlrmEnabled
mRtcclsOn
mRtcclsSync
mRtcclsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.4.2.2.13 SW_RTCC (85K90 Set)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.4.2.2.14 SPI (853K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.4.2.2.15 USART (85K90 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART

putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.4.2.2.16 Timers (85K90 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)

OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)
OpenTimer8 (see page 1134)
CloseTimer8 (see page 1145)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6
WriteTimer8
ReadTimer8

Timers Examples (see page 1146)

7.4.2.2.17 Flash (85K90 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.5 18F87J90 Family

PIC18F66J90 / PIC18F67J90 / PIC18F86J90 / PIC18F87J90

7.5.1 ADC (87J90 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 862)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CALIB
ADC_NO_CALIB

ADC Examples ([see page 898](#))

7.5.2 Analog Comparator (87J90 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.5.3 CTMU (87J90 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.5.4 Input Capture (87J90 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples ([see page 991](#))

7.5.5 Output Compare (87J90 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.5.6 PWM (87J90 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.5.7 I2C (87J90 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

Writel2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcl2C

getcl2C

putcl2C

I2C Examples (see page 1072)

7.5.8 I2C EEPROM (87J90 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.5.9 IO Ports (87J90 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups

ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.5.10 RTCC (87J90 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable

mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccIs2ndHalfSecond
mRtccIsAlrmEnabled
mRtccIsOn
mRtccIsSync
mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.5.11 SW_RTCC (87J90 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.5.12 SPI (87J90 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.5.13 Timers (87J90 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (🔗 see page 1120)
ReadTimer1 (🔗 see page 1138)
WriteTimer1 (🔗 see page 1140)
CloseTimer1 (🔗 see page 1143)
OpenTimer2 (🔗 see page 1123)
CloseTimer2 (🔗 see page 1144)
OpenTimer3 (🔗 see page 1124)
ReadTimer3 (🔗 see page 1138)
WriteTimer3 (🔗 see page 1140)
CloseTimer3 (🔗 see page 1144)
SetTmrCCPSrc (🔗 see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (🔗 see page 1146)

7.5.14 USART (87J90 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (🔗 see page 1148)
Write1USART (🔗 see page 1154)
baud1USART (🔗 see page 1155)
gets1USART (🔗 see page 1159)
puts1USART (🔗 see page 1160)
puts1USART (🔗 see page 1162)
Read1USART (🔗 see page 1163)
Open2USART (🔗 see page 1150)
Write2USART (🔗 see page 1155)
baud2USART (🔗 see page 1157)

gets2USART (🔗 see page 1159)
puts2USART (🔗 see page 1161)
puts2USART (🔗 see page 1162)
Read2USART (🔗 see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (🔗 see page 1164)

7.5.15 MWIRE (87J90 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (🔗 see page 1167)
ReadMwire (🔗 see page 1171)
WriteMwire (🔗 see page 1170)
getsMwire (🔗 see page 1168)

Macros

CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.5.16 Flash (87J90 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.6 18F87J93 Family

PIC18F66J93 / PIC18F67J93 / PIC18F86J93 / PIC18F87J93

7.6.1 ADC (87J93 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 862)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples (see page 898)

7.6.2 Analog Comparator (87J93 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.6.3 CTMU (87J93 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.6.4 Input Capture (87J93 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

OpenCapture2 (see page 955)

ReadCapture1 (see page 983)

ReadCapture2 (see page 984)

CloseCapture1 (see page 987)

CloseCapture2 (see page 987)

Input Capture Examples (see page 991)

7.6.5 Output Compare (87J93 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.6.6 PWM (87J93 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.6.7 I2C (87J93 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.

- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.6.8 I2C EEPROM (87J93 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.6.9 SPI (87J93 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.6.10 IO Ports (87J93 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.6.11 RTCC (87J93 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock ([see page 1095](#))

RtccWriteTime ([see page 1099](#))

RtccWriteDate ([see page 1100](#))

RtccWriteAlrmTimeDate ([see page 1101](#))

RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccIs2ndHalfSecond
mRtccIsAlrmEnabled
mRtccIsOn
mRtccIsSync
mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples ([↗](#) see page 1104)

7.6.12 SW_RTCC (87J93 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (↗ see page 1106)
update_RTCC (↗ see page 1106)
Close_RTCC (↗ see page 1106)

SW_RTCC Examples ([↗](#) see page 1107)

7.6.13 Timers (87J93 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (↗ see page 1119)
ReadTimer0 (↗ see page 1137)
WriteTimer0 (↗ see page 1139)
CloseTimer0 (↗ see page 1143)
OpenTimer1 (↗ see page 1120)
ReadTimer1 (↗ see page 1138)
WriteTimer1 (↗ see page 1140)
CloseTimer1 (↗ see page 1143)
OpenTimer2 (↗ see page 1123)
CloseTimer2 (↗ see page 1144)
OpenTimer3 (↗ see page 1124)
ReadTimer3 (↗ see page 1138)
WriteTimer3 (↗ see page 1140)
CloseTimer3 (↗ see page 1144)
SetTmrCCPSrc (↗ see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples ([see page 1146](#))

7.6.14 USART (87J93 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpen1USART ([see page 1148](#))Write1USART ([see page 1154](#))baud1USART ([see page 1155](#))gets1USART ([see page 1159](#))putrs1USART ([see page 1160](#))puts1USART ([see page 1162](#))Read1USART ([see page 1163](#))Open2USART ([see page 1150](#))Write2USART ([see page 1155](#))baud2USART ([see page 1157](#))gets2USART ([see page 1159](#))putrs2USART ([see page 1161](#))puts2USART ([see page 1162](#))Read2USART ([see page 1163](#))**Macros**

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (🔗 see page 1164)

7.6.15 MWIRE (87J93 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (🔗 see page 1167)
ReadMwire (🔗 see page 1171)
WriteMwire (🔗 see page 1170)
getsMwire (🔗 see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (🔗 see page 1172)

7.6.16 Flash (87J93 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteWordFlash (see page 1202)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.7 18F/LF46J50 Family

PIC18F24J50 / PIC18F25J50 / PIC18F26J50

PIC18F44J50 / PIC18F45J50 / PIC18F46J50

PIC18LF24J50 / PIC18LF25J50 / PIC18LF26J50

PIC18LF44J50 / PIC18LF45J50 / PIC18LF46J50

7.7.1 PIC18F/LF26J50 Set

PIC18F24J50 / PIC18F25J50 / PIC18F26J50

PIC18LF24J50 / PIC18LF25J50 / PIC18LF26J50

7.7.1.1 ADC (2xJ50 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.

- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 860)
SetChanADC (see page 884)
SelChanConvADC (see page 891)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CALIB
ADC_NO_CALIB

ADC Examples (see page 898)

7.7.1.2 Analog Comparator (2xJ50 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 906)
Open_ancomp2 (see page 914)
Close_ancomp1 (see page 923)
Close_ancomp2 (see page 923)

Comparator Examples (see page 924)

7.7.1.3 CTMU (2xJ50 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU (see page 925)

CurrentControlCTMU (see page 927)

CloseCTMU (see page 927)

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples (see page 928)

7.7.1.4 Deep Sleep (2xJ50 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions

DeepSleepWakeUpSource (see page 930)

GotoDeepSleep (see page 930)

IsResetFromDeepSleep (see page 931)

ReadDSGPR (see page 931)

ULPWakeUpEnable (see page 931)

Macros

ReleaseDeepSleep

Write_DSGPR

Deep Sleep Examples (see page 932)

7.7.1.5 PPS (2xJ50 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros
iPPSInput (see page 936)
PPSInput (see page 935)
iPPSOutput (see page 944)
PPSOutput (see page 941)

PPS Examples ([see page 953](#))

7.7.1.6 Input Capture (2xJ50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples ([see page 991](#))

7.7.1.7 Output Compare (2xJ50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)

OpenCompare2 (see page 993)

CloseCompare1 (see page 1021)

CloseCompare2 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.7.1.8 PWM (2xJ50 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.7.1.9 I2C (2xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putSI2C1 (see page 1069)
getSI2C1 (see page 1070)

Macros
EnableIntI2C1

DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1

I2C Examples (see page 1072)

7.7.1.10 I2C EEPROM (2xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)

I2C EEPROM Examples (see page 1088)

7.7.1.11 SPI (2xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.7.1.12 IO Ports (2xJ50 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.7.1.13 RTCC (2xJ50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)

RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)
RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccls2ndHalfSecond
mRtcclsAlrmEnabled
mRtcclsOn
mRtcclsSync
mRtcclsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.7.1.14 SW_RTCC (2xJ50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.7.1.15 Timers (2xJ50 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples ([see page 1146](#))

7.7.1.16 USART (2xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART

Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.7.1.17 MWIRE (2xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros

CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples ([see page 1172](#))

7.7.1.18 Flash (2xJ50 Set)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))

- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteWordFlash (see page 1202)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.7.2 PIC18F/LF46J50 Set

PIC18F44J50 / PIC18F45J50 / PIC18F46J50

PIC18LF44J50 / PIC18LF45J50 / PIC18LF46J50

7.7.2.1 ADC (4xJ50 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 860)

SetChanADC (see page 884)

SelChanConvADC (see page 891)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples ([see page 898](#))

7.7.2.2 Analog Comparator (4xJ50 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 ([see page 906](#))

Open_ancomp2 ([see page 914](#))

Close_ancomp1 ([see page 923](#))

Close_ancomp2 ([see page 923](#))

Comparator Examples ([see page 924](#))

7.7.2.3 CTMU (4xJ50 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU ([see page 925](#))

CurrentControlCTMU ([see page 927](#))

CloseCTMU ([see page 927](#))

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2
Disbl_CTMUEdge1
Disbl_CTMUEdge2
CTMUEdge1_Status
CTMUEdge2_Status
CTMU_INT_ENABLE
CTMU_INT_DISABLE
CTMU_Clear_Intr_Status_Bit
CTMU_Intr_Status

CTMU Examples (see page 928)

7.7.2.4 Deep Sleep (4xJ50 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions
DeepSleepWakeUpSource (see page 930)
GotoDeepSleep (see page 930)
IsResetFromDeepSleep (see page 931)
ReadDSGPR (see page 931)
ULPWakeUpEnable (see page 931)

Macros
ReleaseDeepSleep
Write_DSGPR

Deep Sleep Examples (see page 932)

7.7.2.5 PPS (4xJ50 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros
iPPSInput (see page 936)

PPSInput (see page 935)

iPPSOutput (see page 945)

PPSOutput (see page 942)

PPS Examples (see page 953)

7.7.2.6 Input Capture (4xJ50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples (see page 991)

7.7.2.7 Output Compare (4xJ50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)

Output Compare Examples (see page 1025)

7.7.2.8 PWM (4xJ50 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.7.2.9 I2C (4xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putI2C1 (see page 1069)
getI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putI2C2 (see page 1069)
getI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status

StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putI2C1
getI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.7.2.10 I2C EEPROM (4xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)

EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.7.2.11 IO Ports (4xJ50 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.7.2.12 RTCC (4xJ50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)

RtccWriteAlrmTimeDate (see page 1101)

RtccWriteAlrmTime (see page 1101)

RtccWriteAlrmDate (see page 1102)

RtccSetChimeEnable (see page 1103)

RtccSetCalibration (see page 1103)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRpt (see page 1098)

RtccReadTimeDate (see page 1097)

RtccReadTime (see page 1097)

RtccReadDate (see page 1096)

RtccReadAlrmTimeDate (see page 1096)

RtccReadAlrmTime (see page 1096)

RtccReadAlrmDate (see page 1095)

RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit

mRtccAlrmDisable

mRtccAlrmEnable

mRtccClearAlrmPtr

mRtccClearRtcPtr

mRtccGetAlarmRpt

mRtccGetAlarmRptCount

mRtccGetCalibration

mRtccGetChimeEnable

mRtccGetClockOe

mRtccls2ndHalfSecond
mRtcclsAlrmEnabled
mRtcclsOn
mRtcclsSync
mRtcclsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples ([see page 1104](#))

7.7.2.13 SW_RTCC (4xJ50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.7.2.14 SPI (4xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI1 (see page 1109)
WriteSPI1 (see page 1112)
ReadSPI1 (see page 1113)
putsSPI1 (see page 1114)

getsSPI1 (see page 1115)
OpenSPI2 (see page 1110)
WriteSPI2 (see page 1112)
ReadSPI2 (see page 1113)
putsSPI2 (see page 1114)
getsSPI2 (see page 1116)

Macros
EnableIntSPI1
DisableIntSPI1
SetPriorityIntSPI1
SPI1_Clear_Intr_Status_Bit
SPI1_Intr_Status
SPI1_Clear_Recv_OV
CloseSPI1
DataRdySPI1
getcSPI1
putcSPI1
EnableIntSPI2
DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.7.2.15 Timers (4xJ50 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.7.2.16 USART (4xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)

putrs1USART (🔗 see page 1160)
puts1USART (🔗 see page 1162)
Read1USART (🔗 see page 1163)
Open2USART (🔗 see page 1150)
Write2USART (🔗 see page 1155)
baud2USART (🔗 see page 1157)
gets2USART (🔗 see page 1159)
putrs2USART (🔗 see page 1161)
puts2USART (🔗 see page 1162)
Read2USART (🔗 see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (🔗 see page 1164)

7.7.2.17 PMP (4xJ50 Family)

This peripheral library module:

- Is a parallel 8-bit I/O module. PMP stands for Parallel Master Port module.
- Is specifically designed to communicate with a wide variety of parallel devices, such as communications peripherals, LCDs, external memory devices and microcontrollers.
- Is highly configurable, since the interface to parallel peripherals varies significantly.

Functions
PMPOpen (🔗 see page 1187)
PMPIsBufferNEmpty (🔗 see page 1189)
PMPIsBufferNFull (🔗 see page 1189)
PMPMasterRead (🔗 see page 1188)
PMPMasterWrite (🔗 see page 1189)
PMPSetAddress (🔗 see page 1187)

PMPSlaveReadBufferN ([see page 1190](#))

PMPSlaveReadBuffers ([see page 1190](#))

PMPSlaveWriteBufferN ([see page 1191](#))

PMPSlaveWriteBuffers ([see page 1191](#))

PMPClose ([see page 1192](#))

Macros

mPMPIsBufferFull

mPMPSetWriteStrobePolarity

mPMPSetWriteStrobeEnable

mPMPSetWaitEnd

mPMPSetWaitMiddle

mPMPIsBufferOverflow

mPMPSetWaitBegin

mPMPSetReadStrobePolarity

mPMPSetReadStrobeEnable

mPMPSetPortPins

mPMPSetPortMode

mPMPSetPortEnable

mPMPSetAddrLatchPolarity

mPMPSetInterruptMode

mPMPClearBufferOverflow

mPMPSetIdle

mPMP_Clear_Intr_Status_Bit

mPMPSetInterruptPriority

mPMPSetInterruptEnable

mPMPSetDataMode

mPMPClearBufferUnderflow

mPMPSetChipSelect2Polarity

mPMPSetChipSelect2Enable

mPMPSetChipSelect1Polarity

mPMPSetChipSelect1Enable

mPMPSetChipSelectMode

mPMPSetByteEnable

mPMPSetByteEnablePolarity

mPMPIsBufferEmpty

mPMPIsBufferUnderflow

mPMPSetAddrIncMode

mPMPSetAddrMux

PMP Examples ([↗](#) see page 1192)

7.7.2.18 MWIRE (4xJ50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMWire1 (↗ see page 1166)
ReadMWire1 (↗ see page 1170)
WriteMWire1 (↗ see page 1169)
getsMWire1 (↗ see page 1167)
OpenMWire2 (↗ see page 1166)
ReadMWire2 (↗ see page 1171)
WriteMWire2 (↗ see page 1169)
getsMWire2 (↗ see page 1168)

Macros
CloseMWire1
getcMWire1
putcMWire1
DataRdyMWire1
CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples ([↗](#) see page 1172)

7.7.2.19 Flash (4xJ50 Set)

This peripheral module supports functionalities:

- Erasing Flash ([↗](#) see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteWordFlash (see page 1202)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.8 18F/LF46J11 Family

PIC18F24J11 / PIC18F25J11 / PIC18F26J11

PIC18F44J11 / PIC18F45J11 / PIC18F46J11

PIC18LF24J11 / PIC18LF25J11 / PIC18LF26J11

PIC18LF44J11 / PIC18LF45J11 / PIC18LF46J11

7.8.1 PIC18F/LF26J11 Set

PIC18F24J11 / PIC18F25J11 / PIC18F26J11

PIC18LF24J11 / PIC18LF25J11 / PIC18LF26J11

7.8.1.1 ADC (2xJ11 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.

- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 860)
SetChanADC (see page 884)
SelChanConvADC (see page 891)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CALIB
ADC_NO_CALIB

ADC Examples (see page 898)

7.8.1.2 Analog Comparator (2xJ11 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 906)
Open_ancomp2 (see page 914)
Close_ancomp1 (see page 923)
Close_ancomp2 (see page 923)

Comparator Examples (see page 924)

7.8.1.3 CTMU (2xJ11 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions

OpenCTMU ([see page 925](#))

CurrentControlCTMU ([see page 927](#))

CloseCTMU ([see page 927](#))

Macros

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status

CTMUEdge2_Status

CTMU_INT_ENABLE

CTMU_INT_DISABLE

CTMU_Clear_Intr_Status_Bit

CTMU_Intr_Status

CTMU Examples ([see page 928](#))

7.8.1.4 Deep Sleep (2xJ11 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions

DeepSleepWakeUpSource ([see page 930](#))

GotoDeepSleep ([see page 930](#))

IsResetFromDeepSleep ([see page 931](#))

ReadDSGPR ([see page 931](#))

ULPWakeUpEnable ([see page 931](#))

Macros

ReleaseDeepSleep

Write_DSGPR

Deep Sleep Examples ([see page 932](#))

7.8.1.5 PPS (2xJ11 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros
iPPSInput (see page 938)
PPSInput (see page 939)
iPPSOutput (see page 944)
PPSOutput (see page 941)

PPS Examples ([see page 953](#))

7.8.1.6 Input Capture (2xJ11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples ([see page 991](#))

7.8.1.7 Output Compare (2xJ11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)

OpenCompare2 (see page 993)

CloseCompare1 (see page 1021)

CloseCompare2 (see page 1022)

Output Compare Examples (see page 1025)

7.8.1.8 PWM (2xJ11 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

OpenPWM2 (see page 1027)

SetDCPWM1 (see page 1047)

SetDCPWM2 (see page 1048)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.8.1.9 I2C (2xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 (see page 1064)

WriteI2C1 (see page 1066)

ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

Macros

EnableIntI2C1

DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1

I2C Examples (see page 1072)

7.8.1.10 I2C EEPROM (2xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)

I2C EEPROM Examples (see page 1088)

7.8.1.11 IO Ports (2xJ11 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.

- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.8.1.12 RTCC (2xJ11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU.
- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions
RtccInitClock (see page 1095)
RtccWriteTime (see page 1099)
RtccWriteDate (see page 1100)
RtccWriteAlrmTimeDate (see page 1101)
RtccWriteAlrmTime (see page 1101)
RtccWriteAlrmDate (see page 1102)
RtccSetChimeEnable (see page 1103)
RtccSetCalibration (see page 1103)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRptCount (see page 1104)
RtccSetAlarmRpt (see page 1098)

RtccReadTimeDate (see page 1097)
RtccReadTime (see page 1097)
RtccReadDate (see page 1096)
RtccReadAlrmTimeDate (see page 1096)
RtccReadAlrmTime (see page 1096)
RtccReadAlrmDate (see page 1095)
RtccWrOn (see page 1098)

Macros
mRtcc_Clear_Intr_Status_Bit
mRtccAlrmDisable
mRtccAlrmEnable
mRtccClearAlrmPtr
mRtccClearRtcPtr
mRtccGetAlarmRpt
mRtccGetAlarmRptCount
mRtccGetCalibration
mRtccGetChimeEnable
mRtccGetClockOe
mRtccIs2ndHalfSecond
mRtccIsAlrmEnabled
mRtccIsOn
mRtccIsSync
mRtccIsWrEn
mRtccOff
mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples (see page 1104)

7.8.1.13 SW_RTCC (2xJ11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.8.1.14 SPI (2xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.8.1.15 Timers (2xJ11 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2

ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.8.1.16 USART (2xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART

getc2USART

USART Examples (see page 1164)

7.8.1.17 MWIRE (2xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)

ReadMwire1 (see page 1170)

WriteMwire1 (see page 1169)

getsMwire1 (see page 1167)

OpenMwire2 (see page 1166)

ReadMwire2 (see page 1171)

WriteMwire2 (see page 1169)

getsMwire2 (see page 1168)

Macros

CloseMwire1

getcMwire1

putcMwire1

DataRdyMwire1

CloseMwire2

getcMwire2

putcMwire2

DataRdyMwire2

MWIRE Examples (see page 1172)

7.8.1.18 Flash (2xJ11 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.8.2 PIC18F/LF46J11 Set

PIC18F44J11 / PIC18F45J11 / PIC18F46J11

PIC18LF44J11 / PIC18LF45J11 / PIC18LF46J11

7.8.2.1 ADC (4xJ11 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 860)
SetChanADC (see page 884)
SelChanConvADC (see page 891)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples ([see page 898](#))

7.8.2.2 Analog Comparator (4xJ11 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output**Functions**Open_ancomp1 ([see page 906](#))Open_ancomp2 ([see page 914](#))Close_ancomp1 ([see page 923](#))Close_ancomp2 ([see page 923](#))Comparator Examples ([see page 924](#))

7.8.2.3 CTMU (4xJ11 Family)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

FunctionsOpenCTMU ([see page 925](#))CurrentControlCTMU ([see page 927](#))CloseCTMU ([see page 927](#))**Macros**

Enbl_CTMUEdge1

Enbl_CTMUEdge2

Disbl_CTMUEdge1

Disbl_CTMUEdge2

CTMUEdge1_Status
CTMUEdge2_Status
CTMU_INT_ENABLE
CTMU_INT_DISABLE
CTMU_Clear_Intr_Status_Bit
CTMU_Intr_Status

CTMU Examples ([see page 928](#))

7.8.2.4 Deep Sleep (4xJ11 Family)

This peripheral library module:

- Functions to configure Deep Sleep.
- Function to put device to Deep Sleep & check the wake up source.
- Provides macros to write to deep sleep GPR and Release from Deep Sleep.

Functions
DeepSleepWakeUpSource (see page 930)
GotoDeepSleep (see page 930)
IsResetFromDeepSleep (see page 931)
ReadDSGPR (see page 931)
ULPWakeUpEnable (see page 931)

Macros
ReleaseDeepSleep
Write_DSGPR

Deep Sleep Examples ([see page 932](#))

7.8.2.5 PPS (4xJ11 Family)

This peripheral library module provides:

- Assign given pin as input or output pin.
- Provides macros to Lock and unlock the PPS

Macros
iPPSInput (see page 938)
PPSInput (see page 939)
iPPSOutput (see page 945)
PPSOutput (see page 942)

PPS Examples ([↗](#) see page 953)

7.8.2.6 Input Capture (4xJ11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (↗ see page 954)
OpenCapture2 (↗ see page 955)
ReadCapture1 (↗ see page 983)
ReadCapture2 (↗ see page 984)
CloseCapture1 (↗ see page 987)
CloseCapture2 (↗ see page 987)

Input Capture Examples ([↗](#) see page 991)

7.8.2.7 Output Compare (4xJ11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (↗ see page 992)
OpenCompare2 (↗ see page 993)
CloseCompare1 (↗ see page 1021)
CloseCompare2 (↗ see page 1022)

Output Compare Examples ([↗](#) see page 1025)

7.8.2.8 PWM (4xJ11 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.8.2.9 I2C (4xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putsI2C1 (see page 1069)
getsI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putsI2C2 (see page 1069)
getsI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1

NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putI2C1
getI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.8.2.10 I2C EEPROM (4xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)

EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.8.2.11 IO Ports (4xJ11 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.8.2.12 RTCC (4xJ11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function.
- Is intended for applications where accurate time must be maintained for extended periods of time with minimum to no

intervention from the CPU.

- Is optimized for low-power usage in order to provide extended battery lifetime while keeping track of time

Functions

RtccInitClock (see page 1095)

RtccWriteTime (see page 1099)

RtccWriteDate (see page 1100)

RtccWriteAlrmTimeDate (see page 1101)

RtccWriteAlrmTime (see page 1101)

RtccWriteAlrmDate (see page 1102)

RtccSetChimeEnable (see page 1103)

RtccSetCalibration (see page 1103)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRptCount (see page 1104)

RtccSetAlarmRpt (see page 1098)

RtccReadTimeDate (see page 1097)

RtccReadTime (see page 1097)

RtccReadDate (see page 1096)

RtccReadAlrmTimeDate (see page 1096)

RtccReadAlrmTime (see page 1096)

RtccReadAlrmDate (see page 1095)

RtccWrOn (see page 1098)

Macros

mRtcc_Clear_Intr_Status_Bit

mRtccAlrmDisable

mRtccAlrmEnable

mRtccClearAlrmPtr

mRtccClearRtcPtr

mRtccGetAlarmRpt

mRtccGetAlarmRptCount

mRtccGetCalibration

mRtccGetChimeEnable

mRtccGetClockOe

mRtccls2ndHalfSecond

mRtcclsAlrmEnabled

mRtcclsOn

mRtcclsSync

mRtcclsWrEn

mRtccOff

mRtccOn
mRtccSetAlrmPtr
mRtccSetClockOe
mRtccSetInt
mRtccSetRtcPtr
mRtccWaitSync
mRtccWrOff

RTCC Examples ([see page 1104](#))

7.8.2.13 SW_RTCC (4xJ11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.8.2.14 SPI (4xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI1 (see page 1109)
WriteSPI1 (see page 1112)
ReadSPI1 (see page 1113)
putsSPI1 (see page 1114)
getsSPI1 (see page 1115)
OpenSPI2 (see page 1110)
WriteSPI2 (see page 1112)
ReadSPI2 (see page 1113)
putsSPI2 (see page 1114)
getsSPI2 (see page 1116)

Macros
EnableIntSPI1
DisableIntSPI1
SetPriorityIntSPI1
SPI1_Clear_Intr_Status_Bit
SPI1_Intr_Status
SPI1_Clear_Recv_OV
CloseSPI1
DataRdySPI1
getcSPI1
putcSPI1
EnableIntSPI2
DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.8.2.15 Timers (4xJ11 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1121)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1125)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.8.2.16 USART (4xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)

puts2USART (see page 1162)

Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART

Busy2USART

putc2USART

getc2USART

USART Examples (see page 1164)

7.8.2.17 PMP (4xJ11 Family)

This peripheral library module:

- Is a parallel 8-bit I/O module. PMP stands for Parallel Master Port module.
- Is specifically designed to communicate with a wide variety of parallel devices, such as communications peripherals, LCDs, external memory devices and microcontrollers.
- Is highly configurable, since the interface to parallel peripherals varies significantly.

Functions

PMPOpen (see page 1187)

PMPIsBufferNEmpty (see page 1189)

PMPIsBufferNFull (see page 1189)

PMPMasterRead (see page 1188)

PMPMasterWrite (see page 1189)

PMPSetAddress (see page 1187)

PMPSlaveReadBufferN (see page 1190)

PMPSlaveReadBuffers (see page 1190)

PMPSlaveWriteBufferN (see page 1191)

PMPSlaveWriteBuffers (see page 1191)

PMPClose (see page 1192)

Macros
mPMPIsBufferFull
mPMPSetWriteStrobePolarity
mPMPSetWriteStrobeEnable
mPMPSetWaitEnd
mPMPSetWaitMiddle
mPMPIsBufferOverflow
mPMPSetWaitBegin
mPMPSetReadStrobePolarity
mPMPSetReadStrobeEnable
mPMPSetPortPins
mPMPSetPortMode
mPMPSetPortEnable
mPMPSetAddrLatchPolarity
mPMPSetInterruptMode
mPMPClearBufferOverflow
mPMPSetIdle
mPMP_Clear_Intr_Status_Bit
mPMPSetInterruptPriority
mPMPSetInterruptEnable
mPMPSetDataMode
mPMPClearBufferUnderflow
mPMPSetChipSelect2Polarity
mPMPSetChipSelect2Enable
mPMPSetChipSelect1Polarity
mPMPSetChipSelect1Enable
mPMPSetChipSelectMode
mPMPSetByteEnable
mPMPSetByteEnablePolarity
mPMPIsBufferEmpty
mPMPIsBufferUnderflow
mPMPSetAddrIncMode
mPMPSetAddrMux

PMP Examples (see page 1192)

7.8.2.18 MWIRE (4xJ11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire1 (see page 1166)

ReadMWire1 (see page 1170)

WriteMWire1 (see page 1169)

getsMWire1 (see page 1167)

OpenMWire2 (see page 1166)

ReadMWire2 (see page 1171)

WriteMWire2 (see page 1169)

getsMWire2 (see page 1168)

Macros

CloseMWire1

getcMWire1

putcMWire1

DataRdyMWire1

CloseMWire2

getcMWire2

putcMWire2

DataRdyMWire2

MWIRE Examples (see page 1172)

7.8.2.19 Flash (4xJ11 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteWordFlash (see page 1202)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.9 18F/LF14K50 Family

PIC18F13K50 / PIC18F14K50**PIC18LF13K50 / PIC18LF14K50**

7.9.1 ADC (14K50 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 858)

SetChanADC (see page 883)

SelChanConvADC (see page 890)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.9.2 Analog Comparator (14K50 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 900)
Close_ancomp1 (see page 923)
Open_ancomp2 (see page 912)
Close_ancomp2 (see page 923)

Comparator Examples ([see page 924](#))

7.9.3 Input Capture (14K50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.9.4 Output Compare (14K50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.9.5 PWM (14K50 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

SetDCPWM1 (see page 1047)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

PWM Examples (see page 1060)

7.9.6 I2C (14K50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.9.7 I2C EEPROM (14K50 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.9.8 EEP (14K50 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.9.9 IO Ports (14K50 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.9.10 SW_RTCC (14K50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.9.11 SPI (14K50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI ([see page 1108](#))

WriteSPI ([see page 1111](#))

ReadSPI ([see page 1113](#))

putsSPI ([see page 1114](#))

getsSPI ([see page 1115](#))

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.9.12 Timers (14K50 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1120)

ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)

CloseTimer2 (see page 1144)

OpenTimer3 (see page 1124)

ReadTimer3 (see page 1138)

WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.9.13 USART (14K50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART ([see page 1152](#))

WriteUSART ([see page 1155](#))

baudUSART ([see page 1158](#))

getsUSART ([see page 1160](#))

putrsUSART ([see page 1161](#))

putsUSART ([see page 1163](#))

ReadUSART ([see page 1164](#))

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples ([see page 1164](#))

7.9.14 MWIRE (14K50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire (↗ see page 1170)

getsMwire (↗ see page 1168)
--

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([↗](#) see page 1172)

7.9.15 Flash (14K50 Family)

This peripheral module supports functionalities:

- Erasing Flash ([↗](#) see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (↗ see page 1198)
--

EraseFlash (↗ see page 1196)

WriteBlockFlash (↗ see page 1198)
--

WriteBytesFlash (↗ see page 1202)
--

Macros

LoadFlashAddr

TableRead

Flash Examples ([↗](#) see page 1204)

7.10 18F/LF14K22 Family

PIC18F13K22 / PIC18F14K22 / PIC18LF13K22 / PIC18LF14K22

7.10.1 ADC (14K22 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 858)

SetChanADC (see page 883)

SelChanConvADC (see page 890)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.10.2 Analog Comparator (14K22 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 900)

Close_ancomp1 (see page 923)

Open_ancomp2 (see page 912)

Close_ancomp2 (see page 923)

Comparator Examples (see page 924)

7.10.3 Input Capture (14K22 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 ([see page 954](#))

ReadCapture1 ([see page 983](#))

CloseCapture1 ([see page 987](#))

Input Capture Examples ([see page 991](#))

7.10.4 Output Compare (14K22 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 ([see page 992](#))

CloseCompare1 ([see page 1021](#))

Output Compare Examples ([see page 1025](#))

7.10.5 PWM (14K22 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

SetDCPWM1 ([see page 1047](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 (see page 1056)

PWM Examples (see page 1060)

7.10.6 I2C (14K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

Writel2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcl2C

getcl2C

putcl2C

I2C Examples (see page 1072)

7.10.7 I2C EEPROM (14K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.10.8 EEP (14K22 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.10.9 IO Ports (14K22 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples (see page 1093)

7.10.10 SW_RTCC (14K22 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.10.11 SPI (14K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.10.12 Timers (14K22 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1120)

ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.10.13 USART (14K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART

getcUSART

USART Examples (see page 1164)

7.10.14 MWIRE (14K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.10.15 Flash (14K22 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)

WriteBlockFlash (see page 1198)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples ([see page 1204](#))

7.11 18F4553 Family

PIC18F2458 / PIC18F2553 / PIC18F4458 / PIC18F4553

7.11.1 ADC (4553 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)
--

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)
--

ReadADC (see page 897)
--

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.11.2 Analog Comparator (4553 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.11.3 Input Capture (4553 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.11.4 Output Compare (4553 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([↗](#) see page 1025)

7.11.5 PWM (4553 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (↗ see page 1026)
SetDCPWM1 (↗ see page 1047)
SetOutputPWM1 (↗ see page 1043)
ClosePWM1 (↗ see page 1056)
OpenPWM2 (↗ see page 1027)
SetDCPWM2 (↗ see page 1048)
ClosePWM2 (↗ see page 1056)

PWM Examples ([↗](#) see page 1060)

7.11.6 I2C (4553 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (↗ see page 1063)
WriteI2C (↗ see page 1066)
ReadI2C (↗ see page 1067)
putsI2C (↗ see page 1068)
getsI2C (↗ see page 1070)
IdleI2C (↗ see page 1072)
CloseI2C (↗ see page 1071)

Macros
EnableIntI2C

DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.11.7 I2C EEPROM (4553 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.11.8 EEP (4553 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.11.9 IO Ports (4553 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.11.10 SW_RTCC (4553 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.11.11 SPI (4553 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenSPI ([see page 1108](#))WriteSPI ([see page 1111](#))ReadSPI ([see page 1113](#))putsSPI ([see page 1114](#))getsSPI ([see page 1115](#))**Macros**

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.11.12 Timers (4553 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.11.13 USART (4553 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.11.14 MWIRE (4553 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.11.15 Flash (4553 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1199)
WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.12 18F8493 Family

PIC18F8493 / PIC18F8393 / PIC18F6493 / PIC18F6393

7.12.1 ADC (8493 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.

- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SetChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.12.2 Analog Comparator (8493 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.12.3 Input Capture (8493 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))ReadCapture1 ([see page 983](#))CloseCapture1 ([see page 987](#))Input Capture Examples ([see page 991](#))

7.12.4 Output Compare (8493 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

FunctionsOpenCompare1 ([see page 992](#))CloseCompare1 ([see page 1021](#))Output Compare Examples ([see page 1025](#))

7.12.5 PWM (8493 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

FunctionsOpenPWM1 ([see page 1026](#))SetDCPWM1 ([see page 1047](#))SetOutputPWM1 ([see page 1043](#))ClosePWM1 ([see page 1056](#))OpenPWM2 ([see page 1027](#))SetDCPWM2 ([see page 1048](#))ClosePWM2 ([see page 1056](#))PWM Examples ([see page 1060](#))

7.12.6 I2C (8493 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.12.7 I2C EEPROM (8493 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.12.8 IO Ports (8493 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.12.9 SW_RTCC (8493 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.12.10 SPI (8493 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI ([see page 1108](#))

WriteSPI ([see page 1111](#))

ReadSPI ([see page 1113](#))

putsSPI ([see page 1114](#))

getsSPI ([see page 1115](#))

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.12.11 Timers (8493 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)

ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1120)

ReadTimer1 (see page 1138)

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)

CloseTimer2 (see page 1144)

OpenTimer3 (see page 1124)

ReadTimer3 (see page 1138)

WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.12.12 USART (8493 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.12.13 MWIRE (8493 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.12.14 Flash (8493 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.13 18F8723 Family

PIC18F8723 / PIC18F8628 / PIC18F6723 / PIC18F6628

7.13.1 ADC (8723 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.13.2 Analog Comparator (8723 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

FunctionsOpen_ancomp ([see page 899](#))Close_ancomp ([see page 923](#))Comparator Examples ([see page 924](#))

7.13.3 Input Capture (8723 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))OpenCapture2 ([see page 955](#))OpenCapture3 ([see page 957](#))OpenCapture4 ([see page 959](#))OpenCapture5 ([see page 963](#))ReadCapture1 ([see page 983](#))ReadCapture2 ([see page 984](#))ReadCapture3 ([see page 984](#))ReadCapture4 ([see page 984](#))ReadCapture5 ([see page 984](#))CloseCapture1 ([see page 987](#))CloseCapture2 ([see page 987](#))CloseCapture3 ([see page 988](#))CloseCapture4 ([see page 988](#))CloseCapture5 ([see page 988](#))Input Capture Examples ([see page 991](#))

7.13.4 Output Compare (8723 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)
OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.13.5 PWM (8723 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.13.6 I2C (8723 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenI2C1 ([see page 1064](#))Writel2C1 ([see page 1066](#))ReadI2C1 ([see page 1067](#))putsI2C1 ([see page 1069](#))getsI2C1 ([see page 1070](#))OpenI2C2 ([see page 1065](#))Writel2C2 ([see page 1067](#))ReadI2C2 ([see page 1068](#))putsI2C2 ([see page 1069](#))getsI2C2 ([see page 1071](#))**Macros**

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit

I2C2_Intr_Status

StopI2C2

StartI2C2

RestartI2C2

NotAckI2C2

AckI2C2

DataRdyI2C2

IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.13.7 I2C EEPROM (8723 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddrRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.13.8 EEP (8723 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM

- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.13.9 IO Ports (8723 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.13.10 SW_RTCC (8723 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.13.11 SPI (8723 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples ([see page 1116](#))

7.13.12 Timers (8723 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

WriteTimer4

ReadTimer4

Timers Examples (see page 1146)

7.13.13 USART (8723 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)

Write1USART (see page 1154)

baud1USART (see page 1155)

gets1USART (see page 1159)

putrs1USART (see page 1160)

puts1USART (see page 1162)

Read1USART (see page 1163)

Open2USART (see page 1150)

Write2USART (see page 1155)

baud2USART (see page 1157)

gets2USART (see page 1159)

putrs2USART (see page 1161)

puts2USART (see page 1162)

Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.13.14 MWIRE (8723 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples ([see page 1172](#))

7.13.15 Flash (8723 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.14 18F84J15 Family

PIC18F84J15 / PIC18F64J15

7.14.1 ADC (84J15 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.14.2 Analog Comparator (84J15 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (🔗 see page 899)
Close_ancomp (🔗 see page 923)

Comparator Examples (🔗 see page 924)

7.14.3 Input Capture (84J15 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (🔗 see page 954)
OpenCapture2 (🔗 see page 955)
OpenCapture3 (🔗 see page 957)

OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples (see page 991)

7.14.4 Output Compare (84J15 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)
OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples (see page 1025)

7.14.5 PWM (84J15 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.14.6 I2C (84J15 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putI2C1 (see page 1069)
getI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putI2C2 (see page 1069)
getI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status

StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.14.7 I2C EEPROM (84J15 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)

EEAckPolling1 (see page 1076)

EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.14.8 IO Ports (84J15 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.14.9 SW_RTCC (84J15 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.14.10 SPI (84J15 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 ([see page 1109](#))

WriteSPI1 ([see page 1112](#))

ReadSPI1 ([see page 1113](#))

putsSPI1 ([see page 1114](#))

getsSPI1 ([see page 1115](#))

OpenSPI2 ([see page 1110](#))

WriteSPI2 ([see page 1112](#))

ReadSPI2 ([see page 1113](#))

putsSPI2 ([see page 1114](#))

getsSPI2 ([see page 1116](#))

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status
SPI1_Clear_Recv_OV
CloseSPI1
DataRdySPI1
getcSPI1
putcSPI1
EnableIntSPI2
DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples ([see page 1116](#))

7.14.11 Timers (84J15 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)

ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.14.12 USART (84J15 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.14.13 MWIRE (84J15 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2

getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.14.14 Flash (84J15 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.15 18F87J50 Family

PIC18F65J50 / PIC18F66J50 / PIC18F66J55 / PIC18F67J50

PIC18F85J50 / PIC18F86J50 / PIC18F86J55 / PIC18F87J50

7.15.1 ADC (87J50 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 865)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples (see page 898)

7.15.2 Analog Comparator (87J50 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 902)

Open_ancomp2 (see page 911)

Close_ancomp1 (see page 923)

Close_ancomp2 (see page 923)

Comparator Examples (see page 924)

7.15.3 Input Capture (87J50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.15.4 Output Compare (87J50 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)

OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples (see page 1025)

7.15.5 PWM (87J50 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples (see page 1060)

7.15.6 I2C (87J50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 ([see page 1064](#))

WriteI2C1 ([see page 1066](#))

ReadI2C1 ([see page 1067](#))

putsI2C1 ([see page 1069](#))

getsI2C1 ([see page 1070](#))

OpenI2C2 ([see page 1065](#))

WriteI2C2 ([see page 1067](#))

ReadI2C2 ([see page 1068](#))

putsI2C2 ([see page 1069](#))

getsI2C2 ([see page 1071](#))

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.15.7 I2C EEPROM (87J50 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.15.8 IO Ports (87J50 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.15.9 SW_RTCC (87J50 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.15.10 SPI (87J50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2
getcSPI2
putcSPI2

SPI Examples ([see page 1116](#))

7.15.11 Timers (87J50 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))ReadTimer0 ([see page 1137](#))WriteTimer0 ([see page 1139](#))CloseTimer0 ([see page 1143](#))OpenTimer1 ([see page 1120](#))ReadTimer1 ([see page 1138](#))WriteTimer1 ([see page 1140](#))CloseTimer1 ([see page 1143](#))OpenTimer2 ([see page 1123](#))CloseTimer2 ([see page 1144](#))OpenTimer3 ([see page 1124](#))ReadTimer3 ([see page 1138](#))WriteTimer3 ([see page 1140](#))CloseTimer3 ([see page 1144](#))OpenTimer4 ([see page 1130](#))CloseTimer4 ([see page 1144](#))SetTmrCCPSrc ([see page 1141](#))

Macros

WriteTimer2

ReadTimer2

WriteTimer4

ReadTimer4

Timers Examples ([↗](#) see page 1146)

7.15.12 USART (87J50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (↗ see page 1148)
Write1USART (↗ see page 1154)
baud1USART (↗ see page 1155)
gets1USART (↗ see page 1159)
putrs1USART (↗ see page 1160)
puts1USART (↗ see page 1162)
Read1USART (↗ see page 1163)
Open2USART (↗ see page 1150)
Write2USART (↗ see page 1155)
baud2USART (↗ see page 1157)
gets2USART (↗ see page 1159)
putrs2USART (↗ see page 1161)
puts2USART (↗ see page 1162)
Read2USART (↗ see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([↗](#) see page 1164)

7.15.13 MWIRE (87J50 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire1 ([see page 1166](#))

ReadMWire1 ([see page 1170](#))

WriteMWire1 ([see page 1169](#))

getsMWire1 ([see page 1167](#))

OpenMWire2 ([see page 1166](#))

ReadMWire2 ([see page 1171](#))

WriteMWire2 ([see page 1169](#))

getsMWire2 ([see page 1168](#))

Macros

CloseMWire1

getcMWire1

putcMWire1

DataRdyMWire1

CloseMWire2

getcMWire2

putcMWire2

DataRdyMWire2

MWIRE Examples ([see page 1172](#))

7.15.14 Flash (87J50 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash

- Loading table pointers and performing table-read operation

Functions
ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1200)
WriteBytesFlash (🔗 see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.16 18F87J11 Family

PIC18F65J11 / PIC18F66J11 / PIC18F66J16 / PIC18F67J11

PIC18F85J11 / PIC18F86J11 / PIC18F86J16 / PIC18F87J11

7.16.1 ADC (87J11 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (🔗 see page 865)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CALIB
ADC_NO_CALIB

ADC Examples ([see page 898](#))

7.16.2 Analog Comparator (87J11 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp1 (see page 902)
Open_ancomp2 (see page 911)
Close_ancomp1 (see page 923)
Close_ancomp2 (see page 923)

Comparator Examples ([see page 924](#))

7.16.3 Input Capture (87J11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)

ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.16.4 Output Compare (87J11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)
OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.16.5 PWM (87J11 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples (see page 1060)

7.16.6 I2C (87J11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putSI2C1 (see page 1069)
getSI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putSI2C2 (see page 1069)

getsI2C2 (see page 1071)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit

I2C2_Intr_Status

StopI2C2

StartI2C2

RestartI2C2

NotAckI2C2

AckI2C2

DataRdyI2C2

IdleI2C2

CloseI2C2

putcI2C2

getcI2C2

I2C Examples (see page 1072)

7.16.7 I2C EEPROM (87J11 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples ([see page 1088](#))

7.16.8 IO Ports (87J11 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.16.9 SW_RTCC (87J11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.16.10 SPI (87J11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI1 (see page 1109)
WriteSPI1 (see page 1112)
ReadSPI1 (see page 1113)
putsSPI1 (see page 1114)
getsSPI1 (see page 1115)
OpenSPI2 (see page 1110)
WriteSPI2 (see page 1112)

ReadSPI2 (🔗 see page 1113)
putsSPI2 (🔗 see page 1114)
getsSPI2 (🔗 see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (🔗 see page 1116)

7.16.11 Timers (87J11 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (🔗 see page 1119)

ReadTimer0 (🔗 see page 1137)

WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.16.12 USART (87J11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)

Read1USART (🔗 see page 1163)
Open2USART (🔗 see page 1150)
Write2USART (🔗 see page 1155)
baud2USART (🔗 see page 1157)
gets2USART (🔗 see page 1159)
putrs2USART (🔗 see page 1161)
puts2USART (🔗 see page 1162)
Read2USART (🔗 see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (🔗 see page 1164)

7.16.13 MWIRE (87J11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (🔗 see page 1166)
ReadMwire1 (🔗 see page 1170)
WriteMwire1 (🔗 see page 1169)
getsMwire1 (🔗 see page 1167)
OpenMwire2 (🔗 see page 1166)
ReadMwire2 (🔗 see page 1171)
WriteMwire2 (🔗 see page 1169)

getsMwire2 (see page 1168)

Macros

CloseMwire1

getcMwire1

putcMwire1

DataRdyMwire1

CloseMwire2

getcMwire2

putcMwire2

DataRdyMwire2

MWIRE Examples (see page 1172)

7.16.14 Flash (87J11 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.17 18F46K20 Family

PIC18F23K20 / PIC18F24K20 / PIC18F25K20 / PIC18F26K20

PIC18F43K20 / PIC18F44K20 / PIC18F45K20 / PIC18F46K20

7.17.1 ADC (46K20 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 851)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CALIB

ADC_NO_CALIB

ADC Examples (see page 898)

7.17.2 Analog Comparator (46K20 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 900)
Open_ancomp2 (see page 909)
Close_ancomp1 (see page 923)
Close_ancomp2 (see page 923)

Comparator Examples ([see page 924](#))

7.17.3 Input Capture (46K20 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.17.4 Output Compare (46K20 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([see page 1025](#))

7.17.5 PWM (46K20 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.17.6 I2C (46K20 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C

SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.17.7 I2C EEPROM (46K20 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.17.8 EEP (46K20 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM

- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.17.9 IO Ports (46K20 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.17.10 SW_RTCC (46K20 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.17.11 SPI (46K20 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenSPI ([see page 1108](#))WriteSPI ([see page 1111](#))ReadSPI ([see page 1113](#))putsSPI ([see page 1114](#))getsSPI ([see page 1115](#))**Macros**

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.17.12 Timers (46K20 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (🔗 see page 1119)
ReadTimer0 (🔗 see page 1137)
WriteTimer0 (🔗 see page 1139)
CloseTimer0 (🔗 see page 1143)
OpenTimer1 (🔗 see page 1120)
ReadTimer1 (🔗 see page 1138)
WriteTimer1 (🔗 see page 1140)
CloseTimer1 (🔗 see page 1143)
OpenTimer2 (🔗 see page 1123)
CloseTimer2 (🔗 see page 1144)
OpenTimer3 (🔗 see page 1124)
ReadTimer3 (🔗 see page 1138)
WriteTimer3 (🔗 see page 1140)
CloseTimer3 (🔗 see page 1144)
SetTmrCCPSrc (🔗 see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF




Timers Examples (🔗 see page 1146)

7.17.13 USART (46K20 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenUSART ( see page 1152)WriteUSART ( see page 1155)baudUSART ( see page 1158)getsUSART ( see page 1160)putsUSART ( see page 1161)putsUSART ( see page 1163)ReadUSART ( see page 1164)**Macros**

DataRdyUSART

CloseUSART

BusyUSART

putcUSART



getcUSART

USART Examples ( see page 1164)

7.17.14 MWIRE (46K20 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenMwire ( see page 1167)ReadMwire ( see page 1171)WriteMwire ( see page 1170)getsMwire ( see page 1168)**Macros**

CloseMwire

getcMwire
putcMwire
DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.17.15 Flash (46K20 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1199)
WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples ([see page 1204](#))

7.18 18F97J60 Family

PIC18F66J60 / PIC18F66J65 / PIC18F67J60

PIC18F86J60 / PIC18F86J65 / PIC18F87J60

PIC18F96J60 / PIC18F96J65 / PIC18F97J60

7.18.1 PIC18F67J60 Set

PIC18F66J60 / PIC18F66J65 / PIC18F67J60

7.18.1.1 ADC (67J60 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.18.1.2 Analog Comparator (67J60 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.18.1.3 Input Capture (67J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.18.1.4 Output Compare (67J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([see page 1025](#))

7.18.1.5 PWM (67J60 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.18.1.6 I2C (67J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

I2C Examples (see page 1072)

7.18.1.7 I2C EEPROM (67J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)

EEAckPolling1 (see page 1076)

EEByteWrite1 (see page 1080)

EECurrentAddRead1 (see page 1078)

EEPageWrite1 (see page 1082)

EERandomRead1 (see page 1086)

EESequentialRead1 (see page 1084)

I2C EEPROM Examples (see page 1088)

7.18.1.8 IO Ports (67J60 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.18.1.9 SW_RTCC (67J60 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.18.1.10 SPI (67J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2
putcSPI2

SPI Examples ([see page 1116](#))

7.18.1.11 Timers (67J60 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples ([see page 1146](#))

7.18.1.12 USART (67J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART ([see page 1148](#))

Write1USART ([see page 1154](#))

baud1USART ([see page 1155](#))

gets1USART ([see page 1159](#))

puts1USART ([see page 1160](#))

puts1USART ([see page 1162](#))

Read1USART ([see page 1163](#))

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

USART Examples ([see page 1164](#))

7.18.1.13 MWIRE (67J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 ([see page 1166](#))

ReadMwire1 ([see page 1170](#))

WriteMwire1 ([see page 1169](#))

getsMwire1 ([see page 1167](#))

OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.18.1.14 Flash (67J60 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.18.2 PIC18F87J60 Set

PIC18F86J60 / PIC18F86J65 / PIC18F87J60

7.18.2.1 ADC (87J60 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC ([see page 848](#))

SetChanADC ([see page 882](#))

SelChanConvADC ([see page 889](#))

ConvertADC ([see page 896](#))

BusyADC ([see page 897](#))

ReadADC ([see page 897](#))

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.18.2.2 Analog Comparator (87J60 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp ([see page 899](#))

Close_ancomp ([see page 923](#))

Comparator Examples ([see page 924](#))

7.18.2.3 Input Capture (87J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.18.2.4 Output Compare (87J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([see page 1025](#))

7.18.2.5 PWM (87J60 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.18.2.6 I2C (87J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

I2C Examples (see page 1072)

7.18.2.7 I2C EEPROM (87J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)

EEAckPolling1 (see page 1076)

EEByteWrite1 (see page 1080)

EECurrentAddRead1 (see page 1078)

EEPageWrite1 (see page 1082)

EERandomRead1 (see page 1086)

EESequentialRead1 (see page 1084)

I2C EEPROM Examples (see page 1088)

7.18.2.8 IO Ports (87J60 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.18.2.9 SW_RTCC (87J60 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.18.2.10 SPI (87J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2
putcSPI2

SPI Examples ([see page 1116](#))

7.18.2.11 Timers (87J60 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples ([see page 1146](#))

7.18.2.12 USART (87J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART ([see page 1148](#))

Write1USART ([see page 1154](#))

baud1USART ([see page 1155](#))

gets1USART ([see page 1159](#))

putrs1USART ([see page 1160](#))

puts1USART ([see page 1162](#))

Read1USART ([see page 1163](#))

Open2USART ([see page 1150](#))

Write2USART ([see page 1155](#))

baud2USART ([see page 1157](#))

gets2USART ([see page 1159](#))

putrs2USART ([see page 1161](#))

puts2USART ([see page 1162](#))

Read2USART ([see page 1163](#))

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART

Busy2USART

putc2USART

getc2USART

USART Examples ([see page 1164](#))

7.18.2.13 MWIRE (87J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire1 (see page 1166)

ReadMWire1 (see page 1170)

WriteMWire1 (see page 1169)

getsMWire1 (see page 1167)

OpenMWire2 (see page 1166)

ReadMWire2 (see page 1171)

WriteMWire2 (see page 1169)

getsMWire2 (see page 1168)

Macros

CloseMWire1

getcMWire1

putcMWire1

DataRdyMWire1

CloseMWire2

getcMWire2

putcMWire2

DataRdyMWire2

MWIRE Examples (see page 1172)

7.18.2.14 Flash (87J60 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1200)
WriteBytesFlash (🔗 see page 1203)

Macros

LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.18.3 PIC18F97J60 Set

PIC18F96J60 / PIC18F96J65 / PIC18F97J60

7.18.3.1 ADC (97J60 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (🔗 see page 848)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.18.3.2 Analog Comparator (97J60 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.18.3.3 Input Capture (97J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.18.3.4 Output Compare (97J60 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([see page 1025](#))

7.18.3.5 PWM (97J60 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.18.3.6 I2C (97J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 ([see page 1064](#))

Writel2C1 ([see page 1066](#))

ReadI2C1 ([see page 1067](#))

putsI2C1 ([see page 1069](#))

getsI2C1 ([see page 1070](#))

OpenI2C2 ([see page 1065](#))

Writel2C2 ([see page 1067](#))

ReadI2C2 ([see page 1068](#))

putsI2C2 ([see page 1069](#))

getsI2C2 ([see page 1071](#))

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.18.3.7 I2C EEPROM (97J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddrRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.18.3.8 IO Ports (97J60 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.18.3.9 SW_RTCC (97J60 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.18.3.10 SPI (97J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.18.3.11 Timers (97J60 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.18.3.12 USART (97J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.18.3.13 MWIRE (97J60 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.

- Provides simple interface macros to enable/disable interrupts

Functions
OpenMWire1 (see page 1166)
ReadMWire1 (see page 1170)
WriteMWire1 (see page 1169)
getsMWire1 (see page 1167)
OpenMWire2 (see page 1166)
ReadMWire2 (see page 1171)
WriteMWire2 (see page 1169)
getsMWire2 (see page 1168)

Macros
CloseMWire1
getcMWire1
putcMWire1
DataRdyMWire1
CloseMWire2
getcMWire2
putcMWire2
DataRdyMWire2

MWIRE Examples (see page 1172)

7.18.3.14 Flash (97J60 Set)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.19 18F87J10 Family

PIC18F65J10 / PIC18F65J15 / PIC18F66J10 / PIC18F66J15 / PIC18F67J10

PIC18F85J10 / PIC18F85J15 / PIC18F86J10 / PIC18F86J15 / PIC18F87J10

7.19.1 ADC (87J10 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE


ADC Examples (see page 898)

7.19.2 Analog Comparator (87J10 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output
















Functions
Open_ancomp ( see page 899)
Close_ancomp ( see page 923)

Comparator Examples ( see page 924)

7.19.3 Input Capture (87J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 ( see page 954)
OpenCapture2 ( see page 955)
OpenCapture3 ( see page 957)
OpenCapture4 ( see page 959)
OpenCapture5 ( see page 963)
ReadCapture1 ( see page 983)
ReadCapture2 ( see page 984)
ReadCapture3 ( see page 984)
ReadCapture4 ( see page 984)
ReadCapture5 ( see page 984)
CloseCapture1 ( see page 987)
CloseCapture2 ( see page 987)
CloseCapture3 ( see page 988)
CloseCapture4 ( see page 988)
CloseCapture5 ( see page 988)

Input Capture Examples ([↗](#) see page 991)

7.19.4 Output Compare (87J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (↗ see page 992)
OpenCompare2 (↗ see page 993)
OpenCompare3 (↗ see page 995)
OpenCompare4 (↗ see page 997)
OpenCompare5 (↗ see page 1000)
CloseCompare1 (↗ see page 1021)
CloseCompare2 (↗ see page 1022)
CloseCompare3 (↗ see page 1022)
CloseCompare4 (↗ see page 1022)
CloseCompare5 (↗ see page 1022)

Output Compare Examples ([↗](#) see page 1025)

7.19.5 PWM (87J10 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (↗ see page 1026)
OpenPWM2 (↗ see page 1027)
OpenPWM3 (↗ see page 1027)
OpenPWM4 (↗ see page 1028)
OpenPWM5 (↗ see page 1030)
SetDCPWM1 (↗ see page 1047)
SetDCPWM2 (↗ see page 1048)
SetDCPWM3 (↗ see page 1048)
SetDCPWM4 (↗ see page 1049)
SetDCPWM5 (↗ see page 1050)

SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples (see page 1060)

7.19.6 I2C (87J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)
WriteI2C1 (see page 1066)
ReadI2C1 (see page 1067)
putsI2C1 (see page 1069)
getsI2C1 (see page 1070)
OpenI2C2 (see page 1065)
WriteI2C2 (see page 1067)
ReadI2C2 (see page 1068)
putsI2C2 (see page 1069)
getsI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1

RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putI2C1
getI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putI2C2
getI2C2

I2C Examples (see page 1072)

7.19.7 I2C EEPROM (87J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)

EEAckPolling1 (see page 1076)

EEByteWrite1 (see page 1080)

EECurrentAddrRead1 (see page 1078)

EEPPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.19.8 IO Ports (87J10 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.19.9 SW_RTCC (87J10 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.19.10 SPI (87J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 ([see page 1109](#))

WriteSPI1 ([see page 1112](#))

ReadSPI1 ([see page 1113](#))

putsSPI1 ([see page 1114](#))

getsSPI1 ([see page 1115](#))

OpenSPI2 ([see page 1110](#))

WriteSPI2 ([see page 1112](#))

ReadSPI2 ([see page 1113](#))

putsSPI2 ([see page 1114](#))

getsSPI2 ([see page 1116](#))

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status
SPI1_Clear_Recv_OV
CloseSPI1
DataRdySPI1
getcSPI1
putcSPI1
EnableIntSPI2
DisableIntSPI2
SetPriorityIntSPI2
SPI2_Clear_Intr_Status_Bit
SPI2_Intr_Status
SPI2_Clear_Recv_OV
CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.19.11 Timers (87J10 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)

ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.19.12 USART (87J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros
DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.19.13 MWIRE (87J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2

getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.19.14 Flash (87J10 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.20 18F85J90/11 Family

PIC18F63J90 / PIC18F64J90 / PIC18F65J90

PIC18F83J90 / PIC18F84J90 / PIC18F85J90

PIC18F63J11 / PIC18F64J11 / PIC18F65J11

PIC18F83J11 / PIC18F84J11 / PIC18F85J11

7.20.1 ADC (85J90/11 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.20.2 Analog Comparator (85J90/11 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.20.3 Input Capture (85J90/11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)

Input Capture Examples ([see page 991](#))

7.20.4 Output Compare (85J90/11 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.20.5 PWM (85J90/11 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

FunctionsOpenPWM1 ([see page 1026](#))OpenPWM2 ([see page 1027](#))SetDCPWM1 ([see page 1047](#))SetDCPWM2 ([see page 1048](#))SetOutputPWM1 ([see page 1043](#))ClosePWM1 ([see page 1056](#))ClosePWM2 ([see page 1056](#))PWM Examples ([see page 1060](#))

7.20.6 I2C (85J90/11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenI2C ([see page 1063](#))WriteI2C ([see page 1066](#))ReadI2C ([see page 1067](#))putsI2C ([see page 1068](#))getsI2C ([see page 1070](#))IdleI2C ([see page 1072](#))CloseI2C ([see page 1071](#))**Macros**

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.20.7 I2C EEPROM (85J90/11 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.20.8 IO Ports (85J90/11 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)
--

OpenRB3INT (see page 1092)
--

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.20.9 SW_RTCC (85J90/11 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)
--

SW_RTCC Examples ([see page 1107](#))

7.20.10 SPI (85J90/11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)
--

ReadSPI (see page 1113)

putsSPI ([see page 1114](#))

getsSPI ([see page 1115](#))

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.20.11 Timers (85J90/11 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1120](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 ([see page 1144](#))

OpenTimer3 ([see page 1124](#))

ReadTimer3 ([see page 1138](#))

WriteTimer3 ([see page 1140](#))

CloseTimer3 (see page 1144)

SetTmrCCPSrc (see page 1141)
--

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples ([see page 1146](#))

7.20.12 USART (85J90/11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
--

Write1USART (see page 1154)

baud1USART (see page 1155)
--

gets1USART (see page 1159)
--

putrs1USART (see page 1160)

puts1USART (see page 1162)
--

Read1USART (see page 1163)
--

Open2USART (see page 1150)
--

Write2USART (see page 1155)

gets2USART (see page 1159)
--

putrs2USART (see page 1161)

puts2USART (see page 1162)
--

Read2USART (see page 1163)
--

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.20.13 MWIRE (85J90/11 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.20.14 Flash (85J90/11 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.21 18F8621 Family

PIC18F6525 / PIC18F6621

PIC18F8525 / PIC18F8621

7.21.1 ADC (8621 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.21.2 Analog Comparator (8621 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

FunctionsOpen_ancomp ([see page 899](#))Close_ancomp ([see page 923](#))Comparator Examples ([see page 924](#))

7.21.3 Input Capture (8621 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))OpenCapture2 ([see page 955](#))OpenCapture3 ([see page 957](#))OpenCapture4 ([see page 959](#))OpenCapture5 ([see page 963](#))ReadCapture1 ([see page 983](#))ReadCapture2 ([see page 984](#))ReadCapture3 ([see page 984](#))ReadCapture4 ([see page 984](#))ReadCapture5 ([see page 984](#))

CloseCapture1 (see page 987)
--

CloseCapture2 (see page 987)
--

CloseCapture3 (see page 988)
--

CloseCapture4 (see page 988)
--

CloseCapture5 (see page 988)
--

Input Capture Examples ([see page 991](#))

7.21.4 Output Compare (8621 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

OpenCompare2 (see page 993)

OpenCompare3 (see page 995)

OpenCompare4 (see page 997)

OpenCompare5 (see page 1000)
--

CloseCompare1 (see page 1021)

CloseCompare2 (see page 1022)

CloseCompare3 (see page 1022)

CloseCompare4 (see page 1022)

CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.21.5 PWM (8621 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)
--

OpenPWM2 (see page 1027)
--

OpenPWM3 (see page 1027)
--

OpenPWM4 (see page 1028)
--

OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples (see page 1060)

7.21.6 I2C (8621 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros

EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit

I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.21.7 I2C EEPROM (8621 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.21.8 EEP (8621 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.21.9 IO Ports (8621 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))OpenRB3INT ([see page 1092](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.21.10 SW_RTCC (8621 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.21.11 SPI (8621 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenSPI ([see page 1108](#))WriteSPI ([see page 1111](#))ReadSPI ([see page 1113](#))putsSPI ([see page 1114](#))getsSPI ([see page 1115](#))**Macros**

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.21.12 Timers (8621 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.21.13 USART (8621 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
putrs1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
baud2USART (see page 1157)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.21.14 MWIRE (8621 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.21.15 Flash (8621Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))

EraseFlash ([see page 1196](#))

WriteBlockFlash ([see page 1198](#))

WriteBytesFlash ([see page 1202](#))

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.22 18F8680 Family

PIC18F6585 / PIC18F6680

PIC18F8585 / PIC18F8680

7.22.1 ADC (8680 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.22.2 Analog Comparator (8680 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.22.3 Input Capture (8680 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.22.4 Output Compare (8680 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([↗](#) see page 1025)

7.22.5 PWM (8680 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (↗ see page 1026)
OpenPWM2 (↗ see page 1027)
SetDCPWM1 (↗ see page 1047)
SetDCPWM2 (↗ see page 1048)
SetOutputPWM1 (↗ see page 1043)
ClosePWM1 (↗ see page 1056)
ClosePWM2 (↗ see page 1056)

PWM Examples ([↗](#) see page 1060)

7.22.6 I2C (8680 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (↗ see page 1063)
WriteI2C (↗ see page 1066)
ReadI2C (↗ see page 1067)
putsI2C (↗ see page 1068)
getsI2C (↗ see page 1070)
IdleI2C (↗ see page 1072)
CloseI2C (↗ see page 1071)

Macros
EnableIntI2C

DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.22.7 I2C EEPROM (8680 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.22.8 SPI (8680 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.22.9 EEP (8680 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)

Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.22.10 SW_RTCC (8680 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.22.11 IO Ports (8680 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.22.12 Timers (8680 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples ([see page 1146](#))

7.22.13 USART (8680 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.22.14 MWIRE (8680 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.22.15 Flash (8680 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1198)
WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.23 18F8722 Family

PIC18F66J90 / PIC18F67J90 / PIC18F86J90 / PIC18F87J90

7.23.1 ADC (8722 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.

- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC ([see page 848](#))

SetChanADC ([see page 882](#))

SetChanConvADC ([see page 889](#))

ConvertADC ([see page 896](#))

BusyADC ([see page 897](#))

ReadADC ([see page 897](#))

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.23.2 Analog Comparator (8722 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp ([see page 899](#))

Close_ancomp ([see page 923](#))

Comparator Examples ([see page 924](#))

7.23.3 Input Capture (8722 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
OpenCapture4 (see page 959)
OpenCapture5 (see page 963)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.23.4 Output Compare (8722 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)
OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.23.5 PWM (8722 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
SetOutputPWM2 (see page 1043)
SetOutputPWM3 (see page 1044)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.23.6 I2C (8722 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C1 (see page 1064)

Writel2C1 (see page 1066)
Readl2C1 (see page 1067)
putsI2C1 (see page 1069)
getsI2C1 (see page 1070)
OpenI2C2 (see page 1065)
Writel2C2 (see page 1067)
ReadI2C2 (see page 1068)
putsI2C2 (see page 1069)
getsI2C2 (see page 1071)

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1
EnableIntI2C2
DisableIntI2C2
SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2

getI2C2

I2C Examples (see page 1072)

7.23.7 I2C EEPROM (8722 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.23.8 SPI (8722 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.23.9 EEP (8722 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)

Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.23.10 IO Ports (8722 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.23.11 SW_RTCC (8722 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.23.12 Timers (8722 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1120](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 ([see page 1144](#))

OpenTimer3 ([see page 1124](#))

ReadTimer3 ([see page 1138](#))

WriteTimer3 ([see page 1140](#))

CloseTimer3 ([see page 1144](#))

OpenTimer4 ([see page 1130](#))

CloseTimer4 ([see page 1144](#))

SetTmrCCPSrc ([see page 1141](#))

Macros

WriteTimer2

ReadTimer2

WriteTimer4

ReadTimer4

Timers Examples (see page 1146)

7.23.13 USART (8722 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)

Write1USART (see page 1154)

baud1USART (see page 1155)

gets1USART (see page 1159)

puts1USART (see page 1160)

puts1USART (see page 1162)

Read1USART (see page 1163)

Open2USART (see page 1150)

Write2USART (see page 1155)

baud2USART (see page 1157)

gets2USART (see page 1159)

puts2USART (see page 1161)

puts2USART (see page 1162)

Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART

Busy2USART

putc2USART

getc2USART

USART Examples (🔗 see page 1164)

7.23.14 MWIRE (8722 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire1 (🔗 see page 1166)

ReadMWire1 (🔗 see page 1170)

WriteMWire1 (🔗 see page 1169)

getsMWire1 (🔗 see page 1167)

OpenMWire2 (🔗 see page 1166)

ReadMWire2 (🔗 see page 1171)

WriteMWire2 (🔗 see page 1169)

getsMWire2 (🔗 see page 1168)

Macros

CloseMWire1

getcMWire1

putcMWire1

DataRdyMWire1

CloseMWire2

getcMWire2

putcMWire2

DataRdyMWire2

MWIRE Examples (🔗 see page 1172)

7.23.15 Flash (8722 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.24 18F8490 Family

PIC18F6390 / PIC18F6490

PIC18F8390 / PIC18F8490

7.24.1 ADC (8490 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.24.2 Analog Comparator (8490 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output**Functions**Open_ancomp ([see page 899](#))Close_ancomp ([see page 923](#))Comparator Examples ([see page 924](#))

7.24.3 Input Capture (8490 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))ReadCapture1 ([see page 983](#))CloseCapture1 ([see page 987](#))Input Capture Examples ([see page 991](#))

7.24.4 Output Compare (8490 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.24.5 PWM (8490 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

OpenPWM2 (see page 1027)

SetDCPWM1 (see page 1047)

SetDCPWM2 (see page 1048)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.24.6 I2C (8490 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putSI2C (see page 1068)

getSI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putI2C

getI2C

putI2C

I2C Examples (see page 1072)

7.24.7 I2C EEPROM (8490 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.24.8 SPI (8490 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.24.9 SW_RTCC (8490 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (🔗 see page 1106)

Close_RTCC (🔗 see page 1106)

SW_RTCC Examples (🔗 see page 1107)

7.24.10 IO Ports (8490 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (🔗 see page 1089)

OpenRB0INT (🔗 see page 1089)

OpenRB1INT (🔗 see page 1090)

OpenRB2INT (🔗 see page 1091)

OpenRB3INT (🔗 see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (🔗 see page 1093)

7.24.11 Timers (8490 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.24.12 USART (8490 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
Open1USART (see page 1148)
Write1USART (see page 1154)
baud1USART (see page 1155)
gets1USART (see page 1159)
puts1USART (see page 1160)

puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
gets2USART (see page 1159)
putrs2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART
Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (see page 1164)

7.24.13 MWIRE (8490 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.24.14 Flash (8490 Family)

This peripheral module supports functionalities:

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)

Macros
LoadFlashAddr
TableRead

Flash Examples ([see page 1204](#))

7.25 18F8410 Family

PIC18F6310 / PIC18F6410

PIC18F8310 / PIC18F8410

7.25.1 ADC (8410 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.25.2 Analog Comparator (8410 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.25.3 Input Capture (8410 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenCapture2 (see page 955)
OpenCapture3 (see page 957)
ReadCapture1 (see page 983)
ReadCapture2 (see page 984)
ReadCapture3 (see page 984)
CloseCapture1 (see page 987)
CloseCapture2 (see page 987)
CloseCapture3 (see page 988)

Input Capture Examples ([see page 991](#))

7.25.4 Output Compare (8410 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.25.5 PWM (8410 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)

OpenPWM3 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)

PWM Examples (see page 1060)

7.25.6 I2C (8410 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C

DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.25.7 I2C EEPROM (8410 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.25.8 SPI (8410 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (see page 1108)
WriteSPI (see page 1111)
ReadSPI (see page 1113)

putsSPI ([see page 1114](#))

getsSPI ([see page 1115](#))

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.25.9 IO Ports (8410 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.25.10 SW_RTCC (8410 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)
--

SW_RTCC Examples ([see page 1107](#))

7.25.11 Timers (8410 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (see page 1119)
--

ReadTimer0 (see page 1137)
--

WriteTimer0 (see page 1139)

CloseTimer0 (see page 1143)

OpenTimer1 (see page 1120)
--

ReadTimer1 (see page 1138)
--

WriteTimer1 (see page 1140)

CloseTimer1 (see page 1143)

OpenTimer2 (see page 1123)
--

CloseTimer2 (see page 1144)

OpenTimer3 (see page 1124)
--

ReadTimer3 (see page 1138)
--

WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

SetTmrCCPsrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.25.12 USART (8410 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)

Write1USART (see page 1154)

baud1USART (see page 1155)

gets1USART (see page 1159)

putrs1USART (see page 1160)

puts1USART (see page 1162)

Read1USART (see page 1163)

Open2USART (see page 1150)

Write2USART (see page 1155)

gets2USART (see page 1159)

putrs2USART (see page 1161)

puts2USART (see page 1162)

Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples (🔗 see page 1164)

7.25.13 MWIRE (8410 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (🔗 see page 1167)
ReadMwire (🔗 see page 1171)
WriteMwire (🔗 see page 1170)
getsMwire (🔗 see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (🔗 see page 1172)

7.25.14 Flash (8410 Family)

This peripheral module supports functionalities:

- Reading data from flash
- Loading table pointers and performing table-read operation

FunctionsReadFlash ([see page 1198](#))**Macros**

LoadFlashAddr

TableRead

Flash Examples ([see page 1204](#))

7.26 18F4321 Family

PIC18F2221 / PIC18F2321

PIC18F4221 / PIC18F4321

7.26.1 PIC18F4x21 Set

PIC18F4221 / PIC18F4321

7.26.1.1 ADC (4x21 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

FunctionsOpenADC ([see page 848](#))SetChanADC ([see page 882](#))SelChanConvADC ([see page 889](#))ConvertADC ([see page 896](#))BusyADC ([see page 897](#))ReadADC ([see page 897](#))CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.26.1.2 Analog Comparator (4x21 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output**Functions**

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.26.1.3 Input Capture (4x21 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.26.1.4 Output Compare (4x21 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

FunctionsOpenCompare1 ([see page 992](#))CloseCompare1 ([see page 1021](#))Output Compare Examples ([see page 1025](#))

7.26.1.5 PWM (4x21 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

FunctionsOpenPWM1 ([see page 1026](#))OpenPWM2 ([see page 1027](#))SetDCPWM1 ([see page 1047](#))SetDCPWM2 ([see page 1048](#))SetOutputPWM1 ([see page 1043](#))ClosePWM1 ([see page 1056](#))ClosePWM2 ([see page 1056](#))PWM Examples ([see page 1060](#))

7.26.1.6 I2C (4x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenI2C ([see page 1063](#))WriteI2C ([see page 1066](#))ReadI2C ([see page 1067](#))putSI2C ([see page 1068](#))getSI2C ([see page 1070](#))IdleI2C ([see page 1072](#))CloseI2C ([see page 1071](#))

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.26.1.7 I2C EEPROM (4x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.26.1.8 SPI (4x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.26.1.9 IO Ports (4x21 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.26.1.10 SW_RTCC (4x21 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.26.1.11 EEP (4x21 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

7.26.1.12 Timers (4x21 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.26.1.13 USART (4x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)

putrsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.26.1.14 MWIRE (4x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.26.1.15 Flash (4x21 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)

- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

FunctionsReadFlash ([↗](#) see page 1198)EraseFlash ([↗](#) see page 1196)WriteBlockFlash ([↗](#) see page 1198)WriteBytesFlash ([↗](#) see page 1202)**Macros**

LoadFlashAddr

TableRead

Flash Examples ([↗](#) see page 1204)

7.26.2 PIC18F2x21 Set

PIC18F2221 / PIC18F2321

7.26.2.1 ADC (2x21 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

FunctionsOpenADC ([↗](#) see page 848)SetChanADC ([↗](#) see page 882)SelChanConvADC ([↗](#) see page 889)ConvertADC ([↗](#) see page 896)BusyADC ([↗](#) see page 897)ReadADC ([↗](#) see page 897)CloseADC ([↗](#) see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.26.2.2 Analog Comparator (2x21 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output**Functions**

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.26.2.3 Input Capture (2x21 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.26.2.4 Output Compare (2x21 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

FunctionsOpenCompare1 ([see page 992](#))CloseCompare1 ([see page 1021](#))Output Compare Examples ([see page 1025](#))

7.26.2.5 PWM (2x21 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

FunctionsOpenPWM1 ([see page 1026](#))OpenPWM2 ([see page 1027](#))SetDCPWM1 ([see page 1047](#))SetDCPWM2 ([see page 1048](#))SetOutputPWM1 ([see page 1043](#))ClosePWM1 ([see page 1056](#))ClosePWM2 ([see page 1056](#))PWM Examples ([see page 1060](#))

7.26.2.6 I2C (2x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenI2C ([see page 1063](#))WriteI2C ([see page 1066](#))ReadI2C ([see page 1067](#))putSI2C ([see page 1068](#))getSI2C ([see page 1070](#))IdleI2C ([see page 1072](#))CloseI2C ([see page 1071](#))

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.26.2.7 I2C EEPROM (2x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.26.2.8 SPI (2x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.26.2.9 IO Ports (2x21 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples (see page 1093)

7.26.2.10 EEP (2x21 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.26.2.11 SW_RTCC (2x21 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.26.2.12 Timers (2x21 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.

- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.26.2.13 USART (2x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)

getsUSART (see page 1160)
putsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros

DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.26.2.14 MWIRE (2x21 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros

CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.26.2.15 Flash (2x21 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1198)
WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.27 18F4x80/5 Family

PIC18F2682 / PIC18F2685

PIC18F4682 / PIC18F4685

PIC18F2585 / PIC18F2680

PIC18F4585 / PIC18F4680

PIC18F2480 / PIC18F2580

PIC18F4480 / PIC18F4580

7.27.1 PIC18F4x80/5 Set

PIC18F4682 / PIC18F4685

PIC18F4585 / PIC18F4680

PIC18F4480 / PIC18F4580

7.27.1.1 ADC (4x8x Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (↗ see page 848)
SetChanADC (↗ see page 882)
SelChanConvADC (↗ see page 889)
ConvertADC (↗ see page 896)
BusyADC (↗ see page 897)
ReadADC (↗ see page 897)
CloseADC (↗ see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples ([↗](#) see page 898)

7.27.1.2 Analog Comparator (4x8x Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.27.1.3 Input Capture (4x8x Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
OpenECapture1 (see page 966)
ReadCapture1 (see page 983)
ReadECapture1 (see page 986)
CloseCapture1 (see page 987)
CloseECapture1 (see page 990)

Input Capture Examples ([see page 991](#))

7.27.1.4 Output Compare (4x8x Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenECompare1 (see page 1003)
CloseCompare1 (see page 1021)
CloseECompare1 (see page 1024)

Output Compare Examples ([see page 1025](#))

7.27.1.5 PWM (4x8x Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenEPWM1 (see page 1036)
SetDCPWM1 (see page 1047)
SetDCEPWM1 (see page 1054)
SetOutputPWM1 (see page 1043)
SetOutputEPWM1 (see page 1045)
ClosePWM1 (see page 1056)
CloseEPWM1 (see page 1059)

PWM Examples ([see page 1060](#))

7.27.1.6 I2C (4x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
Writel2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.27.1.7 I2C EEPROM (4x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.27.1.8 SPI (4x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (↗ see page 1108)
WriteSPI (↗ see page 1111)
ReadSPI (↗ see page 1113)
putsSPI (↗ see page 1114)
getsSPI (↗ see page 1115)

Macros
EnableIntSPI
DisableIntSPI
SetPriorityIntSPI
SPI_Clear_Intr_Status_Bit
SPI_Intr_Status
SPI_Clear_Recv_OV
CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples ([↗](#) see page 1116)

7.27.1.9 EEP (4x8x Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (↗ see page 1061)
Read_b_eep (↗ see page 1061)
Busy_eep (↗ see page 1062)

EEP Examples ([↗](#) see page 1062)

7.27.1.10 SW_RTCC (4x8x Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.27.1.11 IO Ports (4x8x Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.27.1.12 Timers (4x8x Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

FunctionsOpenTimer0 ([see page 1119](#))

ReadTimer0 (🔗 see page 1137)
WriteTimer0 (🔗 see page 1139)
CloseTimer0 (🔗 see page 1143)
OpenTimer1 (🔗 see page 1120)
ReadTimer1 (🔗 see page 1138)
WriteTimer1 (🔗 see page 1140)
CloseTimer1 (🔗 see page 1143)
OpenTimer2 (🔗 see page 1123)
CloseTimer2 (🔗 see page 1144)
OpenTimer3 (🔗 see page 1124)
ReadTimer3 (🔗 see page 1138)
WriteTimer3 (🔗 see page 1140)
CloseTimer3 (🔗 see page 1144)
SetTmrCCPSrc (🔗 see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (🔗 see page 1146)

7.27.1.13 USART (4x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (🔗 see page 1152)
WriteUSART (🔗 see page 1155)
baudUSART (🔗 see page 1158)
getsUSART (🔗 see page 1160)
putrsUSART (🔗 see page 1161)
putsUSART (🔗 see page 1163)
ReadUSART (🔗 see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.27.1.14 MWIRE (4x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.27.1.15 Flash (4x80/5 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1200)
WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.27.2 PIC18F2x80/5 Set

PIC18F2682 / PIC18F2685

PIC18F2585 / PIC18F2680

PIC18F2480 / PIC18F2580

7.27.2.1 ADC (2x8x Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (🔗 see page 848)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.27.2.2 Input Capture (2x8x Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.27.2.3 Output Compare (2x8x Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.27.2.4 PWM (2x8x Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

SetDCPWM1 (see page 1047)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

PWM Examples (see page 1060)

7.27.2.5 I2C (2x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putI2C (see page 1068)

getI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putI2C

getI2C

putI2C

I2C Examples (see page 1072)

7.27.2.6 I2C EEPROM (2x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.27.2.7 SPI (2x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (see page 1108)
WriteSPI (see page 1111)
ReadSPI (see page 1113)
putsSPI (see page 1114)
getsSPI (see page 1115)

Macros
EnableIntSPI
DisableIntSPI

SetPriorityIntSPI
SPI_Clear_Intr_Status_Bit
SPI_Intr_Status
SPI_Clear_Recv_OV
CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.27.2.8 EEP (2x8x Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.27.2.9 IO Ports (2x8x Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)

Macros
EnablePullups

DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.27.2.10 SW_RTCC (2x8x Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.27.2.11 Timers (2x8x Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)

WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.27.2.12 USART (2x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.27.2.13 MWIRE (2x8x Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.27.2.14 Flash (2x80/5 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))

EraseFlash ([see page 1196](#))

WriteBlockFlash ([see page 1200](#))

WriteBytesFlash ([see page 1202](#))

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.28 18F4610 Family

PIC18F2410 / PIC18F2510 / PIC18F2515 / PIC18F2610

PIC18F4410 / PIC18F4510 / PIC18F4515 / PIC18F4610

7.28.1 PIC18F4610 Set

PIC18F4410 / PIC18F4510 / PIC18F4515 / PIC18F4610

7.28.1.1 ADC (4610 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.28.1.2 Analog Comparator (4610 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (🔗 see page 899)
Close_ancomp (🔗 see page 923)

Comparator Examples (🔗 see page 924)

7.28.1.3 Input Capture (4610 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (🔗 see page 954)
ReadCapture1 (🔗 see page 983)
CloseCapture1 (🔗 see page 987)

Input Capture Examples (🔗 see page 991)

7.28.1.4 Output Compare (4610 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (🔗 see page 992)
CloseCompare1 (🔗 see page 1021)

Output Compare Examples (🔗 see page 1025)

7.28.1.5 PWM (4610 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
SetDCPWM1 (see page 1047)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
OpenPWM2 (see page 1027)
SetDCPWM2 (see page 1048)
ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.28.1.6 I2C (4610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit

I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.28.1.7 I2C EEPROM (4610 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.28.1.8 SPI (4610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.28.1.9 IO Ports (4610 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.28.1.10 SW_RTCC (4610 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.28.1.11 Timers (4610 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)

CloseTimer3 (see page 1144)

SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.28.1.12 USART (4610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.28.1.13 MWIRE (4610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.28.1.14 Flash (4610Family)

This peripheral module supports functionalities:

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.28.2 PIC18F2610 Set

PIC18F2410 / PIC18F2510 / PIC18F2515 / PIC18F2610

7.28.2.1 ADC (2610 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.28.2.2 Analog Comparator (2610 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.28.2.3 Input Capture (2610 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules

- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 ([see page 954](#))

ReadCapture1 ([see page 983](#))

CloseCapture1 ([see page 987](#))

Input Capture Examples ([see page 991](#))

7.28.2.4 Output Compare (2610 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 ([see page 992](#))

CloseCompare1 ([see page 1021](#))

Output Compare Examples ([see page 1025](#))

7.28.2.5 PWM (2610 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

OpenPWM2 ([see page 1027](#))

SetDCPWM1 ([see page 1047](#))

SetDCPWM2 ([see page 1048](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 ([see page 1056](#))

ClosePWM2 ([see page 1056](#))

PWM Examples ([see page 1060](#))

7.28.2.6 I2C (2610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.28.2.7 I2C EEPROM (2610 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.28.2.8 SPI (2610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.28.2.9 IO Ports (2610 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.28.2.10 SW_RTCC (2610 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.28.2.11 Timers (2610 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples ([see page 1146](#))

7.28.2.12 USART (2610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.

- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.28.2.13 MWIRE (2610 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (🔗 see page 1172)

7.28.2.14 Flash (2610 Family)

This peripheral module supports functionalities:

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (🔗 see page 1198)

Macros
LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.29 18F4620 Family

PIC18F2525 / PIC18F2620

PIC18F4525 / PIC18F4620

7.29.1 PIC18F4620 Set

PIC18F4525 / PIC18F4620

7.29.1.1 ADC (4620 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (🔗 see page 848)

SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.29.1.2 Analog Comparator (4620 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (see page 899)
Close_ancomp (see page 923)

Comparator Examples ([see page 924](#))

7.29.1.3 Input Capture (4620 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.29.1.4 Output Compare (4620 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.29.1.5 PWM (4620 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
SetDCPWM1 (see page 1047)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
OpenPWM2 (see page 1027)
SetDCPWM2 (see page 1048)
ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.29.1.6 I2C (4620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)

ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.29.1.7 I2C EEPROM (4620 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.29.1.8 SPI (4620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.29.1.9 EEP (4620 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.29.1.10 IO Ports (4620 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))OpenRB3INT ([see page 1092](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.29.1.11 SW_RTCC (4620 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.29.1.12 Timers (4620 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples ([see page 1146](#))

7.29.1.13 USART (4620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.29.1.14 MWIRE (4620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.29.1.15 Flash (4620Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.29.2 PIC18F2620 Set

PIC18F2525 / PIC18F2620

7.29.2.1 ADC (2620 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.29.2.2 Analog Comparator (2620 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output**Functions**

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.29.2.3 Input Capture (2620 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples ([↗](#) see page 991)

7.29.2.4 Output Compare (2620 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (↗ see page 992)
CloseCompare1 (↗ see page 1021)

Output Compare Examples ([↗](#) see page 1025)

7.29.2.5 I2C (2620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (↗ see page 1063)
WriteI2C (↗ see page 1066)
ReadI2C (↗ see page 1067)
putsI2C (↗ see page 1068)
getsI2C (↗ see page 1070)
IdleI2C (↗ see page 1072)
CloseI2C (↗ see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C

NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.29.2.6 I2C EEPROM (2620 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.29.2.7 SPI (2620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (see page 1108)
WriteSPI (see page 1111)
ReadSPI (see page 1113)

putsSPI ([see page 1114](#))

getsSPI ([see page 1115](#))

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples ([see page 1116](#))

7.29.2.8 PWM (2620 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

OpenPWM2 ([see page 1027](#))

SetDCPWM1 ([see page 1047](#))

SetDCPWM2 ([see page 1048](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 ([see page 1056](#))

ClosePWM2 ([see page 1056](#))

PWM Examples ([see page 1060](#))

7.29.2.9 EEP (2620 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.29.2.10 IO Ports (2620 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))OpenRB3INT ([see page 1092](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.29.2.11 SW_RTCC (2620 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))

update_RTCC (↗ see page 1106)
--

Close_RTCC (↗ see page 1106)

SW_RTCC Examples ([↗](#) see page 1107)

7.29.2.12 Timers (2620 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 (↗ see page 1119)

ReadTimer0 (↗ see page 1137)

WriteTimer0 (↗ see page 1139)
--

CloseTimer0 (↗ see page 1143)
--

OpenTimer1 (↗ see page 1120)

ReadTimer1 (↗ see page 1138)

WriteTimer1 (↗ see page 1140)
--

CloseTimer1 (↗ see page 1143)
--

OpenTimer2 (↗ see page 1123)

CloseTimer2 (↗ see page 1144)
--

OpenTimer3 (↗ see page 1124)

ReadTimer3 (↗ see page 1138)

WriteTimer3 (↗ see page 1140)
--

CloseTimer3 (↗ see page 1144)
--

SetTmrCCPSrc (↗ see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples ([↗](#) see page 1146)

7.29.2.13 USART (2620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART ([see page 1152](#))

WriteUSART ([see page 1155](#))

baudUSART ([see page 1158](#))

getsUSART ([see page 1160](#))

putsUSART ([see page 1161](#))

putsUSART ([see page 1163](#))

ReadUSART ([see page 1164](#))

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples ([see page 1164](#))

7.29.2.14 MWIRE (2620 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.29.2.15 Flash (2620 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples ([see page 1204](#))

7.30 18F4550/23/20 Family

PIC18F2455 / PIC18F2550

PIC18F4455 / PIC18F4550

PIC18F2423 / PIC18F2523

PIC18F4423 / PIC18F4523

PIC18F2420 / PIC18F2520

PIC18F4420 / PIC18F4520

7.30.1 ADC (45xx Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 848)
SetChanADC (see page 882)
SetChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.30.2 Analog Comparator (45xx Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.

- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.30.3 Input Capture (45xx Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.30.4 Output Compare (45xx Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.30.5 PWM (45xx Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
SetDCPWM1 (see page 1047)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
OpenPWM2 (see page 1027)
SetDCPWM2 (see page 1048)
ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.30.6 I2C (45xx Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C

SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.30.7 I2C EEPROM (45xx Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.30.8 SPI (45xx Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.30.9 IO Ports (45xx Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.30.10 EEP (45xx Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)

Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.30.11 SW_RTCC (45xx Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.30.12 Timers (45xx Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.30.13 USART (45xx Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.30.14 MWIRE (45xx Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.30.15 Flash (45xx Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1199)
WriteBytesFlash (see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.31 18F4450 Family

PIC18F2450

PIC18F4450

7.31.1 ADC (4450 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

FunctionsOpenADC ([see page 848](#))SetChanADC ([see page 882](#))SelChanConvADC ([see page 889](#))ConvertADC ([see page 896](#))BusyADC ([see page 897](#))ReadADC ([see page 897](#))CloseADC ([see page 897](#))**Macros**

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.31.2 Input Capture (4450 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))ReadCapture1 ([see page 983](#))CloseCapture1 ([see page 987](#))Input Capture Examples ([see page 991](#))

7.31.3 Output Compare (4450 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.

- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 ([see page 992](#))

CloseCompare1 ([see page 1021](#))

Output Compare Examples ([see page 1025](#))

7.31.4 PWM (4450 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

SetDCPWM1 ([see page 1047](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 ([see page 1056](#))

PWM Examples ([see page 1060](#))

7.31.5 IO Ports (4450 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.31.6 SW_RTCC (4450 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.31.7 Timers (4450 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)

CloseTimer2 (see page 1144)

Macros

WriteTimer2

ReadTimer2

7.31.8 USART (4450 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putrsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.31.9 Flash (4450 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data

- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1199)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.32 18F4320 Family

PIC18F2450

PIC18F4450

7.32.1 ADC (4320 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.32.2 Analog Comparator (4320 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp ([see page 899](#))

Close_ancomp ([see page 923](#))

Comparator Examples ([see page 924](#))

7.32.3 Input Capture (4320 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 ([see page 954](#))

ReadCapture1 ([see page 983](#))

CloseCapture1 ([see page 987](#))

Input Capture Examples ([see page 991](#))

7.32.4 Output Compare (4320 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.32.5 PWM (4320 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

SetDCPWM1 (see page 1047)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

OpenPWM2 (see page 1027)

SetDCPWM2 (see page 1048)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.32.6 I2C (4320 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.32.7 I2C EEPROM (4320 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.32.8 SPI (4320 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.32.9 IO Ports (4320 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB ([see page 1089](#))

OpenRB0INT ([see page 1089](#))

OpenRB1INT ([see page 1090](#))

OpenRB2INT ([see page 1091](#))

OpenRB3INT ([see page 1092](#))

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.32.10 EEP (4320 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.32.11 SW_RTCC (4320 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.32.12 Timers (4320 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1120](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 ([see page 1144](#))

OpenTimer3 ([see page 1124](#))

ReadTimer3 ([see page 1138](#))

WriteTimer3 ([see page 1140](#))

CloseTimer3 ([see page 1144](#))

SetTmrCCPSrc ([see page 1141](#))

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.32.13 USART (4320 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.32.14 MWIRE (4320 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire (🔗 see page 1167)

ReadMWire (🔗 see page 1171)

WriteMWire (🔗 see page 1170)

getsMWire (🔗 see page 1168)

Macros

CloseMWire

getcMWire

putcMWire

DataRdyMWire

MWIRE Examples (🔗 see page 1172)

7.32.15 Flash (4320 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)

EraseFlash (🔗 see page 1196)

WriteBlockFlash (🔗 see page 1198)

WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (🔗 see page 1204)

7.33 18F8720 Family

PIC18F6520 / PIC18F6620 / PIC18F6720

PIC18F8520 / PIC18F8620 / PIC18F8720

7.33.1 ADC (8720 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (🔗 see page 842)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.33.2 Analog Comparator (8720 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

FunctionsOpen_ancomp ([see page 899](#))Close_ancomp ([see page 923](#))Comparator Examples ([see page 924](#))

7.33.3 Input Capture (8720 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))OpenCapture2 ([see page 955](#))OpenCapture3 ([see page 957](#))OpenCapture4 ([see page 959](#))OpenCapture5 ([see page 963](#))ReadCapture1 ([see page 983](#))ReadCapture2 ([see page 984](#))ReadCapture3 ([see page 984](#))ReadCapture4 ([see page 984](#))ReadCapture5 ([see page 984](#))CloseCapture1 ([see page 987](#))CloseCapture2 ([see page 987](#))CloseCapture3 ([see page 988](#))CloseCapture4 ([see page 988](#))CloseCapture5 ([see page 988](#))Input Capture Examples ([see page 991](#))

7.33.4 Output Compare (8720 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
OpenCompare2 (see page 993)
OpenCompare3 (see page 995)
OpenCompare4 (see page 997)
OpenCompare5 (see page 1000)
CloseCompare1 (see page 1021)
CloseCompare2 (see page 1022)
CloseCompare3 (see page 1022)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.33.5 PWM (8720 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
OpenPWM3 (see page 1027)
OpenPWM4 (see page 1028)
OpenPWM5 (see page 1030)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetDCPWM3 (see page 1048)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)
ClosePWM3 (see page 1056)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.33.6 I2C (8720 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.33.7 I2C EEPROM (8720 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.33.8 SPI (8720 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.33.9 IO Ports (8720 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.33.10 EEP (8720 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.33.11 SW_RTCC (8720 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.33.12 Timers (8720 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

FunctionsOpenTimer0 ([see page 1119](#))ReadTimer0 ([see page 1137](#))WriteTimer0 ([see page 1139](#))CloseTimer0 ([see page 1143](#))OpenTimer1 ([see page 1120](#))ReadTimer1 ([see page 1138](#))WriteTimer1 ([see page 1140](#))CloseTimer1 ([see page 1143](#))OpenTimer2 ([see page 1123](#))CloseTimer2 ([see page 1144](#))OpenTimer3 ([see page 1124](#))ReadTimer3 ([see page 1138](#))

WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
OpenTimer4 (see page 1130)
CloseTimer4 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4

Timers Examples (see page 1146)

7.33.13 USART (8720 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART (see page 1148)
Write1USART (see page 1154)
gets1USART (see page 1159)
puts1USART (see page 1160)
puts1USART (see page 1162)
Read1USART (see page 1163)
Open2USART (see page 1150)
Write2USART (see page 1155)
gets2USART (see page 1159)
puts2USART (see page 1161)
puts2USART (see page 1162)
Read2USART (see page 1163)

Macros

DataRdy1USART

Close1USART
Busy1USART
putc1USART
getc1USART
DataRdy2USART
Close2USART
Busy2USART
putc2USART
getc2USART

USART Examples ([see page 1164](#))

7.33.14 MWIRE (8720 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.33.15 Flash (8720 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))

- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1198)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.34 18C858 Family

PIC18C658 / PIC18C858

7.34.1 ADC (858 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 842)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.34.2 Analog Comparator (858 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (see page 899)

Close_ancomp (see page 923)

Comparator Examples (see page 924)

7.34.3 Input Capture (858 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.34.4 Output Compare (858 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.34.5 PWM (858 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

SetDCPWM1 (see page 1047)

SetOutputPWM1 (see page 1043)

ClosePWM1 (see page 1056)

OpenPWM2 (see page 1027)

SetDCPWM2 (see page 1048)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.34.6 I2C (858 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getI2C (see page 1070)
--

IdleI2C (see page 1072)

CloseI2C (see page 1071)
--

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putI2C

getI2C

putI2C

I2C Examples ([see page 1072](#))

7.34.7 I2C EEPROM (858 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)
--

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)
--

EESequentialRead (see page 1083)
--

I2C EEPROM Examples (see page 1088)

7.34.8 SPI (858 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.34.9 IO Ports (858 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))OpenRB3INT ([see page 1092](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.34.10 SW_RTCC (858 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.34.11 Timers (858 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.34.12 USART (858 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)

ReadUSART (🔗 see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (🔗 see page 1164)

7.34.13 MWIRE (858 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (🔗 see page 1167)
ReadMwire (🔗 see page 1171)
WriteMwire (🔗 see page 1170)
getsMwire (🔗 see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (🔗 see page 1172)

7.35 18C801 Family

PIC18C601 / PIC18C801

7.35.1 ADC (801 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC ([see page 842](#))

SetChanADC ([see page 882](#))

SelChanConvADC ([see page 889](#))

ConvertADC ([see page 896](#))

BusyADC ([see page 897](#))

ReadADC ([see page 897](#))

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.35.2 Input Capture (801 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 ([see page 954](#))

ReadCapture1 ([see page 983](#))

CloseCapture1 ([see page 987](#))

Input Capture Examples ([see page 991](#))

7.35.3 Output Compare (801 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.35.4 PWM (801 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
SetDCPWM1 (see page 1047)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
OpenPWM2 (see page 1027)
SetDCPWM2 (see page 1048)
ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.35.5 I2C (801 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.35.6 I2C EEPROM (801 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.35.7 SPI (801 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.35.8 IO Ports (801 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.35.9 SW_RTCC (801 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.35.10 Timers (801 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

7.35.11 USART (801 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.35.12 MWIRE (801 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.36 18F4539 Family

PIC18F2439 / PIC18F2539

PIC18F4439 / PIC18F4539

7.36.1 ADC (4539 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 840)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros

ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.36.2 I2C (4539 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

Writel2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.36.3 I2C EEPROM (4539 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.36.4 SPI (4539 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)
WriteSPI (see page 1111)
ReadSPI (see page 1113)
putsSPI (see page 1114)
getsSPI (see page 1115)

Macros

EnableIntSPI
DisableIntSPI
SetPriorityIntSPI
SPI_Clear_Intr_Status_Bit
SPI_Intr_Status
SPI_Clear_Recv_OV
CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.36.5 IO Ports (4539 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.36.6 EEP (4539 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)

Busy_eep (see page 1062)

7.36.7 SW_RTCC (4539 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.36.8 Timers (4539 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.36.9 USART (4539 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.36.10 MWIRE (4539 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMWire (see page 1167)

ReadMWire (see page 1171)

WriteMWire (see page 1170)

getsMWire (see page 1168)

Macros

CloseMWire

getcMWire

putcMWire

DataRdyMWire

MWIRE Examples (see page 1172)

7.36.11 Flash (4539 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1198)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.37 18F452 Family

PIC18F242 / PIC18F252

PIC18F442 / PIC18F452

7.37.1 ADC (F452 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (🔗 see page 840)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.37.2 Input Capture (F452 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

FunctionsOpenCapture1 ([see page 954](#))ReadCapture1 ([see page 983](#))CloseCapture1 ([see page 987](#))Input Capture Examples ([see page 991](#))

7.37.3 Output Compare (F452 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

FunctionsOpenCompare1 ([see page 992](#))CloseCompare1 ([see page 1021](#))Output Compare Examples ([see page 1025](#))

7.37.4 PWM (F452 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

FunctionsOpenPWM1 ([see page 1026](#))SetDCPWM1 ([see page 1047](#))SetOutputPWM1 ([see page 1043](#))ClosePWM1 ([see page 1056](#))OpenPWM2 ([see page 1027](#))SetDCPWM2 ([see page 1048](#))ClosePWM2 ([see page 1056](#))PWM Examples ([see page 1060](#))

7.37.5 I2C (F452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

WriteI2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

CloseI2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcI2C

getcI2C

putcI2C

I2C Examples (see page 1072)

7.37.6 I2C EEPROM (F452 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions

OpenI2C (see page 1063)

EEAckPolling (see page 1075)

EEByteWrite (see page 1079)

EECurrentAddrRead (see page 1077)

EEPageWrite (see page 1081)

EERandomRead (see page 1086)

EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.37.7 SPI (F452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.37.8 IO Ports (F452 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.37.9 EEP (F452 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.37.10 SW_RTCC (F452 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

FunctionsOpen_RTCC ([see page 1106](#))update_RTCC ([see page 1106](#))Close_RTCC ([see page 1106](#))SW_RTCC Examples ([see page 1107](#))

7.37.11 Timers (F452 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

FunctionsOpenTimer0 ([see page 1119](#))ReadTimer0 ([see page 1137](#))WriteTimer0 ([see page 1139](#))CloseTimer0 ([see page 1143](#))OpenTimer1 ([see page 1120](#))ReadTimer1 ([see page 1138](#))WriteTimer1 ([see page 1140](#))CloseTimer1 ([see page 1143](#))OpenTimer2 ([see page 1123](#))CloseTimer2 ([see page 1144](#))

OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.37.12 USART (F452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.37.13 MWIRE (F452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire ([see page 1167](#))

ReadMwire ([see page 1171](#))

WriteMwire ([see page 1170](#))

getsMwire ([see page 1168](#))

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples ([see page 1172](#))

7.37.14 Flash (F452 Family)

This peripheral module supports functionalities:

- Erasing Flash ([see page 1196](#))
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))

EraseFlash ([see page 1196](#))

WriteBlockFlash ([see page 1198](#))

WriteBytesFlash ([see page 1202](#))

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.38 18C452 Family

PIC18C242 / PIC18C252

PIC18C442 / PIC18C452

7.38.1 ADC (C452 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 840)
SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.38.2 Input Capture (C452 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 ([see page 954](#))

ReadCapture1 ([see page 983](#))

CloseCapture1 ([see page 987](#))

Input Capture Examples ([see page 991](#))

7.38.3 Output Compare (C452 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 ([see page 992](#))

CloseCompare1 ([see page 1021](#))

Output Compare Examples ([see page 1025](#))

7.38.4 PWM (C452 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

SetDCPWM1 ([see page 1047](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 (see page 1056)

OpenPWM2 (see page 1027)

SetDCPWM2 (see page 1048)

ClosePWM2 (see page 1056)

PWM Examples (see page 1060)

7.38.5 I2C (C452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C (see page 1063)

Writel2C (see page 1066)

ReadI2C (see page 1067)

putsI2C (see page 1068)

getsI2C (see page 1070)

IdleI2C (see page 1072)

Closel2C (see page 1071)

Macros

EnableIntI2C

DisableIntI2C

SetPriorityIntI2C

I2C_Clear_Intr_Status_Bit

I2C_Intr_Status

StopI2C

StartI2C

RestartI2C

NotAckI2C

AckI2C

DataRdyI2C

putcl2C

getcl2C

putcl2C

I2C Examples ([see page 1072](#))

7.38.6 I2C EEPROM (C452 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples ([see page 1088](#))

7.38.7 SPI (C452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (see page 1108)
WriteSPI (see page 1111)
ReadSPI (see page 1113)
putsSPI (see page 1114)
getsSPI (see page 1115)

Macros
EnableIntSPI
DisableIntSPI
SetPriorityIntSPI
SPI_Clear_Intr_Status_Bit
SPI_Intr_Status
SPI_Clear_Recv_OV
CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.38.8 IO Ports (C452 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples (see page 1093)

7.38.9 SW_RTCC (C452 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.38.10 Timers (C452 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1120](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 ([see page 1144](#))

OpenTimer3 ([see page 1124](#))

ReadTimer3 ([see page 1138](#))

WriteTimer3 ([see page 1140](#))

CloseTimer3 ([see page 1144](#))

SetTmrCCPSrc ([see page 1141](#))

Macros

WriteTimer2

ReadTimer2

T3_OSC1EN_ON

T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.38.11 USART (C452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

getsUSART (see page 1160)

putsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.38.12 MWIRE (C452 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.39 18F458 Family

PIC18F248 / PIC18F258

PIC18F448 / PIC18F458

7.39.1 PIC18F258 Set

PIC18F248 / PIC18F258

7.39.1.1 ADC (258 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 840)

SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (see page 898)

7.39.1.2 Input Capture (258 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.39.1.3 Output Compare (258 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.39.1.4 PWM (258 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
SetDCPWM1 (see page 1047)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)

PWM Examples (see page 1060)

7.39.1.5 I2C (258 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C

RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putcI2C
getcI2C
putcI2C

I2C Examples (see page 1072)

7.39.1.6 I2C EEPROM (258 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.39.1.7 SPI (258 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenSPI (see page 1108)
WriteSPI (see page 1111)

ReadSPI (see page 1113)
putsSPI (see page 1114)
getsSPI (see page 1115)

Macros
EnableIntSPI
DisableIntSPI
SetPriorityIntSPI
SPI_Clear_Intr_Status_Bit
SPI_Intr_Status
SPI_Clear_Recv_OV
CloseSPI
DataRdySPI
getcSPI
putcSPI

SPI Examples (see page 1116)

7.39.1.8 IO Ports (258 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.39.1.9 EEP (258 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples ([see page 1062](#))

7.39.1.10 SW_RTCC (258 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.39.1.11 Timers (258 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)

WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.39.1.12 USART (258 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.39.1.13 MWIRE (258 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros
CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.39.1.14 Flash (258 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1198)
WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.39.2 PIC18F458 Set

PIC18F448 / PIC18F458

7.39.2.1 ADC (458 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (🔗 see page 840)
SetChanADC (🔗 see page 882)
SelChanConvADC (🔗 see page 889)
ConvertADC (🔗 see page 896)
BusyADC (🔗 see page 897)
ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE
ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.39.2.2 Analog Comparator (258 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp (🔗 see page 899)
Close_ancomp (🔗 see page 923)

Comparator Examples (🔗 see page 924)

7.39.2.3 Input Capture (458 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (🔗 see page 954)
OpenECapture1 (🔗 see page 966)
ReadCapture1 (🔗 see page 983)
ReadECapture1 (🔗 see page 986)
CloseCapture1 (🔗 see page 987)
CloseECapture1 (🔗 see page 990)

Input Capture Examples (🔗 see page 991)

7.39.2.4 Output Compare (458 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (🔗 see page 992)
OpenECompare1 (🔗 see page 1003)

CloseCompare1 (see page 1021)

CloseECompare1 (see page 1024)

Output Compare Examples (see page 1025)

7.39.2.5 PWM (458 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenEPWM1 (see page 1036)
SetDCPWM1 (see page 1047)
SetDCEPWM1 (see page 1054)
SetOutputPWM1 (see page 1043)
SetOutputEPWM1 (see page 1045)
ClosePWM1 (see page 1056)
CloseEPWM1 (see page 1059)

PWM Examples (see page 1060)

7.39.2.6 I2C (458 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
getsI2C (see page 1070)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
StopI2C
StartI2C
RestartI2C
NotAckI2C
AckI2C
DataRdyI2C
putI2C
getI2C
putI2C

I2C Examples (see page 1072)

7.39.2.7 I2C EEPROM (458 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C (see page 1063)
EEAckPolling (see page 1075)
EEByteWrite (see page 1079)
EECurrentAddrRead (see page 1077)
EEPageWrite (see page 1081)
EERandomRead (see page 1086)
EESequentialRead (see page 1083)

I2C EEPROM Examples (see page 1088)

7.39.2.8 SPI (458 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.39.2.9 IO Ports (458 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.39.2.10 EEP (458 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions

Write_b_eep (see page 1061)

Read_b_eep (see page 1061)

Busy_eep (see page 1062)

EEP Examples (see page 1062)

7.39.2.11 SW_RTCC (458 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)

update_RTCC (see page 1106)

Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.39.2.12 Timers (458 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers

- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.39.2.13 USART (458 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)

getsUSART (see page 1160)
putsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros

DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.39.2.14 MWIRE (458 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)
ReadMwire (see page 1171)
WriteMwire (see page 1170)
getsMwire (see page 1168)

Macros

CloseMwire
getcMwire
putcMwire
DataRdyMwire

MWIRE Examples (see page 1172)

7.39.2.15 Flash (458 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1198)

WriteBytesFlash (see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.40 18F1320 Family

PIC18F1220 / PIC18F1320

7.40.1 ADC (1320 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 844)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (see page 898)

7.40.2 Input Capture (1320 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (see page 954)

ReadCapture1 (see page 983)

CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.40.3 Output Compare (1320 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (see page 992)

CloseCompare1 (see page 1021)

Output Compare Examples (see page 1025)

7.40.4 PWM (1320 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 (see page 1026)

SetDCPWM1 (see page 1047)

ClosePWM1 (see page 1056)

PWM Examples (see page 1060)

7.40.5 IO Ports (1320 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

OpenRB3INT (see page 1092)

Macros

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

CloseRB3INT

IO Ports Examples (see page 1093)

7.40.6 EEP (1320 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM

- Function to check the status of the EEPROM

Functions

Write_b_eep ([see page 1061](#))

Read_b_eep ([see page 1061](#))

Busy_eep ([see page 1062](#))

EEP Examples ([see page 1062](#))

7.40.7 SW_RTCC (1320 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.40.8 Timers (1320 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1120](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 (see page 1144)
OpenTimer3 (see page 1124)
ReadTimer3 (see page 1138)
WriteTimer3 (see page 1140)
CloseTimer3 (see page 1144)
SetTmrCCPSrc (see page 1141)

Macros
WriteTimer2
ReadTimer2
T3_OSC1EN_ON
T3_OSC1EN_OFF

Timers Examples (see page 1146)

7.40.9 USART (1320 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (🔗 see page 1164)

7.40.10 Flash (1320 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (🔗 see page 1198)
EraseFlash (🔗 see page 1196)
WriteBlockFlash (🔗 see page 1198)
WriteBytesFlash (🔗 see page 1202)

Macros
LoadFlashAddr
TableRead

Flash Examples (🔗 see page 1204)

7.41 18F45J10 Family

PIC18F24J10 / PIC18F25J10

PIC18F44J10 / PIC18F45J10

7.41.1 PIC18F45J10 Set

PIC18F44J10 / PIC18F45J10

7.41.1.1 ADC (45J10 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular

device pinout.

- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC ([see page 848](#))

SetChanADC ([see page 882](#))

SelChanConvADC ([see page 889](#))

ConvertADC ([see page 896](#))

BusyADC ([see page 897](#))

ReadADC ([see page 897](#))

CloseADC ([see page 897](#))

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples ([see page 898](#))

7.41.1.2 Analog Comparator (45J10 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp ([see page 899](#))

Close_ancomp ([see page 923](#))

Comparator Examples ([see page 924](#))

7.41.1.3 Input Capture (45J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples ([see page 991](#))

7.41.1.4 Output Compare (45J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([see page 1025](#))

7.41.1.5 PWM (45J10 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.41.1.6 I2C (45J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 (see page 1064)

Writel2C1 (see page 1066)

ReadI2C1 (see page 1067)

putsI2C1 (see page 1069)

getsI2C1 (see page 1070)

OpenI2C2 (see page 1065)

Writel2C2 (see page 1067)

ReadI2C2 (see page 1068)

putsI2C2 (see page 1069)

getsI2C2 (see page 1071)

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2

I2C2_Clear_Intr_Status_Bit

I2C2_Intr_Status

StopI2C2

StartI2C2

RestartI2C2

NotAckI2C2

AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.41.1.7 I2C EEPROM (45J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddrRead2 (see page 1078)
EEPPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.41.1.8 SPI (45J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.

- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2

putcSPI2

SPI Examples (see page 1116)

7.41.1.9 IO Ports (45J10 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.41.1.10 SW_RTCC (45J10 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.41.1.11 Timers (45J10 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.

- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)

Macros
WriteTimer2
ReadTimer2

Timers Examples ([see page 1146](#))

7.41.1.12 USART (45J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)
ReadUSART (see page 1164)

Macros
DataRdyUSART
CloseUSART
BusyUSART
putcUSART
getcUSART

USART Examples (see page 1164)

7.41.1.13 MWIRE (45J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenMwire1 (see page 1166)
ReadMwire1 (see page 1170)
WriteMwire1 (see page 1169)
getsMwire1 (see page 1167)
OpenMwire2 (see page 1166)
ReadMwire2 (see page 1171)
WriteMwire2 (see page 1169)
getsMwire2 (see page 1168)

Macros
CloseMwire1
getcMwire1
putcMwire1
DataRdyMwire1
CloseMwire2
getcMwire2
putcMwire2
DataRdyMwire2

MWIRE Examples (see page 1172)

7.41.1.14 Flash (45J10 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (see page 1196)

WriteBlockFlash (see page 1200)

WriteBytesFlash (see page 1203)

Macros

LoadFlashAddr

TableRead

Flash Examples (see page 1204)

7.41.2 PIC18F25J10 Set

PIC18F24J10 / PIC18F25J10

7.41.2.1 ADC (25J10 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 848)

SetChanADC (see page 882)

SelChanConvADC (see page 889)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (🔗 see page 897)

CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC Examples (🔗 see page 898)

7.41.2.2 Analog Comparator (25J10 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions

Open_ancomp (🔗 see page 899)

Close_ancomp (🔗 see page 923)

Comparator Examples (🔗 see page 924)

7.41.2.3 Input Capture (25J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (🔗 see page 954)

ReadCapture1 (🔗 see page 983)

CloseCapture1 (🔗 see page 987)

Input Capture Examples (🔗 see page 991)

7.41.2.4 Output Compare (25J10 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 ([see page 992](#))

CloseCompare1 ([see page 1021](#))

Output Compare Examples ([see page 1025](#))

7.41.2.5 PWM (25J10 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions

OpenPWM1 ([see page 1026](#))

OpenPWM2 ([see page 1027](#))

SetDCPWM1 ([see page 1047](#))

SetDCPWM2 ([see page 1048](#))

SetOutputPWM1 ([see page 1043](#))

ClosePWM1 ([see page 1056](#))

ClosePWM2 ([see page 1056](#))

PWM Examples ([see page 1060](#))

7.41.2.6 I2C (25J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 ([see page 1064](#))

WriteI2C1 ([see page 1066](#))

ReadI2C1 ([see page 1067](#))

putsI2C1 ([see page 1069](#))

getsI2C1 ([see page 1070](#))

Macros
EnableIntI2C1
DisableIntI2C1
SetPriorityIntI2C1
I2C1_Clear_Intr_Status_Bit
I2C1_Intr_Status
StopI2C1
StartI2C1
RestartI2C1
NotAckI2C1
AckI2C1
DataRdyI2C1
IdleI2C1
CloseI2C1
putcI2C1
getcI2C1

I2C Examples (see page 1072)

7.41.2.7 I2C EEPROM (25J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)

I2C EEPROM Examples (see page 1088)

7.41.2.8 SPI (25J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2

DataRdySPI2

getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.41.2.9 IO Ports (25J10 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)

Macros

EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples (see page 1093)

7.41.2.10 SW_RTCC (25J10 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.41.2.11 Timers (25J10 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)

Macros
WriteTimer2
ReadTimer2

Timers Examples ([see page 1146](#))

7.41.2.12 USART (25J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)

putrsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.41.2.13 MWIRE (25J10 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire1 (see page 1166)

ReadMwire1 (see page 1170)

WriteMwire1 (see page 1169)

getsMwire1 (see page 1167)

OpenMwire2 (see page 1166)

ReadMwire2 (see page 1171)

WriteMwire2 (see page 1169)

getsMwire2 (see page 1168)

Macros

CloseMwire1

getcMwire1

putcMwire1

DataRdyMwire1

CloseMwire2

getcMwire2

putcMwire2

DataRdyMwire2

MWIRE Examples (see page 1172)

7.41.2.14 Flash (25J10 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples (see page 1204)

7.42 18F1330 Family

PIC18F1230 / PIC18F1330

7.42.1 ADC (1330 Family)

This peripheral module supports functionalities:

- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions
OpenADC (see page 846)

SetChanADC (see page 882)
SelChanConvADC (see page 889)
ConvertADC (see page 896)
BusyADC (see page 897)
ReadADC (see page 897)
CloseADC (see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_SEVT_ENABLE
ADC_SEVT_DISABLE

ADC Examples ([see page 898](#))

7.42.2 PCPWM (1330 Family)

This peripheral module supports functionalities:

- Up to eight PWM I/O pins with four duty cycle generators. Pins can be paired to get a complete half-bridge control.
- Up to 14-bit resolution, depending upon the PWM period.
- On-the-fly PWM frequency changes.

Functions
Openpcpwm (see page 1174)
Setdc0pcpwm (see page 1182)
Setdc1pcpwm (see page 1182)
Setdc2pcpwm (see page 1183)
pcpwm_OVD_CTRL (see page 1184)
pcpwm_OVD_IO_STA (see page 1184)
pcpwm_dt_clk_source (see page 1185)
pcpwm_dt_assignment (see page 1185)
Closepcpwm (see page 1186)

Macros
PCPWM_TMR_INT_EN
PCPWM_TMR_INT_DIS
BRK_FLT_EN
BRK_FLT_DIS

FLT_A_CY_CY
FLT_A_CATAS
FLT_A_EN
FLT_A_DIS

7.42.3 EEP (1330 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples ([see page 1062](#))

7.42.4 IO Ports (1330 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)
OpenRB3INT (see page 1092)

Macros
EnablePullups
DisablePullups
ClosePORTB

CloseRB0INT
CloseRB1INT
CloseRB2INT
CloseRB3INT

IO Ports Examples ([see page 1093](#))

7.42.5 SW_RTCC (1330 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.42.6 Timers (1330 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)

Timers Examples ([see page 1146](#))

7.42.7 USART (1330 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putrsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.42.8 Flash (1330 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (see page 1198)

EraseFlash (🔗 see page 1196)

WriteBlockFlash (🔗 see page 1198)

WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (🔗 see page 1204)

7.43 18F4431 Family

PIC18F2331 / PIC18F2431

PIC18F4331 / PIC18F4431

7.43.1 PIC18F4431 Set

PIC18F4331 / PIC18F4431

7.43.1.1 ADC (4431 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (🔗 see page 854)

SetChanADC (🔗 see page 885)

SelChanConvADC (🔗 see page 892)

ConvertADC (🔗 see page 896)

BusyADC (🔗 see page 897)

ReadADC (🔗 see page 897)

CloseADC (🔗 see page 897)

Macros
ADC_INT_ENABLE
ADC_INT_DISABLE
ADC_CH_GRA_AN0()
ADC_CH_GRA_AN4()
ADC_CH_GRA_AN8()
ADC_CH_GRB_AN1()
ADC_CH_GRB_AN5()
ADC_CH_GRC_AN2()
ADC_CH_GRC_AN6()
ADC_CH_GRD_AN3()
ADC_CH_GRD_AN7()
ALL_CH_DIGITAL()

ADC Examples (see page 898)

7.43.1.2 Input Capture (4431 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenCapture1 (see page 954)
ReadCapture1 (see page 983)
CloseCapture1 (see page 987)

Input Capture Examples (see page 991)

7.43.1.3 Output Compare (4431 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions
OpenCompare1 (see page 992)
CloseCompare1 (see page 1021)

Output Compare Examples ([↗](#) see page 1025)

7.43.1.4 PWM (4431 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (↗ see page 1026)
OpenPWM2 (↗ see page 1027)
SetDCPWM1 (↗ see page 1047)
SetDCPWM2 (↗ see page 1048)
SetOutputPWM1 (↗ see page 1043)
ClosePWM1 (↗ see page 1056)
ClosePWM2 (↗ see page 1056)

PWM Examples ([↗](#) see page 1060)

7.43.1.5 PCPWM (4431 Family)

This peripheral module supports functionalities:

- Up to eight PWM I/O pins with four duty cycle generators. Pins can be paired to get a complete half-bridge control.
- Up to 14-bit resolution, depending upon the PWM period.
- On-the-fly PWM frequency changes.

Functions
Openpcpwm (↗ see page 1178)
Setdc0pcpwm (↗ see page 1182)
Setdc1pcpwm (↗ see page 1182)
Setdc2pcpwm (↗ see page 1183)
Setdc3pcpwm (↗ see page 1183)
pcpwm_OVD_CTRL (↗ see page 1184)
pcpwm_OVD_IO_STA (↗ see page 1184)
pcpwm_dt_clk_source (↗ see page 1185)
pcpwm_dt_assignment (↗ see page 1185)
Closepcpwm (↗ see page 1186)

Macros
PCPWM_TMR_INT_EN

PCPWM_TMR_INT_DIS
BRK_FLT_EN
BRK_FLT_DIS
FLT_A_CY_CY
FLT_A_CATAS
FLT_A_EN
FLT_A_DIS
FLT_B_CY_CY
FLT_B_CATAS
FLT_B_EN
FLT_B_DIS
FLT_AB_DEACT_ALL
FLT_AB_DEACT_0TO5

7.43.1.6 I2C (4431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Supports only **slave mode** in SSP Module.
- Master mode can be implemented in Firmware.

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putSI2C (see page 1068)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
putCI2C

getcI2C

I2C Examples (see page 1072)

7.43.1.7 SPI (4431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.43.1.8 IO Ports (4431 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

FunctionsOpenPORTB ([see page 1089](#))OpenRB0INT ([see page 1089](#))OpenRB1INT ([see page 1090](#))OpenRB2INT ([see page 1091](#))**Macros**

EnablePullups

DisablePullups

ClosePORTB

CloseRB0INT

CloseRB1INT

CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.43.1.9 EEP (4431 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.43.1.10 Timers (4431 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator

FunctionsOpenTimer0 ([see page 1119](#))

ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer5 (see page 1127)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)

Macros
WriteTimer2
ReadTimer2

Timers Examples (see page 1146)

7.43.1.11 SW_RTCC (4431 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples (see page 1107)

7.43.1.12 USART (4431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenUSART (see page 1152)

WriteUSART (see page 1155)

baudUSART (see page 1158)

getsUSART (see page 1160)

putrsUSART (see page 1161)

putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.43.1.13 MWIRE (4431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.43.1.14 Flash (4431 Family)

This peripheral module supports functionalities:

- Erasing Flash (🔗 see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash (🔗 see page 1198)

EraseFlash (🔗 see page 1196)

WriteBlockFlash (🔗 see page 1198)

WriteBytesFlash (🔗 see page 1202)

Macros

LoadFlashAddr

TableRead

Flash Examples (🔗 see page 1204)

7.43.2 PIC18F2431 Set

PIC18F2331 / PIC18F2431

7.43.2.1 ADC (2431 Family)

This peripheral module supports functionalities:

- Up to 16 analog input pins, designated AN0 through AN15, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (🔗 see page 855)

SetChanADC (🔗 see page 886)

SelChanConvADC (🔗 see page 893)

ConvertADC (🔗 see page 896)

BusyADC (🔗 see page 897)

ReadADC (🔗 see page 897)
CloseADC (🔗 see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

ADC_CH_GRA_AN0()

ADC_CH_GRA_AN4()

ADC_CH_GRB_AN1()

ADC_CH_GRC_AN2()

ADC_CH_GRD_AN3()

ADC Examples (🔗 see page 898)

7.43.2.2 Input Capture (2431 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions

OpenCapture1 (🔗 see page 954)

ReadCapture1 (🔗 see page 983)

CloseCapture1 (🔗 see page 987)

Input Capture Examples (🔗 see page 991)

7.43.2.3 Output Compare (2431 Family)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available

Functions

OpenCompare1 (🔗 see page 992)

CloseCompare1 (🔗 see page 1021)

Output Compare Examples (🔗 see page 1025)

7.43.2.4 PWM (2431 Family)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenPWM1 (see page 1026)
OpenPWM2 (see page 1027)
SetDCPWM1 (see page 1047)
SetDCPWM2 (see page 1048)
SetOutputPWM1 (see page 1043)
ClosePWM1 (see page 1056)
ClosePWM2 (see page 1056)

PWM Examples ([see page 1060](#))

7.43.2.5 PCPWM (2431 Family)

This peripheral module supports functionalities:

- Up to eight PWM I/O pins with four duty cycle generators. Pins can be paired to get a complete half-bridge control.
- Up to 14-bit resolution, depending upon the PWM period.
- On-the-fly PWM frequency changes.

Functions
Openpcpwm (see page 1178)
Setdc0pcpwm (see page 1182)
Setdc1pcpwm (see page 1182)
Setdc2pcpwm (see page 1183)
pcpwm_OVD_CTRL (see page 1184)
pcpwm_OVD_IO_STA (see page 1184)
pcpwm_dt_clk_source (see page 1185)
pcpwm_dt_assignment (see page 1185)
Closepcpwm (see page 1186)

Macros
PCPWM_TMR_INT_EN
PCPWM_TMR_INT_DIS
BRK_FLT_EN

BRK_FLT_DIS
FLT_A_CY_CY
FLT_A_CATAS
FLT_A_EN
FLT_A_DIS

7.43.2.6 I2C (2431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Supports only **slave mode** in SSP Module.
- Master mode can be implemented in Firmware.

Functions
OpenI2C (see page 1063)
WriteI2C (see page 1066)
ReadI2C (see page 1067)
putsI2C (see page 1068)
IdleI2C (see page 1072)
CloseI2C (see page 1071)

Macros
EnableIntI2C
DisableIntI2C
SetPriorityIntI2C
I2C_Clear_Intr_Status_Bit
I2C_Intr_Status
putcI2C
getcI2C

I2C Examples (see page 1072)

7.43.2.7 SPI (2431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.

- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI (see page 1108)

WriteSPI (see page 1111)

ReadSPI (see page 1113)

putsSPI (see page 1114)

getsSPI (see page 1115)

Macros

EnableIntSPI

DisableIntSPI

SetPriorityIntSPI

SPI_Clear_Intr_Status_Bit

SPI_Intr_Status

SPI_Clear_Recv_OV

CloseSPI

DataRdySPI

getcSPI

putcSPI

SPI Examples (see page 1116)

7.43.2.8 IO Ports (2431 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions

OpenPORTB (see page 1089)

OpenRB0INT (see page 1089)

OpenRB1INT (see page 1090)

OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples ([see page 1093](#))

7.43.2.9 EEP (2431 Family)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

Functions
Write_b_eep (see page 1061)
Read_b_eep (see page 1061)
Busy_eep (see page 1062)

EEP Examples ([see page 1062](#))

7.43.2.10 SW_RTCC (2431 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions
Open_RTCC (see page 1106)
update_RTCC (see page 1106)
Close_RTCC (see page 1106)

SW_RTCC Examples ([see page 1107](#))

7.43.2.11 Timers (2431 Family)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.

- Timer1 oscillator acts as low power oscillator

Functions
OpenTimer0 (see page 1119)
ReadTimer0 (see page 1137)
WriteTimer0 (see page 1139)
CloseTimer0 (see page 1143)
OpenTimer1 (see page 1120)
ReadTimer1 (see page 1138)
WriteTimer1 (see page 1140)
CloseTimer1 (see page 1143)
OpenTimer2 (see page 1123)
CloseTimer2 (see page 1144)
OpenTimer5 (see page 1127)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)

Macros
WriteTimer2
ReadTimer2

Timers Examples ([see page 1146](#))

7.43.2.12 USART (2431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions
OpenUSART (see page 1152)
WriteUSART (see page 1155)
baudUSART (see page 1158)
getsUSART (see page 1160)
putrsUSART (see page 1161)
putsUSART (see page 1163)

ReadUSART (see page 1164)

Macros

DataRdyUSART

CloseUSART

BusyUSART

putcUSART

getcUSART

USART Examples (see page 1164)

7.43.2.13 MWIRE (2431 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenMwire (see page 1167)

ReadMwire (see page 1171)

WriteMwire (see page 1170)

getsMwire (see page 1168)

Macros

CloseMwire

getcMwire

putcMwire

DataRdyMwire

MWIRE Examples (see page 1172)

7.43.2.14 Flash (2431 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word

- Reading data from flash
- Loading table pointers and performing table-read operation

Functions

ReadFlash ([see page 1198](#))
 EraseFlash ([see page 1196](#))
 WriteBlockFlash ([see page 1198](#))
 WriteBytesFlash ([see page 1202](#))

Macros

LoadFlashAddr
 TableRead

Flash Examples ([see page 1204](#))

7.44 18F45K22 Family

PIC18F23K22 ([see page 273](#)) \ PIC18LF23K22 ([see page 274](#)) \ PIC18F24K22 ([see page 271](#)) \ PIC18LF24K22 ([see page 272](#))

PIC18F25K22 ([see page 269](#)) \ PIC18LF25K22 ([see page 270](#)) \ PIC18F26K22 ([see page 267](#)) \ PIC18LF26K22 ([see page 268](#))

PIC18F43K22 ([see page 263](#)) \ PIC18LF43K22 ([see page 264](#)) \ PIC18F44K22 ([see page 261](#)) \ PIC18LF44K22 ([see page 262](#))

PIC18F45K22 ([see page 259](#)) \ PIC18LF45K22 ([see page 260](#)) \ PIC18F46K22 ([see page 265](#)) \ PIC18LF46K22 ([see page 266](#))

7.44.1 ADC (45K22 Set)

This peripheral module supports functionalities:

- Up to 32 analog input pins, designated AN0 through AN31, on the 10-bit A/D converter, depending on the particular device pinout.
- Two analog input pins for external voltage reference connections.
- Voltage reference inputs that may be shared with other analog input pins

Functions

OpenADC (see page 879)

SetChanADC (see page 889)

SelChanConvADC (see page 896)

ConvertADC (see page 896)

BusyADC (see page 897)

ReadADC (see page 897)

CloseADC (see page 897)

Macros

ADC_INT_ENABLE

ADC_INT_DISABLE

* ENABLE_AN0_DIG()
 * ENABLE_AN0_ANA()
 * ENABLE_AN1_DIG()
 * ENABLE_AN1_ANA()
 * ENABLE_AN2_DIG()
 * ENABLE_AN2_ANA()
 * ENABLE_AN3_DIG()
 * ENABLE_AN3_ANA()
 * ENABLE_AN4_DIG()
 * ENABLE_AN4_ANA()
 * ENABLE_AN5_DIG()
 * ENABLE_AN5_ANA()
 * ENABLE_AN6_DIG()
 * ENABLE_AN6_ANA()
 * ENABLE_AN7_DIG()
 * ENABLE_AN7_ANA()
 * ENABLE_ALL_ANA_0_7()
 * ENABLE_ALL_DIG_0_7()

 * ENABLE_AN8_DIG()
 * ENABLE_AN8_ANA()
 * ENABLE_AN9_DIG()
 * ENABLE_AN9_ANA()
 * ENABLE_AN10_DIG()
 * ENABLE_AN10_ANA()

ADC Examples (see page 898)

7.44.2 Analog Comparator (45K22 Family)

This peripheral library module:

- Provides two ranges of output voltage, each with 16 distinct levels.
- Can come from either VDD and VSS, or the external VREF+ and VREF-.

Note: The settling time of the comparator voltage reference must be considered when changing the CVREF output

Functions
Open_ancomp1 (see page 900)
Close_ancomp1 (see page 923)
Open_ancomp2 (see page 912)
Close_ancomp2 (see page 923)

Comparator Examples (see page 924)

7.44.3 CTMU (45K22 Set)

This peripheral library module:

- Functions to configure CTMU.
- Function to control current.
- Provides macros to control edge sequence and to configure interrupt.

Functions
OpenCTMU (see page 925)
CurrentControlCTMU (see page 927)
CloseCTMU (see page 927)

Macros
Enbl_CTMUEdge1
Enbl_CTMUEdge2
Disbl_CTMUEdge1
Disbl_CTMUEdge2
CTMUEdge1_Status
CTMUEdge2_Status
CTMU_INT_ENABLE
CTMU_INT_DISABLE
CTMU_Clear_Intr_Status_Bit
CTMU_Intr_Status

CTMU Examples (see page 928)

7.44.4 Input Capture (45K22 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECapture1 (see page 979)
OpenECapture2 (see page 980)
OpenECapture3 (see page 981)
OpenCapture4 (see page 961)
OpenCapture5 (see page 982)
ReadECapture1 (see page 986)
ReadECapture2 (see page 986)
ReadECapture3 (see page 987)
ReadCapture4 (see page 984)
ReadCapture5 (see page 984)
CloseECapture1 (see page 990)
CloseECapture2 (see page 990)
CloseECapture3 (see page 991)
CloseCapture4 (see page 988)
CloseCapture5 (see page 988)

Input Capture Examples ([see page 991](#))

7.44.5 Output Compare (45K22 Set)

This peripheral library module provides:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- Configurable interrupt generation

Functions
OpenECompare1 (see page 1016)
OpenECompare2 (see page 1017)

OpenECompare3 (see page 1018)
OpenCompare4 (see page 1019)
OpenCompare5 (see page 1020)
CloseECompare1 (see page 1024)
CloseECompare2 (see page 1024)
CloseECompare3 (see page 1025)
CloseCompare4 (see page 1022)
CloseCompare5 (see page 1022)

Output Compare Examples ([see page 1025](#))

7.44.6 PWM (45K22 Set)

This peripheral library module:

- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

Functions
OpenEPWM1 (see page 1040)
OpenEPWM2 (see page 1040)
OpenEPWM3 (see page 1041)
OpenPWM4 (see page 1041)
OpenPWM5 (see page 1042)
SetDCEPWM1 (see page 1054)
SetDCEPWM2 (see page 1054)
SetDCEPWM3 (see page 1055)
SetDCPWM4 (see page 1049)
SetDCPWM5 (see page 1050)
SetOutputEPWM1 (see page 1045)
SetOutputEPWM2 (see page 1046)
SetOutputEPWM3 (see page 1046)
CloseEPWM1 (see page 1059)
CloseEPWM2 (see page 1059)
CloseEPWM3 (see page 1059)
ClosePWM4 (see page 1057)
ClosePWM5 (see page 1057)

PWM Examples ([see page 1060](#))

7.44.7 I2C (45K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenI2C1 ([see page 1064](#))

WriteI2C1 ([see page 1066](#))

ReadI2C1 ([see page 1067](#))

putsI2C1 ([see page 1069](#))

getsI2C1 ([see page 1070](#))

OpenI2C2 ([see page 1065](#))

WriteI2C2 ([see page 1067](#))

ReadI2C2 ([see page 1068](#))

putsI2C2 ([see page 1069](#))

getsI2C2 ([see page 1071](#))

Macros

EnableIntI2C1

DisableIntI2C1

SetPriorityIntI2C1

I2C1_Clear_Intr_Status_Bit

I2C1_Intr_Status

StopI2C1

StartI2C1

RestartI2C1

NotAckI2C1

AckI2C1

DataRdyI2C1

IdleI2C1

CloseI2C1

putcI2C1

getcI2C1

EnableIntI2C2

DisableIntI2C2

SetPriorityIntI2C2
I2C2_Clear_Intr_Status_Bit
I2C2_Intr_Status
StopI2C2
StartI2C2
RestartI2C2
NotAckI2C2
AckI2C2
DataRdyI2C2
IdleI2C2
CloseI2C2
putcI2C2
getcI2C2

I2C Examples (see page 1072)

7.44.8 I2C EEPROM (44K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception to I2C EEPROM.
- Incorporates multiple BYTE transmission and reception in a single function call to EEPROM.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts
- Provides ACK polling sequence

Functions
OpenI2C1 (see page 1064)
EEAckPolling1 (see page 1076)
EEByteWrite1 (see page 1080)
EECurrentAddrRead1 (see page 1078)
EEPageWrite1 (see page 1082)
EERandomRead1 (see page 1086)
EESequentialRead1 (see page 1084)
OpenI2C2 (see page 1065)
EEAckPolling2 (see page 1076)
EEByteWrite2 (see page 1080)
EECurrentAddrRead2 (see page 1078)
EEPageWrite2 (see page 1083)
EERandomRead2 (see page 1087)
EESequentialRead2 (see page 1085)

I2C EEPROM Examples (see page 1088)

7.44.9 SPI (45K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

OpenSPI1 (see page 1109)

WriteSPI1 (see page 1112)

ReadSPI1 (see page 1113)

putsSPI1 (see page 1114)

getsSPI1 (see page 1115)

OpenSPI2 (see page 1110)

WriteSPI2 (see page 1112)

ReadSPI2 (see page 1113)

putsSPI2 (see page 1114)

getsSPI2 (see page 1116)

Macros

EnableIntSPI1

DisableIntSPI1

SetPriorityIntSPI1

SPI1_Clear_Intr_Status_Bit

SPI1_Intr_Status

SPI1_Clear_Recv_OV

CloseSPI1

DataRdySPI1

getcSPI1

putcSPI1

EnableIntSPI2

DisableIntSPI2

SetPriorityIntSPI2

SPI2_Clear_Intr_Status_Bit

SPI2_Intr_Status

SPI2_Clear_Recv_OV

CloseSPI2
DataRdySPI2
getcSPI2
putcSPI2

SPI Examples (see page 1116)

7.44.10 IO Ports (45K22 Family)

For this peripheral library:

- A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral.
- When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled.

Functions
OpenPORTB (see page 1089)
OpenRB0INT (see page 1089)
OpenRB1INT (see page 1090)
OpenRB2INT (see page 1091)

Macros
EnablePullups
DisablePullups
ClosePORTB
CloseRB0INT
CloseRB1INT
CloseRB2INT

IO Ports Examples (see page 1093)

7.44.11 EEP (45K22 Set)

This peripheral library module:

- Byte Read from the specified location in EEPROM
- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

FunctionsWrite_b_eep ([see page 1061](#))Read_b_eep ([see page 1061](#))Busy_eep ([see page 1062](#))EEP Examples ([see page 1062](#))

7.44.12 MWIRE (45K22 Family)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

FunctionsOpenMwire1 ([see page 1166](#))ReadMwire1 ([see page 1170](#))WriteMwire1 ([see page 1169](#))getsMwire1 ([see page 1167](#))OpenMwire2 ([see page 1166](#))ReadMwire2 ([see page 1171](#))WriteMwire2 ([see page 1169](#))getsMwire2 ([see page 1168](#))**Macros**

CloseMwire1

getcMwire1

putcMwire1

DataRdyMwire1

CloseMwire2

getcMwire2

putcMwire2

DataRdyMwire2

MWIRE Examples ([see page 1172](#))

7.44.13 SW_RTCC (45K22 Family)

This peripheral library module:

- Provides a Real-Time Clock and Calendar (RTCC) function simulated using Timer1.

Functions

Open_RTCC ([see page 1106](#))

update_RTCC ([see page 1106](#))

Close_RTCC ([see page 1106](#))

SW_RTCC Examples ([see page 1107](#))

7.44.14 Timers (45K22 Set)

For this peripheral library module:

- The Timer module is a 8/16-bit timer , with one 8 bit timer and three 16 bit timers
- Timer 1 operates in CPU Idle modes and Sleep modes.
- Timer1 oscillator acts as low power oscillator
- Timer1/3/5 are 16 bit timers/counters
- Timer2/4/6/8 are 8 bit timers

Functions

OpenTimer0 ([see page 1119](#))

ReadTimer0 ([see page 1137](#))

WriteTimer0 ([see page 1139](#))

CloseTimer0 ([see page 1143](#))

OpenTimer1 ([see page 1121](#))

ReadTimer1 ([see page 1138](#))

WriteTimer1 ([see page 1140](#))

CloseTimer1 ([see page 1143](#))

OpenTimer2 ([see page 1123](#))

CloseTimer2 ([see page 1144](#))

OpenTimer3 ([see page 1125](#))

ReadTimer3 ([see page 1138](#))

WriteTimer3 ([see page 1140](#))

CloseTimer3 ([see page 1144](#))

OpenTimer4 ([see page 1130](#))

CloseTimer4 (see page 1144)
OpenTimer5 (see page 1129)
ReadTimer5 (see page 1139)
WriteTimer5 (see page 1140)
CloseTimer5 (see page 1144)
OpenTimer6 (see page 1131)
CloseTimer6 (see page 1145)

Macros
WriteTimer2
ReadTimer2
WriteTimer4
ReadTimer4
WriteTimer6
ReadTimer6

Timers Examples (see page 1146)

7.44.15 Flash (45K22 Family)

This peripheral module supports functionalities:

- Erasing Flash (see page 1196)
- Writing blocks of data
- Writing a word
- Reading data from flash
- Loading table pointers and performing table-read operation

Functions
ReadFlash (see page 1198)
EraseFlash (see page 1196)
WriteBlockFlash (see page 1200)
WriteWordFlash (see page 1202)
WriteBytesFlash (see page 1203)

Macros
LoadFlashAddr
TableRead

Flash Examples ([↗](#) see page 1204)

7.44.16 USART (45K22 Set)

This peripheral library module:

- Supports BYTE transmission and reception.
- Incorporates multiple BYTE transmission and reception in a single function call.
- Provides simple functions to read from and write to the buffers.
- Provides simple interface macros to enable/disable interrupts

Functions

Open1USART ([↗](#) see page 1148)

Write1USART ([↗](#) see page 1154)

baud1USART ([↗](#) see page 1155)

gets1USART ([↗](#) see page 1159)

putrs1USART ([↗](#) see page 1160)

puts1USART ([↗](#) see page 1162)

Read1USART ([↗](#) see page 1163)

Open2USART ([↗](#) see page 1150)

Write2USART ([↗](#) see page 1155)

baud2USART ([↗](#) see page 1157)

gets2USART ([↗](#) see page 1159)

putrs2USART ([↗](#) see page 1161)

puts2USART ([↗](#) see page 1162)

Read2USART ([↗](#) see page 1163)

Macros

DataRdy1USART

Close1USART

Busy1USART

putc1USART

getc1USART

DataRdy2USART

Close2USART

Busy2USART

putc2USART

getc2USART

USART Examples ([↗](#) see page 1164)

8 Modules

The PIC18F peripheral library supports the hardware and software peripheral modules.

8.1 A/D Converter (ADC)

The 10 bit A/D Converter has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 500 ksp/s
- up to 16 analog input pins
- External voltage reference input pins
- Internal band gap reference inputs
- Selectable conversion trigger source
- Selectable Buffer Fill modes
- Two result alignment options

8.1.1 ADC Functions

8.1.1.1 Open_ADC

8.1.1.1.1 OpenADC_Page1

```
void OpenADC( unsigned char config, unsigned char config2)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ADCON1 register as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none"> * ADC_FOSC_2 * ADC_FOSC_4 * ADC_FOSC_8 * ADC_FOSC_16 * ADC_FOSC_32 * ADC_FOSC_64 * ADC_FOSC_RC * ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none"> * ADC_RIGHT_JUST * ADC_LEFT_JUST * ADC_RESULT_MASK <p>A/D voltage reference source</p> <ul style="list-style-type: none"> * ADC_8ANA_0REF * ADC_7ANA_1REF * ADC_6ANA_2REF * ADC_6ANA_0REF * ADC_5ANA_1REF * ADC_5ANA_0REF * ADC_4ANA_2REF * ADC_4ANA_1REF * ADC_3ANA_2REF * ADC_3ANA_0REF * ADC_2ANA_2REF * ADC_2ANA_1REF * ADC_1ANA_2REF * ADC_1ANA_0REF * ADC_0ANA_0REF * ADC_CONFIG_MASK

config2	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel</div> <div><div>* ADC_CH0</div><div>* ADC_CH1</div><div>* ADC_CH2</div><div>* ADC_CH3</div><div>* ADC_CH4</div><div>* ADC_CH5</div><div>* ADC_CH6</div><div>* ADC_CH7</div></div> <div>A/D Interrupts</div> <div><div>* ADC_INT_ON</div><div>* ADC_INT_OFF</div><div>* ADC_INT_MASK</div></div>
---------	---

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.2 OpenADC_Page2

void OpenADC(unsigned char config, unsigned char config2)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ADCON1 register as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none"> * ADC_FOSC_2 * ADC_FOSC_4 * ADC_FOSC_8 * ADC_FOSC_16 * ADC_FOSC_32 * ADC_FOSC_64 * ADC_FOSC_RC * ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none"> * ADC_RIGHT_JUST * ADC_LEFT_JUST * ADC_RESULT_MASK <p>A/D port configuration</p> <ul style="list-style-type: none"> * ADC_0ANA * ADC_1ANA * ADC_2ANA * ADC_3ANA * ADC_4ANA * ADC_5ANA * ADC_6ANA * ADC_7ANA * ADC_8ANA * ADC_9ANA * ADC_10ANA * ADC_11ANA * ADC_12ANA * ADC_13ANA * ADC_14ANA * ADC_15ANA * ADC_CONFIG_MASK

config2	<p>This contains the parameters to be configured in the ADCON0 register as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none">* ADC_REF_VDD_VREFMINUS* ADC_REF_VREFPLUS_VREFMINUS* ADC_REF_VREFPLUS_VSS* ADC_REF_VDD_VSS* ADC_REF_MASK
---------	--

Returns

None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.3 OpenADC_Page3

void OpenADC(unsigned char config, unsigned char config2, unsigned char portconfig)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none"> * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF * ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS * ADC_REF_MASK
portconfig	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D port configuration</p> <ul style="list-style-type: none"> * ADC_0ANA * ADC_1ANA * ADC_2ANA * ADC_3ANA * ADC_4ANA * ADC_5ANA * ADC_6ANA * ADC_7ANA

Returns

None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.4 OpenADC_Page4

```
void OpenADC( unsigned char config, unsigned char config2, unsigned char portconfig)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<div>This contains the parameters to be configured in the as defined below</div> <div>Channel</div> <div>* ADC_CH0</div> <div>* ADC_CH1</div> <div>* ADC_CH2</div> <div>* ADC_CH3</div> <div>A/D Interrupts</div> <div>* ADC_INT_ON</div> <div>* ADC_INT_OFF</div> <div>* ADC_INT_MASK</div> <div>A/D Vref configuration</div> <div>* ADC_REF_VDD_VREFMINUS</div> <div>* ADC_REF_VREFPLUS_VREFMINUS</div> <div>* ADC_REF_VREFPLUS_VSS</div> <div>* ADC_REF_VDD_VSS</div> <div>* ADC_REF_MASK</div>
portconfig	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D port configuration</div> <div>* ADC_0ANA</div> <div>* ADC_1ANA</div> <div>* ADC_2ANA</div> <div>* ADC_3ANA</div> <div>* ADC_4ANA</div>

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.5 OpenADC_Page5

```
void OpenADC( unsigned char config, unsigned char config2, unsigned char portconfig)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none">* ADC_REF_VDD_VREFMINUS* ADC_REF_VREFPLUS_VREFMINUS* ADC_REF_VREFPLUS_VSS* ADC_REF_VDD_VSS* ADC_REF_MASK
---------	--

portconfig	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D port configuration</div> <div><div>* ADC_0ANA</div><div>* ADC_1ANA</div><div>* ADC_2ANA</div><div>* ADC_3ANA</div><div>* ADC_4ANA</div><div>* ADC_5ANA</div><div>* ADC_6ANA</div><div>* ADC_7ANA</div><div>* ADC_8ANA</div><div>* ADC_9ANA</div><div>* ADC_10ANA</div><div>* ADC_11ANA</div><div>* ADC_12ANA</div><div>* ADC_13ANA</div><div>* ADC_14ANA</div><div>* ADC_15ANA</div></div>
------------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.6 OpenADC_Page6

void OpenADC(unsigned char config, unsigned char config2, unsigned int portconfig)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none"> * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 * ADC_CH12 <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF * ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS * ADC_REF_MASK
portconfig	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D port configuration</p> <ul style="list-style-type: none"> * ADC_0ANA * ADC_1ANA * ADC_2ANA * ADC_3ANA * ADC_4ANA * ADC_5ANA * ADC_6ANA * ADC_7ANA * ADC_8ANA * ADC_9ANA * ADC_10ANA * ADC_11ANA * ADC_12ANA * ADC_13ANA

Returns

None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.7 OpenADC_Page7

```
void OpenADC( unsigned char config1, unsigned char config2, unsigned char config3)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config1	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D conversion type</p> <ul style="list-style-type: none">* ADC_CONV_CONTINUOUS* ADC_CONV_SINGLE_SHOT <p>A/D conversion mode</p> <ul style="list-style-type: none">* ADC_MODE_MULTI_CH* ADC_MODE_SINGLE_CH <p>A/D conversion sequence select</p> <ul style="list-style-type: none">* ADC_CONV_SEQ_SEQM1* ADC_CONV_SEQ_SEQM2* ADC_CONV_SEQ_STNM1* ADC_CONV_SEQ_STNM2 <p>A/D result buffer depth Interrupt select control</p> <ul style="list-style-type: none">* INT_EACH_WR_BUF* INT_2_4_WR_BUF* INT_4_WR_BUF <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF

config2	<div><div>This contains the parameters to be configured in the as defined below</div><div><div>A/D Vref configuration</div><div><div>* ADC_REF_VDD_VREFMINUS</div><div>* ADC_REF_VREFPLUS_VREFMINUS</div><div>* ADC_REF_VREFPLUS_VSS</div><div>* ADC_REF_VDD_VSS</div><div>* ADC_REF_MASK</div></div><div>A/D FIFO buffer control</div><div><div>* ADC_FIFO_EN</div><div>* ADC_FIFO_DIS</div></div><div>A/D Trigger Source</div><div><div>* ADC_TRIG_EXT_INT0</div><div>* ADC_TRIG_TMR_5</div><div>* ADC_TRIG_INP_CAP</div><div>* ADC_TRIG_CCP2_COM</div><div>* ADC_TRIG_PCPWM</div></div></div></div>
---------	--

8.1.1.1.8 OpenADC_Page8

```
void OpenADC( unsigned char config1, unsigned char config2, unsigned char config3)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config1	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D conversion type</p> <ul style="list-style-type: none"> * ADC_CONV_CONTINUOUS * ADC_CONV_SINGLE_SHOT <p>A/D conversion mode</p> <ul style="list-style-type: none"> * ADC_MODE_MULTI_CH * ADC_MODE_SINGLE_CH <p>A/D conversion sequence select</p> <ul style="list-style-type: none"> * ADC_CONV_SEQ_SEQM1 * ADC_CONV_SEQ_SEQM2 * ADC_CONV_SEQ_STNM1 * ADC_CONV_SEQ_STNM2 <p>A/D result buffer depth Interrupt select control</p> <ul style="list-style-type: none"> * INT_EACH_WR_BUF * INT_2_4_WR_BUF * INT_4_WR_BUF <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF
config2	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS <p>A/D FIFO buffer control</p> <ul style="list-style-type: none"> * ADC_FIFO_EN * ADC_FIFO_DIS <p>A/D Trigger Source</p> <ul style="list-style-type: none"> * ADC_TRIG_EXT_INT0 * ADC_TRIG_TMR_5 * ADC_TRIG_INP_CAP * ADC_TRIG_CCP2_COM * ADC_TRIG_PCPWM

config3	<div><p>This contains the parameters to be configured in the as defined below</p><p>A/D result justification</p><ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST<p>A/D acquisition time select</p><ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_10_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_24_TAD* ADC_28_TAD* ADC_32_TAD* ADC_36_TAD* ADC_40_TAD* ADC_48_TAD* ADC_64_TAD<p>A/D clock source</p><ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC<p>A/D Channel selection</p><p>Channel from group A</p><ul style="list-style-type: none">* ADC_CH_GRA_AN0()* ADC_CH_GRA_AN4()<p>Channel from group B</p><ul style="list-style-type: none">* ADC_CH_GRB_AN1()<p>Channel from group C</p><ul style="list-style-type: none">* ADC_CH_GRC_AN2()<p>Channel from group D</p><ul style="list-style-type: none">* ADC_CH_GRD_AN3()</div>
---------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.9 OpenADC_Page9

```
void OpenADC( unsigned char config, unsigned char config2, unsigned char config3, unsigned int portconfig)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D clock source</div> <div>* ADC_FOSC_2</div> <div>* ADC_FOSC_4</div> <div>* ADC_FOSC_8</div> <div>* ADC_FOSC_16</div> <div>* ADC_FOSC_32</div> <div>* ADC_FOSC_64</div> <div>* ADC_FOSC_RC</div> <div>* ADC_FOSC_MASK</div> <div>A/D result justification</div> <div>* ADC_RIGHT_JUST</div> <div>* ADC_LEFT_JUST</div> <div>* ADC_RESULT_MASK</div> <div>A/D acquisition time select</div> <div>* ADC_0_TAD</div> <div>* ADC_2_TAD</div> <div>* ADC_4_TAD</div> <div>* ADC_6_TAD</div> <div>* ADC_8_TAD</div> <div>* ADC_12_TAD</div> <div>* ADC_16_TAD</div> <div>* ADC_20_TAD</div> <div>* ADC_TAD_MASK</div>

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none"> * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 * FVR1 * DAC1 <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF * ADC_INT_MASK
config3	<p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS * ADC_REF_MASK
portconfig	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D port configuration</p> <ul style="list-style-type: none"> * ADC_3ANA * ADC_4ANA * ADC_5ANA * ADC_6ANA * ADC_7ANA * ADC_8ANA * ADC_9ANA * ADC_10ANA * ADC_11ANA <p>Below macros are applicable only to F1xK50 Family</p> <ul style="list-style-type: none"> * ADC_0ANA * ADC_1ANA * ADC_2ANA

Returns

None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.10 OpenADC_Page10

```
void OpenADC( unsigned char config, unsigned char config2, unsigned int portconfig)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none"> * ADC_FOSC_2 * ADC_FOSC_4 * ADC_FOSC_8 * ADC_FOSC_16 * ADC_FOSC_32 * ADC_FOSC_64 * ADC_FOSC_RC * ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none"> * ADC_RIGHT_JUST * ADC_LEFT_JUST * ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none"> * ADC_0_TAD * ADC_2_TAD * ADC_4_TAD * ADC_6_TAD * ADC_8_TAD * ADC_12_TAD * ADC_16_TAD * ADC_20_TAD * ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH_CTMU* ADC_CH_VDDCORE* ADC_CH_VBG <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none">* ADC_REF_VDD_VREFMINUS* ADC_REF_VREFPLUS_VREFMINUS* ADC_REF_VREFPLUS_VSS* ADC_REF_VDD_VSS* ADC_REF_MASK
---------	---

portconfig	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D port configuration</div> <div><div>* ADC_0ANA</div><div>* ADC_1ANA</div><div>* ADC_2ANA</div><div>* ADC_3ANA</div><div>* ADC_4ANA</div><div>* ADC_5ANA</div><div>* ADC_6ANA</div><div>* ADC_7ANA</div><div>* ADC_8ANA</div><div>* ADC_9ANA</div><div>* ADC_10ANA</div><div>* ADC_11ANA</div><div>* ADC_12ANA</div><div>* ADC_13ANA</div></div> <div>Band Gap selection:</div> <div><div>* ADC_VBG_ON</div><div>* ADC_VBG_OFF</div></div>
------------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.11 OpenADC_Page11

void OpenADC(unsigned char config, unsigned char config2, unsigned char portconfig)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none"> * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF * ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS * ADC_REF_MASK
portconfig	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D port configuration</p> <ul style="list-style-type: none"> * ADC_0ANA * ADC_1ANA * ADC_2ANA * ADC_3ANA * ADC_4ANA * ADC_5ANA * ADC_6ANA * ADC_7ANA * ADC_8ANA * ADC_9ANA * ADC_10ANA * ADC_11ANA <p>Special Trigger configuration</p> <ul style="list-style-type: none"> * ADC_TRIG_CTMU * ADC_TRIG_CCP2

Returns

None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format, Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.12 OpenADC_Page12

void OpenADC(unsigned char config, unsigned char config2, unsigned int portconfig)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none">* ADC_REF_VDD_VREFMINUS* ADC_REF_VREFPLUS_VREFMINUS* ADC_REF_VREFPLUS_VSS* ADC_REF_VDD_VSS* ADC_REF_MASK
---------	--

portconfig	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D port configuration</div> <div>* ADC_0ANA</div> <div>* ADC_1ANA</div> <div>* ADC_2ANA</div> <div>* ADC_3ANA</div> <div>* ADC_4ANA</div> <div>* ADC_5ANA</div> <div>* ADC_6ANA</div> <div>* ADC_7ANA</div> <div>* ADC_8ANA</div> <div>* ADC_9ANA</div> <div>* ADC_10ANA</div> <div>* ADC_11ANA</div> <div>* ADC_12ANA</div> <div>* ADC_13ANA</div> <div>* ADC_14ANA</div> <div>* ADC_15ANA</div>
------------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.13 OpenADC_Page13

void OpenADC(unsigned char config, unsigned char config2, unsigned int portconfig)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH16* ADC_CH17* ADC_CH18* ADC_CH19* ADC_CH30* ADC_CH31 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK
---------	---

config3	<div>This contains the parameters to be configured in the as defined below</div> <div>Special Trigger Select<ul style="list-style-type: none">* ADC_TRIG_RTCC* ADC_TRIG_TIMER1* ADC_TRIG_CTMU* ADC_TRIG_CCP2</div> <div>Analog Negative Channel Select<ul style="list-style-type: none">* ADC_NEG_CH0* ADC_NEG_CH1* ADC_NEG_CH2* ADC_NEG_CH3* ADC_NEG_CH4* ADC_NEG_CH5* ADC_NEG_CH6* ADC_NEG_CH7</div> <div>A/D VREF+ Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VDD* ADC_REF_VDD_VREFPLUS* ADC_REF_VDD_INT_VREF_2* ADC_REF_VDD_INT_VREF_4</div> <div>A/D VREF- Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VSS* ADC_REF_VDD_VREFMINUS</div>
---------	---

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.14 OpenADC_Page14

void OpenADC(unsigned char config, unsigned char config2, unsigned char config3)
--

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15* ADC_CH16* ADC_CH17* ADC_CH18* ADC_CH19* ADC_CH20* ADC_CH21* ADC_CH22* ADC_CH23* ADC_CH28* ADC_CH29* ADC_CH30* ADC_CH31 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK
---------	---

config3	<div>This contains the parameters to be configured in the as defined below</div> <div>Special Trigger Select<ul style="list-style-type: none">* ADC_TRIG_RTCC* ADC_TRIG_TIMER1* ADC_TRIG_CTMU* ADC_TRIG_CCP2</div> <div>Analog Negative Channel Select<ul style="list-style-type: none">* ADC_NEG_CH0* ADC_NEG_CH1* ADC_NEG_CH2* ADC_NEG_CH3* ADC_NEG_CH4* ADC_NEG_CH5* ADC_NEG_CH6* ADC_NEG_CH7</div> <div>A/D VREF+ Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VDD* ADC_REF_VDD_VREFPLUS* ADC_REF_VDD_INT_VREF_2* ADC_REF_VDD_INT_VREF_4</div> <div>A/D VREF- Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VSS* ADC_REF_VDD_VREFMINUS</div>
---------	---

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.15 OpenADC_Page15

```
void OpenADC( unsigned char config, unsigned char config2, unsigned char config3, unsigned int portconfig)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none"> * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 * ADC_CH12 * ADC_CH_CTMU * ADC_CH_VDDCORE * ADC_CH_VBG <p>A/D Interrupts</p> <ul style="list-style-type: none"> * ADC_INT_ON * ADC_INT_OFF * ADC_INT_MASK <p>A/D Vref configuration</p> <ul style="list-style-type: none"> * ADC_REF_VDD_VREFMINUS * ADC_REF_VREFPLUS_VREFMINUS * ADC_REF_VREFPLUS_VSS * ADC_REF_VDD_VSS * ADC_REF_MASK
config3	<p>Special Trigger Select bit</p> <ul style="list-style-type: none"> * ADC_TRIG_RTCC * ADC_TRIG_TIMER1 * ADC_TRIG_CTMU * ADC_TRIG_CCP2

portconfig	<div>This contains the parameters to be configured in the as defined below</div> <div>A/D port configuration</div> <div><div>* ADC_0ANA</div><div>* ADC_1ANA</div><div>* ADC_2ANA</div><div>* ADC_3ANA</div><div>* ADC_4ANA</div><div>* ADC_5ANA</div><div>* ADC_6ANA</div><div>* ADC_7ANA</div><div>* ADC_8ANA</div><div>* ADC_9ANA</div><div>* ADC_10ANA</div><div>* ADC_11ANA</div><div>* ADC_12ANA</div><div>* ADC_13ANA</div></div> <div>Band Gap selection:</div> <div><div>* ADC_VBG_ON</div><div>* ADC_VBG_OFF</div></div>
------------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.16 OpenADC_Page16

void OpenADC(unsigned char config, unsigned char config2, unsigned char config3)

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH29* ADC_CH30* ADC_CH31 <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK
---------	---

config3	<div>This contains the parameters to be configured in the as defined below</div> <div>Special Trigger Select<ul style="list-style-type: none">* ADC_TRIG_RTCC* ADC_TRIG_TIMER1* ADC_TRIG_CTMU* ADC_TRIG_CCP2</div> <div>Analog Negative Channel Select<ul style="list-style-type: none">* ADC_NEG_CH0* ADC_NEG_CH1* ADC_NEG_CH2* ADC_NEG_CH3* ADC_NEG_CH4* ADC_NEG_CH5* ADC_NEG_CH6</div> <div>A/D VREF+ Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VDD* ADC_REF_VDD_VREFPLUS* ADC_REF_VDD_INT_VREF_2* ADC_REF_VDD_INT_VREF_4</div> <div>A/D VREF- Configuration<ul style="list-style-type: none">* ADC_REF_VDD_VSS* ADC_REF_VDD_VREFMINUS</div>
---------	---

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.1.17 OpenADC_Page17

```
void OpenADC( unsigned char config, unsigned char config2, unsigned char config3)
```

This function configures A/D & starts the conversion.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the as defined below</p> <p>A/D clock source</p> <ul style="list-style-type: none">* ADC_FOSC_2* ADC_FOSC_4* ADC_FOSC_8* ADC_FOSC_16* ADC_FOSC_32* ADC_FOSC_64* ADC_FOSC_RC* ADC_FOSC_MASK <p>A/D result justification</p> <ul style="list-style-type: none">* ADC_RIGHT_JUST* ADC_LEFT_JUST* ADC_RESULT_MASK <p>A/D acquisition time select</p> <ul style="list-style-type: none">* ADC_0_TAD* ADC_2_TAD* ADC_4_TAD* ADC_6_TAD* ADC_8_TAD* ADC_12_TAD* ADC_16_TAD* ADC_20_TAD* ADC_TAD_MASK

config2	<p>This contains the parameters to be configured in the as defined below</p> <p>Channel</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15* ADC_CH16* ADC_CH17* ADC_CH18* ADC_CH19* ADC_CH20* ADC_CH21* ADC_CH22* ADC_CH23* ADC_CH24* ADC_CH25* ADC_CH26* ADC_CH27* ADC_CH_CTMU* ADC_CH_DAC* ADC_CH_FRV <p>A/D Interrupts</p> <ul style="list-style-type: none">* ADC_INT_ON* ADC_INT_OFF* ADC_INT_MASK
---------	---

config3	<div>This contains the parameters to be configured in the as defined below</div> <div>Special Trigger Select</div> <div>* ADC_TRIG_CTMU</div> <div>* ADC_TRIG_CCP5</div> <div>A/D VREF+ Configuration</div> <div>* ADC_REF_VDD_VDD</div> <div>* ADC_REF_VDD_VREFPLUS</div> <div>* ADC_REF_FVR_BUF</div> <div>A/D VREF- Configuration</div> <div>* ADC_REF_VDD_VSS</div> <div>* ADC_REF_VDD_VREFMINUS</div>
---------	--

Returns
None

Remarks

This function configures the ADC for the following parameters: Operating mode, Data o/p format,Sample Clk Source, VREF source, No of samples/int, sample mode, Conv clock source, Conv Clock Select bits, Port Config Control bits.

8.1.1.2 SetChan_ADC

8.1.1.2.1 SetChanADC_Page1

void SetChanADC(unsigned char channel)
--

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div>* ADC_CH0</div> <div>* ADC_CH1</div> <div>* ADC_CH2</div> <div>* ADC_CH3</div> <div>* ADC_CH4</div> <div>* ADC_CH5</div> <div>* ADC_CH6</div> <div>* ADC_CH7</div> <div>* ADC_CH8</div> <div>* ADC_CH9</div> <div>* ADC_CH10</div> <div>* ADC_CH11</div> <div>* ADC_CH12</div> <div>* ADC_CH13</div> <div>* ADC_CH14</div> <div>* ADC_CH15</div>

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.2 SetChanADC_Page2

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	This contains the parameters to be configured in the ADCON0 register as defined below Channel Select * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 * ADC_CH12 * DAC1 * FVR1

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.3 SetChanADC_Page3

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div><div><div>* ADC_CH0</div><div>* ADC_CH1</div><div>* ADC_CH2</div><div>* ADC_CH3</div><div>* ADC_CH4</div><div>* ADC_CH5</div><div>* ADC_CH6</div><div>* ADC_CH7</div><div>* ADC_CH8</div><div>* ADC_CH9</div><div>* ADC_CH10</div><div>* ADC_CH11</div><div>* ADC_CH12</div><div>* ADC_CH_CTMU</div><div>* ADC_CH_VDDCORE</div><div>* ADC_CH_VBG</div></div></div>

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.4 SetChanADC_Page4

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	This contains the parameters to be configured in the ADCON0 register as defined below Channel Select * ADC_CH_GRA_AN0() * ADC_CH_GRA_AN4() * ADC_CH_GRB_AN1() * ADC_CH_GRC_AN2() * ADC_CH_GRD_AN3() * ADC_CH_GRB_AN5() * ADC_CH_GRC_AN6() * ADC_CH_GRD_AN7() * ADC_CH_GRA_AN8() * ALL_CH_DIGITAL()

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.5 SetChanADC_Page5

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	This contains the parameters to be configured in the ADCON0 register as defined below Channel Select * ADC_CH_GRA_AN0() * ADC_CH_GRA_AN4() * ADC_CH_GRB_AN1() * ADC_CH_GRC_AN2() * ADC_CH_GRD_AN3()

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.6 SetChanADC_Page6

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div>* ADC_CH0</div> <div>* ADC_CH1</div> <div>* ADC_CH2</div> <div>* ADC_CH3</div> <div>* ADC_CH4</div> <div>* ADC_CH5</div> <div>* ADC_CH6</div> <div>* ADC_CH7</div> <div>* ADC_CH8</div> <div>* ADC_CH9</div> <div>* ADC_CH10</div> <div>* ADC_CH11</div> <div>* ADC_CH16</div> <div>* ADC_CH17</div> <div>* ADC_CH18</div> <div>* ADC_CH19</div> <div>* ADC_CH30</div> <div>* ADC_CH31</div>

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.7 SetChanADC_Page7

void SetChanADC(unsigned char channel)

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div><ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH13* ADC_CH14* ADC_CH15* ADC_CH16* ADC_CH17* ADC_CH18* ADC_CH19* ADC_CH20* ADC_CH21* ADC_CH22* ADC_CH23* ADC_CH28* ADC_CH29* ADC_CH30* ADC_CH31</div>

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.2.8 SetChanADC_Page8

```
void SetChanADC(unsigned char channel)
```

This function Select the channel used as input to the A/D converter.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	This contains the parameters to be configured in the ADCON0 register as defined below Channel Select * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH29 * ADC_CH30 * ADC_CH31

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter.

8.1.1.3 SelChanConv_ADC

8.1.1.3.1 SelChanConvADC_Page1

```
void SelChanConvADC(unsigned char channel)
```

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	This contains the parameters to be configured in the ADCON0 register as defined below Channel Select * ADC_CH0 * ADC_CH1 * ADC_CH2 * ADC_CH3 * ADC_CH4 * ADC_CH5 * ADC_CH6 * ADC_CH7 * ADC_CH8 * ADC_CH9 * ADC_CH10 * ADC_CH11 * ADC_CH12 * ADC_CH13 * ADC_CH14 * ADC_CH15

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.2 SelChanConvADC_Page2

```
void SelChanConvADC(unsigned char channel)
```

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<p>This contains the parameters to be configured in the ADCON0 register as defined below</p> <p>Channel Select</p> <ul style="list-style-type: none">* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* DAC1* FVR1

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.3 SelChanConvADC_Page3

void SelChanConvADC(unsigned char channel)

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<p>This contains the parameters to be configured in the ADCON0 register as defined below</p> <p>Channel Select</p> <ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH11* ADC_CH12* ADC_CH_CTMU* ADC_CH_VDDCORE* ADC_CH_VBG

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.4 SelChanConvADC_Page4

void SelChanConvADC(unsigned char channel)

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<p>This contains the parameters to be configured in the ADCON0 register as defined below</p> <p>Channel Select</p> <ul style="list-style-type: none"> * ADC_CH_GRA_AN0() * ADC_CH_GRA_AN4() * ADC_CH_GRB_AN1() * ADC_CH_GRC_AN2() * ADC_CH_GRD_AN3() * ADC_CH_GRB_AN5() * ADC_CH_GRC_AN6() * ADC_CH_GRD_AN7() * ADC_CH_GRA_AN8() * ALL_CH_DIGITAL()

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.5 SelChanConvADC_Page5

void SelChanConvADC(unsigned char channel)
--

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<p>This contains the parameters to be configured in the ADCON0 register as defined below</p> <p>Channel Select</p> <ul style="list-style-type: none"> * ADC_CH_GRA_AN0() * ADC_CH_GRA_AN4() * ADC_CH_GRB_AN1() * ADC_CH_GRC_AN2() * ADC_CH_GRD_AN3()

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.6 SelChanConvADC_Page6

void SelChanConvADC(unsigned char channel)

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div>* ADC_CH0</div> <div>* ADC_CH1</div> <div>* ADC_CH2</div> <div>* ADC_CH3</div> <div>* ADC_CH4</div> <div>* ADC_CH5</div> <div>* ADC_CH6</div> <div>* ADC_CH7</div> <div>* ADC_CH8</div> <div>* ADC_CH9</div> <div>* ADC_CH10</div> <div>* ADC_CH11</div> <div>* ADC_CH16</div> <div>* ADC_CH17</div> <div>* ADC_CH18</div> <div>* ADC_CH19</div> <div>* ADC_CH30</div> <div>* ADC_CH31</div>

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.7 SelChanConvADC_Page7

void SelChanConvADC(unsigned char channel)

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div>* ADC_CH0</div> <div>* ADC_CH1</div> <div>* ADC_CH2</div> <div>* ADC_CH3</div> <div>* ADC_CH4</div> <div>* ADC_CH5</div> <div>* ADC_CH6</div> <div>* ADC_CH7</div> <div>* ADC_CH8</div> <div>* ADC_CH9</div> <div>* ADC_CH10</div> <div>* ADC_CH11</div> <div>* ADC_CH12</div> <div>* ADC_CH13</div> <div>* ADC_CH14</div> <div>* ADC_CH15</div> <div>* ADC_CH16</div> <div>* ADC_CH17</div> <div>* ADC_CH18</div> <div>* ADC_CH19</div> <div>* ADC_CH20</div> <div>* ADC_CH21</div> <div>* ADC_CH22</div> <div>* ADC_CH23</div> <div>* ADC_CH28</div> <div>* ADC_CH29</div> <div>* ADC_CH30</div> <div>* ADC_CH31</div>

Returns

None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.3.8 SelChanConvADC_Page8

```
void SelChanConvADC(unsigned char channel)
```

This function Select the channel used as input to the A/D converter and starts the A/D conversion process.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
channel	<div>This contains the parameters to be configured in the ADCON0 register as defined below</div> <div>Channel Select</div> <div><ul style="list-style-type: none">* ADC_CH0* ADC_CH1* ADC_CH2* ADC_CH3* ADC_CH4* ADC_CH5* ADC_CH6* ADC_CH7* ADC_CH8* ADC_CH9* ADC_CH10* ADC_CH29* ADC_CH30* ADC_CH31</div>

Returns
None

Remarks

Selects the pin that will be used as input to the A/D converter, and starts an A/D conversion. The BusyADC() function or A/D interrupt may be used to detect completion of the conversion.

8.1.1.4 Convert_ADC

```
void ConvertADC(void)
```

This function starts the A/D conversion.

Returns

None

Remarks

This function sets the ADCON0<GO> bit and thus starts conversion.

8.1.1.5 Busy_ADC

char BusyADC(void)

This function returns the ADC conversion status

Returns

If the value of GO is '1', then '1' is returned, indicating that the ADC is busy in conversion. If the value of GO is '0', then '0' is returned, indicating that the ADC has completed conversion.

Remarks

This function returns the complement of the ADCON0 <GO/~DONE> bit status which indicates whether the ADC is busy in conversion.

8.1.1.6 Read_ADC

int ReadADC(void)

This function reads the ADC Buffer register which contains the conversion value.

Returns

Returns the ADC Buffer value

Remarks

None

8.1.1.7 Close_ADC

void CloseADC(void)

This function turns off the ADC module and disables the ADC interrupts

Returns

None

Remarks

This function first disables the ADC interrupt and then turns off the ADC module. The Interrupt Flag bit (ADIF) is also cleared.

8.1.2 ADC Structs, Records, Enums

8.1.3 ADC Macros

8.1.4 ADC Examples

8.1.4.1 ADC Example 1

Example Source Code demonstrating peripheral library usage

```

/*****
 * NOTES:
 * Code uses the Peripheral library support available with MCC18 Compiler
 * Code Tested on:
 * PicDem2+ demo board with PIC18F4685 controller
 *
 * signal to be converted is fed to AN0 channel.
 *****/

#define USE_OR_MASKS
#include <pl18cxxx.h>
#include "adc.h"

unsigned int ADCResult=0;
float voltage=0;

void main(void)
{
    unsigned char channel=0x00,config1=0x00,config2=0x00,config3=0x00,portconfig=0x00,i=0;

    //-- clear adc interrupt and turn off adc if in case was on prerviously--
    CloseADC();

    //--initialize adc--
    /*** ADC configured for:
     * FOSC/2 as conversion clock
     * Result is right justified
     * Aquisition time of 2 AD
     * Channel 1 for sampling
     * ADC interrupt on
     * ADC reference voltage from VDD & VSS
     */
    config1 = ADC_FOSC_2 | ADC_RIGHT_JUST | ADC_2_TAD ;
    config2 = ADC_CH0 | ADC_INT_ON | ADC_REF_VDD_VSS ;
    portconfig = ADC_15ANA ;
    OpenADC(config1,config2,portconfig);

```

```
//---initialize the adc interrupt and enable them---
ADC_INT_ENABLE();

//---sample and convert---
for(i=0;i<16;i++)
{
    ConvertADC();
    while(BusyADC());
    ADCResult += (unsigned int) ReadADC();
}
ADCResult /= 16;
voltage = (ADCResult*5.0)/1024; // convert ADC count into voltage

CloseADC();           //turn off ADC

while(1);              //End of program
}
```

8.2 Analog Comparator (ANCOMP)

Analog Comparators with Programmable Input/Output Configuration. The comparator module provides dual input comparators. The inputs to the comparator can be configured to use any one of four external analog inputs as well, as a voltage reference input from either the internal band gap reference divided by two (VBG/2) or the comparator voltage reference generator.

8.2.1 Comparator Functions

8.2.1.1 Open_ANCOMP

8.2.1.1.1 Open_ancomp_Page1

```
void Open_ancomp(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator Output Inversion:</p> <ul style="list-style-type: none"> * COMP_1_2_OP_INV * COMP_1_OP_INV * COMP_2_OP_INV * COMP_OP_INV_NONE * COMP_OP_MASK <p>Comparator Mode Select:</p> <ul style="list-style-type: none"> * COMP_1_2_INDP * COMP_1_2_INDP_OP * COMP_1_2_COMN_REF * COMP_1_2_COMN_REF_OP * COMP_1_INDP_OP * COMP_INT_REF_SAME_IP * COMP_INT_REF_MUX_IP * COMP_SELECT_MASK <p>Comparator Interrupts:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.2 Open_ancomp1_Page1

```
void Open_ancomp1(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator1 Ref (C1VREF)select:</p> <ul style="list-style-type: none"> * COMP_REF_FVR * COMP_REF_CVREF * COMP_REF_MASK <p>Comparator1 Speed/Power select:</p> <ul style="list-style-type: none"> * COMP_HSPEED * COMP_LSPEED * COMP_HSPEED_MASK <p>Comparator1 Ref (C1VIN+) select:</p> <ul style="list-style-type: none"> * COMP_VINP_PIN * COMP_VINP_VREF * COMP_VINP_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_IN0 * COMP_VINM_IN1 * COMP_VINM_IN2 * COMP_VINM_IN3 * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.3 Open_ancomp1_Page2

```
void Open_ancomp1(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none">* COMP_INT_EN* COMP_INT_DIS* COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none">* COMP_OP_EN* COMP_OP_DIS* COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none">* COMP_OP_INV* COMP_OP_NINV* COMP_OP_INV_MASK <p>Comparator1 Interrupt polarity select:</p> <ul style="list-style-type: none">* COMP_INT_ALL_EDGE* COMP_INT_FALL_EDGE* COMP_INT_RISE_EDGE* COMP_INT_NOGEN* COMP_INT_EDGE_MASK <p>Comparator1 Ref (C1VIN+)select:</p> <ul style="list-style-type: none">* COMP_REF_CVREF* COMP_REF_CINA* COMP_REF_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none">* COMP_VINM_VIRV* COMP_VINM_CIND* COMP_VINM_CINC* COMP_VINM_CINB* COMP_VINM_MASK

Returns
None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.4 Open_ancomp1_Page3

```
void Open_ancomp1(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator1 Ref (C1VREF)select:</p> <ul style="list-style-type: none"> * COMP_REF_FVR * COMP_REF_DAC * COMP_REF_MASK <p>Comparator1 Speed/Power select:</p> <ul style="list-style-type: none"> * COMP_HSPEED * COMP_LSPEED * COMP_HSPEED_MASK <p>Comparator1 Ref (C1VP) select:</p> <ul style="list-style-type: none"> * COMP_C1VP_VREF * COMP_C1VP_PIN * COMP_C1VP_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_GND * COMP_VINM_IN1 * COMP_VINM_IN2 * COMP_VINM_IN3 * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.5 Open_ancomp1_Page4

```
void Open_ancomp1(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator1 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator1 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VBG * COMP_VINM_CM1_C2INB * COMP_VINM_GND * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.6 Open_ancomp1_Page5

void Open_ancomp1(unsigned char config)

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator1 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator1 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VBG * COMP_VINM_CM1_CM3_C2INB * COMP_VINM_CM2_C2IND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.7 Open_amcomp2_Page1

```
void Open_amcomp2(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Ref (C1VREF)select:</p> <ul style="list-style-type: none"> * COMP_REF_FVR * COMP_REF_CVREF * COMP_REF_MASK <p>Comparator2 Speed/Power select:</p> <ul style="list-style-type: none"> * COMP_HSPEED * COMP_LSPEED * COMP_HSPEED_MASK <p>Comparator2 Ref (C1VIN+) select:</p> <ul style="list-style-type: none"> * COMP_VINP_PIN * COMP_VINP_VREF * COMP_VINP_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_IN0 * COMP_VINM_IN1 * COMP_VINM_IN2 * COMP_VINM_IN3 * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.8 Open_ancomp2_Page2

```
void Open_ancomp2(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator2 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VIRV * COMP_VINM_CIND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.9 Open_ancomp2_Page3

void Open_ancomp2(unsigned char config)

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Ref (C1VREF)select:</p> <ul style="list-style-type: none"> * COMP_REF_FVR * COMP_REF_DAC * COMP_REF_MASK <p>Comparator2 Speed/Power select:</p> <ul style="list-style-type: none"> * COMP_HSPEED * COMP_LSPEED * COMP_HSPEED_MASK <p>Comparator2 Ref (C1VP) select:</p> <ul style="list-style-type: none"> * COMP_C1VP_VREF * COMP_C1VP_PIN * COMP_C1VP_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_GND * COMP_VINM_IN1 * COMP_VINM_IN2 * COMP_VINM_IN3 * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.10 Open_ancomp2_Page4

```
void Open_ancomp2(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator2 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VBG * COMP_VINM_CM1_C2INB * COMP_VINM_GND * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.11 **Open_ancomp2_Page5**

void Open_ancomp2(unsigned char config)

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator2 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VBG * COMP_VINM_CM1_CM3_C2INB * COMP_VINM_CM2_C2IND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.12 Open_ancomp3_Page1

```
void Open_ancomp3(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator3 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator3 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator3 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator3 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator3 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator3 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VBG * COMP_VINM_CM1_CM3_C2INB * COMP_VINM_CM2_C2IND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.13 Open_ancomp1_Page6

void Open_ancomp1(unsigned char config)

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator1 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator1 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator1 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator1 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator1 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator1 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VIRV * COMP_VINM_CIND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.14 Open_ancomp2_Page6

```
void Open_ancomp2(unsigned char config)
```

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator2 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator2 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator2 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator2 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator2 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator2 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VIRV * COMP_VINM_CIND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.1.15 Open_ancomp3_Page2

void Open_ancomp3(unsigned char config)

This routine configures Reference Voltage level and Comparator module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CVRCON & CMxCON Registers.</p> <p>Comparator3 Interrupt:</p> <ul style="list-style-type: none"> * COMP_INT_EN * COMP_INT_DIS * COMP_INT_MASK <p>Comparator3 Output enable:</p> <ul style="list-style-type: none"> * COMP_OP_EN * COMP_OP_DIS * COMP_OP_MASK <p>Comparator3 output polarity select:</p> <ul style="list-style-type: none"> * COMP_OP_INV * COMP_OP_NINV * COMP_OP_INV_MASK <p>Comparator3 Interrupt polarity select:</p> <ul style="list-style-type: none"> * COMP_INT_ALL_EDGE * COMP_INT_FALL_EDGE * COMP_INT_RISE_EDGE * COMP_INT_NOGEN * COMP_INT_EDGE_MASK <p>Comparator3 Ref (C1VIN+)select:</p> <ul style="list-style-type: none"> * COMP_REF_CVREF * COMP_REF_CINA * COMP_REF_MASK <p>Comparator3 channel select:</p> <ul style="list-style-type: none"> * COMP_VINM_VIRV * COMP_VINM_CIND * COMP_VINM_CINC * COMP_VINM_CINB * COMP_VINM_MASK

Returns

None

Remarks

This function configures comparator with channel of selection for positive and negative input terminals, configures interrupts, reference voltage module is configured based on input parameters.

8.2.1.2 Close_ANCOMP

8.2.1.2.1 Close_ancomp_Page1

```
void Close_ancomp(void)
```

This function turns off the Comparator module and disables the Comparator interrupts

Returns
None

[Remarks](#)

This function first disables the Comparator interrupt and then turns off the Comparator module. The Interrupt Flag bit (CM1IF) is also cleared.

8.2.1.2.2 Close_ancomp1_Page1

```
void Close_ancomp1(void)
```

This function turns off the Comparator module and disables the Comparator interrupts

Returns
None

[Remarks](#)

This function first disables the Comparator interrupt and then turns off the Comparator module. The Interrupt Flag bit (CM11IF) is also cleared.

8.2.1.2.3 Close_ancomp2_Page1

```
void Close_ancomp2(void)
```

This function turns off the Comparator module and disables the Comparator interrupts

Returns
None

[Remarks](#)

This function first disables the Comparator interrupt and then turns off the Comparator module. The Interrupt Flag bit (CM2IF) is also cleared.

8.2.1.2.4 Close_ancomp3_Page1

```
void Close_ancomp3(void)
```

This function turns off the Comparator module and disables the Comparator interrupts

Returns

None

Remarks

This function first disables the Comparator interrupt and then turns off the Comparator module. The Interrupt Flag bit (CM3IF) is also cleared.

8.2.2 Comparator Macros

8.2.3 Comparator Examples

8.2.3.1 Comparator Example1

Example Source Code demonstrating peripheral library usage

```
/*
 * NOTES:
 * Code uses the Peripheral library support available with MCC18 Compiler
 * Code Tested on:
 * PicDem2+ demo board with PIC18F4685 controller
 *
 * signal to be compared are applied to CxIN+ & CxIN- and output is tapped at CxOUT
 */

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "ancomp.h"

unsigned char CMPResult;

void main(void)
{
    unsigned char config=0x00;

    /******Configure Analog Comparator *****/
    /*** Analog comparator is configured for:
     * Output is non-inverted
     * comparator 1 & 2 are configured as independent comparators
     * comparator interrupt is eabled
     */
    config = COMP_OP_INV_NONE | COMP_1_2_INDP | COMP_INT_EN ;
    Open_ancomp(config);

    while(1)
    {
        if( (CMCON&0x40) ) //check for the comparator output status or comparartor
interrupt flag status
            CMPResult=1; //flag the logic 1 status of comparartor output
        else
            CMPResult=0; // flag the logic 0 status of comparator output
    }
    /*--- Turn off comparator ---

```

```
    Close_ancomp();  
  
    while(1); //End of program  
}
```

8.3 Charge Time Measurement Unit (CTMU)

The Charge Time Measurement Unit is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. Its key features include:

- Four edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- Accurate current source suitable for capacitive measurement

Together with other on-chip analog modules, the CTMU can be used to precisely measure time, measure capacitance, measure relative changes in capacitance, or generate output pulses that are independent of the system clock. The CTMU module is ideal for interfacing with capacitive-based sensors.

8.3.1 CTMU Functions

8.3.1.1 Open_CTMU

```
void OpenCTMU(unsigned char config1, unsigned char config2, unsigned char config3)
```

This function configures the CTMU.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing bitwise OR operation ('|'), configurable in the way shown in the examples

Input Parameters	Description
config1	<p>config1 - This contains the parameters to be configured in the CTMUCONH register as defined below</p> <p>Enable/Disable bit</p> <p>*CTMU_ENABLE</p> <p>*CTMU_DISABLE</p> <p>Idle Mode bit</p> <p>*CTMU_IDLE_STOP</p> <p>*CTMU_IDLE_CONTINUE</p> <p>Time Generation Enable bit</p> <p>*CTMU_TIME_GEN_ENABLE</p> <p>*CTMU_TIME_GEN_DISABLE</p> <p>Edge Enable bit</p> <p>*CTMU_EDGE_ENABLE</p> <p>*CTMU_EDGE_DISABLE</p> <p>Edge sequence Enable bit</p> <p>*CTMU_EDGE_SEQUENCE_ON</p> <p>*CTMU_EDGE_SEQUENCE_OFF</p> <p>Analog Current Source Control bit</p> <p>*CTMU_ANA_CURR_SOURCE_GND</p> <p>*CTMU_ANA_CURR_SOURCE_NOT_GND</p> <p>Trigger Control bit</p> <p>*CTMU_TRIG_OUTPUT_ENABLE</p> <p>*CTMU_TRIG_OUTPUT_DISABLE</p> <p>Enable/Disable Interrupt</p> <p>*CTMU_INT_ON</p> <p>*CTMU_INT_OFF</p>

config2	<div><div>This contains the parameters to be configured in the CTMUCONL register as defined below</div><div>Edge2 Polarity select bit</div><div>*CTMU_EDGE2_POLARITY_POS</div><div>*CTMU_EDGE2_POLARITY_NEG</div><div>Edge2 Source Select bit</div><div>*CTMU_EDGE2_SOURCE_CTED1</div><div>*CTMU_EDGE2_SOURCE_CTED2</div><div>*CTMU_EDGE2_SOURCE_OC1</div><div>*CTMU_EDGE2_SOURCE_TIMER1</div><div>Edge1 Polarity Select bit</div><div>*CTMU_EDGE1_POLARITY_POS</div><div>*CTMU_EDGE1_POLARITY_NEG</div><div>Edge1 Source Select bits</div><div>*CTMU_EDGE1_SOURCE_CTED1</div><div>*CTMU_EDGE1_SOURCE_CTED2</div><div>*CTMU_EDGE1_SOURCE_OC1</div><div>*CTMU_EDGE1_SOURCE_TIMER1</div></div>
---------	--

8.3.1.2 CurrentControl_CTMU

void CurrentControlCTMU(unsigned char config)

This function selects the current source range and trims the current source of CTMU.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing bitwise OR operation ('|'), configurable in the way shown in the examples

Input Parameters	Description
------------------	-------------

8.3.1.3 Close_CTMU

void CloseCTMU (void)

This function turns off the CTMU module and disables the CTMU interrupts.

Returns
None

Remarks

This function first disables the CTMU interrupt and then turns off the CTMU module. The Interrupt Flag bit is also cleared.

8.3.2 CTMU Macros

8.3.3 CTMU Examples

8.3.3.1 CTMU Example1

Example Source Code demonstrating peripheral library usage

```

/*****
 *
 * NOTES:
 * Code uses the Peripheral library support available with MCC18 Compiler
 * Code Tested on:
 * PIC18F46J50 controller
 *
 * Capacitor(for capacitance measurement) is connected on AN7.
 * "capacitance" variable contains the final measured value of capacitance.
 *****/

#include "p18cxxx.h"
#define USE_OR_MASKS
#include "adc.h"
#include "ctmu.h"
#include "delays.h"

float voltage,current, capacitance,time;

void main(void)
{
    unsigned char ctmucon1=0,ctmucon2=0,ctmuicon=0,config1=0,config2=0,config3=0,i=0;
    unsigned int adccount=0;

    // current = 0.000055 ; // 55uA - 100_BASE_CURR
    //current = 0.000055 ; // 5.5uA - 10_BASE_CURR
    current = 0.0000055 ; // 0.55uA - BASE_CURR

    TRISB= TRISB | 0x0002; //Configure RB1 as input pin

    /*Configure ADC to read channel 1*/
    //---initialize adc---
    /**** ADC configured for:
        * FOSC-RC as source of conversion clock
        * Result is right justified
        * Aquisition time of 2 AD
        * Channel 7 for sampling
        * ADC interrupt off
        * ADC reference voltage from VDD & VSS
    */
    config1 = ADC_FOSC_RC | ADC_RIGHT_JUST | ADC_2_TAD ;
    config2 = ADC_CH7 | ADC_INT_OFF | ADC_REF_VDD_VSS ;
    config3 = ADC_13ANA | ADC_VBG_OFF;
    OpenADC(config1,config2,config3);
    ADRESH=0; //clear the ADC result register
    ADRESL=0; //clear the ADC result register

```

```

/*Configure the CTMU*/
//-----
/***** CTMU configured for:
 * Edge 1 programmed for a positive edge response
 * Edge 2 programmed for a positive edge response
 * CTED1 is a source select for Edge
 * trigger output disaled
 * Edge sequence of CTMU disabled
 * no edge delay generation
 * CTMU edges blocked
 * Current of 0.55uA
 */
ctmucon2 = CTMU_EDGE1_POLARITY_POS | CTMU_EDGE2_POLARITY_POS | CTMU_EDGE1_SOURCE_CTED1
          | CTMU_EDGE2_SOURCE_CTED1 ;
ctmucon1 = CTMU_TRIG_OUTPUT_DISABLE | CTMU_EDGE_SEQUENCE_OFF | CTMU_TIME_GEN_DISABLE
          | CTMU_EDGE_DISABLE ;
ctmuicon = CTMU_NOMINAL_CURRENT | CTMU_CURR_RANGE_BASE_CURR;
OpenCTMU(ctmucon1,ctmucon2,ctmuicon);

Enbl_CTMUEdge1;    //Enable current source

/* Wait for 50 usec*/
Delay10TCYx(0x05);

Disbl_CTMUEdge1;    //Disable current source

PIR1bits.ADIF=0;    //clear the ADC interrupt

/* Read ADC*/
ConvertADC();        // stop sampling and starts adc conversion
while(BusyADC());    //wait untill the conversion is completed
adccount = ReadADC(); //read the result of conversion
/* Capacitance calculation */
time = 0.00005;
voltage = (adccount*3.3)/1024; // convert ADC count into voltage
capacitance = (current * time)/voltage; // calculate the Capacitance value

CloseADC();          // disable ADC
CloseCTMU();         //disable CTMU

while(1);            //End of program
}

```

8.4 Deep Sleep (DPSLP)

Deep Sleep mode brings the device into its lowest power consumption state without requiring the use of external switches to remove power from the device. During deep sleep, the on-chip VDDCORE voltage regulator is powered down, effectively disconnecting power to the core logic of the microcontroller.

8.4.1 Deep Sleep Functions

8.4.1.1 Goto_DeepSleep

```
void gotoDeepSleep( unsigned int config )
```

This function saves content to DPGPRx registers, enables the sources of deep sleep wake up and puts the device to deep sleep.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing bitwise OR operation ('|'), configurable as shown in the examples

Input Parameters	Description
config	contains parameters to configure the deep sleep wake up sources Deep Sleep Ultra Low Power Wake Up *DPSLP_ULPWU_ENABLE *DPSLP_ULPWU_DISABLE RTCC wake up source enable/disable *DPSLP_RTCC_WAKEUP_ENABLE *DPSLP_RTCC_WAKEUP_DISABLE

Returns

None

Remarks

Affects Ultra Low Power wake up module configuration.

8.4.1.2 DeepSleep_WakeUpSource

```
void DeepSleepWakeUpSource( SRC* ptr)
```

This function updates source of wake up of device from deep sleep.

Input Parameters

Input Parameters	Description
SRC	Pointer to union that reflects the status of deep sleep wake up source

Returns

Updates the SRC union with the source of wake up from deep sleep

Remarks

This function updates union SRC reflects the wake up source of deep sleep.

8.4.1.3 IsResetFrom_DeepSleep

```
unsigned char IsResetFromDeepSleep( void)
```

This function returns the source of reset.

Returns

Reset source

* -1 : Reset source is Deep Sleep Wake up

* 0 : Reset source is pure Power on Reset/BOR during sleep/some other source

Remarks

Clears DS bit in WDTCON.

8.4.1.4 Read_DSGPR

```
void ReadDSGPR( CONTEXT* ptr)
```

This function reads context saved in DSGPRx registers and updates in CONTEXT structure.

Input Parameters

Input Parameters	Description
CONTEXT	Pointer to union to which the state of device stored before going to Deep Sleep is read back

Returns

Updates the CONTEXT union with the state of device stored before entering to Deep Sleep

8.4.1.5 ULPWakeUp_Enable

```
void ULPWakeUpEnable( void)
```

This function enables the Ultra Low power wake up module.

Remarks

User must have charged the capacitor prior to enabling the deep sleep wake up from ultra low power wake up and must have called "EnableIntULPWU" function.

8.4.2 DeepSleep Structs,Records,Enums

8.4.2.1 _BOOL

Boolean value with Logic 1 and Logic 0 values defined

8.4.3 Deep Sleep Macros

8.4.4 Deep Sleep Examples

8.4.4.1 Deep Sleep Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PIC18F46J50 controller
*****/

#define USE_OR_MASKS
#include <p18cxxx.h>
#include "dpslp.h"
#include "portb.h"
#include "rtcc.h"
#define TRUE 1

//Function Prototypes
void user_main(void);
void USER_Function(void);
void RTCC_configure(void);
void Check_INT_SRV_Routine(void);

//Global structures used in deep sleep library
SRC ptr;
CONTEXT read_state;
rtccTimeDate RtccTimeDate ,RtccAlrmTimeDate, Rtcc_read_TimeDate ;

//main function
int main(void)
{
    if(IsResetFromDeepSleep()==0xFF) //if this is the reset after the
                                    //deep_sleep wakeup...then do this
    {
        ReadDSGPR(&read_state); //Read the deep sleep GPR
        DeepSleepWakeUpSource(&ptr); //Check the deep sleep wakup soruce (if required)
        ReleaseDeepSleep(); //Release the Deep sleep (IO configuration)
        if((ptr.WK_SRC.DS_POR)==TRUE)
        {

```

```

        //deep sleep wakeup source is DSPOR
        USER_Function();
    }
    if(ptr.WK_SRC.DS_MCLR==TRUE)
    {
        //deep sleep wakeup source is MCLR
        USER_Function();
    }
    if(ptr.WK_SRC.DS_RTC==TRUE)
    {
        //deep sleep wakeup source is RTCC
        USER_Function();
    }
    if(ptr.WK_SRC.DS_WDT==TRUE)
    {
        //deep sleep wakeup source is DSWDT
        USER_Function();
    }
    if(ptr.WK_SRC.DS_FLT==TRUE)
    {
        //deep sleep wakeup source is Falut in deep sleep configuration
        USER_Function();
    }
    if(ptr.WK_SRC.DS_INT0==TRUE)
    {
        //deep sleep wakeup source is INT0
        USER_Function();
    }
    if(ptr.WK_SRC.DS_BOR==TRUE)
    {
        //deep sleep wakeup source is DSBOR
        USER_Function();
    }
    if(ptr.WK_SRC.DS_ULP==TRUE)
    {
        //deep sleep wakeup source is DSULP
        USER_Function();
    }
    user_main(); //call the "user_main" -- The User application program
}
else //else... this is the Normal (pure) Power_on Reset...do the normal init
{
    //Normal POR init program here
    user_main(); //call the "user_main" -- The User application program
}
while(1);
}

void user_main(void)
{
    while(1)
    {
        {
            Write_DSGPR(0x67,0x7A); //Save state of system prior to deep sleep
            RTCC_configure(); //Configure RTCC as one of sources of wake up
            /*** Charge the capacitor on RA0 for ultra low power
            /** wake up as source of wake up from deep sleep****
            TRISAbits.TRISA0 = 0;
            PORTAbits.RA0 = 1;
            for(i = 0; i < 10000; i++) Nop();
            /*** configure INT0 with pullups enabled, falling edge ***
            config = PORTB_CHANGE_INT_ON | FALLING_EDGE_INT | PORTB_PULLUPS_ON;
            OpenRB0INT(config); //configures INT0 & enables it
            /******* configure deep sleep wake up sources
            /*******
            config = ( DPSLP_ULPWU_ENABLE | DPSLP_RTCC_WAKEUP_DISABLE);
            GotoDeepSleep(config); //This function puts the device into deep sleep

            Check_INT_SRV_Routine(); //Interrupt occured simultaneously while going to deep

```

```

    sleep.Hence service it.
    }
}

void USER_Function(void);
{
    Nop();
    //USER written application to process the source of wake up from deep sleep
}

void RTCC_configure(void)
{
    unsigned int i=0,j=0;

    RtccInitClock(); //turn on clock source
    for(i=0;i<4;i++) //Delay for Secondary oscillator to stabilize
    for(j=0;j<60000;j++);

    RtccWrOn(); //write enable the rtcc registers
    RtccTimeDate.f.hour = 9; //Set Date and time
    RtccTimeDate.f.min = 10;
    RtccTimeDate.f.sec = 9;
    RtccTimeDate.f.mday = 18;
    RtccTimeDate.f.mon = 1;
    RtccTimeDate.f.year = 8;
    RtccAlrmTimeDate.f.hour = RtccTimeDate.f.hour;
    RtccAlrmTimeDate.f.min = RtccTimeDate.f.min ;
    RtccAlrmTimeDate.f.sec = RtccTimeDate.f.sec + 9;
    RtccAlrmTimeDate.f.mday = RtccTimeDate.f.mday;
    RtccAlrmTimeDate.f.mon = RtccTimeDate.f.mon;
    RtccAlrmTimeDate.f.year = RtccTimeDate.f.year;

    RtccWriteTimeDate(&RtccTimeDate,1); //write into registers
    RtccSetAlarmRpt(RTCC_RPT_TEN_SEC,1); //Set the alarm repeat to every 10 seconds
    RtccSetAlarmRptCount(5,1);
    RtccWriteAlrmTimeDate(&RtccAlrmTimeDate);

    mRtccOn(); //enable the rtcc
    mRtccAlrmEnable(); //enable the rtcc alarm to wake the device up from deep sleep
}

void Check_INT_SRV_Routine(void)
{
    while(1); //User written INT service routine to handle interrupt
}

```

8.5 Peripheral Pin Select (PPS)

The peripheral pin select feature provides an enabling the user's peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The peripheral pin select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. Peripheral pin select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

8.5.1 PPS Macros

8.5.1.1 PPS_Input_Page1/2

```
#define PPSInput(fn,pin) iPPSInput(IN_FN_##fn,IN_PIN_##pin)
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	function to be assigned for particular pin: * PPS_INT1 * PPS_INT2 * PPS_INT3 * PPS_T0CK * PPS_T3CK * PPS_IC1 * PPS_IC2 * PPS_T1G * PPS_T3G * PPS_RX2DT2 * PPS_CK2 * PPS_SDI2 * PPS_SCK2IN * PPS_SS2IN * PPS_FLT0

pin	pin number(x) for which functionality has to be assigned: * PPS_RP0 * PPS_RP1 * PPS_RP2 * PPS_RP3 * PPS_RP4 * PPS_RP5 * PPS_RP6 * PPS_RP7 * PPS_RP8 * PPS_RP9 * PPS_RP10 * PPS_RP11 * PPS_RP12 * PPS_RP13 * PPS_RP17 * PPS_RP18 * PPS_RP19 * PPS_RP20 * PPS_RP21 * PPS_RP22 * PPS_RP23 * PPS_RP24 * PPS_VSS
-----	--

Remarks

Equivalent to iPPSInput (see page 936) Macro

8.5.1.2 iPPS_Input_Page1/2

```
#define iPPSInput(fn,pin) fn=pin
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	function to be assigned for particular pin: * IN_FN_PPS_INT1 * IN_FN_PPS_INT2 * IN_FN_PPS_INT3 * IN_FN_PPS_T0CK * IN_FN_PPS_T3CK * IN_FN_PPS_IC1 * IN_FN_PPS_IC2 * IN_FN_PPS_T1G * IN_FN_PPS_T3G * IN_FN_PPS_RX2DT2 * IN_FN_PPS_CK2 * IN_FN_PPS_SDI2 * IN_FN_PPS_SCK2IN * IN_FN_PPS_SS2IN * IN_FN_PPS_FLT0
pin	pin number(x) for which functionality has to be assigned: * IN_PIN_PPS_RP0 * IN_PIN_PPS_RP1 * IN_PIN_PPS_RP2 * IN_PIN_PPS_RP3 * IN_PIN_PPS_RP4 * IN_PIN_PPS_RP5 * IN_PIN_PPS_RP6 * IN_PIN_PPS_RP7 * IN_PIN_PPS_RP8 * IN_PIN_PPS_RP9 * IN_PIN_PPS_RP10 * IN_PIN_PPS_RP11 * IN_PIN_PPS_RP12 * IN_PIN_PPS_RP13 * IN_PIN_PPS_RP17 * IN_PIN_PPS_RP18 * IN_PIN_PPS_RP19 * IN_PIN_PPS_RP20 * IN_PIN_PPS_RP21 * IN_PIN_PPS_RP22 * IN_PIN_PPS_RP23 * IN_PIN_PPS_RP24 * IN_PIN_PPS_VSS

Remarks

Equivalent to PPSInput (see page 935) Macro

8.5.1.3 iPPS_Input_Page1/2_1

```
#define iPPSInput(fn,pin) fn=pin
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	function to be assigned for particular pin: * IN_FN_PPS_INT1 * IN_FN_PPS_INT2 * IN_FN_PPS_INT3 * IN_FN_PPS_T0CK * IN_FN_PPS_T3CK * IN_FN_PPS_IC1 * IN_FN_PPS_IC2 * IN_FN_PPS_T1G * IN_FN_PPS_T3G * IN_FN_PPS_RX2DT2 * IN_FN_PPS_CK2 * IN_FN_PPS_SDI2 * IN_FN_PPS_SCK2IN * IN_FN_PPS_SS2IN * IN_FN_PPS_FLT0

pin	<div>pin number(x) for which functionality has to be assigned:<ul style="list-style-type: none">* IN_PIN_PPS_RP0* IN_PIN_PPS_RP1* IN_PIN_PPS_RP2* IN_PIN_PPS_RP3* IN_PIN_PPS_RP4* IN_PIN_PPS_RP5* IN_PIN_PPS_RP6* IN_PIN_PPS_RP7* IN_PIN_PPS_RP8* IN_PIN_PPS_RP9* IN_PIN_PPS_RP10* IN_PIN_PPS_RP11* IN_PIN_PPS_RP12* IN_PIN_PPS_RP13* IN_PIN_PPS_RP14* IN_PIN_PPS_RP15* IN_PIN_PPS_RP16* IN_PIN_PPS_RP17* IN_PIN_PPS_RP18* IN_PIN_PPS_RP19* IN_PIN_PPS_RP20* IN_PIN_PPS_RP21* IN_PIN_PPS_RP22* IN_PIN_PPS_RP23* IN_PIN_PPS_RP24* IN_PIN_PPS_VSS</div>
-----	--

Remarks

Equivalent to PPSInput (see page 939) Macro

8.5.1.4 PPS_Input_Page1/2_1

```
#define PPSInput(fn,pin) iPPSInput(IN_FN_##fn,IN_PIN_##pin)
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	<p>function to be assigned for particular pin:</p> <ul style="list-style-type: none"> * IN_FN_PPS_INT1 * IN_FN_PPS_INT2 * IN_FN_PPS_INT3 * IN_FN_PPS_T0CK * IN_FN_PPS_T3CK * IN_FN_PPS_IC1 * IN_FN_PPS_IC2 * IN_FN_PPS_T1G * IN_FN_PPS_T3G * IN_FN_PPS_RX2DT2 * IN_FN_PPS_CK2 * IN_FN_PPS_SDI2 * IN_FN_PPS_SCK2IN * IN_FN_PPS_SS2IN * IN_FN_PPS_FLT0
pin	<p>pin number(x) for which functionality has to be assigned:</p> <ul style="list-style-type: none"> * IN_PIN_PPS_RP0 * IN_PIN_PPS_RP1 * IN_PIN_PPS_RP2 * IN_PIN_PPS_RP3 * IN_PIN_PPS_RP4 * IN_PIN_PPS_RP5 * IN_PIN_PPS_RP6 * IN_PIN_PPS_RP7 * IN_PIN_PPS_RP8 * IN_PIN_PPS_RP9 * IN_PIN_PPS_RP10 * IN_PIN_PPS_RP11 * IN_PIN_PPS_RP12 * IN_PIN_PPS_RP13 * IN_PIN_PPS_RP14 * IN_PIN_PPS_RP15 * IN_PIN_PPS_RP16 * IN_PIN_PPS_RP17 * IN_PIN_PPS_RP18 * IN_PIN_PPS_RP19 * IN_PIN_PPS_RP20 * IN_PIN_PPS_RP21 * IN_PIN_PPS_RP22 * IN_PIN_PPS_RP23 * IN_PIN_PPS_RP24 * IN_PIN_PPS_VSS

Remarks

Equivalent to iPPSInput (see page 938) Macro

8.5.1.5 PPS_Output_Page1

```
#define PPSOutput(pin,fn) iPPSOutput(OUT_PIN_##pin,OUT_FN_##fn)
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	pin number(x) for which functionality has to be assigned: * PPS_RP0 * PPS_RP1 * PPS_RP2 * PPS_RP3 * PPS_RP4 * PPS_RP5 * PPS_RP6 * PPS_RP7 * PPS_RP8 * PPS_RP9 * PPS_RP10 * PPS_RP11 * PPS_RP12 * PPS_RP13 * PPS_RP14 * PPS_RP15 * PPS_RP16 * PPS_RP17 * PPS_RP18

fn	function to be assigned for particular pin: * PPS_NULL * PPS_C1OUT * PPS_C2OUT * PPS_TX2CK2 * PPS_DT2 * PPS_SDO2 * PPS_SCK2 * PPS_SSDMA * PPS_ULPWU * PPS_CCP1P1A * PPS_P1B * PPS_P1C * PPS_P1D * PPS_CCP2P2A * PPS_P2B * PPS_P2C * PPS_P2D
----	--

Remarks

Equivalent to iPPSOutput Macro

8.5.1.6 PPS_Output_Page2

```
#define PPSOutput(pin,fn) iPPSOutput(OUT_PIN_##pin,OUT_FN_##fn)
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	pin number(x) for which functionality has to be assigned: * PPS_RP0 * PPS_RP1 * PPS_RP2 * PPS_RP3 * PPS_RP4 * PPS_RP5 * PPS_RP6 * PPS_RP7 * PPS_RP8 * PPS_RP9 * PPS_RP10 * PPS_RP11 * PPS_RP12 * PPS_RP13 * PPS_RP14 * PPS_RP15 * PPS_RP16 * PPS_RP17 * PPS_RP18 * PPS_RP19 * PPS_RP20 * PPS_RP21 * PPS_RP22 * PPS_RP23 * PPS_RP24
fn	function to be assigned for particular pin: * PPS_NULL * PPS_C1OUT * PPS_C2OUT * PPS_TX2CK2 * PPS_DT2 * PPS_SDO2 * PPS_SCK2 * PPS_SSDMA * PPS_ULPWU * PPS_CCP1P1A * PPS_P1B * PPS_P1C * PPS_P1D * PPS_CCP2P2A * PPS_P2B * PPS_P2C * PPS_P2D

Remarks

Equivalent to iPPSOutput Macro

8.5.1.7 iPPS_Output_Page1

```
#define iPPSOutput(pin,fn) pin=fn
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	pin number(x) for which functionality has to be assigned: * OUT_PIN_PPS_RP0 * OUT_PIN_PPS_RP1 * OUT_PIN_PPS_RP2 * OUT_PIN_PPS_RP3 * OUT_PIN_PPS_RP4 * OUT_PIN_PPS_RP5 * OUT_PIN_PPS_RP6 * OUT_PIN_PPS_RP7 * OUT_PIN_PPS_RP8 * OUT_PIN_PPS_RP9 * OUT_PIN_PPS_RP10 * OUT_PIN_PPS_RP11 * OUT_PIN_PPS_RP12 * OUT_PIN_PPS_RP13 * OUT_PIN_PPS_RP14 * OUT_PIN_PPS_RP15 * OUT_PIN_PPS_RP16 * OUT_PIN_PPS_RP17 * OUT_PIN_PPS_RP18

fn	<div>function to be assigned for particular pin:<ul style="list-style-type: none">* OUT_FN_PPS_NULL* OUT_FN_PPS_C1OUT* OUT_FN_PPS_C2OUT* OUT_FN_PPS_TX2CK2* OUT_FN_PPS_DT2* OUT_FN_PPS_SDO2* OUT_FN_PPS_SCK2* OUT_FN_PPS_SSDMA* OUT_FN_PPS_ULPWU* OUT_FN_PPS_CCP1P1A* OUT_FN_PPS_P1B* OUT_FN_PPS_P1C* OUT_FN_PPS_P1D* OUT_FN_PPS_CCP2P2A* OUT_FN_PPS_P2B* OUT_FN_PPS_P2C* OUT_FN_PPS_P2D</div>
----	--

Remarks

Equivalent to iPPSOutput Macro

8.5.1.8 iPPS_Output_Page2

```
#define iPPSOutput(pin,fn) pin=fn
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	<p>pin number(x) for which functionality has to be assigned:</p> <ul style="list-style-type: none"> * OUT_PIN_PPS_RP0 * OUT_PIN_PPS_RP1 * OUT_PIN_PPS_RP2 * OUT_PIN_PPS_RP3 * OUT_PIN_PPS_RP4 * OUT_PIN_PPS_RP5 * OUT_PIN_PPS_RP6 * OUT_PIN_PPS_RP7 * OUT_PIN_PPS_RP8 * OUT_PIN_PPS_RP9 * OUT_PIN_PPS_RP10 * OUT_PIN_PPS_RP11 * OUT_PIN_PPS_RP12 * OUT_PIN_PPS_RP13 * OUT_PIN_PPS_RP14 * OUT_PIN_PPS_RP15 * OUT_PIN_PPS_RP16 * OUT_PIN_PPS_RP17 * OUT_PIN_PPS_RP18 * OUT_PIN_PPS_RP19 * OUT_PIN_PPS_RP20 * OUT_PIN_PPS_RP21 * OUT_PIN_PPS_RP22 * OUT_PIN_PPS_RP23 * OUT_PIN_PPS_RP24
fn	<p>function to be assigned for particular pin:</p> <ul style="list-style-type: none"> * OUT_FN_PPS_NULL * OUT_FN_PPS_C1OUT * OUT_FN_PPS_C2OUT * OUT_FN_PPS_TX2CK2 * OUT_FN_PPS_DT2 * OUT_FN_PPS_SDO2 * OUT_FN_PPS_SCK2 * OUT_FN_PPS_SSDMA * OUT_FN_PPS_ULPWU * OUT_FN_PPS_CCP1P1A * OUT_FN_PPS_P1B * OUT_FN_PPS_P1C * OUT_FN_PPS_P1D * OUT_FN_PPS_CCP2P2A * OUT_FN_PPS_P2B * OUT_FN_PPS_P2C * OUT_FN_PPS_P2D

Remarks

Equivalent to iPPSOutput Macro

8.5.1.9 iPPS_Input_Page3

```
#define iPPSInput(fn,pin) fn=pin
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	<div>function to be assigned for particular pin:</div> <div><ul style="list-style-type: none">* IN_FN_PPS_INT1* IN_FN_PPS_INT2* IN_FN_PPS_INT3* IN_FN_PPS_T0CK* IN_FN_PPS_T3CK* IN_FN_PPS_T5CK* IN_FN_PPS_IC1* IN_FN_PPS_IC2* IN_FN_PPS_IC3* IN_FN_PPS_T1G* IN_FN_PPS_T3G* IN_FN_PPS_T5G* IN_FN_PPS_RX2DT2* IN_FN_PPS_CK2* IN_FN_PPS_SDI2* IN_FN_PPS_SCK2IN* IN_FN_PPS_SS2IN* IN_FN_PPS_FLT0</div>

pin	<div>pin number(x) for which functionality has to be assigned:<ul style="list-style-type: none">* IN_PIN_PPS_RP0* IN_PIN_PPS_RP1* IN_PIN_PPS_RP2* IN_PIN_PPS_RP3* IN_PIN_PPS_RP4* IN_PIN_PPS_RP5* IN_PIN_PPS_RP6* IN_PIN_PPS_RP7* IN_PIN_PPS_RP8* IN_PIN_PPS_RP9* IN_PIN_PPS_RP10* IN_PIN_PPS_RP11* IN_PIN_PPS_RP12* IN_PIN_PPS_RP13* IN_PIN_PPS_RP17* IN_PIN_PPS_RP18* IN_PIN_PPS_RP19* IN_PIN_PPS_RP20* IN_PIN_PPS_RP21* IN_PIN_PPS_RP22* IN_PIN_PPS_RP23* IN_PIN_PPS_RP24* IN_PIN_PPS_VSS</div>
-----	--

Remarks

Equivalent to PPSInput (see page 935) Macro

8.5.1.10 PPS_Input_Page3

```
#define PPSInput(fn,pin) iPPSInput(IN_FN_##fn,IN_PIN_##pin)
```

The macro assigns given pin as input pin by configuring register RPINRx

Input Parameters

Input Parameters	Description
fn	function to be assigned for particular pin: * PPS_INT1 * PPS_INT2 * PPS_INT3 * PPS_T0CK * PPS_T3CK * PPS_T5CK * PPS_IC1 * PPS_IC2 * PPS_IC3 * PPS_T1G * PPS_T3G * PPS_T5G * PPS_RX2DT2 * PPS_CK2 * PPS_SDI2 * PPS_SCK2IN * PPS_SS2IN * PPS_FLT0
pin	pin number(x) for which functionality has to be assigned: * PPS_RP0 * PPS_RP1 * PPS_RP2 * PPS_RP3 * PPS_RP4 * PPS_RP5 * PPS_RP6 * PPS_RP7 * PPS_RP8 * PPS_RP9 * PPS_RP10 * PPS_RP11 * PPS_RP12 * PPS_RP13 * PPS_RP17 * PPS_RP18 * PPS_RP19 * PPS_RP20 * PPS_RP21 * PPS_RP22 * PPS_RP23 * PPS_RP24 * PPS_VSS

Remarks

Equivalent to iPPSInput (see page 936) Macro

8.5.1.11 iPPS_Output_Page3

```
#define iPPSOutput(pin,fn) pin=fn
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	pin number(x) for which functionality has to be assigned: * OUT_PIN_PPS_RP0 * OUT_PIN_PPS_RP1 * OUT_PIN_PPS_RP2 * OUT_PIN_PPS_RP3 * OUT_PIN_PPS_RP4 * OUT_PIN_PPS_RP5 * OUT_PIN_PPS_RP6 * OUT_PIN_PPS_RP7 * OUT_PIN_PPS_RP8 * OUT_PIN_PPS_RP9 * OUT_PIN_PPS_RP10 * OUT_PIN_PPS_RP11 * OUT_PIN_PPS_RP12 * OUT_PIN_PPS_RP13 * OUT_PIN_PPS_RP17 * OUT_PIN_PPS_RP18 * OUT_PIN_PPS_RP19 * OUT_PIN_PPS_RP20 * OUT_PIN_PPS_RP21 * OUT_PIN_PPS_RP22 * OUT_PIN_PPS_RP23 * OUT_PIN_PPS_RP24

fn	<div>function to be assigned for particular pin:<ul style="list-style-type: none">* OUT_FN_PPS_NULL* OUT_FN_PPS_C1OUT* OUT_FN_PPS_C2OUT* OUT_FN_PPS_TX2CK2* OUT_FN_PPS_DT2* OUT_FN_PPS_SDO2* OUT_FN_PPS_SCK2* OUT_FN_PPS_SSDMA* OUT_FN_PPS_ULPWU* OUT_FN_PPS_CCP1P1A* OUT_FN_PPS_P1B* OUT_FN_PPS_P1C* OUT_FN_PPS_P1D* OUT_FN_PPS_CCP2P2A* OUT_FN_PPS_P2B* OUT_FN_PPS_P2C* OUT_FN_PPS_P2D* OUT_FN_PPS_CCP3P3A* OUT_FN_PPS_P3B* OUT_FN_PPS_P3C* OUT_FN_PPS_P3D</div>
----	--

Remarks

Equivalent to iPPSOutput Macro

8.5.1.12 PPS_Output_Page3

```
#define PPSOutput(pin,fn) iPPSOutput(OUT_PIN_##pin,OUT_FN_##fn)
```

The macro assigns given pin as output pin by configuring register RPORx.

Input Parameters

Input Parameters	Description
pin	pin number(x) for which functionality has to be assigned: * PPS_RP0 * PPS_RP1 * PPS_RP2 * PPS_RP3 * PPS_RP4 * PPS_RP5 * PPS_RP6 * PPS_RP7 * PPS_RP8 * PPS_RP9 * PPS_RP10 * PPS_RP11 * PPS_RP12 * PPS_RP13 * PPS_RP17 * PPS_RP18 * PPS_RP19 * PPS_RP20 * PPS_RP21 * PPS_RP22 * PPS_RP23 * PPS_RP24
fn	function to be assigned for particular pin: * PPS_NULL * PPS_C1OUT * PPS_C2OUT * PPS_TX2CK2 * PPS_DT2 * PPS_SDO2 * PPS_SCK2 * PPS_SSDMA * PPS_ULPWU * PPS_CCP1P1A * PPS_P1B * PPS_P1C * PPS_P1D * PPS_CCP2P2A * PPS_P2B * PPS_P2C * PPS_P2D * PPS_CCP3P3A * PPS_P3B * PPS_P3C * PPS_P3D

Remarks

Equivalent to iPPSOutput Macro

8.5.2 PPS Examples

8.5.2.1 PPS Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PIC18F46J50 controller
*
* signal to be compared are applied to C1INA & C1INB and output is tapped at C1OUT mapped
to pin RP24 using PPS
*****/
#define USE_OR_MASKS
#include <p18f46j50.h>
#include "ancomp.h"
#include "pps.h"

unsigned char CMPResult;

void main(void)
{
    unsigned char config=0x00,h=0;
    TRISD=0x00;

    //---PPS Configuration---
    PPSUnlock();
    iPPSOutput(OUT_PIN_PPS_RP24,OUT_FN_PPS_C1OUT);           //Configure RP24 as C1OUT
    pin
    PPSLock();

    //*****Configure Analog Comparator *****
    /*** Analog comparator is configured for:
    * Output is non-inverted
    * comparator 1 configured with C1INA as positive input terminal and C1INB as negative
input terminal
    * comparator output is enabled onto pin C1OUT mapped to RP24 using PPS
    * comparator interrupt is enabled
    * interrupt on both the edges
    */
    config = COMP_OP_NINV | COMP_REF_CINA | COMP_VINM_CINB | COMP_OP_EN ;
    Open_ancomp1(config);

    while(1)
    {
        if( CMSTAT&0x01 )    //check for the comparator output status or comparator
interrupt flag status
            CMPResult=1;        //flag the logic 1 status of comparator output
        else
            CMPResult=0;        // flag the logic 0 status of comparator output
    }
    //--- Turn off comparator ---
    Close_ancomp1();

```

```
}
```

8.6 Input Capture (INCAP)

Input Capture modules offers a wide range of configuration and operating options for capturing external pulse events and generating interrupts. Key features of the input capture module include:

- Hardware-configurable for 16-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with many user-selectable trigger/sync sources available
- A buffer for capturing and holding timer values for several events
- Configurable interrupt generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

8.6.1 Input Capture Function

8.6.1.1 Open_Capture

8.6.1.1.1 Open_Capture1

```
void OpenCapture1(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.2 Open_Capture2

void OpenCapture2(unsigned char config)

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.3 Open_Capture2_Page2

void OpenCapture2(unsigned char config)

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer source selection</p> <ul style="list-style-type: none"> * CCP_2_SEL_TMR12 * CCP_2_SEL_TMR34 * CCP_2_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.4 Open_Capture3

```
void OpenCapture3(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.5 Open_Capture3_Page2

void OpenCapture3(unsigned char config)
--

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer source selection</p> <ul style="list-style-type: none"> * CCP_3_SEL_TMR12 * CCP_3_SEL_TMR34 * CCP_3_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.6 Open_Capture4

void OpenCapture4(unsigned char config)

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.7 Open_Capture4_Page2

```
void OpenCapture4(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR36 * CCP_4_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.7.1 Open_Capture4_Page4
void OpenCapture4(unsigned char config)

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR56 * CCP_4_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.8 Open_Capture4_Page3

```
void OpenCapture4(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer source selection</p> <ul style="list-style-type: none"> * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.9 Open_Capture5

```
void OpenCapture5(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.10 Open_Capture5_Page2

```
void OpenCapture5(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK</div> <div>Timer Selection<ul style="list-style-type: none">* CCP_5_SEL_TMR12* CCP_5_SEL_TMR54* CCP_5_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.11 Open_Capture5_Page3

```
void OpenCapture5(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer source selection</p> <ul style="list-style-type: none"> * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR34 * CCP_5_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.12 Open_ECapture1

void OpenECapture1(unsigned char config)

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

NOTE: In addition to opening the capture, the appropriate timer module must be enabled before any of the captures will operate. See the data sheet for CCP and timer interconnect configurations and Timer Functions for the arguments used with CCP in OpenTimer3.

8.6.1.1.13 Open_ECapture1_Page2

void OpenECapture1(unsigned char config)

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * ECAP_EVERY_FALL_EDGE * ECAP_EVERY_RISE_EDGE * ECAP_EVERY_4_RISE_EDGE * ECAP_EVERY_16_RISE_EDGE * ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none"> * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR36 * ECCP_1_SEL_TMR38 * ECCP_1_SEL_TMR310 * ECCP_1_SEL_TMR312 * ECCP_1_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.14 Open_ECapture1_Page3

```
void OpenECapture1(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the ECCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK</div> <div>Timer Source Selection<ul style="list-style-type: none">* ECCP_1_SEL_TMR12* ECCP_1_SEL_TMR34* ECCP_1_SEL_TMR36* ECCP_1_SEL_TMR38* ECCP_1_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.15 Open_ECapture1_Page4

void OpenECapture1(unsigned char config)

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * ECAP_EVERY_FALL_EDGE * ECAP_EVERY_RISE_EDGE * ECAP_EVERY_4_RISE_EDGE * ECAP_EVERY_16_RISE_EDGE * ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none"> * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.16 Open_ECapture2_Page1

```
void OpenECapture2(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none">* ECCP_2_SEL_TMR12* ECCP_2_SEL_TMR34* ECCP_2_SEL_TMR36* ECCP_2_SEL_TMR38* ECCP_2_SEL_TMR310* ECCP_2_SEL_TMR_MASK

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.17 Open_ECapture2_Page2

```
void OpenECapture2(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * ECAP_EVERY_FALL_EDGE * ECAP_EVERY_RISE_EDGE * ECAP_EVERY_4_RISE_EDGE * ECAP_EVERY_16_RISE_EDGE * ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none"> * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR36 * ECCP_2_SEL_TMR38 * ECCP_2_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.18 Open_ECapture3_Page1

```
void OpenECapture2(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the ECCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none">* ECCP_3_SEL_TMR12* ECCP_3_SEL_TMR34* ECCP_3_SEL_TMR36* ECCP_3_SEL_TMR38* ECCP_3_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.19 Open_Capture6_Page1

void OpenCapture6(unsigned char config)

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_6_SEL_TMR12 * CCP_6_SEL_TMR52 * CCP_6_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.20 Open_Capture7_Page1

```
void OpenCapture7(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK</div> <div>Timer Selection<ul style="list-style-type: none">* CCP_7_SEL_TMR12* CCP_7_SEL_TMR54* CCP_7_SEL_TMR56* CCP_7_SEL_TMR58* CCP_7_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.21 Open_Capture8_Page1

```
void OpenCapture8(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR74 * CCP_8_SEL_TMR76 * CCP_8_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.22 Open_Capture8_Page2

```
void OpenCapture8(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR14 * CCP_8_SEL_TMR16 * CCP_8_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.23 Open_Capture9_Page1

```
void OpenCapture9(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none"> * CAPTURE_INT_ON * CAPTURE_INT_OFF * CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none"> * CAP_EVERY_FALL_EDGE * CAP_EVERY_RISE_EDGE * CAP_EVERY_4_RISE_EDGE * CAP_EVERY_16_RISE_EDGE * CAP_MODE_MASK <p>Timer Selection</p> <ul style="list-style-type: none"> * CCP_9_SEL_TMR12 * CCP_9_SEL_TMR74 * CCP_9_SEL_TMR_MASK

Returns

None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.24 Open_Capture10_Page1

```
void OpenCapture10(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK</div> <div>Timer Selection<ul style="list-style-type: none">* CCP_10_SEL_TMR12* CCP_10_SEL_TMR72* CCP_10_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.25 Open_ECapture1_Page5

```
void OpenECapture1(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>This contains the parameters to be configured in the CCPxCON register as defined below</p> <p>Enable CCP Interrupts:</p> <ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK <p>Capture configuration</p> <ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK <p>Timer Source Selection</p> <ul style="list-style-type: none">* ECCP_1_SEL_TMR12* ECCP_1_SEL_TMR34* ECCP_1_SEL_TMR56* ECCP_1_SEL_TMR_MASK

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.26 Open_ECapture2_Page4

```
void OpenECapture2(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK</div> <div>Timer Source Selection<ul style="list-style-type: none">* ECCP_2_SEL_TMR12* ECCP_2_SEL_TMR34* ECCP_2_SEL_TMR56* ECCP_2_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.27 Open_ECapture3_Page4

```
void OpenECapture3(unsigned char config)
```

This function configures the Extended Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* ECAP_EVERY_FALL_EDGE* ECAP_EVERY_RISE_EDGE* ECAP_EVERY_4_RISE_EDGE* ECAP_EVERY_16_RISE_EDGE* ECAP_MODE_MASK</div> <div>Timer Source Selection<ul style="list-style-type: none">* ECCP_3_SEL_TMR12* ECCP_3_SEL_TMR34* ECCP_3_SEL_TMR56* ECCP_3_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.1.28 Open_Capture5_Page4

```
void OpenCapture5(unsigned char config)
```

This function configures the Input Capture module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>This contains the parameters to be configured in the CCPxCON register as defined below</div> <div>Enable CCP Interrupts:<ul style="list-style-type: none">* CAPTURE_INT_ON* CAPTURE_INT_OFF* CAPTURE_INT_MASK</div> <div>Capture configuration<ul style="list-style-type: none">* CAP_EVERY_FALL_EDGE* CAP_EVERY_RISE_EDGE* CAP_EVERY_4_RISE_EDGE* CAP_EVERY_16_RISE_EDGE* CAP_MODE_MASK</div> <div>Timer Selection<ul style="list-style-type: none">* CCP_5_SEL_TMR12* CCP_5_SEL_TMR34* CCP_5_SEL_TMR56* CCP_5_SEL_TMR_MASK</div>

Returns
None

Remarks

This function configures the input capture for idle mode, clock select, capture per interrupt and mode select

8.6.1.2 Read_Capture

8.6.1.2.1 Read_Capture1

unsigned int ReadCapture1(void)

This function reads the pending Input Capture buffer.

Returns
This routine reads the CCPR1L andCCPR1H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.2 Read_Capture2

unsigned int ReadCapture1(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR1L and CCPR1H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.3 Read_Capture3

unsigned int ReadCapture3(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR3L and CCPR3H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.4 Read_Capture4

unsigned int ReadCapture4(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR4L and CCPR4H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.5 Read_Capture5

unsigned int ReadCapture5(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR5L and CCPR5H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.6 Read_Capture6

unsigned int ReadCapture6(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR6L and CCPR6H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.7 Read_Capture7

unsigned int ReadCapture7(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR7L and CCPR7H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.8 Read_Capture8

unsigned int ReadCapture8(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR8L and CCPR8H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.9 Read_Capture9

unsigned int ReadCapture9(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR9L and CCPR9H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.10 Read_Capture10

unsigned int ReadCapture10(void)

This function reads the pending Input Capture buffer.

Returns

This routine reads the CCPR10L and CCPR10H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.11 Read_ECapture1

unsigned int ReadECapture1(void)

This function reads the pending Extended Input Capture buffer.

Returns

This routine reads the ECCPR1L and ECCPR1H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.12 Read_ECapture2

unsigned int ReadECapture2(void)

This function reads the pending Extended Input Capture buffer.

Returns

This routine reads the ECCPR2L and ECCPR2H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.2.13 Read_ECapture3

```
unsigned int ReadECapture3(void)
```

This function reads the pending Extended Input Capture buffer.

Returns

This routine reads the ECCPR3L and ECCPR3H into the union Cap of type CapResult that is defined in global data space. The int result is then returned.

Remarks

This function reads the pending Input Capture buffer

8.6.1.3 Close_Capture

8.6.1.3.1 Close_Capture1

```
void CloseCapture1(void)
```

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.2 Close_Capture2

```
void CloseCapture2(void)
```

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.3 Close_Capture3

void CloseCapture3(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.4 Close_Capture4

void CloseCapture4(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.5 Close_Capture5

void CloseCapture5(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.6 Close_Capture6

```
void CloseCapture6(void)
```

This function turns off the Input Capture module

Returns
None

[Remarks](#)

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.7 Close_Capture7

```
void CloseCapture7(void)
```

This function turns off the Input Capture module

Returns
None

[Remarks](#)

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.8 Close_Capture8

```
void CloseCapture8(void)
```

This function turns off the Input Capture module

Returns
None

[Remarks](#)

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.9 Close_Capture9

```
void CloseCapture9(void)
```

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.10 Close_Capture10

void CloseCapture10(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.11 Close_ECapture1

void CloseECapture1(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.12 Close_ECapture2

void CloseECapture2(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.1.3.13 Close_ECapture3

void CloseECapture3(void)

This function turns off the Input Capture module

Returns

None

Remarks

This function disables the Input Capture interrupt and then turns off the module.

8.6.2 Input Capture Structs, Records, Enums

8.6.3 Input Capture Macros

8.6.4 Input Capture Examples

8.6.4.1 Input Capture Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*
* signal to be captured is fed to CCP1 pin.
*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "capture.h"           //header file for using library API for input capture
#include "timers.h"           //header file for using the library APIs for timers
that act as source for input capture

unsigned int  INCAPResult;

void main(void)
{
    unsigned char config1=0x00,timer_value = 0x00;

    //---Configure timer for corresponding timer module selection for capture module---
    SetTmrCCPSrc(T1_SOURCE CCP);           //Set Timer 1 as source for
input capture module

```

```

//---Configure input capture ---
    config1 = CAP_EVERY_RISE_EDGE | CAPTURE_INT_OFF ;           //configure input capture
for capture on every rising edge and its interrupt off
    OpenCapture1(config1 );
    OpenTimer1(0);                                              //start the timer

//---wait till input is captured---
    while(!PIR1bits.CCP1IF);                                   // Wait for event

//--- read result---
    INCAPResult = ReadCapture1();                               // read result

//---close capture module---
    CloseCapture1();

    while(1);                                                  //End of program
}

```

8.7 Output Compare (OCMP)

Output Compare module offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce pulse-width modulated waveforms for driving power applications. Key features of the output compare module include:

- Hardware-configurable for 16-bit operation in all modes.
- Synchronous and Trigger modes of output compare operation, with user-selectable trigger/sync sources available
- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation

8.7.1 Output Compare Functions

8.7.1.1 Open_Compare

8.7.1.1.1 Open_Compare1

```
void OpenCompare1(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.2 Open_Compare2

```
void OpenCompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.3 Open_Compare2_Page2

```
void OpenCompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_2_SEL_TMR12 * CCP_2_SEL_TMR34 * CCP_2_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.4 Open_Compare3

```
void OpenCompare3(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.5 Open_Compare3_Page2

void OpenCompare3(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_3_SEL_TMR12 * CCP_3_SEL_TMR34 * CCP_3_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.6 Open_Compare4

```
void OpenCompare4(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.7 Open_Compare4_Page2

void OpenCompare4(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR36 * CCP_4_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.8 Open_Compare4_Page3

void OpenCompare4(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.9 Open_Compare5

```
void OpenCompare5(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.10 Open_Compare5_Page2

```
void OpenCompare5(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR54 * CCP_5_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.11 Open_Compare5_Page3

```
void OpenCompare5(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR34 * CCP_5_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.12 Open_ECompare1

```
void OpenECompare1(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.13 Open_ECompare1_Page2

```
void OpenECompare1(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR36 * ECCP_1_SEL_TMR38 * ECCP_1_SEL_TMR310 * ECCP_1_SEL_TMR312 * ECCP_1_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.14 Open_ECompare1_Page3

```
void OpenECompare1(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR36 * ECCP_1_SEL_TMR38 * ECCP_1_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.15 Open_ECompare1_Page4

void OpenECompare1(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.16 Open_ECompare2_Page1

```
void OpenECompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR36 * ECCP_2_SEL_TMR38 * ECCP_2_SEL_TMR310 * ECCP_2_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.17 Open_ECompare2_Page2

```
void OpenECompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR36 * ECCP_2_SEL_TMR38 * ECCP_2_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.18 Open_ECompare3_Page1

```
void OpenECompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_3_SEL_TMR12 * ECCP_3_SEL_TMR34 * ECCP_3_SEL_TMR36 * ECCP_3_SEL_TMR38 * ECCP_3_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.19 Open_Compare6_Page1

void OpenCompare6(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_6_SEL_TMR12 * CCP_6_SEL_TMR52 * CCP_6_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.20 Open_Compare7_Page1

```
void OpenCompare7(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_7_SEL_TMR12 * CCP_7_SEL_TMR54 * CCP_7_SEL_TMR56 * CCP_7_SEL_TMR58 * CCP_7_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.21 Open_Compare8_Page1

```
void OpenCompare8(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR74 * CCP_8_SEL_TMR76 * CCP_8_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.22 Open_Compare8_Page2

```
void OpenCompare8(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR14 * CCP_8_SEL_TMR16 * CCP_8_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.23 Open_Compare9_Page1

void OpenCompare9(unsigned char config,unsigned int period)

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_9_SEL_TMR12 * CCP_9_SEL_TMR74 * CCP_9_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.24 Open_Compare10_Page1

```
void OpenCompare10(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_10_SEL_TMR12 * CCP_10_SEL_TMR72 * CCP_10_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns
None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

8.7.1.1.25 Open_ECompare1_Page5

```
void OpenECompare1(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable ECCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR56 * ECCP_1_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.26 Open_ECompare2_Page4

```
void OpenECompare2(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR56 * ECCP_2_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.27 Open_ECompare3_Page4

```
void OpenECompare3(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * ECOM_TOGG_MATCH * ECOM_LO_MATCH * ECOM_UNCHG_MATCH * ECOM_TRIG_SEVNT * ECOM_MODE_MASK Timer Selection * ECCP_3_SEL_TMR12 * ECCP_3_SEL_TMR34 * ECCP_3_SEL_TMR56 * ECCP_3_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.28 Open_Compare4_Page4

```
void OpenCompare4(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR56 * CCP_4_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.1.29 Open_Compare5_Page4

```
void OpenCompare5(unsigned char config,unsigned int period)
```

This routine configures the compare for interrupt, output signal and compare period.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Bit definitions to configure compare unit Enable CCP Interrupts: * COM_INT_ON * COM_INT_OFF * COM_INT_MASK Compare configuration * COM_TOGG_MATCH * COM_LO_MATCH * COM_UNCHG_MATCH * COM_TRIG_SEVNT * COM_MODE_MASK Timer Selection * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR34 * CCP_5_SEL_TMR56 * CCP_5_SEL_TMR_MASK
period	This is the 16bit integer data, to be compared with TMRx values, it can have any value from 1 to 65535.

Returns

None

Remarks

This function first resets the compare module to the POR state and then configures the output compare for the specified event, also the 16bit period data is assigned to the CCPRx registers.

NOTE: In addition to opening the compare, the appropriate timer module must be enabled before any of the compares will operate. See the data sheet for CCP and timer interconnect configurations.

8.7.1.2 Close_Compare

8.7.1.2.1 Close_Compare1

```
void CloseCompare1(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.2 Close_Compare2

```
void CloseCompare2(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.3 Close_Compare3

```
void CloseCompare3(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.4 Close_Compare4

```
void CloseCompare4(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.5 Close_Compare5

```
void CloseCompare5(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.6 Close_Compare6

void CloseCompare6(void)

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.7 Close_Compare7

void CloseCompare7(void)

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.8 Close_Compare8

void CloseCompare8(void)

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.9 Close_Compare9

```
void CloseCompare9(void)
```

This function turns off the Output Compare module.

Returns
None

[Remarks](#)

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.10 Close_Compare10

```
void CloseCompare10(void)
```

This function turns off the Output Compare module.

Returns
None

[Remarks](#)

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.11 Close_ECompare1

```
void CloseECompare1(void)
```

This function turns off the Output Compare module.

Returns
None

[Remarks](#)

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.12 Close_ECompare2

```
void CloseECompare2(void)
```

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.1.2.13 Close_ECompare3

void CloseECompare3(void)

This function turns off the Output Compare module.

Returns

None

Remarks

This function disables the Output Compare interrupt and then turns off the module.

8.7.2 Output Compare Macros

8.7.3 Output Compare Examples

8.7.3.1 Output Compare Example1

Example Source Code demonstrating peripheral library usage

```
/* *****  
 * NOTES:  
 * Code uses the Peripheral library support available with MCC18 Compiler  
 * Code Tested on:  
 * PicDem2+ demo board with PIC18F4685 controller  
 *  
 * output compare module output is observed at CCP1 pin  
 ***** */  
  
#define USE_OR_MASKS  
#include <p18cxxx.h>  
#include "compare.h"           //header file for using library API for output compare  
#include "timers.h"           //header file for using the library APIs for timers  
that act as source for output compare  
  
void main(void)  
{  
    unsigned int period=0x00;  
    unsigned char config=0x00,config1=0x00;
```

```
//---Configure timer for corresponding timer module selection for capture module---
SetTmrCCPSrc(T1_SOURCE_CCP);
OpenTimer1(config1);

//---Configure output compare ---
config = COM_TOGG_MATCH | COM_INT_OFF ;
period = 0x0101;
OpenCompare1( config, period);

/**/ On match of period the CCP1 pin toggles **/
while(1); //End of program

}
```

8.8 Pulse Width Modulation (PWM)

Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. PWM has a time base (period) register and associated Timer2 against which the value in time period register is constantly compared and upon match, the event can be triggered which is user selectable.

8.8.1 PWM Functions

8.8.1.1 Open_PWM

8.8.1.1.1 Open_PWM1

void OpenPWM1(char period)

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123)

function before the PWM configuration

8.8.1.1.2 Open_PWM2

```
void OpenPWM2( char period )
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123) function before the PWM configuration

8.8.1.1.3 Open_PWM3

```
void OpenPWM3( char period )
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123)

function before the PWM configuration

8.8.1.1.4 Open_PWM3_Page2

```
void OpenPWM3 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRS register to select the source of timers for CCP module * CCP_3_SEL_TMR12 * CCP_3_SEL_TMR34 * CCP_3_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.5 Open_PWM4

```
void OpenPWM4( char period )
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123) function before the PWM configuration

8.8.1.1.6 Open_PWM4_Page2

```
void OpenPWM4 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR36 * CCP_4_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.7 Open_PWM4_Page3

```
void OpenPWM3 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler

timer_source	<p>These parameters configures CCPTMRS register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * CCP_3_SEL_TMR12 * CCP_3_SEL_TMR34 * CCP_3_SEL_TMR_MASK
--------------	--

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.8 Open_PWM2_Page2

```
void OpenPWM2 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	<p>Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula:</p> $\text{PWM period} = [(\text{period}) + 1] \times 4 \times T_{\text{osc}} \times \text{TMRx prescaler}$
timer_source	<p>These parameters configures CCPTMRS register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * CCP_2_SEL_TMR12 * CCP_2_SEL_TMR34 * CCP_2_SEL_TMR_MASK

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.9 Open_PWM5

```
void OpenPWM5( char period )
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123) function before the PWM configuration

8.8.1.1.10 Open_PWM5_Page2

```
void OpenPWM5 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR54 * CCP_5_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.11 Open_PWM5_Page3

```
void OpenPWM5 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRS register to select the source of timers for CCP module * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR34 * CCP_5_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.12 Open_PWM6_Page1

```
void OpenPWM6 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_6_SEL_TMR12 * CCP_6_SEL_TMR52 * CCP_6_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.13 Open_PWM7_Page1

```
void OpenPWM7 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: $\text{PWM period} = [(\text{period}) + 1] \times 4 \times T_{\text{osc}} \times \text{TMRx prescaler}$
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_7_SEL_TMR12 * CCP_7_SEL_TMR54 * CCP_7_SEL_TMR56 * CCP_7_SEL_TMR58 * CCP_7_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.14 Open_PWM8_Page1

```
void OpenPWM8 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR74 * CCP_8_SEL_TMR76 * CCP_8_SEL_TMR_MASK

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.15 Open_PWM8_Page2

```
void OpenPWM8 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_8_SEL_TMR12 * CCP_8_SEL_TMR14 * CCP_8_SEL_TMR16 * CCP_8_SEL_TMR_MASK

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.16 Open_PWM9_Page1

```
void OpenPWM9 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_9_SEL_TMR12 * CCP_9_SEL_TMR74 * CCP_9_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.17 Open_PWM10_Page1

```
void OpenPWM10 ( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_10_SEL_TMR12 * CCP_10_SEL_TMR72 * CCP_10_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timerx.

8.8.1.1.18 Open_EPWM1

void OpenEPWM1(char period)

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMR2 prescaler

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base. PWM uses only Timer2.

NOTE : In addition to configuring the PWM, Timer2 must also be configured with an OpenTimer2 (see page 1123) function before the PWM configuration

8.8.1.1.19 Open_EPWM1_Page2

void OpenEPWM1(unsigned char period, unsigned char timer_source);

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler

timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR36 * ECCP_1_SEL_TMR38 * ECCP_1_SEL_TMR310 * ECCP_1_SEL_TMR312 * ECCP_1_SEL_TMR_MASK
--------------	--

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.20 Open_EPWM1_Page3

```
void OpenEPWM1( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	<p>Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula:</p> $\text{PWM period} = [(\text{period}) + 1] \times 4 \times T_{\text{osc}} \times \text{TMRx prescaler}$
timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR36 * ECCP_1_SEL_TMR38 * ECCP_1_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.21 Open_EPWM2_Page1

```
void OpenEPWM2( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module <ul style="list-style-type: none"> * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR36 * ECCP_2_SEL_TMR38 * ECCP_2_SEL_TMR310 * ECCP_2_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.22 Open_EPWM2_Page2

```
void OpenEPWM2( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler

timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR36 * ECCP_2_SEL_TMR38 * ECCP_2_SEL_TMR_MASK
--------------	--

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.23 Open_EPWM3_Page1

```
void OpenEPWM3( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	<p>Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula:</p> $\text{PWM period} = [(period) + 1] \times 4 \times T_{osc} \times TMRx \text{ prescaler}$
timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_3_SEL_TMR12 * ECCP_3_SEL_TMR34 * ECCP_3_SEL_TMR36 * ECCP_3_SEL_TMR38 * ECCP_3_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.24 Open_EPWM1_Page4

```
void OpenEPWM1( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * ECCP_1_SEL_TMR12 * ECCP_1_SEL_TMR34 * ECCP_1_SEL_TMR56 * ECCP_1_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.25 Open_EPWM2_Page3

```
void OpenEPWM2( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler

timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_2_SEL_TMR12 * ECCP_2_SEL_TMR34 * ECCP_2_SEL_TMR56 * ECCP_2_SEL_TMR_MASK
--------------	--

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.26 Open_EPWM3_Page3

```
void OpenEPWM3( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	<p>Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula:</p> $\text{PWM period} = [(\text{period}) + 1] \times 4 \times T_{\text{osc}} \times \text{TMRx prescaler}$
timer_source	<p>These parameters configures CCPTMRSx register to select the source of timers for CCP module</p> <ul style="list-style-type: none"> * ECCP_3_SEL_TMR12 * ECCP_3_SEL_TMR34 * ECCP_3_SEL_TMR56 * ECCP_3_SEL_TMR_MASK

Returns
None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.27 Open_PWM4_Page4

```
void OpenPWM4( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_4_SEL_TMR12 * CCP_4_SEL_TMR34 * CCP_4_SEL_TMR56 * CCP_4_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.1.28 Open_PWM5_Page4

```
void OpenPWM5( unsigned char period, unsigned char timer_source );
```

Configure PWM channel

Input Parameters

Input Parameters	Description
period	Can be any value from 0x00 to 0xff. This value determines the PWM frequency by using the following formula: PWM period = [(period) + 1] x 4 x T _{osc} x TMRx prescaler
timer_source	These parameters configures CCPTMRSx register to select the source of timers for CCP module * CCP_5_SEL_TMR12 * CCP_5_SEL_TMR34 * CCP_5_SEL_TMR56 * CCP_5_SEL_TMR_MASK

Returns

None

Remarks

This function configures the specified PWM channel for period and for time base.

8.8.1.2 SetOutput_PWM

8.8.1.2.1 SetOutput_PWM1

```
void SetOutputPWM1(unsigned char outputconfig, unsigned char outputmode)
```

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: * SINGLE_OUT * FULL_OUT_FWD * HALF_OUT * FULL_OUT_REV * PWM_OP_MODE_MASK
outputmode	PWM Output mode selection: * PWM_MODE_1 * PWM_MODE_2 * PWM_MODE_3 * PWM_MODE_4 * PWM_MODE_MASK

Returns

None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.2.2 SetOutput_PWM2

```
void SetOutputPWM2(unsigned char outputconfig, unsigned char outputmode)
```

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: * SINGLE_OUT * FULL_OUT_FWD * HALF_OUT * FULL_OUT_REV * PWM_OP_MODE_MASK
outputmode	PWM Output mode selection: * PWM_MODE_1 * PWM_MODE_2 * PWM_MODE_3 * PWM_MODE_4 * PWM_MODE_MASK

Returns

None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.2.3 SetOutput_PWM3

```
void SetOutputPWM3(unsigned char outputconfig, unsigned char outputmode)
```

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: * SINGLE_OUT * FULL_OUT_FWD * HALF_OUT * FULL_OUT_REV * PWM_OP_MODE_MASK

outputmode	PWM Output mode selection: <ul style="list-style-type: none">* PWM_MODE_1* PWM_MODE_2* PWM_MODE_3* PWM_MODE_4* PWM_MODE_MASK
------------	--

Returns
None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.2.4 SetOutput_EPWM1

void SetOutputEPWM1(unsigned char outputconfig, unsigned char outputmode)
--

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: <ul style="list-style-type: none">* SINGLE_OUT* FULL_OUT_FWD* HALF_OUT* FULL_OUT_REV* PWM_OP_MODE_MASK
outputmode	PWM Output mode selection: <ul style="list-style-type: none">* PWM_MODE_1* PWM_MODE_2* PWM_MODE_3* PWM_MODE_4* PWM_MODE_MASK

Returns
None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.2.5 SetOutput_EPWM2_Page1

```
void SetOutputEPWM2(unsigned char outputconfig, unsigned char outputmode)
```

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: * SINGLE_OUT * FULL_OUT_FWD * HALF_OUT * FULL_OUT_REV * PWM_OP_MODE_MASK
outputmode	PWM Output mode selection: * PWM_MODE_1 * PWM_MODE_2 * PWM_MODE_3 * PWM_MODE_4 * PWM_MODE_MASK

Returns

None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.2.6 SetOutput_EPWM3_Page1

```
void SetOutputEPWM3(unsigned char outputconfig, unsigned char outputmode)
```

Sets the PWM output configuration bits for ECCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
outputconfig	PWM Output configuration Selection: * SINGLE_OUT * FULL_OUT_FWD * HALF_OUT * FULL_OUT_REV * PWM_OP_MODE_MASK
outputmode	PWM Output mode selection: * PWM_MODE_1 * PWM_MODE_2 * PWM_MODE_3 * PWM_MODE_4 * PWM_MODE_MASK

Returns
None

Remarks

This function is only applicable to those devices with Extended or Enhanced CCP (ECCP)

8.8.1.3 SetDC_PWM

8.8.1.3.1 SetDC_PWM1

```
void SetDCPWM1(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula: $\text{PWM} \times \text{Duty cycle} = (\text{DCx} < 9:0 >) \times T_{\text{osc}}$ where DCx<9:0> is the 10-bit value specified in the call to this function

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.2 SetDC_PWM2

```
void SetDCPWM2(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns

None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.3 SetDC_PWM3

```
void SetDCPWM3(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.4 SetDC_PWM4

void SetDCPWM4(unsigned int dutycycle)

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$$

8.8.1.3.5 SetDC_PWM5

```
void SetDCPWM5(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> <p>$\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$</p> <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$$

8.8.1.3.6 SetDC_PWM6_Page1

```
void SetDCPWM6(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times T_{\text{osc}}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(F_{\text{osc}}/F_{\text{pwm}}) / \log(2)$

8.8.1.3.7 SetDC_PWM7_Page1

```
void SetDCPWM7(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times T_{\text{osc}}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.8 SetDC_PWM8_Page1

void SetDCPWM8(unsigned int dutycycle)

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> <p>$\text{PWM x Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$</p> <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.9 SetDC_PWM9_Page1

void SetDCPWM9(unsigned int dutycycle)

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times T_{\text{osc}}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(F_{\text{osc}}/F_{\text{pwm}}) / \log(2)$

8.8.1.3.10 SetDC_PWM10_Page1

```
void SetDCPWM10(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times T_{\text{osc}}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$$

8.8.1.3.11 SetDC_EPWM1

void SetDCEPWM1(unsigned int dutycycle)

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> <p>$\text{PWM x Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$</p> <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$$

8.8.1.3.12 SetDC_EPWM2_Page1

void SetDCEPWM2(unsigned int dutycycle)

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

Resolution (bits) = $\log(\text{Fosc}/\text{Fpwm}) / \log(2)$

8.8.1.3.13 SetDC_EPWM3_Page1

```
void SetDCEPWM3(unsigned int dutycycle)
```

Write a new duty cycle value to the specified PWM channel duty cycle registers

Input Parameters

Input Parameters	Description
dutycycle	<p>The value of dutycycle can be any 10-bit number. Only the lower 10-bits of dutycycle are written into the duty cycle registers. The duty cycle, or more specifically the high time of the PWM waveform, can be calculated from the following formula:</p> $\text{PWM} \times \text{Duty cycle} = (\text{DCx}<9:0>) \times \text{Tosc}$ <p>where DCx<9:0> is the 10-bit value specified in the call to this function</p>

Returns
None

Remarks

This function writes the new value for dutycycle to the specified PWM channel duty cycle registers.

The maximum resolution of the PWM waveform can be calculated from the period using the following formula:

$$\text{Resolution (bits)} = \log(\text{Fosc}/\text{Fpwm}) / \log(2)$$

8.8.1.4 Close_PWM

8.8.1.4.1 Close_PWM1

```
void ClosePWM1(void)
```

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.2 Close_PWM2

```
void ClosePWM2(void)
```

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.3 Close_PWM3

```
void ClosePWM3(void)
```

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.4 Close_PWM4

```
void ClosePWM4(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.1.4.5 Close_PWM5

```
void ClosePWM5(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.1.4.6 Close_PWM6_Page1

```
void ClosePWM6(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.1.4.7 Close_PWM7_Page1

```
void ClosePWM7(void)
```

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.8 Close_PWM8_Page1**void ClosePWM8(void)**

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.9 Close_PWM9_Page1**void ClosePWM9(void)**

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.10 Close_PWM10_Page1**void ClosePWM10(void)**

Disable PWM channel

Returns

None

Remarks

This function disables the specified PWM channel

8.8.1.4.11 Close_EPWM1

```
void CloseEPWM1(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.1.4.12 Close_EPWM3_Page1

```
void CloseEPWM3(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.1.4.13 Close_EPWM2_Page1

```
void CloseEPWM2(void)
```

Disable PWM channel

Returns
None

[Remarks](#)

This function disables the specified PWM channel

8.8.2 PWM Structs,Records,Enums

8.8.3 PWM Macros

8.8.4 PWM Examples

8.8.4.1 PWM Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
* PWM output is obtained on CCP1 pin. duty cycle is gievn by
*
*Formula for Period and Duty cycle calculatio
*
*      PWM period =  [(period  ) + 1] x 4 x Tosc x TMR2 prescaler
*
*      PWM x Duty cycle = (DCx<9:0>) x Tosc
*
*      Resolution (bits) = log(Fosc/Fpwm) / log(2)
*****/

#define USE_OR_MASKS
#include <p18cxxx.h>
#include "pwm.h"

void main(void)
{
    char period=0x00;
    unsigned char outputconfig=0,outputmode=0,config=0;
    unsigned int duty_cycle=0;

    //----Configure pwm ----
    period = 0xFF;
    OpenPWM1( period);           //Configure PWM module and initialize PWM period

    //-----set duty cycle----
    duty_cycle = 0x0F00;
    SetDCPWM1(duty_cycle);       //set the duty cycle

    //----set pwm output----
    outputconfig = FULL_OUT_FWD ;
    outputmode = PWM_MODE_1;
    SetOutputPWM1( outputconfig, outputmode);    //output PWM in respective modes

    while(1);                    //observe output on CCP1 pin

    //-----close pwm-----
    ClosePWM1();
}

```

8.9 EEPROM (EEP)

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. Peripheral library supports the EEPROM operations that includes:

- Byte Read from the specified location in EEPROM

- Byte Write to specified location in EEPROM
- Function to check the status of the EEPROM

8.9.1 EEP Functions

8.9.1.1 Write_Byte_EEP

```
void Write_b_eep( unsigned int badd,unsigned char bdata )
```

This function Writes a single byte of data to Internal EEP at the user specified location

Input Parameters

Input Parameters	Description
badd	Address of EEPROM location at which the byte of data has to be stored
bdata	byte of data to be stored in EEPROM

Returns

None

Remarks

User has to refer data sheet to get the address range and availability of internal EEPROM on specific device.

8.9.1.2 Read_Byte_EEP

```
unsigned char Read_b_eep( unsigned int badd )
```

Reads a single byte from Internal EEPROM from the user specified location

Input Parameters

Input Parameters	Description
badd	Address of EEPROM location at which the byte of data has to be stored

Returns

Returns byte of data stored at the specified address

Remarks

None

8.9.1.3 Busy__EEP

void Busy_eeep (void)

Checks & waits the status of ER bit in EECON1 register

Returns

None

Remarks

If the EEPROM is busy, then the function will be in continuous loop till the EEPROM is free.

8.9.2 EEP Examples

8.9.2.1 EEP Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*****/

#define USE_OR_MASKS
#include <p18cxxx.h>
#include "EEP.h"

unsigned char EEPWrite[15] = "MICROCHIP_TECH", EEPRead[15], Error=0 ;

void main(void)
{
    unsigned char q=0;
    unsigned int address;

    address = 0x0200;

    /* Write single byte to Internal EEP*/
    for(q=0;q<16;q++)
    {
        Write_b_eeep (address, EEPWrite[q]);    // write into to EEPROM
        address++;                               //increment the address of EEPROM to next
location
    /* Checks & waits the status of ER bit in EECON1 register */
    Busy_eeep ();
    }

    address = 0x0200;        // initialize the starting address
    Error = 0;               //clear the error flag
    /* Read single byte from Internal EEP*/
    for(q=0;q<16;q++)
    {
        EEPRead[q] = Read_b_eeep (address++);    //read the EEPROM data written previously
    }
}

```

```
from corresponding address
    if ( EEPRead[q] != EEPWrite[q] )           //check if the data read abck is same as
that was written
    {
        Error=1;                             //if the data read/ write match does not
    occur, then flag the error status
        while(1);                             //error occured
    }
    while(1);                                 //End of program
}
```

8.10 Inter Integrated Circuit Communication (I2C)

The Inter-Integrated Circuit (I2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers,

A/D Converters, etc. The I2C module supports these features:

- 7-bit and 10-bit device addresses
- General call address, as defined in the I2C protocol
- Clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications.
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

8.10.1 I2C Functions

8.10.1.1 I2C_Open

8.10.1.1.1 Open_I2C

```
void OpenI2C( unsigned char sync_mode, unsigned char slew )
```

Configures the I2C in SSP module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameter to configure the SSPCONx register: Mode Select * SLAVE_7 * SLAVE_10 * MASTER
slew	This contains the parameter to configure the SSPCONx register: Slew Rate Control * SLEW_OFF * SLEW_ON

Returns
None

Remarks

OpenI2C resets the SSP module to the POR state and then configures the module for Master/Slave mode and the selected slew rate.

8.10.1.1.2 Open_I2C1

```
void OpenI2C1( unsigned char sync_mode, unsigned char slew )
```

Configures the I2C in SSP1 module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameter to configure the SSPCONx register: Mode Select * SLAVE_7 * SLAVE_10 * MASTER
slew	This contains the parameter to configure the SSPCONx register: Slew Rate Control * SLEW_OFF * SLEW_ON

Returns

None

Remarks

OpenI2C1 resets the SSP1 module to the POR state and then configures the module for Master/Slave mode and the selected slew rate.

8.10.1.1.3 Open_I2C2

```
void OpenI2C2( unsigned char sync_mode, unsigned char slew )
```

Configures the I2C in SSP2 module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameter to configure the SSPCONx register: Mode Select * SLAVE_7 * SLAVE_10 * MASTER
slew	This contains the parameter to configure the SSPCONx register: Slew Rate Control * SLEW_OFF * SLEW_ON

Returns

None

Remarks

OpenI2C2 resets the SSP2 module to the POR state and then configures the module for Master/Slave mode and the selected slew rate.

8.10.1.2 I2C_Write

8.10.1.2.1 Write_I2C

```
unsigned char WriteI2C( unsigned char data_out )
```

This function is used to write out a single data byte to the I2C device.

Input Parameters

Input Parameters	Description
data_out	A single data byte to be written to the I2C bus device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This function writes a single byte to the I2C bus.

This function performs the same function as putcI2Cx.

8.10.1.2.2 Write_I2C1

```
unsigned char WriteI2C1( unsigned char data_out )
```

This function is used to write out a single data byte to the I2C device.

Input Parameters

Input Parameters	Description
data_out	A single data byte to be written to the I2C bus device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This function writes a single byte to the I2C bus.

This function performs the same function as putcI2Cx.

8.10.1.2.3 Write_I2C2

```
unsigned char WriteI2C2( unsigned char data_out )
```

This function is used to write out a single data byte to the I2C device.

Input Parameters

Input Parameters	Description
data_out	A single data byte to be written to the I2C bus device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This function writes a single byte to the I2C bus.

This function performs the same function as putcI2Cx.

8.10.1.3 I2C_Read

8.10.1.3.1 Read_I2C

```
unsigned char ReadI2C( void )
```

This function is used to read a single byte from I2C bus

Returns

The return value is the data byte read from the I2C bus.

Remarks

This function reads in a single byte from the I2C bus.

This function performs the same function as getcI2Cx.

8.10.1.3.2 Read_I2C1

```
unsigned char ReadI2C1( void )
```

This function is used to read a single byte from I2C bus

Returns

The return value is the data byte read from the I2C bus.

Remarks

This function reads in a single byte from the I2C bus.

This function performs the same function as `getcI2Cx`.

8.10.1.3.3 Read_I2C2

```
unsigned char ReadI2C2( void )
```

This function is used to read a single byte from I2C bus

Returns

The return value is the data byte read from the I2C bus.

Remarks

This function reads in a single byte from the I2C bus.

This function performs the same function as `getcI2Cx`.

8.10.1.4 I2C_puts

8.10.1.4.1 puts_I2C

```
unsigned char putsI2C( unsigned char *wrptr )
```

This function is used to write out a data string to the I2C bus.

Input Parameters

Input Parameters	Description
wrptr	Character type pointer to data objects in RAM. The data objects are written to the I2C device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This routine writes a data string to the I2C bus until a null character is reached. The null character itself is not transmitted.

This routine can operate in both Master or Slave mode.

8.10.1.4.2 puts_I2C1

```
unsigned char putsI2C1( unsigned char *wrptr )
```

This function is used to write out a data string to the I2C bus.

Input Parameters

Input Parameters	Description
wrptr	Character type pointer to data objects in RAM. The data objects are written to the I2C device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This routine writes a data string to the I2C bus until a null character is reached. The null character itself is not transmitted. This routine can operate in both Master or Slave mode.

8.10.1.4.3 puts_I2C2

```
unsigned char putsI2C2( unsigned char *wrptr )
```

This function is used to write out a data string to the I2C bus.

Input Parameters

Input Parameters	Description
wrptr	Character type pointer to data objects in RAM. The data objects are written to the I2C device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)
- * -2 -- if negative acknowledgement received from other I2C device

Remarks

This routine writes a data string to the I2C bus until a null character is reached. The null character itself is not transmitted. This routine can operate in both Master or Slave mode.

8.10.1.5 I2C_gets

8.10.1.5.1 gets_I2C

```
unsigned char getsI2C( unsigned char *rdptr, unsigned char length )
```

This function reads predetermined data string length from the I2C bus.

Input Parameters

Input Parameters	Description
rdptr	Character type pointer to RAM for storage of data read from I2C device
length	Number of bytes to read from I2C device.

Returns

Returns error condition status:

* 0 -- if the write was successful

* -1 -- if there was a write collision (for Master Mode only)

Remarks

This routine reads a predefined data string from the I2C bus.

NOTE:

User has to send a ACK or NACK signal after this function in application to complete the acknowledgement sequence for the last byte of data received.

8.10.1.5.2 gets_I2C1

```
unsigned char getsI2C1( unsigned char *rdptr, unsigned char length )
```

This function reads predetermined data string length from the I2C bus.

Input Parameters

Input Parameters	Description
rdptr	Character type pointer to RAM for storage of data read from I2C device
length	Number of bytes to read from I2C device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)

Remarks

This routine reads a predefined data string from the I2C bus.

NOTE:

User has to send a ACK or NACK signal after this function in application to complete the acknowledgement sequence for the last byte of data received.

8.10.1.5.3 gets_I2C2

```
unsigned char getsI2C2( unsigned char *rdptr, unsigned char length )
```

This function reads predetermined data string length from the I2C bus.

Input Parameters

Input Parameters	Description
rdptr	Character type pointer to RAM for storage of data read from I2C device
length	Number of bytes to read from I2C device.

Returns

Returns error condition status:

- * 0 -- if the write was successful
- * -1 -- if there was a write collision (for Master Mode only)

Remarks

This routine reads a predefined data string from the I2C bus.

NOTE:

User has to send a ACK or NACK signal after this function in application to complete the acknowledgement sequence for the last byte of data received.

8.10.1.6 Close_I2C

```
void CloseI2C( void )
```

This function turns off the I2C module

Returns

None

Remarks

None

8.10.1.7 Idle_I2C

void IdleI2C(void)

This function generates Wait condition until I2C bus is Idle

Returns

None

Remarks

This function will be in a wait state until Start Condition Enable bit, Stop Condition Enable bit, Receive Enable bit, Acknowledge Sequence Enable bit of I2C Control register and Transmit Status bit I2C Status register are clear. The IdleI2C function is required since the hardware I2C peripheral does not allow for spooling of bus sequence. The I2C peripheral must be in Idle state before an I2C operation can be initiated or write collision will be generated

8.10.2 I2C Macros

8.10.3 I2C Examples

8.10.3.1 I2C Example1_Master

Example Source Code demonstrating peripheral library usage

```
/* *****  
 * NOTES:  
 * Code uses the Peripheral library support available with MCC18 Compiler  
 * Code Tested on:  
 * PicDem2+ demo board with PIC18F4685 controller  
 *  
 * SCA & SCL must be connected to the corresponding in slave  
 *Note: If not using PicDem2+ demo board, then SCA & SCL lines have to be pulled up by  
connecting 4.7KOhms resister to Vdd.  
 ***** */  
  
#define USE_OR_MASKS  
#include <p18cxxx.h>  
#include "i2c.h"  
  
unsigned char I2C_Send[ 21 ] = "MICROCHIP:I2C_MASTER" ;
```

```

unsigned char I2C_Recv[21];

***** I2C MASTER *****
void main(void)
{

unsigned char sync_mode=0, slew=0, addl,w,data,status,length;

    for(w=0;w<20;w++)
        I2C_Recv[w]=0;

    addl=0xA2;           //address of the device (slave) under communication

    CloseI2C();          //close i2c if was operating earlier

    ---INITIALISE THE I2C MODULE FOR MASTER MODE WITH 100KHz ---
    sync_mode = MASTER;
    slew = SLEW_OFF;

    OpenI2C(sync_mode,slew);

    SSPADD=0x0A;          //400kHz Baud clock(9) @8MHz

    //check for bus idle condition in multi master communication
    IdleI2C();

    ---START I2C---
    StartI2C();

    ****write the address of the device for communication****
    data = SSPBUF;         //read any previous stored content in buffer to clear buffer full
    do
    {
        status = WriteI2C( addl | 0x00 );    //write the address of slave
        if(status == -1)          //check if bus collision happened
        {
            data = SSPBUF;         //upon bus collision detection clear the buffer,
            SSPCON1bits.WCOL=0;    // clear the bus collision status bit
        }
    }
    while(status!=0);           //write untill successful communication
    //R/W BIT IS '0' FOR FURTHER WRITE TO SLAVE

    ***WRITE THE THE DATA TO BE SENT FOR SLAVE***
    while(putsI2C(I2C_Send)!=0);    //write string of data to be transmitted to slave

    ---TERMINATE COMMUNICATION FROM MASTER SIDE---
    IdleI2C();

    ---RESTART I2C COMMUNICATION---
    RestartI2C();
    IdleI2C();

    ****write the address of the device for communication****
    data = SSPBUF;         //read any previous stored content in buffer to clear buffer
    full status

    //R/W BIT IS '1' FOR READ FROM SLAVE
    addl = 0xA2;
    do
    {
        status = WriteI2C( addl | 0x01 );    //write the address of slave
        if(status == -1)          //check if bus collision happened
        {
            data = SSPBUF;         //upon bus collision detection clear the buffer,
            SSPCON1bits.WCOL=0;    // clear the bus collision status bit
        }
    }
    while(status!=0);           //write untill successful communication

```

```

/** Recieve data from slave */
while( getsI2C(I2C_Recv,20) );           //recieve data string of lenght 20 from slave
I2C_Recv[20] = '\0' ;

    NotAckI2C();                         //send the end of transmission signal through nack
    while( SSPCON2bits.ACKEN!=0);        //wait till ack sequence is complete

/** close I2C */
CloseI2C();                             //close I2C module

while(1);                               //End of program
}

```

8.10.3.2 I2C Example1_Slave

Example Source Code demonstrating peripheral library usage

```

*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*
* signal to be compared are applied to CxIN+ & CxIN- and output is tapped at CxOUT
*****

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "i2c.h"

unsigned char I2C_Send[21] = "MICROCHIP:I2C_SLAVE" ;
unsigned char I2C_Recv[21];

***** I2C SLAVE *****
void main(void)
{

    unsigned char sync_mode=0, slew=0, addl,status,temp,w,length=0;

    for(w=0;w<20;w++)
        I2C_Recv[w]=0;

        CloseI2C();                     //close i2c if was operating earlier

    /--INITIALISE THE I2C MODULE FOR MASTER MODE WITH 100KHz --
        sync_mode = SLAVE_7;
        slew = SLEW_OFF;
        OpenI2C(sync_mode,slew);

        SSPADD = 0xA2;                  //initialize slave address
    **** Read the address sent by master from buffer ****
        while(DataRdyI2C()==0);        //WAIT UNTILL THE DATA IS TRANSMITTED FROM master
        temp = ReadI2C();

    **** Data reception from master by slave ****

        do
        {
            while(DataRdyI2C()==0);    //WAIT UNTILL THE DATA IS TRANSMITTED FROM master
            I2C_Recv[length++]=getcI2C(); // save byte received
        }
        while(length!=20);

    **** write sequence from slave ****
        while(SSPSTATbits.S!=1);       //wait untill STOP CONDITION

```

```

    /** Read the address sent by master from buffer */
    while(DataRdyI2C()==0);           //WAIT UNTILL THE DATA IS TRANSMITTED FROM master
    temp = ReadI2C();

    /** Slave transmission */
    if(SSPSTAT & 0x04)                 //check if master is ready for reception
    while(putsI2C(I2C_Send));          // send the data to master

    /**--TERMINATE COMMUNICATION FROM MASTER SIDE--
    CloseI2C();                        //close I2C module

    while(1);                          //End of program
}
```

8.11 I2C EEPROM (I2C_EEP)

The Inter-Integrated Circuit (I2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. This section provides peripheral library support to serial I2C compatible EEPROMs

8.11.1 I2C_EEP Functions

8.11.1.1 EEAckPolling_I2C

8.11.1.1.1 I2C_EEAckPolling

unsigned char EEAckPolling(unsigned char control)

Acknowledge polling of I2C EE memory device.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns
Returns error condition status: * 0 -- if the control/address write was successful * -1 -- if there was a bus collision * -3 -- if there was a bus collision

Remarks

This function is used to generate the Acknowledge polling sequence for EE I2C memory devices that utilize Acknowledge polling.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.1.2 I2C_EEAckPolling1

unsigned char EEAckPolling1(unsigned char control)

Acknowledge polling of I2C EE memory device.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -3 -- if there was a bus collision

Remarks

This function is used to generate the Acknowledge polling sequence for EE I2C memory devices that utilize Acknowledge polling.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.1.3 I2C_EEAckPolling2

unsigned char EEAckPolling2(unsigned char control)

Acknowledge polling of I2C EE memory device.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -3 -- if there was a bus collision

Remarks

This function is used to generate the Acknowledge polling sequence for EE I2C memory devices that utilize Acknowledge polling.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.2 EECurrentAddRead_I2C

8.11.1.2.1 I2C_EECurrentAddRead

unsigned int EECurrentAddRead(unsigned char control)

This function Reads data at current address of EE memory.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- The read byte of **data** is returned as an unsigned 16-bit quantity. Since the buffer itself is only 8-bits wide, this means that the Most Significant Byte will be zero and the Least Significant Byte will contain the read buffer contents.

Remarks

This function reads a byte of data from the current address loaded previously to the I2C EE memory device. The address location of the data to read is that of the current pointer within the I2C EE device. The memory device contains an address counter that maintains the address of the last word accessed, incremented by one.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of

document prior to using the I2C EEPROM functions

- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.2.2 I2C_EECurrentAddRead1

unsigned int EECurrentAddRead1(unsigned char control)

This function Reads data at current address of EE memory.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- The read byte of **data** is returned as an unsigned 16-bit quantity. Since the buffer itself is only 8-bits wide, this means that the Most Significant Byte will be zero and the Least Significant Byte will contain the read buffer contents.

Remarks

This function reads a byte of data from the current address loaded previously to the I2C EE memory device. The address location of the data to read is that of the current pointer within the I2C EE device. The memory device contains an address counter that maintains the address of the last word accessed, incremented by one.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.2.3 I2C_EECurrentAddRead2

unsigned int EECurrentAddRead2(unsigned char control)

This function Reads data at current address of EE memory.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- The read byte of **data** is returned as an unsigned 16-bit quantity. Since the buffer itself is only 8-bits wide, this means that the Most Significant Byte will be zero and the Least Significant Byte will contain the read buffer contents.

Remarks

This function reads a byte of data from the current address loaded previously to the I2C EE memory device. The address location of the data to read is that of the current pointer within the I2C EE device. The memory device contains an address counter that maintains the address of the last word accessed, incremented by one.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.3 EEByteWrite_I2C

8.11.1.3.1 I2C_EEByteWrite

unsigned char EEByteWrite(unsigned char control, unsigned char address, unsigned char data)

This function Write a byte of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
data	Data to write to EEPROM address specified in function parameter address.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function writes a byte of data to the I2C EE memory device. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.3.2 I2C_EEByteWrite1

unsigned char EEByteWrite1(unsigned char control, unsigned char address, unsigned char data)

This function Write a byte of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
data	Data to write to EEPROM address specified in function parameter address.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function writes a byte of data to the I2C EE memory device. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.3.3 I2C_EEByteWrite2

unsigned char EEByteWrite2(unsigned char control, unsigned char address, unsigned char data)

This function Write a byte of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
data	Data to write to EEPROM address specified in function parameter address.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function writes a byte of data to the I2C EE memory device. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.4 EEPPageWrite_I2C

8.11.1.4.1 I2C_EEPPageWrite

```
unsigned char EEPPageWrite( unsigned char control, unsigned char address, unsigned char *wrptr )
```

This function Write a string of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
wrptr	Character type pointer to PICmicro MCU RAM area from where the data to be written to EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- * -4 -- if there was a bus device responded possible error

Remarks

This function writes a null terminated string of data to the I2C EE memory device. The null character itself is not transmitted. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.4.2 I2C_EEPPageWrite1

unsigned char EEPPageWrite1(unsigned char control, unsigned char address, unsigned char *wrptr)

This function Write a string of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
wrptr	Character type pointer to PICmicro MCU RAM area from where the data to be written to EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- * -4 -- if there was a bus device responded possible error

Remarks

This function writes a null terminated string of data to the I2C EE memory device. The null character itself is not transmitted. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of

document prior to using the I2C EEPROM functions

- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.4.3 I2C_EEPageWrite2

unsigned char EEPageWrite2(unsigned char control, unsigned char address, unsigned char *wrptr)

This function Write a string of data to the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
wrptr	Character type pointer to PICmicro MCU RAM area from where the data to be written to EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision
- * -4 -- if there was a bus device responded possible error

Remarks

This function writes a null terminated string of data to the I2C EE memory device. The null character itself is not transmitted. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.5 EESequentialRead_I2C

8.11.1.5.1 I2C_EESequentialRead

unsigned char EESequentialRead(unsigned char control, unsigned char address, unsigned char *rdptr, unsigned char length)

This function reads a string of data from the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
rdptr	Character type pointer to PICmicro MCU RAM area for placement of data read from EEPROM device.
length	Number of bytes to read from EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a predefined string length of data from the I2C bus. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.5.2 I2C_EESequentialRead1

unsigned char EESequentialRead1(unsigned char control, unsigned char address, unsigned char *rdptr, unsigned char length)

This function reads a string of data from the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
rdptr	Character type pointer to PICmicro MCU RAM area for placement of data read from EEPROM device.
length	Number of bytes to read from EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a predefined string length of data from the I2C bus. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.5.3 I2C_EESequentialRead2

unsigned char EESequentialRead2(unsigned char control, unsigned char address, unsigned char *rdptr, unsigned char length)

This function reads a string of data from the I2C EEPROM from the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location
rdptr	Character type pointer to PICmicro MCU RAM area for placement of data read from EEPROM device.
length	Number of bytes to read from EEPROM device.

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a predefined string length of data from the I2C bus. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of

document prior to using the I2C EEPROM functions

- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.6 EERandomRead_I2C

8.11.1.6.1 I2C_EERandomRead

```
unsigned int EERandomRead( unsigned char control, unsigned char address )
```

This function reads a single byte of data from the I2C EEPROM at the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a single byte from the I2C EEPROM. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C (see page 1063) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.6.2 I2C_EERandomRead1

```
unsigned int EERandomRead1( unsigned char control, unsigned char address )
```

This function reads a single byte of data from the I2C EEPROM at the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a single byte from the I2C EEPROM. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C1 (see page 1064) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.1.6.3 I2C_EERandomRead2

unsigned int EERandomRead2(unsigned char control, unsigned char address)

This function reads a single byte of data from the I2C EEPROM at the specified address.

Input Parameters

Input Parameters	Description
control	EEPROM control / bus device select address byte
address	EEPROM internal address location

Returns

Returns error condition status:

- * 0 -- if the control/address write was successful
- * -1 -- if there was a bus collision
- * -2 -- if negative acknowledgement received from other I2C device
- * -3 -- if there was a bus collision

Remarks

This function reads in a single byte from the I2C EEPROM. The routine can be used for Microchip I2C EE memory devices which only require 1 byte of address information.

NOTE:

- I2C module on device must be configured using the OpenI2C2 (see page 1065) function available in I2C section of document prior to using the I2C EEPROM functions
- I2C on device must be configured in MASTER mode only
- Refer I2C compatible EEPROM data sheet for "control" parameter details and "address" range.

8.11.2 I2C_EEP Examples

8.11.2.1 I2C_EEP Example1

Example Source Code demonstrating peripheral library usage

//The following is a simple code example illustrating the SSP module configured for I2C master communication. The routine illustrates I2C communications with a Microchip 24LC01B I2C EE memory device.

```
#include "p18cxx.h"
#include "i2c.h"

unsigned char arraywr[] = {1,2,3,4,5,6,7,8,0};
unsigned char arrayrd[20];

//*****
void main(void)
{
    OpenI2C(MASTER, SLEW_ON); // Initialize I2C module
    SSPADD = 9;               //400kHz Baud clock(9) @16MHz
                             //100kHz Baud clock(39) @16MHz

    while(1)
    {
        EEByteWrite(0xA0, 0x30, 0xA5);

        EEAckPolling(0xA0);

        EECurrentAddRead(0xA0);

        EEPAGEWrite(0xA0, 0x70, arraywr);

        EEAckPolling(0xA0);

        EESequentialRead(0xA0, 0x70, arrayrd, 20);

        EERandomRead(0xA0, 0x30);
    }
}
```

8.12 IO Ports

All of the device pins (except VDD, VSS, MCLR and OSC1/CLKI) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

8.12.1 Ports Functions

8.12.1.1 Open_PORTB

void OpenPORTB(unsigned char config)

Configure the interrupts and internal pull-up resistors on PORTB.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Parameters to configure Interrupt on chnage and Pull Ups Interrupt-on-change: * PORTB_CHANGE_INT_ON * PORTB_CHANGE_INT_OFF * PORTB_CHANGE_INT_MASK Enable/Disable Pullups: * PORTB_PULLUPS_ON * PORTB_PULLUPS_OFF * PORTB_PULLUPS_MASK

Returns
None

Remarks

This function configures the interrupts and internal pull-up resistors on PORTB.

8.12.1.2 Open_RB0INT

void OpenRB0INT(unsigned char config)

Enable interrupts for the INT0 pin.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Parameters to configure Interrupt on change and Pull Ups</p> <p>Interrupt-on-change:</p> <ul style="list-style-type: none">* PORTB_CHANGE_INT_ON* PORTB_CHANGE_INT_OFF* PORTB_CHANGE_INT_MASK <p>Interrupt-on-edge:</p> <ul style="list-style-type: none">* RISING_EDGE_INT* FALLING_EDGE_INT* EDGE_INT_MASK <p>Interrupt Priority Select:</p> <ul style="list-style-type: none">* PORTB_INT_PRIO_HIGH* PORTB_INT_PRIO_LOW* PORTB_INT_PRIO_MASK <p>Enable/Disable Pullups:</p> <ul style="list-style-type: none">* PORTB_PULLUPS_ON* PORTB_PULLUPS_OFF* PORTB_PULLUPS_MASK

Returns

None

Remarks

This function configures the interrupts and internal pull-up resistors on INT0.

8.12.1.3 Open_RB1INT

void OpenRB1INT(unsigned char config)

Enable interrupts for the INT1 pin.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Parameters to configure Interrupt on chnage and Pull Ups</div> <div>Interrupt-on-change:<ul style="list-style-type: none">* PORTB_CHANGE_INT_ON* PORTB_CHANGE_INT_OFF* PORTB_CHANGE_INT_MASK</div> <div>Interrupt-on-edge:<ul style="list-style-type: none">* RISING_EDGE_INT* FALLING_EDGE_INT* EDGE_INT_MASK</div> <div>Interrupt Priority Select:<ul style="list-style-type: none">* PORTB_INT_PRIO_HIGH* PORTB_INT_PRIO_LOW* PORTB_INT_PRIO_MASK</div> <div>Enable/Disable Pullups:<ul style="list-style-type: none">* PORTB_PULLUPS_ON* PORTB_PULLUPS_OFF* PORTB_PULLUPS_MASK</div>

Returns
None

Remarks

This function configures the interrupts and internal pull-up resistors on INT1.

8.12.1.4 Open_RB2INT

void OpenRB2INT(unsigned char config)

Enable interrupts for the INT2 pin.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Parameters to configure Interrupt on change and Pull Ups</p> <p>Interrupt-on-change:</p> <ul style="list-style-type: none">* PORTB_CHANGE_INT_ON* PORTB_CHANGE_INT_OFF* PORTB_CHANGE_INT_MASK <p>Interrupt-on-edge:</p> <ul style="list-style-type: none">* RISING_EDGE_INT* FALLING_EDGE_INT* EDGE_INT_MASK <p>Interrupt Priority Select:</p> <ul style="list-style-type: none">* PORTB_INT_PRIO_HIGH* PORTB_INT_PRIO_LOW* PORTB_INT_PRIO_MASK <p>Enable/Disable Pullups:</p> <ul style="list-style-type: none">* PORTB_PULLUPS_ON* PORTB_PULLUPS_OFF* PORTB_PULLUPS_MASK

Returns

None

Remarks

This function configures the interrupts and internal pull-up resistors on INT2.

8.12.1.5 Open_RB3INT

void OpenRB3INT(unsigned char config)

Enable interrupts for the INT3 pin.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Parameters to configure Interrupt on chnage and Pull Ups</div> <div>Interrupt-on-change: * PORTB_CHANGE_INT_ON * PORTB_CHANGE_INT_OFF * PORTB_CHANGE_INT_MASK</div> <div>Interrupt-on-edge: * RISING_EDGE_INT * FALLING_EDGE_INT * EDGE_INT_MASK</div> <div>Interrupt Priority Select: * PORTB_INT_PRIO_HIGH * PORTB_INT_PRIO_LOW * PORTB_INT_PRIO_MASK</div> <div>Enable/Disable Pullups: * PORTB_PULLUPS_ON * PORTB_PULLUPS_OFF * PORTB_PULLUPS_MASK</div>

Returns
None

Remarks

This function configures the interrupts and internal pull-up resistors on INT3.

8.12.2 Ports Macros

8.12.3 Ports Examples

8.12.3.1 Ports Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* ADDITIONAL NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*
*****/
```

```

* Uses INT0/INT1/INT2/RBx pins to cause interrupts
*****

#define USE_OR_MASKS
#include <pl18cxxx.h>
#include "portb.h"

unsigned char PORTResult[5]={0,0,0,0,0};

void main(void)
{
    unsigned char config=0;
    ADCON1 = 0xFF;

    **** configure INT0 with pullups enabled, falling edge ***
    config = PORTB_CHANGE_INT_ON | FALLING_EDGE_INT | PORTB_PULLUPS_ON;
    OpenRB0INT(config); //configures INT0 & enables it

    **** configure INT1 with pullups enabled, falling edge ***
    config=0;
    config = PORTB_CHANGE_INT_ON | FALLING_EDGE_INT | PORTB_PULLUPS_ON;
    OpenRB1INT( config); //configures and enables INT1

    **** configure INT2 with pullups enabled, falling edge ***
    config=0;
    config = PORTB_CHANGE_INT_ON | FALLING_EDGE_INT | PORTB_PULLUPS_ON;
    OpenRB2INT( config); //configures and enables INT2

    **** configure Change Notification in PORTB with pullups enabled, falling edge ***
    config=0;
    config = PORTB_CHANGE_INT_ON | PORTB_PULLUPS_ON;
    OpenPORTB(config); //configures and enables change notification
    in PORTB

    **** Check for interrupts ***
    while(1)
    {
        if( INTCONbits.INT0IF == 1) //Check for INT0
        {
            PORTResult[0]=1; //Ser flag to indicate occurrence of INT0
            INTCONbits.INT0IF = 0; //Clear INT0
        }

        if(INTCON3bits.INT1IF == 1) //Check for INT1
        {
            PORTResult[1]=1; //Ser flag to indicate occurrence of INT1
            INTCON3bits.INT1IF = 0; //Clear INT1
        }

        if(INTCON3bits.INT2IF == 1) //Check for INT2
        {
            PORTResult[2]=1; //Ser flag to indicate occurrence of INT2
            INTCON3bits.INT2IF = 0; //Clear INT2
        }

        if(INTCONbits.RBIF == 1) //Check for Change Notification interrupt
        {
            PORTResult[4]=1; //Ser flag to indicate occurrence of Change
            INTCONbits.RBIF = 0; //Clear Change Notification interrupt
        }
    }

    CloseRB0INT(); //disable INT0
    CloseRB1INT(); //disable INT1
    CloseRB2INT(); //disable INT2
    ClosePORTB(); //disable Change notification

```

}

8.13 Real Time Clock & Calender (RTCC)

This module implements a full-featured clock and calendar with alarm functions in hardware, freeing up timer resources and program memory space for use of the core application.

8.13.1 RTCC Functions

8.13.1.1 RTCC_InitClock

void RtccInitClock(void)

The function initializes the RTCC device. It starts the RTCC clock, sets the RTCC Off and disables RTCC write. Disables the OE.

Returns
None

Remarks

Enables the secondary oscillator from Timer1

8.13.1.2 RTCC_ReadAlrmDate

void RtccReadAlrmDate(rtccDate* pDt)

The function updates the user supplied union/structure with the current alarm Date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccDate *pDt	pointer to a rtccDate union to store the alarm Date

Returns
Updates the rtccDate structure

Remarks

pDt a valid pointer

8.13.1.3 RTCC_ReadAlarmTime

```
void RtccReadAlarmTime(rtccTime* pTm)
```

The function updates the user supplied union/structure with the current alarm time of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTime* pTm	pointer to a rtccTime union to store the alarm time

Returns

Updates the rtccTime structure

Remarks

pTm a valid pointer

8.13.1.4 RTCC_ReadAlarmTimeDate

```
void RtccReadAlarmTimeDate(rtccTimeDate* pTD)
```

The function updates the user supplied union/structure with the current alarm time and date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTimeDate* pTD	pointer to a rtccTimeDate union to store the alarm time and date

Returns

Updates the rtccTimeDate structure

Remarks

pTD a valid pointer

8.13.1.5 RTCC_ReadDate

```
void RtccReadDate(rtccDate* pDt)
```

The function updates the user supplied union/structure with the current time of the RTCC device.

Input Parameters

Input Parameters	Description
rtccDate* pDt	pointer to a rtccTime union to store the current time

Returns

Updates the rtccDate structure

Remarks

The function makes sure that the read value is valid. It avoids waiting for the RTCSYNC to be clear by performing successive reads.

8.13.1.6 RTCC_ReadTime

```
void RtccReadTime(rtccTime* pTm)
```

The function updates the user supplied union/structure with the current time of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTime* pTm	pointer to a rtccTime union to store the current time

Returns

Updates the rtccTime structure

Remarks

The function makes sure that the read value is valid. It avoids waiting for the RTCSYNC to be clear by performing successive reads.

8.13.1.7 RTCC_ReadTimeDate

```
void RtccReadTimeDate(rtccTimeDate* pTD)
```

The function updates the user supplied union/structure with the current time and date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTimeDate* pTD	pointer to a rtccTimeDate union to store the current time and date

Returns

Updates the pTD structure

Remarks

This firmware solution would consist of reading each register twice and then comparing the two values. If the two values match, then a rollover did not occur.

8.13.1.8 RTCC_SetAlarmRpt

```
void RtccSetAlarmRpt(rtccRepeat rpt, BOOL dsblAlrm)
```

The function sets the RTCC alarm repeat count.

Input Parameters

Input Parameters	Description
rpt	value of the desired alarm repeat rate
dsblAlrm	if TRUE, the API can temporarily disable the alarm when changing the RPT value

Returns

None

Remarks

If alarm is enabled, changing the repeat count can be safely made only when the sync pulse is unasserted. To avoid waiting for the sync pulse, the user can choose to temporarily disable the alarm and then re-enable it. This means that the user has the knowledge that an alarm

event is not imminent.

8.13.1.9 RTCC_WrOn

```
void RtccWrOn(void)
```

Function to set the RTCCFG.RTCWREN

Returns

None

Remarks

The interrupts are disabled in order to have a proper device initialization

8.13.1.10 RTCC_WriteTimeDate

BOOL RtccWriteTimeDate(const rtccTimeDate* pTD , BOOL di)

The function updates the user supplied union/structure with the current time and date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTimeDate* pTD	pointer to a rtccTimeDate union to store the current time and date
di	if interrupts need to be disabled

Returns

TRUE '1' : If all the values are within range
FALSE '0' : If any value is out of above mentioned range.

Remarks

The write is successful only if Wr Enable is set. The function will enable the write itself, if needed.

Also, the Alarm will be temporarily disabled and the device will be stopped (On set to 0) in order to safely perform the update of the RTC time register. However, the device status will be restored.

Usually the disabling of the interrupts is desired, if the user has to have more precise control over the actual moment of the time setting.

PreCondition: rtccTimeDate structure fields have to have proper values:

*sec: BCD codification, 00-59

*min: BCD codification, 00-59

*hour: BCD codification, 00-24

*wday: BCD codification, 00-06

*mday: BCD codification, 01-31

*mon: BCD codification, 01-12

*year: BCD codification, 00-99

8.13.1.11 RTCC_WriteTime

BOOL RtccWriteTime(const rtccTime* pTm , BOOL di)

The function updates the user supplied union/structure with the current time and date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccTime* pTm	pointer to a rtccTime union to store the current time and date
di	if interrupts need to be disabled

Returns

TRUE '1' : If all the values are within range
 FALSE '0' : If any value is out of above mentioned range.

Remarks

The write is successful only if Wr Enable is set. The function will enable the write itself, if needed.

Also, the Alarm will be temporarily disabled and the device will be stopped (On set to 0) in order to safely perform the update of the RTC time register. However, the device status will be restored.

Usually the disabling of the interrupts is desired, if the user has to have more precise control over the actual moment of the time setting.

PreCondition: pTm pointing to a valid rtccTime structure having proper values:

sec: BCD codification, 00-59

min: BCD codification, 00-59

hour: BCD codification, 00-24

8.13.1.12 RTCC_WriteDate

BOOL RtccWriteDate(const rtccDate* pDt , BOOL di)

The function updates the user supplied union/structure with the current time and date of the RTCC device.

Input Parameters

Input Parameters	Description
rtccDate* pDt	pointer to a rtccDate union to store the current time and date
di	if interrupts need to be disabled

Returns

TRUE '1' : If all the values are within range
 FALSE '0' : If any value is out of above mentioned range.

Remarks

The write is successful only if Wr Enable is set. The function will enable the write itself, if needed.

Also, the Alarm will be temporarily disabled and the device will be stopped (On set to 0) in order

to safely perform the update of the RTC time register. However, the device status will be restored.

Usually the disabling of the interrupts is desired, if the user has to have more precise control over the actual moment of the time setting.

PreCondition: pDt is a valid rtccDate pointer having proper values:

wday: BCD codification, 00-06

mday: BCD codification, 01-31

mon: BCD codification, 01-12

year: BCD codification, 00-99

8.13.1.13 RTCC_WriteAlrmTimeDate

BOOL RtccWriteAlrmTimeDate(const rtccTimeDate* pTD)

The function sets the current alarm time and date in the RTCC device.

Input Parameters

Input Parameters	Description
rtccTimeDate* pTD	pointer to a rtccTimeDate union to store the current time and date

Returns

TRUE '1' : If all the values are within range

FALSE '0' : If any value is out of above mentioned range.

Remarks

Note that the alarm time does not contain a year field.

PreCondition: rtccTimeDate structure fields have to have proper values:

sec: BCD codification, 00-59

min: BCD codification, 00-59

hour: BCD codification, 00-24

wday: BCD codification, 00-06

mday: BCD codification, 01-31

mon: BCD codification, 01-12

8.13.1.14 RTCC_WriteAlrmTime

BOOL RtccWriteAlrmTime(const rtccTime* pTm)

The function sets the current time in the RTCC device.

Input Parameters

Input Parameters	Description
rtccTime* pTm	pointer to a rtccTime union to store the current time and date

Returns

TRUE '1' : If all the values are within range
FALSE '0' : If any value is out of above mentioned range.

Remarks

The write is successful only if Wr Enable is set. The function will enable the write itself, if needed.

Also, the Alarm will be temporarily disabled in order to safely perform the update of the ALRMTIME register. However, the device status will be restored.

PreCondition: pTm a valid rtccTime pointer having proper values:

sec: BCD codification, 00-59

min: BCD codification, 00-59

hour: BCD codification, 00-24

8.13.1.15 RTCC_WriteAlrmDate

BOOL RtccWriteAlrmDate(const rtccDate* pDt)

The function sets the alarm date in the RTCC device.

Input Parameters

Input Parameters	Description
rtccDate* pDt	pointer to a rtccDate union to store the current time and date

Returns

TRUE '1' : If all the values are within range
FALSE '0' : If any value is out of above mentioned range.

Remarks

The write is successful only if Wr Enable is set. The function will enable the write itself, if needed.

Also, the Alarm will be temporarily disabled in order to safely perform the update of the ALRMTIME register. However, the device status will be restored.

PreCondition: pDt a valid rtccDate pointer having proper values:

wday: BCD codification, 00-06

mday: BCD codification, 01-31

mon: BCD codification, 01-12

8.13.1.16 RTCC_SetChimeEnable

```
void RtccSetChimeEnable(BOOL enable, BOOL dsblAlrm)
```

The function enables/disables the chime alarm of the RTCC device.

Input Parameters

Input Parameters	Description
enable	boolean to enable/disable the RTCC chime
dsblAlrm	if TRUE, the API can temporarily disable the alarm when changing the Chime status

Returns

None

Remarks

If alarm is enabled, changing the chime status can be safely madewhen the sync pulse is unasserted. To avoid waiting for the sync pulse, the user can choose to temporarily disable the alarm and then

re-enable it. This means that the user has knowledge that an alarm event is not imminent.

8.13.1.17 RTCC_SetCalibration

```
void RtccSetCalibration(int drift)
```

The function updates the value that the RTCC uses in the auto-adjust feature, once every minute.

The drift value acts as a signed value, [-128*4, +127*4], 0 not having any effect.

Input Parameters

Input Parameters	Description
drift	value to be added/subtracted to perform calibration

Returns

None

Remarks

Writes to the RTCCAL[7:0] register should only occur when the timer is turned off or immediately or after the edge of the seconds pulse (except when SECONDS=00 - due to the possibility of the auto-adjust event). In order to speed-up the process, the API function performs the reading of the HALFSEC field. The function may block for half a second, worst case, when called at the start of the minute. Interrupts can not be disabled for such a long period. However, long interrupt routines can interfere with the proper functioning of the device. Care must be taken.

8.13.1.18 RTCC_SetAlarmRptCount

```
void RtccSetAlarmRptCount(unsigned char rptCnt, BOOL dsblAlrm)
```

The function sets the RTCC alarm repeat rate.

Input Parameters

Input Parameters	Description
rptCnt	value of the desired alarm repeat rate
dsblAlrm	if TRUE, the API can temporarily disable the alarm when changing the RPT value

Returns

None

Remarks

rptCnt will be truncated to fit into 8 bit representation. If alarm is enabled, changing the repeat count can be safely made only when the sync pulse is unasserted. To avoid waiting for the sync pulse, the user can choose to temporarily disable the alarm and then re-enable it. This means that the user has the knowledge that an alarm event is not imminent.

8.13.2 RTCC Structs,Records,Enums**8.13.3 RTCC Macros****8.13.4 RTCC Examples****8.13.4.1 RTCC Example1**

Example Source Code demonstrating peripheral library usage

```
/* *****
```

```

* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PIC18F46J50 controller
* NOTE: Device Operates with 3.3V supply
*****/

#include <p18cxxx.h>
#include "rtcc.h"

//---Function Prototypes---
void RTCC_configure(void);

//---Global structures used in deep sleep library---
rtccTimeDate RtccTimeDate ,RtccAlrmTimeDate, Rtcc_read_TimeDate ;

//*****main function*****
void main(void)
{
    mRtcc_Clear_Intr_Status_Bit;           //clears the RTCC interrupt status bit
    RTCC_configure();                     //Configure RTCC using library APIs

    while(PIR3bits.RTCCIF==0)             //wait untill alarm is set
        RtccReadTimeDate(&Rtcc_read_TimeDate); //Rtcc_read_TimeDate will have latest time

    while(1);                             //End of program
}

void RTCC_configure(void)
{
    unsigned int i=0,j=0;

    RtccInitClock();                     //turn on clock source
    RtccWrOn();                          //write enable the rtcc registers
    mRtccSetClockOe(1);                  //enable RTCC output on RTCC

    output pin
    PIE3bits.RTCCIE=1;
    //Set Date and time using global structures defined in libraries
    RtccTimeDate.f.hour = 1;              //Set Hour
    RtccTimeDate.f.min = 0;               //Set minute
    RtccTimeDate.f.sec = 0;               //Set second
    RtccTimeDate.f.mday = 04;             //Set day
    RtccTimeDate.f.mon = 04;              //Se month
    RtccTimeDate.f.year = 09;             //set year
    RtccTimeDate.f.wday = 6;              //Set which day of the week for
    the corrspoding date

    //Set the alarm time and date using gloabl structures defined in libraries
    RtccAlrmTimeDate.f.hour = RtccTimeDate.f.hour; //Set Hour
    RtccAlrmTimeDate.f.min = RtccTimeDate.f.min ; //Set minute
    RtccAlrmTimeDate.f.sec = RtccTimeDate.f.sec + 4; //alarm after ten seconds
    RtccAlrmTimeDate.f.mday = RtccTimeDate.f.mday; //Set day
    RtccAlrmTimeDate.f.wday = RtccTimeDate.f.wday; //Set which day of the week for
    the corrspoding date
    RtccAlrmTimeDate.f.mon = RtccTimeDate.f.mon; //Se month
    RtccAlrmTimeDate.f.year = RtccTimeDate.f.year; //set year

    RtccWriteTimeDate(&RtccTimeDate,1); //write into registers
    RtccSetAlarmRpt(RTCC_RPT_TEN_SEC,1); //Set the alarm repeat to
    every minute
    RtccSetAlarmRptCount(5,1);           //set alarm repeat count
    RtccWriteAlrmTimeDate(&RtccAlrmTimeDate); //write the time for alarm into
    alarm registers
    mRtccOn();                           //enable the rtcc
    mRtccAlrmEnable();                   //enable the rtcc alarm to wake
    the device up from deep sleep

```

```
}
```

8.14 Software RTCC (SW_RTCC)

This peripheral library support provides simulated RTCC support using the normal timer modules available on the device. This module can be made use for those devices on which the RTCC module in hardware is not available.

8.14.1 SW_RTCC Functions

8.14.1.1 Open_SW_RTCC

```
void Open_RTCC(void)
```

Configure and enable Timer1 which is used in the RTCC simulation

Returns

None

Remarks

Configures Timer1 to work as clock source for RTCC, enables Timer1 interrupts, and writes a value into TMR1H & TMR1L registers to get 1second interrupt

8.14.1.2 update_SW_RTCC

```
unsigned char update_RTCC(void)
```

Checks for the Timer1 interrupt flag and refreshes TMR1H register in case of interrupt

Returns

Returns the state of the TMR1IF to check if the 1 second interval is completed or not

Remarks

checks for the TMR1 interrupt flag, refreshes TMR1H if interrupt has occurred and returns the state of TMR1IF

8.14.1.3 Close_SW_RTCC

```
void Close_RTCC(void)
```

Disable Timer1 and ends the RTCC in software

Returns

None

Remarks

Disable Timer1, clear TMR1 interrupt flag, disable TMR1 interrupt

8.14.2 SW_RTCC Examples

8.14.2.1 SW_RTCC Example1

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "rtcc.h"
#include "LCD.h"

/***** Prototypes *****/
void User_Timer(void);

/***** Global variables *****/
unsigned int msec, sec, min, hr, MSD, MdD, LSD;
char day=04, month=04, year=09;

/***** Main function *****/
void main(void)
{
    unsigned char config1=0x00;
    unsigned char config2=0x00;
    unsigned int timer_value=0x00;

    /**** Initialization of Time 00:00:00 and Date 04/04/09 ****
    hr=0;
    min=0;
    sec=0;
    msec=0;
    day = 4;
    month = 4;
    year = 9 ;

    /****Configure SW_RTCC****
    Open_RTCC(); //Configures RTCC using timer in controller

    while(1)
    {
        DisplayTime(hr,min,sec,msec); //Display the Time on LCD
        DisplayDate(day,month,year); //Display Date on LCD
        while(update_RTCC()) ; //wait for timer interruption after one milli

```

```

second completion
    User_Timer();           //Update Timer count
}

/** Close SW_RTCC***/
    Close_RTCC();
}

void User_Timer(void)
{
    msec++;
    if(msec>=100)
    {
        sec++;
        msec=0;
        if(sec>=60)
        {
            min++;
            sec=0;
            if(min>=60)
            {
                hr++;
                min=0;
                if(hr>=24)
                {
                    hr=0;
                }
            }
        }
    }
}

```

8.15 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shiftregisters, display drivers, A/D Converters, etc. The SPI module is compatible with Motorola's SPI and SIOP interfaces. The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer.

8.15.1 SPI Functions

8.15.1.1 SPI_Open

8.15.1.1.1 Open_SPI

```
void OpenSPI(unsigned char sync_mode, unsigned char bus_mode, unsigned char smp_phase)
```

Initialize the SSPx module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameters to be configured in the SSPxCON1 register * SPI_FOSC_4 * SPI_FOSC_16 * SPI_FOSC_64 * SPI_FOSC_TMR2 * SLV_SSON * SLV_SSOFF
bus_mode	SPI bus mode * MODE_00 * MODE_01 * MODE_10 * MODE_11
smp_phase	Data sampling selection * SMPEND * SMPMID

Returns

None

Remarks

This function sets up the SSPx module for use with a SPIx bus device.

8.15.1.1.2 Open_SPI1

```
void OpenSPI1( unsigned char sync_mode, unsigned char bus_mode, unsigned char smp_phase)
```

Initialize the SSPx module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameters to be configured in the SSPxCON1 register * SPI_FOSC_4 * SPI_FOSC_16 * SPI_FOSC_64 * SPI_FOSC_TMR2 * SLV_SSON * SLV_SSOFF
bus_mode	SPI bus mode * MODE_00 * MODE_01 * MODE_10 * MODE_11
smp_phase	Data sampling selection * SMPEND * SMPMID

Returns

None

Remarks

This function sets up the SSPx module for use with a SPIx bus device.

8.15.1.1.3 Open_SPI2

```
void OpenSPI2( unsigned char sync_mode, unsigned char bus_mode, unsigned char smp_phase)
```

Initialize the SSPx module.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	This contains the parameters to be configured in the SSPxCON1 register * SPI_FOSC_4 * SPI_FOSC_16 * SPI_FOSC_64 * SPI_FOSC_TMR2 * SLV_SSON * SLV_SSOFF
bus_mode	SPI bus mode * MODE_00 * MODE_01 * MODE_10 * MODE_11
smp_phase	Data sampling selection * SMPEND * SMPMID

Returns

None

Remarks

This function sets up the SSPx module for use with a SPIx bus device.

8.15.1.2 SPI_Write

8.15.1.2.1 Write_SPI

unsigned char WriteSPI(unsigned char data_out)

This routine writes a single byte to the SPIx bus

Input Parameters

Input Parameters	Description
data_out	Value to be written to the SPIx bus

Returns

Returns fault value:

- * 0 : if no write collision occurred
- * -1 : if a write collision occurred

Remarks

This function writes a single data byte out and then checks for a write collision. putcSPI is alternative name to WriteSPI.

8.15.1.2.2 Write_SPI1

unsigned char WriteSPI1(unsigned char data_out)

This routine writes a single byte to the SPIx bus

Input Parameters

Input Parameters	Description
data_out	Value to be written to the SPIx bus

Returns

Returns fault value:

- * 0 : if no write collision occurred
- * -1 : if a write collision occurred

Remarks

This function writes a single data byte out and then checks for a write collision. putcSPI1 is alternative name to WriteSPI1

8.15.1.2.3 Write_SPI2

unsigned char WriteSPI2(unsigned char data_out)

This routine writes a single byte to the SPIx bus

Input Parameters

Input Parameters	Description
data_out	Value to be written to the SPIx bus

Returns

Returns fault value:

- * 0 : if no write collision occurred
- * -1 : if a write collision occurred

Remarks

This function writes a single data byte out and then checks for a write collision. putcSPI2 is alternative name to WriteSPI2

8.15.1.3 SPI_Read

8.15.1.3.1 Read_SPI

```
unsigned char ReadSPI( void )
```

Read single byte from SPI bus

Returns

This function returns a byte of data read during a SPIx read cycle.

Remarks

This function initiates a SPIx bus cycle for the acquisition of a byte of data. getcSPI is alternative name to ReadSPI

8.15.1.3.2 Read_SPI1

```
unsigned char ReadSPI1( void )
```

Read single byte from SPI1 bus

Returns

This function returns a byte of data read during a SPIx read cycle.

Remarks

This function initiates a SPIx bus cycle for the acquisition of a byte of data. getcSPI1 is alternative name to ReadSPI1

8.15.1.3.3 Read_SPI2

```
unsigned char ReadSPI2( void )
```

Read single byte from SPI2 bus

Returns

This function returns a byte of data read during a SPIx read cycle.

Remarks

This function initiates a SPIx bus cycle for the acquisition of a byte of data. getcSPI2 is alternative name to ReadSPI2

8.15.1.4 SPI_puts

8.15.1.4.1 puts_SPI

```
void putsSPI( unsigned char *wrptr )
```

Write a string to the SPIx bus

Input Parameters

Input Parameters	Description
wrptr	Pointer to the array of data in RAM that will be written to the SPIx bus

Returns

None

Remarks

This function writes out a data string to the SPIx bus device. The routine is terminated by reading a null character in the data string (the null character is not written to the bus).

8.15.1.4.2 puts_SPI1

```
void putsSPI1( unsigned char *wrptr )
```

Write a string to the SPIx bus

Input Parameters

Input Parameters	Description
wrptr	Pointer to the array of data in RAM that will be written to the SPIx bus

Returns

None

Remarks

This function writes out a data string to the SPIx bus device. The routine is terminated by reading a null character in the data string (the null character is not written to the bus).

8.15.1.4.3 puts_SPI2

```
void putsSPI2( unsigned char *wrptr )
```

Write a string to the SPIx bus

Input Parameters

Input Parameters	Description
wrptr	Pointer to the array of data in RAM that will be written to the SPIx bus

Returns

None

Remarks

This function writes out a data string to the SPIx bus device. The routine is terminated by reading a null character in the data string (the null character is not written to the bus).

8.15.1.5 SPI_gets

8.15.1.5.1 gets_SPI

```
void getsSPI( unsigned char *rdptr, unsigned char length )
```

Read a string from the SPIx bus

Input Parameters

Input Parameters	Description
rdptr	Pointer to location in RAM of device to store data read from SPIx device
length	Number of bytes to read from SPIx device

Returns

None

Remarks

This function reads in a predetermined data string length from the SPIx bus

8.15.1.5.2 gets_SPI1

```
void getsSPI1( unsigned char *rdptr, unsigned char length )
```

Read a string from the SPIx bus

Input Parameters

Input Parameters	Description
rdptr	Pointer to location in RAM of device to store data read from SPIx device
length	Number of bytes to read from SPIx device

Returns
None

Remarks

This function reads in a predetermined data string length from the SPIx bus

8.15.1.5.3 gets_SPI2

```
void getsSPI2( unsigned char *rdptr, unsigned char length )
```

Read a string from the SPIx bus

Input Parameters

Input Parameters	Description
rdptr	Pointer to location in RAM of device to store data read from SPIx device
length	Number of bytes to read from SPIx device

Returns
None

Remarks

This function reads in a predetermined data string length from the SPIx bus

8.15.2 SPI Macros

8.15.3 SPI Examples

8.15.3.1 SPI Example1_Master

Example Source Code demonstrating peripheral library usage

```
/******
```

```

* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*
* SDO,SDI & SCK must be connected to the corresponding in slave
*   MASTER          SLAVE
*
*   SDO    <----->    SDI
*   SDI    <----->    SDO
*   SCK    <----->    SCK
* NOTE: Refer Slave source code also
*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "spi.h"

unsigned char SPI_Send[21] = "MICROCHIP_SPI_MASTER";
unsigned char SPI_Recv[21];

//***** SPI MASTER *****/

void main(void)
{
    unsigned char sync_mode=0;
    unsigned char bus_mode=0;
    unsigned char smp_phase=0;
    unsigned char w=0;

    CloseSPI(); // Turn off SPI modules if was previously on

    /***Configure SPI MASTER module to transmit in master mode ***/
    sync_mode = SPI_FOSC_64 ;
    bus_mode = MODE_01;
    smp_phase = SMPMID;
    OpenSPI(sync_mode,bus_mode,smp_phase );

    /*** WRITE INITIAL CHARECTER ***/
    while(WriteSPI(0xF5)); //send initial charecter to use the same as flag
    at slave side and send it till successful transmission

    /*** WRITE THE STRING TO SPI ****
        putsSPI(SPI_Send); //send the string of data to be sent to slave

    /*** Read the initial flag id ****
        if( 0xF5 == ReadSPI() )
        {
            getsSPI(SPI_Recv,20); // read the string sent from slave
            SPI_Recv[20] = '\0' ; //terminate the string with a null charecter
        }

    /* Turn off SPI module and clear IF bit */
    CloseSPI();

    while(1); //End of program
}

```

8.15.3.2 SPI Example1_Slave

Example Source Code demonstrating peripheral library usage

```

/*****
* NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
*

```

```

* SDO,SDI & SCK must be connected to the corresponding in slave
*   MASTER           SLAVE
*
*   SDO   <----->   SDI
*   SDI   <----->   SDO
*   SCK   <----->   SCK
* NOTE: Refer Master source code also
*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "spi.h"

unsigned char SPI_Send[25] = "MICROCHIP_SPI_SLAVE";
unsigned char SPI_Recv[25];

//***** SPI SLAVE *****

void main(void)
{
    unsigned char sync_mode=0;
    unsigned char bus_mode=0;
    unsigned char smp_phase=0;
    unsigned char w=0,temp;

    for(w=0;w<25;w++)
        SPI_Recv[w]=0;

    CloseSPI(); // Turn off SPI modules if was previosly on

    /***Configure SPI SLAVE module ***/
    sync_mode = SLV_SSOF;
    bus_mode = MODE_01;
    smp_phase = SMPMID;
    OpenSPI(sync_mode,bus_mode,smp_phase );

    /***Read the initial flag byte sent by master ***/
    temp = ReadSPI(); //This is for address implementation in software

    if(temp == 0xF5)
    {
        getsSPI(SPI_Recv,21); //recieve the string of data from master
        SPI_Recv[21] = '\0' ; //put null charecter at the end of string recieved
        while( PIR1bits.SSPIF!=1 ); //wait till completion of transmission

        WriteSPI(0xF5); //send the software flag bit to master
        while( PIR1bits.SSPIF!=1 ); //wait till completion of transmission

        /*** WRITE THE STRING TO SPI ***/
        putsSPI(SPI_Send);
    }

    /* Turn off SPI module and clear IF bit */
    CloseSPI();

    while(1); //End of program
}

```

8.16 TIMERS

The Timers module is a 8/16-bit timer which can serve as the time counter for the Real-Time Clock (RTC), or operate as a free-running, interval timer/counter. The Timer2/3 and Timer4/5 modules are 8/16-bit timers, which can also be configured as four independent 8-bit timers with selectable operating modes. Timer 1 operates in CPU Idle modes and Sleep modes.

Individually, timers can function as synchronous timers or counters.

8.16.1 TIMERS Functions

8.16.1.1 Open_Timer

8.16.1.1.1 Open_Timer0

```
void OpenTimer0(unsigned char config)
```

Configure and enable Timer0

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer0 Interrupt:<ul style="list-style-type: none">* TIMER_INT_ON* TIMER_INT_OFF* TIMER_INT_MASK</div> <div>Timer Width:<ul style="list-style-type: none">* T0_8BIT* T0_16BIT* T0_BIT_MASK</div> <div>Clock Source:<ul style="list-style-type: none">* T0_SOURCE_EXT* T0_SOURCE_INT* T0_SOURCE_MASK</div> <div>External Clock Trigger (for T0_SOURCE_EXT):<ul style="list-style-type: none">* T0_EDGE_FALL* T0_EDGE_RISE* T0_EDGE_MASK</div> <div>Prescale Value:<ul style="list-style-type: none">* T0_PS_1_1* NO_T0_PS_MASK* T0_PS_1_2* T0_PS_1_4* T0_PS_1_8* T0_PS_1_16* T0_PS_1_32* T0_PS_1_64* T0_PS_1_128* T0_PS_1_256* T0_PS_MASK</div>

Returns
None

Remarks

This function configures Timer0 according to the options specified and then enables it

8.16.1.1.2 Open_Timer1_Page1

void OpenTimer1(unsigned char config)

Configure and enable Timer1

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer1 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK</div> <div>Timer Width: * T1_8BIT_RW * T1_16BIT_RW * T1_BIT_RW_MASK</div> <div>Clock Source: * T1_SOURCE_EXT * T1_SOURCE_INT * T1_SOURCE_MASK</div> <div>Prescaler: * T1_PS_1_1 * T1_PS_1_2 * T1_PS_1_4 * T1_PS_1_8 * T1_PS_MASK</div> <div>Oscillator Use: * T1_OSC1EN_ON * T1_OSC1EN_OFF * T1_OSC_MASK</div> <div>Synchronize Clock Input: * T1_SYNC_EXT_ON * T1_SYNC_EXT_OFF * T1_SYNC_MASK</div>

Returns
None

Remarks

This function configures Timer1 according to the options specified and then enables it

8.16.1.1.3 Open_Timer1_Page2

```
void OpenTimer1(unsigned char config, unsigned char config1)
```

Configure and enable Timer1

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Enable Timer1 Interrupt:</p> <ul style="list-style-type: none"> * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK <p>Timer Width:</p> <ul style="list-style-type: none"> * T1_8BIT_RW * T1_16BIT_RW * T1_BIT_RW_MASK <p>Clock Source:</p> <ul style="list-style-type: none"> * T1_SOURCE_PINOSC * T1_SOURCE_CAPOSC * T1_SOURCE_FOSC_4 * T1_SOURCE_FOSC * T1_SOURCE_MASK <p>Prescaler:</p> <ul style="list-style-type: none"> * T1_PS_1_1 * T1_PS_1_2 * T1_PS_1_4 * T1_PS_1_8 * T1_PS_MASK <p>Oscillator Use:</p> <ul style="list-style-type: none"> * T1_OSC1EN_ON * T1_OSC1EN_OFF * T1_OSC_MASK <p>Synchronize Clock Input:</p> <ul style="list-style-type: none"> * T1_SYNC_EXT_ON * T1_SYNC_EXT_OFF * T1_SYNC_MASK

config1	<div>Timer1 Gate Enable</div> <div>* TIMER_GATE_ON</div> <div>* TIMER_GATE_OFF</div> <div>* TIMER_GATE_MASK</div> <div>Timer1 Gate Polarity</div> <div>* TIMER_GATE_POL_HI</div> <div>* TIMER_GATE_POL_LO</div> <div>* TIMER_GATE_POL_MASK</div> <div>Timer1 Gate Toggle Mode</div> <div>* TIMER_GATE_TOGGLE_ON</div> <div>* TIMER_GATE_TOGGLE_OFF</div> <div>* TIMER_GATE_TOGGLE_MASK</div> <div>Timer1 Gate One Shot Enable</div> <div>* TIMER_GATE_1SHOT_ON</div> <div>* TIMER_GATE_1SHOT_OFF</div> <div>* TIMER_GATE_1SHOT_MASK</div> <div>Timer1 Gate Source Select</div> <div>* TIMER_GATE_SRC_T1GPIN</div> <div>* TIMER_GATE_SRC_T0</div> <div>* TIMER_GATE_SRC_T2</div> <div>* TIMER_GATE_SRC_MASK</div> <div>Enable Timer1 Gate Interrupt:</div> <div>* TIMER_GATE_INT_OFF</div> <div>* TIMER_GATE_INT_ON</div> <div>* TIMER_GATE_INT_MASK</div>
---------	---

Returns
None

Remarks

This function configures Timer1 according to the options specified and then enables it

8.16.1.1.4 Open_Timer2

void OpenTimer2(unsigned char config)

Configure and enable Timer2

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Enable Timer2 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK Prescale Value: * T2_PS_1_1 * T2_PS_1_4 * T2_PS_1_16 Postscale Value: * T2_POST_1_1 * T2_POST_1_2 * T2_POST_1_3 * T2_POST_1_4 * T2_POST_1_5 * T2_POST_1_6 * T2_POST_1_7 * T2_POST_1_8 * T2_POST_1_9 * T2_POST_1_10 * T2_POST_1_11 * T2_POST_1_12 * T2_POST_1_13 * T2_POST_1_14 * T2_POST_1_15 * T2_POST_1_16 * T2_POST_MASK

Returns
None

Remarks

This function configures Timer2 according to the options specified and then enables it

8.16.1.1.5 Open_Timer3_Page1

void OpenTimer3(unsigned char config)

Configure and enable Timer3

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer3 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK</div> <div>Timer Width: * T3_8BIT_RW * T3_16BIT_RW * T3_BIT_MASK</div> <div>Clock Source: * T3_SOURCE_EXT * T3_SOURCE_INT * T3_SOURCE_MASK</div> <div>Prescale Value: * T3_PS_1_1 * T3_PS_1_2 * T3_PS_1_4 * T3_PS_1_8 * T3_PS_MASK</div> <div>Synchronize Clock Input: * T3_SYNC_EXT_ON * T3_SYNC_EXT_OFF * T3_SYNC_MASK</div>

Returns
None

Remarks

This function configures Timer2 according to the options specified and then enables it

8.16.1.1.6 Open_Timer3_Page2

```
void OpenTimer3(unsigned char config, unsigned char config1)
```

Configure and enable Timer3

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Enable Timer3 Interrupt:</p> <ul style="list-style-type: none"> * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK <p>Timer Width:</p> <ul style="list-style-type: none"> * T3_8BIT_RW * T3_16BIT_RW * T3_BIT_MASK <p>Clock Source:</p> <ul style="list-style-type: none"> * T3_SOURCE_PINOSC * T3_SOURCE_CAPOSC * T3_SOURCE_FOSC_4 * T3_SOURCE_FOSC * T3_SOURCE_MASK <p>Oscillator Use:</p> <ul style="list-style-type: none"> * T3_OSC1EN_ON * T3_OSC1EN_OFF * T3_OSC_MASK <p>Prescale Value:</p> <ul style="list-style-type: none"> * T3_PS_1_1 * T3_PS_1_2 * T3_PS_1_4 * T3_PS_1_8 * T3_PS_MASK <p>Synchronize Clock Input:</p> <ul style="list-style-type: none"> * T3_SYNC_EXT_ON * T3_SYNC_EXT_OFF * T3_SYNC_MASK

config1	<div>Timer3 Gate Enable</div> <div>* TIMER_GATE_ON</div> <div>* TIMER_GATE_OFF</div> <div>* TIMER_GATE_MASK</div> <div>Timer3 Gate Polarity</div> <div>* TIMER_GATE_POL_HI</div> <div>* TIMER_GATE_POL_LO</div> <div>* TIMER_GATE_POL_MASK</div> <div>Timer3 Gate Toggle Mode</div> <div>* TIMER_GATE_TOGGLE_ON</div> <div>* TIMER_GATE_TOGGLE_OFF</div> <div>* TIMER_GATE_TOGGLE_MASK</div> <div>Timer3 Gate One Shot Enable</div> <div>* TIMER_GATE_1SHOT_ON</div> <div>* TIMER_GATE_1SHOT_OFF</div> <div>* TIMER_GATE_1SHOT_MASK</div> <div>Timer3 Gate Source Select</div> <div>* TIMER_GATE_SRC_T1GPIN</div> <div>* TIMER_GATE_SRC_T0</div> <div>* TIMER_GATE_SRC_T2</div> <div>* TIMER_GATE_SRC_MASK</div> <div>Enable Timer3 Gate Interrupt:</div> <div>* TIMER_GATE_INT_OFF</div> <div>* TIMER_GATE_INT_ON</div> <div>* TIMER_GATE_INT_MASK</div>
---------	---

Returns
None

Remarks

This function configures Timer3 according to the options specified and then enables it

8.16.1.1.7 Open_Timer5

void OpenTimer5(unsigned char config,unsigned int t5pr)

Configure and enable Timer5

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Enable Timer5 Interrupt:</p> <ul style="list-style-type: none"> * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK <p>Timer5 Sleep Enable:</p> <ul style="list-style-type: none"> * T5_SLP_EN * T5_SLP_DIS * T5_SLP_MASK <p>Special Event Reset:</p> <ul style="list-style-type: none"> * T5_SP_EVNT_REN * T5_SP_EVNT_RDIS * T5_SP_ENNT_MASK <p>Timer5 Mode:</p> <ul style="list-style-type: none"> * T5_MD_SNGL_SHOT * T5_MD_CONT_COUNT * T5_MD_MASK <p>Prescale Value:</p> <ul style="list-style-type: none"> * T5_PS_1_1 * T5_PS_1_2 * T5_PS_1_4 * T5_PS_1_8 * T5_PS_MASK <p>Synchronize Clock Input:</p> <ul style="list-style-type: none"> * T5_EX_CLK_SYNC * T5_EX_CLK_NOSYNC * T5_EX_CLK_SYNC_MASK <p>Clock Source:</p> <ul style="list-style-type: none"> * T5_CLK_EXTRN * T5_CLK_INT * T5_CLK_SOURCE_MASK
t5pr	<p>t5pr value will be loaded in to low and high byte of the Timer5 Period Register (i.e. PR5L = t5pr, PR5H = (t5pr>>8))</p>

Returns

None

Remarks

This function configures Timer5 according to the options specified and then enables it

8.16.1.1.8 Open_Timer5_Page2

void OpenTimer5(unsigned char config, unsigned char config1)

Configure and enable Timer5

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer5 Interrupt:<ul style="list-style-type: none">* TIMER_INT_ON* TIMER_INT_OFF* TIMER_INT_MASK</div> <div>Timer Width:<ul style="list-style-type: none">* T5_8BIT_RW* T5_16BIT_RW* T5_BIT_MASK</div> <div>Clock Source:<ul style="list-style-type: none">* T5_SOURCE_PINOSC* T5_SOURCE_CAPOSC* T5_SOURCE_FOSC_4* T5_SOURCE_FOSC* T5_SOURCE_MASK</div> <div>Oscillator Use:<ul style="list-style-type: none">* T5_OSC1EN_ON* T5_OSC1EN_OFF* T5_OSC_MASK</div> <div>Prescale Value:<ul style="list-style-type: none">* T5_PS_1_1* T5_PS_1_2* T5_PS_1_4* T5_PS_1_8* T5_PS_MASK</div> <div>Synchronize Clock Input:<ul style="list-style-type: none">* T5_SYNC_EXT_ON* T5_SYNC_EXT_OFF* T5_SYNC_MASK</div>

config1	<div>Timer5 Gate Enable</div> <div>* TIMER_GATE_ON</div> <div>* TIMER_GATE_OFF</div> <div>* TIMER_GATE_MASK</div> <div>Timer5 Gate Polarity</div> <div>* TIMER_GATE_POL_HI</div> <div>* TIMER_GATE_POL_LO</div> <div>* TIMER_GATE_POL_MASK</div> <div>Timer5 Gate Toggle Mode</div> <div>* TIMER_GATE_TOGGLE_ON</div> <div>* TIMER_GATE_TOGGLE_OFF</div> <div>* TIMER_GATE_TOGGLE_MASK</div> <div>Timer5 Gate One Shot Enable</div> <div>* TIMER_GATE_1SHOT_ON</div> <div>* TIMER_GATE_1SHOT_OFF</div> <div>* TIMER_GATE_1SHOT_MASK</div> <div>Timer5 Gate Source Select</div> <div>* TIMER_GATE_SRC_T1GPIN</div> <div>* TIMER_GATE_SRC_T0</div> <div>* TIMER_GATE_SRC_T2</div> <div>* TIMER_GATE_SRC_MASK</div> <div>Enable Timer5 Gate Interrupt:</div> <div>* TIMER_GATE_INT_OFF</div> <div>* TIMER_GATE_INT_ON</div> <div>* TIMER_GATE_INT_MASK</div>
---------	---

Returns
None

Remarks

This function configures Timer5 according to the options specified and then enables it

8.16.1.1.9 Open_Timer4

void OpenTimer4(unsigned char config)

Configure and enable Timer4

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer4 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK</div> <div>Prescale Value: * T4_PS_1_1 * T4_PS_1_4 * T4_PS_1_16</div> <div>Postscale Value: * T4_POST_1_1 * T4_POST_1_2 * T4_POST_1_3 * T4_POST_1_4 * T4_POST_1_5 * T4_POST_1_6 * T4_POST_1_7 * T4_POST_1_8 * T4_POST_1_9 * T4_POST_1_10 * T4_POST_1_11 * T4_POST_1_12 * T4_POST_1_13 * T4_POST_1_14 * T4_POST_1_15 * T4_POST_1_16 * T4_POST_MASK</div>

Returns
None

Remarks

This function configures Timer4 according to the options specified and then enables it

8.16.1.1.10 Open_Timer6_Page1

void OpenTimer6(unsigned char config)

Configure and enable Timer6

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer6 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK</div> <div>Prescale Value: * T6_PS_1_1 * T6_PS_1_4 * T6_PS_1_16</div> <div>Postscale Value: * T6_POST_1_1 * T6_POST_1_2 * T6_POST_1_3 * T6_POST_1_4 * T6_POST_1_5 * T6_POST_1_6 * T6_POST_1_7 * T6_POST_1_8 * T6_POST_1_9 * T6_POST_1_10 * T6_POST_1_11 * T6_POST_1_12 * T6_POST_1_13 * T6_POST_1_14 * T6_POST_1_15 * T6_POST_1_16 * T6_POST_MASK</div>

Returns
None

Remarks

This function configures Timer6 according to the options specified and then enables it

8.16.1.1.11 Open_Timer7_Page1

void OpenTimer7(unsigned char config, unsigned char config1)

Configure and enable Timer7

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Enable Timer7 Interrupt:</p> <ul style="list-style-type: none"> * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK <p>Timer Width:</p> <ul style="list-style-type: none"> * T7_8BIT_RW * T7_16BIT_RW * T7_BIT_MASK <p>Clock Source:</p> <ul style="list-style-type: none"> * T7_SOURCE_PINOSC * T7_SOURCE_CAPOSC * T7_SOURCE_FOSC_4 * T7_SOURCE_FOSC * T7_SOURCE_MASK <p>Oscillator Use:</p> <ul style="list-style-type: none"> * T7_OSC1EN_ON * T7_OSC1EN_OFF * T7_OSC_MASK <p>Prescale Value:</p> <ul style="list-style-type: none"> * T7_PS_1_1 * T7_PS_1_2 * T7_PS_1_4 * T7_PS_1_8 * T7_PS_MASK <p>Synchronize Clock Input:</p> <ul style="list-style-type: none"> * T7_SYNC_EXT_ON * T7_SYNC_EXT_OFF * T7_SYNC_MASK

config1	<div>Timer7 Gate Enable</div> <div>* TIMER_GATE_ON</div> <div>* TIMER_GATE_OFF</div> <div>* TIMER_GATE_MASK</div> <div>Timer7 Gate Polarity</div> <div>* TIMER_GATE_POL_HI</div> <div>* TIMER_GATE_POL_LO</div> <div>* TIMER_GATE_POL_MASK</div> <div>Timer7 Gate Toggle Mode</div> <div>* TIMER_GATE_TOGGLE_ON</div> <div>* TIMER_GATE_TOGGLE_OFF</div> <div>* TIMER_GATE_TOGGLE_MASK</div> <div>Timer7 Gate One Shot Enable</div> <div>* TIMER_GATE_1SHOT_ON</div> <div>* TIMER_GATE_1SHOT_OFF</div> <div>* TIMER_GATE_1SHOT_MASK</div> <div>Timer7 Gate Source Select</div> <div>* TIMER_GATE_SRC_T1GPIN</div> <div>* TIMER_GATE_SRC_T0</div> <div>* TIMER_GATE_SRC_T2</div> <div>* TIMER_GATE_SRC_MASK</div> <div>Enable Timer7 Gate Interrupt:</div> <div>* TIMER_GATE_INT_OFF</div> <div>* TIMER_GATE_INT_ON</div> <div>* TIMER_GATE_INT_MASK</div>
---------	---

Returns
None

Remarks

This function configures Timer7 according to the options specified and then enables it

8.16.1.1.12 Open_Timer8_Page1

void OpenTimer8(unsigned char config)

Configure and enable Timer8

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Timer8 Interrupt:<ul style="list-style-type: none">* TIMER_INT_ON* TIMER_INT_OFF* TIMER_INT_MASK</div> <div>Prescale Value:<ul style="list-style-type: none">* T8_PS_1_1* T8_PS_1_4* T8_PS_1_16</div> <div>Postscale Value:<ul style="list-style-type: none">* T8_POST_1_1* T8_POST_1_2* T8_POST_1_3* T8_POST_1_4* T8_POST_1_5* T8_POST_1_6* T8_POST_1_7* T8_POST_1_8* T8_POST_1_9* T8_POST_1_10* T8_POST_1_11* T8_POST_1_12* T8_POST_1_13* T8_POST_1_14* T8_POST_1_15* T8_POST_1_16* T8_POST_MASK</div>

Returns
None

Remarks

This function configures Timer8 according to the options specified and then enables it

8.16.1.1.13 Open_Timer10_Page1

void OpenTimer10(unsigned char config)

Configure and enable Timer10

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<div>Enable Time10 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK</div> <div>Prescale Value: * T10_PS_1_1 * T10_PS_1_4 * T10_PS_1_16</div> <div>Postscale Value: * T10_POST_1_1 * T10_POST_1_2 * T10_POST_1_3 * T10_POST_1_4 * T10_POST_1_5 * T10_POST_1_6 * T10_POST_1_7 * T10_POST_1_8 * T10_POST_1_9 * T10_POST_1_10 * T10_POST_1_11 * T10_POST_1_12 * T10_POST_1_13 * T10_POST_1_14 * T10_POST_1_15 * T10_POST_1_16 * T10_POST_MASK</div>

Returns
None

Remarks

This function configures Timer10 according to the options specified and then enables it

8.16.1.1.14 Open_Timer12_Page1

void OpenTimer12(unsigned char config)

Configure and enable Timer12

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Enable Time12 Interrupt: * TIMER_INT_ON * TIMER_INT_OFF * TIMER_INT_MASK Prescale Value: * T12_PS_1_1 * T12_PS_1_4 * T12_PS_1_16 Postscale Value: * T12_POST_1_1 * T12_POST_1_2 * T12_POST_1_3 * T12_POST_1_4 * T12_POST_1_5 * T12_POST_1_6 * T12_POST_1_7 * T12_POST_1_8 * T12_POST_1_9 * T12_POST_1_10 * T12_POST_1_11 * T12_POST_1_12 * T12_POST_1_13 * T12_POST_1_14 * T12_POST_1_15 * T12_POST_1_16 * T12_POST_MASK

Returns
None

Remarks

This function configures Timer12 according to the options specified and then enables it

8.16.1.2 Read_Timer

8.16.1.2.1 Read_Timer0

unsigned int ReadTimer0(void)

Reads the value of the Timer0 registers TMR0L,TMR0H

Returns

The current value of the Timer0

Remarks

When using a timer in 8-bit mode that may be configured in 16-bit mode (e.g., timer0), the upper byte is not ensured to be zero. The user may wish to cast the result to a char for correct results. For example:

```
// Example of reading a 16-bit result
// from a 16-bit timer operating in
// 8-bit mode:
```

```
unsigned int result;
result = (unsigned char) ReadTimer0();
```

8.16.1.2.2 Read_Timer1

unsigned int ReadTimer1(void)

Reads the value of the Timer1 registers TMR1L,TMR1H

Returns

The current value of the Timer1

Remarks

None

8.16.1.2.3 Read_Timer3

unsigned int ReadTimer3(void)

Reads the value of the Timer3 registers TMR3L,TMR3H

Returns

The current value of the Timer3

Remarks

None

8.16.1.2.4 Read_Timer5

unsigned int ReadTimer5(void)

Reads the value of the Timer5 registers TMR5L,TMR5H

Returns

The current value of the Timer5

Remarks

None

8.16.1.2.5 Read_Timer7_Page1

unsigned int ReadTimer7(void)

Reads the value of the Timer7 registers TMR7L,TMR7H

Returns

The current value of the Timer7

Remarks

None

8.16.1.3 Write_Timer

8.16.1.3.1 Write_Timer0

void WriteTimer0(unsigned int timer0)

Write a value into the Timer0 registers TMR0L & TMR0H

Input Parameters

Input Parameters	Description
timer0	The value that will be loaded into the specified timer registers (TMR0H = unsigned char(timer0>>8) & TMR0L = unsigned char timer0)

Returns

None

8.16.1.3.2 Write_Timer1

```
void WriteTimer1(unsigned int timer1)
```

Write a value into the Timer1 registers TMR1L & TMR1H

Input Parameters

Input Parameters	Description
timer1	The value that will be loaded into the specified timer registers (TMR1H = unsigned char(timer1>>8) & TMR1L = unsigned char timer1)

Returns

None

8.16.1.3.3 Write_Timer3

```
void WriteTimer3(unsigned int timer3)
```

Write a value into the Timer3 registers TMR3L & TMR3H

Input Parameters

Input Parameters	Description
timer3	The value that will be loaded into the specified timer registers (TMR3H = unsigned char(timer3>>8) & TMR3L = unsigned char timer3)

Returns

None

8.16.1.3.4 Write_Timer5

```
void WriteTimer5(unsigned int timer5)
```

Write a value into the Timer5 registers TMR5L & TMR5H

Input Parameters

Input Parameters	Description
timer5	The value that will be loaded into the specified timer registers (TMR5H = unsigned char(timer5>>8) & TMR5L = unsigned char timer5)

Returns
None

8.16.1.3.5 Write_Timer7_Page1

```
void WriteTimer7(unsigned int timer7)
```

Write a value into the Timer7 registers TMR7L & TMR7H

Input Parameters

Input Parameters	Description
timer7	The value that will be loaded into the specified timer registers (TMR7H = unsigned char(timer7>>8) & TMR7L = unsigned char timer7)

Returns
None

8.16.1.4 Set_TmrCCPSrc

8.16.1.4.1 Set_TmrCCPSrc_Page1

```
void SetTmrCCPSrc(unsigned char config)
```

Enable timer source to CCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Timers as source of CCP * T3_SOURCE_CCP * T1_CCP1_T3_CCP2 * T1_SOURCE_CCP * TMR_SOURCE_CCP_MASK

Returns
None

Remarks

This function configures timer as a clock sources for the CCP module

8.16.1.4.2 Set_TmrCCPSrc_Page2

```
void SetTmrCCPSrc(unsigned char config)
```

Enable timer source to CCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Timers as source of CCP * T34_SOURCE_CCP * T12_CCP12_T34_CCP345 * T12_CCP1_T34_CCP2345 * T12_SOURCE_CCP * TMR_SOURCE_CCP_MASK

Returns
None

Remarks

This function configures timer as a clock sources for the CCP module

8.16.1.4.3 Set_TmrCCPSrc_Page3

```
void SetTmrCCPSrc(unsigned char config)
```

Enable timer source to CCP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	Timers as source of CCP * T34_SOURCE_CCP12 * T12_CCP1_T34_CCP2 * T12_SOURCE_CCP * TMR_SOURCE_CCP_MASK

Returns
None

Remarks

This function configures timer as a clock sources for the CCP module

8.16.1.5 Close_Timer

8.16.1.5.1 Close_Timer0

void CloseTimer0(void)

Disable the Timer0.

Returns
None

Remarks

This function disables the interrupt and the Timer0

8.16.1.5.2 Close_Timer1

void CloseTimer1(void)

Disable the Timer1.

Returns
None

Remarks

This function disables the interrupt and the Timer1

8.16.1.5.3 Close_Timer2

```
void CloseTimer2(void)
```

Disable the Timer2.

Returns
None

Remarks

This function disables the interrupt and the Timer2

8.16.1.5.4 Close_Timer3

```
void CloseTimer3(void)
```

Disable the Timer3.

Returns
None

Remarks

This function disables the interrupt and the Timer3

8.16.1.5.5 Close_Timer4

```
void CloseTimer4(void)
```

Disable the Timer4.

Returns
None

Remarks

This function disables the interrupt and the Timer4

8.16.1.5.6 Close_Timer5

```
void CloseTimer5(void)
```

Disable the Timer5.

Returns

None

Remarks

This function disables the interrupt and the Timer5

8.16.1.5.7 Close_Timer6_Page1

void CloseTimer6(void)

Disable the Timer6.

Returns

None

Remarks

This function disables the interrupt and the Timer6

8.16.1.5.8 Close_Timer7_Page1

void CloseTimer7(void)

Disable the Timer7.

Returns

None

Remarks

This function disables the interrupt and the Timer7

8.16.1.5.9 Close_Timer8_Page1

void CloseTimer8(void)

Disable the Timer8.

Returns

None

Remarks

This function disables the interrupt and the Timer8

8.16.1.5.10 Close_Timer10_Page1

void CloseTimer10(void)

Disable the Timer10.

Returns

None

Remarks

This function disables the interrupt and the Timer10

8.16.1.5.11 Close_Timer12_Page1

void CloseTimer12(void)

Disable the Timer12.

Returns

None

Remarks

This function disables the interrupt and the Timer12

8.16.2 TIMERS Structs,Records,Enums

8.16.3 TIMERS Macros

8.16.4 TIMERS Examples

8.16.4.1 TIMERS Example1

Example Source Code demonstrating peripheral library usage

```
/******  
* NOTES:  
* Code uses the Peripheral library support available with MCC18 Compiler  
* Code Tested on:  
* PicDem2+ demo board with PIC18F4685 controller
```

```

*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "timers.h"
#include "LCD.h"

//**** Prototypes ****
void User_Timer(void);

//**** Global variables ****
unsigned int msec, sec, min, hr, MSD, MdD, LSD;
char day=04, month=04, year=09;

void main(void)
{
    unsigned char config1=0x00;
    unsigned char config2=0x00;
    unsigned int timer_value=0x00;

    //----Configure Timers----
    timer_value = 0x00;
    WriteTimer1(timer_value);           //clear timer if previously contains any value

    config1 = T1_8BIT_RW | T1_SOURCE_EXT | T1_PS_1_8
              | T1_OSC1EN_ON | T1_SYNC_EXT_ON | TIMER_INT_ON;
    OpenTimer1(config1);                //API configures the tmer1 as per user defined
parameters

    while(1)
    {
        DisplayTime(hr,min,sec,msec);   //Display the Time on LCD
        DisplayDate(day,month,year);    //Displays the Data on LCD
        while(!PIR1bits.TMR1IF);       //wait for timer interruption after one milli
second completion
        User_Timer();                   //Update Timer count
    }

    //**** Close Timer ****
    CloseTimer1();
}

void User_Timer(void)
{
    msec++;
    if(msec>=100)
    {
        sec++;
        msec=0;
        if(sec>=60)
        {
            min++;
            sec=0;
            if(min>=60)
            {
                hr++;
                min=0;
                if(hr>=24)
                {
                    hr=0;
                }
            }
        }
    }
}

```

8.17 Universal Asynchronous/Synchronous Receiver Transmitter (USART)

The Universal Asynchronous/Synchronous Receiver Transmitter (usart) module is one of the serial I/O modules available in the PIC18F device family. The usart is a full-duplex system that can communicate with peripheral devices, such as personal computers, LIN, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option and also includes an IrDA® encoder and decoder. The primary features of the usart module are:

- Asynchronous (full duplex) with:
 - Auto-Wake-up on Character Reception
 - Auto-Baud Calibration
 - 12-Bit Break Character Transmission
- Synchronous – Master (half duplex) with selectable Clock Polarity
- Synchronous – Slave (half duplex) with selectable Clock Polarity

8.17.1 USART Functions

8.17.1.1 USART_Open

8.17.1.1.1 Open_1USART

```
void Open1USART( unsigned char config, unsigned int spbrg)
```

Configure the specified usart module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Interrupt on Transmission:</p> <ul style="list-style-type: none"> * USART_TX_INT_ON * USART_TX_INT_OFF * USART_TX_INT_MASK <p>Interrupt on Receipt:</p> <ul style="list-style-type: none"> * USART_RX_INT_ON * USART_RX_INT_OFF * USART_RX_INT_MASK <p>usart Mode:</p> <ul style="list-style-type: none"> * USART_ASYNC_MODE * USART_SYNC_MODE * USART_MODE_MASK <p>Transmission Width:</p> <ul style="list-style-type: none"> * USART_EIGHT_BIT * USART_NINE_BIT * USART_BIT_MASK <p>Slave/Master Select (Applicable to Synchronous mode only):</p> <ul style="list-style-type: none"> * USART_SYNC_SLAVE * USART_SYNC_MASTER * USART_SYNC_MASK <p>Reception mode:</p> <ul style="list-style-type: none"> * USART_SINGLE_RX * USART_CONT_RX * USART_CONT_RX_MASK <p>Baud rate:</p> <ul style="list-style-type: none"> * USART_BRGH_HIGH * USART_BRGH_LOW * USART_BRGH_MASK <p>Address Detect Enable:</p> <ul style="list-style-type: none"> * USART_ADDEN_ON * USART_ADDEN_OFF * USART_ADDEN_MASK

spbrg	<p>This is the value that is written to the baud rate generator register which determines the baud rate at which the usart operates. The formulas for baud rate are:</p> <p>Asynchronous mode, high speed: $F_{osc} / (16 * (spbrg + 1))$</p> <p>Asynchronous mode, low speed: $F_{osc} / (64 * (spbrg + 1))$</p> <p>Synchronous mode: $F_{osc} / (4 * (spbrg + 1))$</p> <p>Where Fosc is the oscillator frequency</p>
-------	---

Returns
None

Remarks

This function configures the usart module according to the specified configuration options

8.17.1.1.2 Open_2USART

void Open2USART(unsigned char config, unsigned int spbrg)

Configure the specified usart module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Interrupt on Transmission:</p> <ul style="list-style-type: none"> * USART_TX_INT_ON * USART_TX_INT_OFF * USART_TX_INT_MASK <p>Interrupt on Receipt:</p> <ul style="list-style-type: none"> * USART_RX_INT_ON * USART_RX_INT_OFF * USART_RX_INT_MASK <p>usart Mode:</p> <ul style="list-style-type: none"> * USART_ASYNC_MODE * USART_SYNC_MODE * USART_MODE_MASK <p>Transmission Width:</p> <ul style="list-style-type: none"> * USART_EIGHT_BIT * USART_NINE_BIT * USART_BIT_MASK <p>Slave/Master Select (Applicable to Synchronous mode only):</p> <ul style="list-style-type: none"> * USART_SYNC_SLAVE * USART_SYNC_MASTER * USART_SYNC_MASK <p>Reception mode:</p> <ul style="list-style-type: none"> * USART_SINGLE_RX * USART_CONT_RX * USART_CONT_RX_MASK <p>Baud rate:</p> <ul style="list-style-type: none"> * USART_BRGH_HIGH * USART_BRGH_LOW * USART_BRGH_MASK <p>Address Detect Enable:</p> <ul style="list-style-type: none"> * USART_ADDEN_ON * USART_ADDEN_OFF * USART_ADDEN_MASK

spbrg	<p>This is the value that is written to the baud rate generator register which determines the baud rate at which the usart operates. The formulas for baud rate are:</p> <p>Asynchronous mode, high speed: $F_{osc} / (16 * (spbrg + 1))$</p> <p>Asynchronous mode, low speed: $F_{osc} / (64 * (spbrg + 1))$</p> <p>Synchronous mode: $F_{osc} / (4 * (spbrg + 1))$</p> <p>Where F_{osc} is the oscillator frequency</p>
-------	---

Returns

None

Remarks

This function configures the usart module according to the specified configuration options

8.17.1.1.3 Open_USART

void OpenUSART(unsigned char config, unsigned int spbrg)

Configure the specified usart module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	<p>Interrupt on Transmission:</p> <ul style="list-style-type: none"> * USART_TX_INT_ON * USART_TX_INT_OFF * USART_TX_INT_MASK <p>Interrupt on Receipt:</p> <ul style="list-style-type: none"> * USART_RX_INT_ON * USART_RX_INT_OFF * USART_RX_INT_MASK <p>usart Mode:</p> <ul style="list-style-type: none"> * USART_ASYNC_MODE * USART_SYNC_MODE * USART_MODE_MASK <p>Transmission Width:</p> <ul style="list-style-type: none"> * USART_EIGHT_BIT * USART_NINE_BIT * USART_BIT_MASK <p>Slave/Master Select (Applicable to Synchronous mode only):</p> <ul style="list-style-type: none"> * USART_SYNC_SLAVE * USART_SYNC_MASTER * USART_SYNC_MASK <p>Reception mode:</p> <ul style="list-style-type: none"> * USART_SINGLE_RX * USART_CONT_RX * USART_CONT_RX_MASK <p>Baud rate:</p> <ul style="list-style-type: none"> * USART_BRGH_HIGH * USART_BRGH_LOW * USART_BRGH_MASK <p>Address Detect Enable:</p> <ul style="list-style-type: none"> * USART_ADDEN_ON * USART_ADDEN_OFF * USART_ADDEN_MASK

spbrg	<p>This is the value that is written to the baud rate generator register which determines the baud rate at which the usart operates. The formulas for baud rate are:</p> <p>Asynchronous mode, high speed: $F_{osc} / (16 * (spbrg + 1))$</p> <p>Asynchronous mode, low speed: $F_{osc} / (64 * (spbrg + 1))$</p> <p>Synchronous mode: $F_{osc} / (4 * (spbrg + 1))$</p> <p>Where Fosc is the oscillator frequency</p>
-------	---

Returns
None

Remarks

This function configures the usart module according to the specified configuration options

8.17.1.2 USART_Write

8.17.1.2.1 Write_1USART

void Write1USART(char data)

Write a byte (one character) to the usart transmit buffer, including the 9th bit if enabled

Input Parameters

Input Parameters	Description
data	The value to be written to the usart

Returns
None

Remarks

This function writes a byte to the usart transmit buffer. If 9-bit mode is enabled, the 9th bit is written from the field TX_NINE, found in a union of type USART1:

8.17.1.2.2 Write_2USART

```
void Write2USART(char data)
```

Write a byte (one character) to the usart transmit buffer, including the 9th bit if enabled

Input Parameters

Input Parameters	Description
data	The value to be written to the usart

Returns

None

Remarks

This function writes a byte to the usart transmit buffer. If 9-bit mode is enabled, the 9th bit is written from the field TX_NINE, found in a union of type USART2:

8.17.1.2.3 Write_USART

```
void WriteUSART(char data)
```

Write a byte (one character) to the usart transmit buffer, including the 9th bit if enabled

Input Parameters

Input Parameters	Description
data	The value to be written to the usart

Returns

None

Remarks

This function writes a byte to the usart transmit buffer. If 9-bit mode is enabled, the 9th bit is written from the field TX_NINE, found in a union of type USART:

8.17.1.3 USART_baud

8.17.1.3.1 baud_1USART

```
void baud1USART (unsigned char baudconfig)
```

Set the baud rate configuration bits for enhanced usart operation

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
baudconfig	RX Idle State: In Asynchronous mode: * BAUD_IDLE_RX_PIN_STATE_HIGH * BAUD_IDLE_RX_PIN_STATE_LOW * BAUD_IDLE_TX_RX_PIN_STATE_MASK In Synchronous mode: * BAUD_IDLE_RX_PIN_STATE_HIGH * BAUD_IDLE_RX_PIN_STATE_LOW * BAUD_IDLE_TX_PIN_STATE_HIGH * BAUD_IDLE_TX_PIN_STATE_LOW * BAUD_IDLE_TX_RX_PIN_STATE_MASK Clock Idle State: (In Synchronous mode) * BAUD_IDLE_CLK_HIGH * BAUD_IDLE_CLK_LOW * BAUD_IDLE_CLK_MASK Baud Rate Generation: * BAUD_16_BIT_RATE * BAUD_8_BIT_RATE * BAUD_BIT_RATE RX Pin Monitoring: * BAUD_WAKEUP_ON * BAUD_WAKEUP_OFF * BAUD_WAKEUP_MASK Baud Rate Measurement: * BAUD_AUTO_ON * BAUD_AUTO_OFF * BAUD_AUTO_MASK

Returns

None

Remarks

These functions are only available for processors with enhanced usart capability

8.17.1.3.2 baud_2USART

void baud2USART (unsigned char baudconfig)

Set the baud rate configuration bits for enhanced usart operation

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
baudconfig	<div>RX Idle State: In Asynchronous mode: * BAUD_IDLE_RX_PIN_STATE_HIGH * BAUD_IDLE_RX_PIN_STATE_LOW * BAUD_IDLE_TX_RX_PIN_STATE_MASK In Synchronous mode: * BAUD_IDLE_RX_PIN_STATE_HIGH * BAUD_IDLE_RX_PIN_STATE_LOW * BAUD_IDLE_TX_PIN_STATE_HIGH * BAUD_IDLE_TX_PIN_STATE_LOW * BAUD_IDLE_TX_RX_PIN_STATE_MASK Clock Idle State: (In Synchronous mode) * BAUD_IDLE_CLK_HIGH * BAUD_IDLE_CLK_LOW * BAUD_IDLE_CLK_MASK Baud Rate Generation: * BAUD_16_BIT_RATE * BAUD_8_BIT_RATE * BAUD_BIT_RATE RX Pin Monitoring: * BAUD_WAKEUP_ON * BAUD_WAKEUP_OFF * BAUD_WAKEUP_MASK Baud Rate Measurement: * BAUD_AUTO_ON * BAUD_AUTO_OFF * BAUD_AUTO_MASK</div>

Returns

None

Remarks

These functions are only available for processors with enhanced usart capability

8.17.1.3.3 baud_USART

void baudUSART (unsigned char baudconfig)

Set the baud rate configuration bits for enhanced usart operation

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
baudconfig	<div>RX Idle State:</div> <div>In Asynchronous mode:</div> <div>* BAUD_IDLE_RX_PIN_STATE_HIGH</div> <div>* BAUD_IDLE_RX_PIN_STATE_LOW</div> <div>* BAUD_IDLE_TX_RX_PIN_STATE_MASK</div> <div>In Synchronous mode:</div> <div>* BAUD_IDLE_RX_PIN_STATE_HIGH</div> <div>* BAUD_IDLE_RX_PIN_STATE_LOW</div> <div>* BAUD_IDLE_TX_PIN_STATE_HIGH</div> <div>* BAUD_IDLE_TX_PIN_STATE_LOW</div> <div>* BAUD_IDLE_TX_RX_PIN_STATE_MASK</div> <div>Clock Idle State: (In Synchronous mode)</div> <div>* BAUD_IDLE_CLK_HIGH</div> <div>* BAUD_IDLE_CLK_LOW</div> <div>* BAUD_IDLE_CLK_MASK</div> <div>Baud Rate Generation:</div> <div>* BAUD_16_BIT_RATE</div> <div>* BAUD_8_BIT_RATE</div> <div>* BAUD_BIT_RATE</div> <div>RX Pin Monitoring:</div> <div>* BAUD_WAKEUP_ON</div> <div>* BAUD_WAKEUP_OFF</div> <div>* BAUD_WAKEUP_MASK</div> <div>Baud Rate Measurement:</div> <div>* BAUD_AUTO_ON</div> <div>* BAUD_AUTO_OFF</div> <div>* BAUD_AUTO_MASK</div>

Returns

None

Remarks

These functions are only available for processors with enhanced usart capability

8.17.1.4 USART_gets

8.17.1.4.1 gets_1USART

void gets1USART(char *buffer, unsigned char len)

Read a fixed-length string of characters from the specified usart

Input Parameters

Input Parameters	Description
buffer	A pointer to the location in RAM where incoming characters are to be stored
len	The number of characters to read from the usart

Returns

Read location in RAM will be updated with the data received

Remarks

This function only works in 8-bit transmit/receive mode. This function waits for and reads len number of characters out of the specified usart. There is no time out when waiting for characters to arrive

8.17.1.4.2 gets_2USART

void gets2USART(char *buffer, unsigned char len)

Read a fixed-length string of characters from the specified usart

Input Parameters

Input Parameters	Description
buffer	A pointer to the location in RAM where incoming characters are to be stored
len	The number of characters to read from the usart

Returns

Read location in RAM will be updated with the data received

Remarks

This function only works in 8-bit transmit/receive mode. This function waits for and reads len number of characters out of the specified usart. There is no time out when waiting for characters to arrive

8.17.1.4.3 gets_USART

```
void getsUSART(char *buffer, unsigned char len)
```

Read a fixed-length string of characters from the specified usart

Input Parameters

Input Parameters	Description
buffer	A pointer to the location in RAM where incoming characters are to be stored
len	The number of characters to read from the usart

Returns

Read location in RAM will be updated with the data received

Remarks

This function only works in 8-bit transmit/receive mode. This function waits for and reads len number of characters out of the specified usart. There is no time out when waiting for characters to arrive

8.17.1.5 USART_putrs

8.17.1.5.1 putrs_1USART

```
void putrs1USART(const rom char *data)
```

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in data memory should be used with the “puts” versions of these functions.

8.17.1.5.2 puts_2USART

void puts2USART(const rom char *data)

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in data memory should be used with the “puts” versions of these functions.

8.17.1.5.3 puts_USART

void putsUSART(const rom char *data)

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in data memory should be used with the “puts” versions of these functions.

8.17.1.6 USART_puts

8.17.1.6.1 puts_1USART

```
void puts1USART( char *data)
```

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in program memory, including string literals, should be used with the “puts” versions of these functions

8.17.1.6.2 puts_2USART

```
void puts2USART( char *data)
```

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in program memory, including string literals, should be used with the “puts” versions of these functions

8.17.1.6.3 puts_USART

```
void putsUSART( char *data)
```

Writes a string of characters to the usart including the null character from a location in Program Memory

Input Parameters

Input Parameters	Description
data	Pointer to a null-terminated string of data in Program Memory

Returns

None

Remarks

This function only works in 8-bit transmit/receive mode. This function writes a string of data to the usart including the null character.

Note : Strings located in program memory, including string literals, should be used with the “puts” versions of these functions

8.17.1.7 USART_Read

8.17.1.7.1 Read_1USART

```
char Read1USART(void)
```

Read a byte (one character) out of the usart receive buffer, including the 9th bit if enabled

Returns

Returns a byte of data read from usart buffer. The 9th bit are saved in a union USART1

Remarks

This function reads a byte out of the usart receive buffer. The Status bits and the 9th data bits are saved in a union USART1
The 9th bit is read-only if 9-bit mode is enabled. The Status bits are however read always.

8.17.1.7.2 Read_2USART

```
char Read2USART(void)
```

Read a byte (one character) out of the usart receive buffer, including the 9th bit if enabled

Returns

Returns a byte of data read from usart buffer. The 9th bit are saved in a union USART2

Remarks

This function reads a byte out of the usart receive buffer. The Status bits and the 9th data bits are saved in a union USART2
The 9th bit is read-only if 9-bit mode is enabled. The Status bits are however read always.

8.17.1.7.3 Read_USART

char ReadUSART(void)

Read a byte (one character) out of the usart receive buffer, including the 9th bit if enabled

Returns

Returns a byte of data read from usart buffer. The 9th bit are saved in a union USART

Remarks

This function reads a byte out of the usart receive buffer. The Status bits and the 9th data bits are saved in a union USART
The 9th bit is read-only if 9-bit mode is enabled. The Status bits are however read always.

8.17.2 USART Structs,Records,Enums

8.17.3 USART Macros

8.17.4 USART Examples

8.17.4.1 USART Example1

Example Source Code demonstrating peripheral library usage

```
/* *****  
* NOTES:  
* Code uses the Peripheral library support available with MCC18 Compiler  
* Code Tested on:  
* PicDem2+ demo board with PIC18F4685 controller  
*  
* Uses Tx pin for transmission and Rx pin for reception.  
* Baud rate of 2400 is configured at 8MHz oscillator frequency  
***** */
```

```

*****/

#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "usart.h"

unsigned char Rxdata[25];
unsigned char Txdata[] = "MICROCHIP_USART";

void main(void)
{
    unsigned char config=0,spbrg=0,baudconfig=0,i=0;

    CloseUSART(); //turn off usart if was previously on

    //-----configure USART -----
    config = USART_TX_INT_OFF | USART_RX_INT_OFF | USART_ASYNC_MODE | USART_EIGHT_BIT |
    USART_CONT_RX | USART_BRGH_LOW;
    //-----SPBRG needs to be changed depending upon oscillator frequency-----
    spbrg = 51; //At 8Mhz of oscillator frequency & baud rate of 2400.

    OpenUSART(config, spbrg); //API configures USART for desired parameters

    baudconfig = BAUD_8_BIT_RATE | BAUD_AUTO_OFF;
    baudUSART (baudconfig);

    //-----USART Transmission ----
    while(BusyUSART()); //Check if Usart is busy or not
    putsUSART((char *)Txdata); //transmit the string

    //---USART Reception ---
    getsUSART((char *)Rxdata,24); //Recieve data upto 24 bytes

    while(BusyUSART()); //Check if Usart is busy or not
    putsUSART((char *)Rxdata); //echo back the data recieved back to host

    CloseUSART();
    while(1); //end of program
}

```

8.18 MicroWire (MWIRE)

Microwire, is a restricted subset of SPI. MWIRE module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shiftregisters, display drivers, A/D Converters, etc. The MWIRE module is compatible with Motorola's SPI and SIOP interfaces. The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer.

8.18.1 MWIRE Functions

8.18.1.1 Mwire_Open

8.18.1.1.1 Open_Mwire1

void OpenMwire1(unsigned char sync_mode)

Configure the SSP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	MWIRE Mode Selection: * MWIRE_FOSC_4 * MWIRE_FOSC_16 * MWIRE_FOSC_64 * MWIRE_FOSC_TMR2

Returns
None

Remarks

Function resets the SSP module to the POR state and then configures the module for Microwire communications

8.18.1.1.2 Open_Mwire2

void OpenMwire2(unsigned char sync_mode)

Configure the SSP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	MWIRE Mode Selection: * MWIRE_FOSC_4 * MWIRE_FOSC_16 * MWIRE_FOSC_64 * MWIRE_FOSC_TMR2

Returns
None

Remarks

Function resets the SSP module to the POR state and then configures the module for Microwire communications

8.18.1.1.3 Open_Mwire

```
void OpenMwire( unsigned char sync_mode )
```

Configure the SSP module

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
sync_mode	MWIRE Mode Selection: * MWIRE_FOSC_4 * MWIRE_FOSC_16 * MWIRE_FOSC_64 * MWIRE_FOSC_TMR2

Returns

None

Remarks

Function resets the SSP module to the POR state and then configures the module for Microwire communications

8.18.1.2 Mwire_gets

8.18.1.2.1 gets_Mwire1

```
void getsMwire1( unsigned char *rdptr, unsigned char length )
```

Read a string from the Microwire device

Input Parameters

Input Parameters	Description
rdptr	Pointer to PICmicro MCU RAM for placement of data read from Microwire device
length	Number of bytes to read from Microwire device

Returns

Updates the location in RAM with the data read from MicroWire Device

Remarks

This function is used to read a predetermined length of data from a Microwire device. Before using this function, a Read command with the appropriate address must be issued

8.18.1.2.2 gets_Mwire2

```
void getsMwire2( unsigned char *rdptr, unsigned char length )
```

Read a string from the Microwire device

Input Parameters

Input Parameters	Description
rdptr	Pointer to PICmicro MCU RAM for placement of data read from Microwire device
length	Number of bytes to read from Microwire device

Returns

Updates the location in RAM with the data read from MicroWire Device

Remarks

This function is used to read a predetermined length of data from a Microwire device. Before using this function, a Read command with the appropriate address must be issued

8.18.1.2.3 gets_Mwire

```
void getsMwire( unsigned char *rdptr, unsigned char length )
```

Read a string from the Microwire device

Input Parameters

Input Parameters	Description
rdptr	Pointer to PICmicro MCU RAM for placement of data read from Microwire device
length	Number of bytes to read from Microwire device

Returns

Updates the location in RAM with the data read from MicroWire Device

Remarks

This function is used to read a predetermined length of data from a Microwire device. Before using this function, a Read command with the appropriate address must be issued

8.18.1.3 Mwire_Write

8.18.1.3.1 Write_Mwire1

```
unsigned char WriteMwire1( unsigned char data_out )
```

This function is used to write out a single data byte (one character)

Input Parameters

Input Parameters	Description
data_out	Single byte of data to write to Microwire device

Returns

None

Remarks

This function writes out single data byte to a Microwire device utilizing the SSP module

putcMwire1 is an alternative name for WriteMwire1

8.18.1.3.2 Write_Mwire2

```
unsigned char WriteMwire2( unsigned char data_out )
```

This function is used to write out a single data byte (one character)

Input Parameters

Input Parameters	Description
data_out	Single byte of data to write to Microwire device

Returns

None

Remarks

This function writes out single data byte to a Microwire device utilizing the SSP module

putcMwire2 is an alternative name for WriteMwire2

8.18.1.3.3 Write_Mwire

```
unsigned char WriteMwire( unsigned char data_out )
```

This function is used to write out a single data byte (one character)

Input Parameters

Input Parameters	Description
data_out	Single byte of data to write to Microwire device

Returns

None

Remarks

This function writes out single data byte to a Microwire device utilizing the SSP module

putcMwire is an alternative name for WriteMwire

8.18.1.4 Mwire_Read

8.18.1.4.1 Read_Mwire1

```
unsigned char ReadMwire1( unsigned char high_byte, unsigned char low_byte )
```

Read a byte from a Microwire device

Input Parameters

Input Parameters	Description
high_byte	First byte of 16-bit instruction word
low_byte	Second byte of 16-bit instruction word

Returns

Returns a byte of data read from MicroWire device

Remarks

This function reads in a single byte from a Microwire device. The Start bit, opcode and address compose the high and low bytes passed into this function

getcMwire1 is an alternative name for ReadMwire1

8.18.1.4.2 Read_Mwire2

```
unsigned char ReadMwire2( unsigned char high_byte, unsigned char low_byte )
```

Read a byte from a Microwire device

Input Parameters

Input Parameters	Description
high_byte	First byte of 16-bit instruction word
low_byte	Second byte of 16-bit instruction word

Returns

Returns a byte of data read from MicroWire device

Remarks

This function reads in a single byte from a Microwire device. The Start bit, opcode and address compose the high and low bytes passed into this function

getcMwire2 is an alternative name for ReadMwire2

8.18.1.4.3 Read_Mwire

```
unsigned char ReadMwire( unsigned char high_byte, unsigned char low_byte )
```

Read a byte from a Microwire device

Input Parameters

Input Parameters	Description
high_byte	First byte of 16-bit instruction word
low_byte	Second byte of 16-bit instruction word

Returns

Returns a byte of data read from MicroWire device

Remarks

This function reads in a single byte from a Microwire device. The Start bit, opcode and address compose the high and low bytes passed into this function

getcMwire is an alternative name for ReadMwire

8.18.2 MWIRE Macros

8.18.3 MWIRE Examples

8.18.3.1 MWIRE Example1

Example Source Code demonstrating peripheral library usage

//The following is a simple code example illustrating the SSP module communicating with a Microchip 93LC66 Microwire EE memory device.

```
#include "p18cxxx.h"

#include "mwire.h"

// 93LC66 x 8

// FUNCTION Prototypes

void main(void);
void ew_enable(void);
void erase_all(void);
void busy_poll(void);
void write_all(unsigned char data);
void byte_read(unsigned char address);
void read_mult(unsigned char address,
               unsigned char *rdptr,
               unsigned char length);
void write_byte(unsigned char address,
               unsigned char data);

// VARIABLE Definitions
unsigned char arrayrd[20];
unsigned char var;

// DEFINE 93LC66 MACROS -- see datasheet for details
#define READ 0x0C
#define WRITE 0x0A
#define ERASE 0x0E
#define EWEN1 0x09
#define EWEN2 0x80
#define ERAL1 0x09
#define ERAL2 0x00
#define WRAL1 0x08
#define WRAL2 0x80
#define EWDS1 0x08
#define EWDS2 0x00
#define W_CS LATCbits.LATC2

void main(void)
{
    TRISCbits.TRISC2 = 0;
    W_CS = 0;           //ensure CS is negated
    OpenMWire(MWIRE_FOSC_16); //enable SSP peripheral
    ew_enable();        //send erase/write enable
    write_byte(0x13, 0x34); //write byte (address, data)
    busy_poll();
    Nop();
    byte_read(0x13);    //read single byte (address)
    read_mult(0x10, arrayrd, 10); //read multiple bytes
    erase_all();        //erase entire array
}
```

```

    CloseMWire();                //disable SSP peripheral
}

void ew_enable(void)
{
    W_CS = 1;                    //assert chip select
    putcMWire(EWEN1);            //enable write command byte 1
    putcMWire(EWEN2);            //enable write command byte 2
    W_CS = 0;                    //negate chip select
}

void busy_poll(void)
{
    W_CS = 1;
    while(! DataRdyMWire() );
    W_CS = 0;
}

void write_byte(unsigned char address,
                unsigned char data)
{
    W_CS = 1;
    putcMWire(WRITE);            //write command
    putcMWire(address);          //address
    putcMWire(data);             //write single byte
    W_CS = 0;
}

void byte_read(unsigned char address)
{
    W_CS = 1;
    getcMWire(READ,address);     //read one byte
    W_CS = 0;
}

void read_mult(unsigned char address,
               unsigned char *rdptr,
               unsigned char length)
{
    W_CS = 1;
    putcMWire(READ);            //read command
    putcMWire(address);          //address (A7 - A0)
    getsMWire(rdptr, length);    //read multiple bytes
    W_CS = 0;
}

void erase_all(void)
{
    W_CS = 1;
    putcMWire(ERAL1);            //erase all command byte 1
    putcMWire(ERAL2);            //erase all command byte 2
    W_CS = 0;
}

```

8.19 Power Control PWM (PCPWM)

The Power Control PWM module simplifies the task of generating multiple, synchronized Pulse-Width Modulated (PWM) outputs for use in the control of motor controllers and power conversion applications.

The PWM module has the following features:

- Up to eight PWM I/O pins with four duty cycle generators. Pins can be paired to get a complete half-bridge control.
- Up to 14-bit resolution, depending upon the PWM period.

- On-the-fly PWM frequency changes.
- Edge and Center-Aligned Output modes.
- Single-Pulse Generation mode.
- Programmable dead-time control between paired PWMs.
- Interrupt support for asymmetrical updates in Center-Aligned mode.
- Output override for Electrically Commutated Motor (ECM) operation; for example, BLDC.
- Special Event Trigger comparator for scheduling other peripheral events.
- PWM outputs disable feature sets PWM outputs to their inactive state when in Debug mode

8.19.1 PCPWM Functions

8.19.1.1 Open_pcpwm_Page1

```
void Openpcpwm(unsigned char config0,unsigned char config1,unsigned char config2,unsigned char config3,unsigned int period,unsigned int sptime)
```

Configure the PCPWM module for period, special event time, pwm mode

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config0	<p>configures PWMCON0:</p> <p>PWM Module Enable:</p> <ul style="list-style-type: none">* PWM_IO_ALL_ODD* PWM_IO_1AND3* PWM_IO_ALL* PWM_IO_0TO5* PWM_IO_0TO3* PWM_IO_0AND1* PWM_IO_1* PWM_DISABLE* PWM_IO_MASK <p>PWM Output Pair Mode:</p> <p>PMOD0:</p> <ul style="list-style-type: none">* PWM_0AND1_INDPEN* PWM_0AND1_COMPLI* PWM_0AND1_MASK <p>PMOD1:</p> <ul style="list-style-type: none">* PWM_2AND3_INDPEN* PWM_2AND3_COMPLI* PWM_2AND3_MASK <p>PMOD2:</p> <ul style="list-style-type: none">* PWM_4AND5_INDPEN* PWM_4AND5_COMPLI* PWM_4AND5_MASK

config1	<p>configures PWMCON1:</p> <p>PWM Special Event Trigger :</p> <ul style="list-style-type: none">* PW_SEVT_POS_1_1* PW_SEVT_POS_1_2* PW_SEVT_POS_1_3* PW_SEVT_POS_1_4* PW_SEVT_POS_1_5* PW_SEVT_POS_1_6* PW_SEVT_POS_1_7* PW_SEVT_POS_1_8* PW_SEVT_POS_1_9* PW_SEVT_POS_1_10* PW_SEVT_POS_1_11* PW_SEVT_POS_1_12* PW_SEVT_POS_1_13* PW_SEVT_POS_1_14* PW_SEVT_POS_1_15* PW_SEVT_POS_1_16* PW_SEVT_POS_MASK <p>PWM Special Event Count Direction:</p> <ul style="list-style-type: none">* PW_SEVT_DIR_UP* PW_SEVT_DIR_DWN* PW_SEVT_DIR_MASK <p>PWM Output Override Synchronization:</p> <ul style="list-style-type: none">* PW_OP_SYNC* PW_OP_ASYNC* PW_OP_SYNC_MASK
---------	--

config2	<p>configures PTCON0:</p> <p>PWM Time Base Output Postscale Select:</p> <ul style="list-style-type: none"> * PT_POS_1_1 * PT_POS_1_2 * PT_POS_1_3 * PT_POS_1_4 * PT_POS_1_5 * PT_POS_1_6 * PT_POS_1_7 * PT_POS_1_8 * PT_POS_1_9 * PT_POS_1_10 * PT_POS_1_11 * PT_POS_1_12 * PT_POS_1_13 * PT_POS_1_14 * PT_POS_1_15 * PT_POS_1_16 * PT_POS_MASK <p>PWM Time Base Input Clock Prescale Select:</p> <ul style="list-style-type: none"> * PT_PRS_1_1 * PT_PRS_1_4 * PT_PRS_1_16 * PT_PRS_1_64 * PT_PRS_MASK <p>PWM Time Base Mode Select:</p> <ul style="list-style-type: none"> * PT_MOD_CNT_UPDN_INT * PT_MOD_CNT_UPDN * PT_MOD_SINGL_SHOT * PT_MOD_FREE_RUN * PT_MOD_MASK
config3	<p>configures PTCON1:</p> <p>PWM Time Base Timer Enable/Disable :</p> <ul style="list-style-type: none"> * PT_ENABLE * PT_DISABLE * PT_MASK <p>PWM Time Base Count Direction Status :</p> <ul style="list-style-type: none"> * PT_CNT_UP * PT_CNT_DWN * PT_CNT_MASK
period	<p>configures PTPERL & PTPERH:</p> <p>16 Bit value loaded to PWM Time Base Period register</p>

sptime	configures SEVTCMPL & SEVTCMPH: 16 bit value loaded to PWM Special Event Trigger Compare Registers
--------	---

Returns
None

Remarks

None

8.19.1.2 Open_pcpwm_Page2

```
void Openpcpwm(unsigned char config0,unsigned char config1,unsigned char config2,unsigned char config3,unsigned int period,unsigned int sptime)
```

Configure the PCPWM module for period, special event time, pwm mode

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config0	<p>configures PWMCON0:</p> <p>PWM Module Enable:</p> <ul style="list-style-type: none">* PWM_IO_ALL_ODD* PWM_IO_1AND3* PWM_IO_ALL* PWM_IO_0TO5* PWM_IO_0TO3* PWM_IO_0AND1* PWM_IO_1* PWM_DISABLE* PWM_IO_MASK <p>PWM Output Pair Mode:</p> <p>PMOD0:</p> <ul style="list-style-type: none">* PWM_0AND1_INDPEN* PWM_0AND1_COMPLI* PWM_0AND1_MASK <p>PMOD1:</p> <ul style="list-style-type: none">* PWM_2AND3_INDPEN* PWM_2AND3_COMPLI* PWM_2AND3_MASK <p>PMOD2:</p> <ul style="list-style-type: none">* PWM_4AND5_INDPEN* PWM_4AND5_COMPLI* PWM_4AND5_MASK <p>PMOD3:</p> <ul style="list-style-type: none">* PWM_6AND7_INDPEN* PWM_6AND7_COMPLI* PWM_6AND7_MASK

config1	<div>configures PWMCON1:</div> <div>PWM Special Event Trigger :</div> <div><div>* PW_SEVT_POS_1_1</div><div>* PW_SEVT_POS_1_2</div><div>* PW_SEVT_POS_1_3</div><div>* PW_SEVT_POS_1_4</div><div>* PW_SEVT_POS_1_5</div><div>* PW_SEVT_POS_1_6</div><div>* PW_SEVT_POS_1_7</div><div>* PW_SEVT_POS_1_8</div><div>* PW_SEVT_POS_1_9</div><div>* PW_SEVT_POS_1_10</div><div>* PW_SEVT_POS_1_11</div><div>* PW_SEVT_POS_1_12</div><div>* PW_SEVT_POS_1_13</div><div>* PW_SEVT_POS_1_14</div><div>* PW_SEVT_POS_1_15</div><div>* PW_SEVT_POS_1_16</div><div>* PW_SEVT_POS_MASK</div></div> <div>PWM Special Event Count Direction:</div> <div><div>* PW_SEVT_DIR_UP</div><div>* PW_SEVT_DIR_DWN</div><div>* PW_SEVT_DIR_MASK</div></div> <div>PWM Output Override Synchronization:</div> <div><div>* PW_OP_SYNC</div><div>* PW_OP_ASYNC</div><div>* PW_OP_SYNC_MASK</div></div>
---------	--

config2	<p>configures PTCON0:</p> <p>PWM Time Base Output Postscale Select:</p> <ul style="list-style-type: none"> * PT_POS_1_1 * PT_POS_1_2 * PT_POS_1_3 * PT_POS_1_4 * PT_POS_1_5 * PT_POS_1_6 * PT_POS_1_7 * PT_POS_1_8 * PT_POS_1_9 * PT_POS_1_10 * PT_POS_1_11 * PT_POS_1_12 * PT_POS_1_13 * PT_POS_1_14 * PT_POS_1_15 * PT_POS_1_16 * PT_POS_MASK <p>PWM Time Base Input Clock Prescale Select:</p> <ul style="list-style-type: none"> * PT_PRS_1_1 * PT_PRS_1_4 * PT_PRS_1_16 * PT_PRS_1_64 * PT_PRS_MASK <p>PWM Time Base Mode Select:</p> <ul style="list-style-type: none"> * PT_MOD_CNT_UPDN_INT * PT_MOD_CNT_UPDN * PT_MOD_SINGL_SHOT * PT_MOD_FREE_RUN * PT_MOD_MASK
config3	<p>configures PTCON1:</p> <p>PWM Time Base Timer Enable/Disable :</p> <ul style="list-style-type: none"> * PT_ENABLE * PT_DISABLE * PT_MASK <p>PWM Time Base Count Direction Status :</p> <ul style="list-style-type: none"> * PT_CNT_UP * PT_CNT_DWN * PT_CNT_MASK
period	<p>configures PTPERL & PTPERH:</p> <p>16 Bit value loaded to PWM Time Base Period register</p>

sptime	configures SEVTCMPL & SEVTCMPH: 16 bit value loaded to PWM Special Event Trigger Compare Registers
--------	---

Returns
None

Remarks

None

8.19.1.3 Setdc0_pcpwm

void Setdc0pcpwm(unsigned int dutycycle)

This function configures the duty cycle

Input Parameters

Input Parameters	Description
dutycycle	Integer value that will be loaded into Duty cycle register

Returns
None

Remarks

This function configures the duty cycle by loading the argument into the PWM DUTY CYCLE REGISTERS (PDC0L and PDC0H). The value in these registers amounts for the time during which the PWM remains high

8.19.1.4 Setdc1_pcpwm

void Setdc1pcpwm(unsigned int dutycycle)

This function configures the duty cycle

Input Parameters

Input Parameters	Description
dutycycle	Integer value that will be loaded into Duty cycle register

Returns
None

Remarks

This function configures the duty cycle by loading the argument into the PWM DUTY CYCLE REGISTERS (PDC1L and PDC1H). The value in these registers amounts for the time during which the PWM remains high

8.19.1.5 Setdc2_pcpwm

```
void Setdc2pcpwm(unsigned int dutycycle)
```

This function configures the duty cycle

Input Parameters

Input Parameters	Description
dutycycle	Integer value that will be loaded into Duty cycle register

Returns

None

Remarks

This function configures the duty cycle by loading the argument into the PWM DUTY CYCLE REGISTERS (PDC2L and PDC2H). The value in these registers amounts for the time during which the PWM remains high

8.19.1.6 Setdc3_pcpwm

```
void Setdc3pcpwm(unsigned int dutycycle)
```

This function configures the duty cycle

Input Parameters

Input Parameters	Description
dutycycle	Integer value that will be loaded into Duty cycle register

Returns

None

Remarks

This function configures the duty cycle by loading the argument into the PWM DUTY CYCLE REGISTERS (PDC3L and PDC3H). The value in these registers amounts for the time during which the PWM remains high

8.19.1.7 OVD_CTRL_pcpwm

```
void pcpwm_OVD_CTRL(unsigned char config)
```

This function configures for channel Override

Input Parameters

Input Parameters	Description
config	(configures OVDCOND) Values of OVDCON Bit<0-7>: 1 :- PWM Output on PCPWM pin is controlled by Value in Duty cycle register. 0 :- PWM output on pin is controlled by POUT value in OVDCONS register

Returns

None

Remarks

Configures OVDCOND register to override PWM output on pin

8.19.1.8 OVD_IO_STA_pcpwm

```
void pcpwm_OVD_IO_STA(unsigned char config)
```

This function configures for channel Override

Input Parameters

Input Parameters	Description
config	(configures OVDCONS) Values of OVDCONS Bit<0-7>: 1 :- PWM Output on PCPWM pin is active when the corresponding PWM output override bit is cleared. 0 :- PWM Output on PCPWM pin is inactive when the corresponding PWM output override bit is cleared

Returns

None

Remarks

Configures OVDCONS register to override PWM output on pin

8.19.1.9 dt_clk_source_pcpwm

```
void pcpwm_dt_clk_source(unsigned char config)
```

This function selects the clock source (prescaler) for Dead time

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description
config	(configures DTCON) Dead-Time Unit A Prescale Select : * DT_CLK_SRC_FOSC_16 * DT_CLK_SRC_FOSC_8 * DT_CLK_SRC_FOSC_4 * DT_CLK_SRC_FOSC_2 * DT_CLK_SRC_MASK

Returns

None

Remarks

Function configures the clock prescaler for Dead time unit

8.19.1.10 dt_assignment_pcpwm

```
void pcpwm_dt_assignment(unsigned char config)
```

This function updates the dead time

Input Parameters

Input Parameters	Description
config	(configures DTCON) Unsigned 6-Bit value (Dead-Time Value) for Dead-Time Unit bits

Returns

None

Remarks

Function configures the Dead time by loading the unsigned 6 bit value into DTCON register

8.19.1.11 Close_pcpwm

void Closepcpwm(void)

This function turns off the PCPWM module and sets the ports as input ports

Returns

None

Remarks

Function disables the interrupts and clears the interrupt flags.

8.19.2 PCPWM Macros

8.20 Parallel Master Port (PMP)

The Parallel Master Port (PMP) module is a parallel 8-bit I/O module, specifically designed to communicate with a wide variety of parallel devices, such as communication peripherals, LCDs, external memory devices and microcontrollers. Because the interface to parallel peripherals varies significantly, the PMP is highly configurable. Key features of the PMP module include:

- Up to 16 Programmable Address Lines
- Up to 2 Chip Select Lines
- Programmable Strobe Options:
 - Individual Read and Write Strobes or;
 - Read/Write Strobe with Enable Strobe
- Address Auto-Increment/Auto-Decrement
- Programmable Address/Data Multiplexing
- Programmable Polarity on Control Signals
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support:

- Address Support
- 4-Byte Deep Auto-Incrementing Buffer
- Programmable Wait States
- Selectable Input Voltage Levels

8.20.1 PMP Functions

8.20.1.1 Open_PMP

```
void PMPOpen(UINT control, UINT mode, UINT port, UINT addr, BYTE interrupt)
```

Provides method for setting PMP registers using bit masks provided in this header file.

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing bitwise OR operation ('|'), configurable as shown in the examples

Input Parameters	Description
------------------	-------------

8.20.1.2 SetAddress_PMP

```
void PMPSetAddress(WORD address)
```

This function sets the PMP port address

Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing bitwise OR operation ('|'), configurable as shown in the examples

Input Parameters	Description
address	<div>This contains the parameters to be configured in the PMADDR register as defined below</div> <div>Chip Select</div> <div><ul style="list-style-type: none">• BIT_CS2_ON• BIT_CS2_OFF• BIT_CS1_ON• BIT_CS1_OFF</div> <div>Destination Address</div> <div><ul style="list-style-type: none">• BIT_A13• BIT_A12• BIT_A11• BIT_A10• BIT_A9• BIT_A8• BIT_A7• BIT_A6• BIT_A5• BIT_A4• BIT_A3• BIT_A2• BIT_A1• BIT_A0</div>

Returns
None

Remarks

Use in Master mode 1 or 2, MODE[1:0] = 10, 11

8.20.1.3 MasterRead_PMP

WORD PMPMasterRead(void)

This function reads the data from PMP data lines latched onto buffer

Returns
8-bit value read from external device

Remarks

Use in Master mode 1 or 2, MODE[1:0] = 10, 11

8.20.1.4 MasterWrite_PMP

void PMPMasterWrite(WORD value)

If 8-bit data mode is selected, the data appears on 8 data lines. If 16-bit data mode, the lower 8 bits of data are written first, followed by the upper 8 bits of data.

Input Parameters

Input Parameters	Description
value	to write to external device

Returns
None

Remarks

Use in Master mode 1 or 2, MODE[1:0] = 10, 11

8.20.1.5 IsBufferNEmpty_PMP

BOOL PMPIsBufferNEmpty(BUFFER buf)

Returns state of PMSTAT.OBnE (output buffer(s) empty bit)

Input Parameters

Input Parameters	Description
buf	buffer(0..3)

Returns
TRUE/FALSE

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 or SLAVE ENHANCED mode, MODE[1:0] = 00 and INCM[1:0]=11

8.20.1.6 IsBufferNFull_PMP

BOOL PMPIsBufferNFull(BUFFER buf)

Returns state of PMSTAT.IBxF (input buffer(s) full bit)

Input Parameters

Input Parameters	Description
buf	buffer(0..3)

Returns

TRUE/FALSE

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 or SLAVE ENHANCED mode, MODE[1:0] = 00 and INCM[1:0]=11

8.20.1.7 SlaveReadBufferN_PMP

```
unsigned char PMPSlaveReadBufferN(BUFFER buf)
```

Reads the value in PMDATA register written by a master device.

Input Parameters

Input Parameters	Description
buf	buffer(0..3)

Returns

The value in selected buffer.

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 and INCM[1:0]=11 or SLAVE ENHANCED mode, MODE[1:0] = 01

8.20.1.8 SlaveReadBuffers_PMP

```
void PMPSlaveReadBuffers(BYTE* ref)
```

Copies 4 bytes from DATAIN buffers to a starting location pointed to by input parameter.

Input Parameters

Input Parameters	Description
ref	BYTE pointer

Returns

The contents of the 4 8-bit slave buffer registers.

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 and INCM[1:0]=11 or SLAVE ENHANCED mode, MODE[1:0] = 01

8.20.1.9 SlaveWriteBufferN_PMP

BOOL PMPSlaveWriteBufferN(BUFFER buf, BYTE value)

writes the desired value into the selected output buffer

Input Parameters

Input Parameters	Description
buf	buffer(0..3)
value	value to be written

Returns

Returns the state of PMSTAT.OBE prior to the write operation.

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 or SLAVE ENHANCED mode, MODE[1:0] = 00 and INCM[1:0]=11

8.20.1.10 SlaveWriteBuffers_PMP

BOOL PMPSlaveWriteBuffers(BYTE* ref)

Copies 4 bytes, addressed by the pointer/ref argument, into the corresponding output registers. Byte[0] -> OUT1[7:0], byte[1] -> OUT1[15:8], ... etc. If entire buffer is empty, (IBF = 0) function returns TRUE, else the bytes are not copied and returns FALSE.

Input Parameters

Input Parameters	Description
ref	BYTE pointer

Returns

Returns the state of PMSTAT.OBE prior to the write operation.

Remarks

Use in SLAVE BUFFERED mode, MODE[1:0] = 00 or SLAVE ENHANCED mode, MODE[1:0] = 00 and INCM[1:0]=11

8.20.1.11 Close_PMP

void PMPClose(void)

disables PMP module, disables interrupt

Returns

None

Remarks

PMCONH.PMPEN, PIE1.PMPIE, PIR1.PMPIF are cleared.

8.20.2 PMP Structs,Records,Enums

8.20.3 PMP Macros

8.20.4 PMP Examples

8.20.4.1 PMP Example1

Example Source Code demonstrating peripheral library usage

```

/*****
*
* NOTES:
* Code uses the PERIPHERAL LIBRARY support available with MCC18 Compiler
* Code Tested on:
* PIC18F46J50 controller
* The Processor starts with the External Crystal (8 Mhz).
*
* PMP module used in LCD driver.
* Tianma TM162JCAWG1 LCD used to display message
* Refer Tianma TM162JCAWG1 LCD for timing diagrams, control instructions and time delay
information
*****/
#define USE_OR_MASKS
#include "p18cxxx.h"
#include "pmp.h"

#if defined(__18F46J50)
/***** CONFIGURATION *****/
* Oscillator is configured as HS
* Fail safe monitor is enabled
* watch dog timer is disabled

```

```

    * Extended instruction mode is disabled
    * oscillator switch over is enabled
    * CPU clock is not devided
    *****/
#pragma config OSC=HS, FCMEN=ON, WDTEN=OFF, IESO=ON, XINST=OFF, CPUDIV=OSC1
#endif

//***** Macros for LCD delays *****/
// Define a fast instruction execution time in terms of loop time
// typically > 43us
#define LCD_F_INSTR 100

// Define a slow instruction execution time in terms of loop time
// typically > 1.35ms
#define LCD_S_INSTR 1500 //150

// Define the startup time for the LCD in terms of loop time
// typically > 30ms
#define LCD_STARTUP 14000 //2000

unsigned int _uLCDloops;
unsigned char LCD_DATA1[] = "MICROCHIP's PIC";

//***** Prototype declarations *****/
void pmp_Init(void);
void LCDInit(void);
void LCDHome(void);
void LCD1Home(void);
void LCD2Home(void);
void LCDClear(void);
void LCDPut(char A);
void DisplayMSG(unsigned char *array);
void Wait(unsigned int B);

void main(void)
{
    LCDInit(); //Initialize LCD module
    DisplayMSG(LCD_DATA1); //Disaplay message in LCD_DATA1 string on
    first row of LCD

    while(1); //end of program
}

/*****
*****
*Function : pmp_Init
*Function initializes PMP module for data transfer to LCD mdule
* Parameter passed: None
* Return value: None
* Affects registers and port pins associated with PMP module
*****/

void pmp_Init(void)
{
    unsigned int mode,control,port,addrs,interrupt;

    PMPClose(); //dsiable PMP if enabled
    previously

    /***** PMP configuration *****/
    *****/
    /***** configuration settings *****/
    *****/

```

```

*
*      PMP module enabled
*      Configure RD, RD/WR strobe = ON; WR, WR/ENB strobe = ON
*      Configure Write/ENB and Read/RW polarity = active high
*      Configure MASTER mode 1
*      Configure 4 Tcy WAIT for Data Setup to Read/Write Wait State
*      Configure 15 Tcy WAIT for Read to Byte Enable Strobe Wait State
*      Configure 4 Tcy WAIT for Data Hold After Strobe Wait State
*****
*****/
    control = BIT_PMP_ON | BIT_RD_WR_ON | BIT_RD_WR_HI ;
    mode = BIT_MODE_MASTER_1 | BIT_WAITB_4_TCY | BIT_WAITM_15_TCY | BIT_WAITE_4_TCY ;
    port = BIT_P0;
    addrs = 0x0000;
    interrupt = 0x0000;
    PMPOpen(control,mode,port,addrs,interrupt);           //Configure PMP and enable it
}

//***** LCD initialization
//*****
void LCDInit(void)
{
    pmp_Init();                                           //Initialize PMP

    //wlat for typically > 30us
    _uLCDloops = LCD_STARTUP;
    Wait(_uLCDloops);

    //wlat for typically > 1.35us
    _uLCDloops = LCD_F_INSTR;
    PMPMasterWrite(0x0038);                             // Set the default function
    Wait(_uLCDloops);

    //wlat for typically > 30us
    _uLCDloops = LCD_STARTUP;
    Wait(_uLCDloops);
    //wlat for typically > 1.35us
    _uLCDloops = LCD_F_INSTR;
    PMPMasterWrite(0x000C);
    Wait(_uLCDloops);
    //wlat for typically > 30us
    _uLCDloops = LCD_STARTUP;
    Wait(_uLCDloops);
    //wlat for typically > 43us
    _uLCDloops = LCD_S_INSTR;
    PMPMasterWrite(0x0001);                             // Clear the display
    Wait(_uLCDloops);
    //wlat for typically > 30us
    _uLCDloops = LCD_STARTUP;
    Wait(_uLCDloops);
    //wlat for typically > 43us
    _uLCDloops = LCD_S_INSTR;
    PMPMasterWrite(0x0006);                             // Set the entry mode

    Wait(_uLCDloops);

    LCDClear();
    LCDHome();
}

void LCDHome(void)
{
    _uLCDloops = LCD_S_INSTR;
    PMPSetAddress(0x0000);
    PMPMasterWrite(0x0002);
}

```

```

    while(_uLCDloops)
        _uLCDloops--;
}

void LCDL1Home(void)
{
    _uLCDloops = LCD_S_INSTR;
    PMPSetAddress(0x0000);
    PMPMasterWrite(0x0080);
    while(_uLCDloops)
        _uLCDloops--;
}

void LCDL2Home(void)
{
    _uLCDloops = LCD_S_INSTR;
    PMPSetAddress(0x0000);
    PMPMasterWrite(0x00C0);
    while(_uLCDloops)
        _uLCDloops--;
}

void LCDClear(void)
{
    _uLCDloops = LCD_S_INSTR;
    PMPSetAddress(0x0000);
    PMPMasterWrite(0x0001);
    while(_uLCDloops)
        _uLCDloops--;
}

void LCDPut(char A)
{
    _uLCDloops = LCD_F_INSTR;
    PMPSetAddress(0x0001);
    PMPMasterWrite( (WORD) A);
    while(_uLCDloops)
        _uLCDloops--;
    Nop();
    Nop();
    Nop();
    Nop();
}

void Wait(unsigned int B)
{
    while(B)
        B--;
}

void DisplayMSG(unsigned char *array)
{
    unsigned char i=0,line=1;
    LCDL1Home();
    while (*array) // Continue display characters from STRING
    {
        LCDPut(*array++); // Display selected character from the STRING.
        if (i>19 && line==1)
        {
            LCDL2Home();
            line++;
        }
        i++;
    }
}

```

```
}  
}
```

8.21 Flash

8.21.1 Flash Functions

8.21.1.1 Erase_Flash_Page1

```
void EraseFlash(unsigned long startaddr, unsigned long endaddr)
```

The function erases flash from starting address in terms of 64 byte till end address or nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be erased
endaddr	End address till which flash has to be erased

Returns

None

Remarks

Note: 1. If number of bytes between strating and end address in not in multiples

of number of bytes of flash specified in data sheet, then excessive memory is erased upto nearest next multiple of number of bytes of flash specified in data sheet.

2. The starting and end address has to be in blocks of number of bytes of flash specified in data sheet

else function will allign the address to nearest previous and next

alligned address respectively

8.21.1.2 Erase_Flash_Page2

```
void EraseFlash(unsigned long startaddr, unsigned long endaddr)
```

The function erases flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be erased
endaddr	End address till which flash has to be erased

Returns
None

Remarks

Note: 1. If number of bytes between strating and end address in not in multiples of number of bytes of flash specified in data sheet, then excessive memory is erased upto nearest next multiple of number of bytes of flash specified in data sheet.

2. The starting and end address has to be in blocks of number of bytes of flash specified in data sheet else function will allign the address to nearest previous and next alligned address respectively

8.21.1.3 Erase_Flash_Page3

```
void EraseFlash(unsigned long startaddr, unsigned long endaddr)
```

The function erases flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be erased
endaddr	End address till which flash has to be erased

Returns
None

Remarks

Note: 1. If number of bytes between strating and end address in not in multiples of number of bytes of flash specified in data sheet, then excessive memory is erased upto nearest next multiple.

2. The starting and end address has to be in blocks of number of bytes of flash specified in data sheet else function will allign the address to nearest previous and next alligned address respectively

8.21.1.4 Read_Flash_Page1

```
void ReadFlash(unsigned long startaddr, unsigned int num_bytes, unsigned char *flash_array)
```

The function reads flash for number of bytes passed as parameter from starting address

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be read
num_bytes	Number of bytes of flash to be read
flash_array	Pointer to array to which the flash has to be read

Returns

None

Remarks

1. Non zero number of bytes has to be passed as parameter for num_bytes
2. Maximum number of bytes that can be read in one call is 64K bytes

8.21.1.5 WriteBlock_Flash_Page1

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address is not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing
if application had written data into this block of flash(after erasing followed by

programming).

2. Necessary to write in terms of block of number of bytes of flash specified in data sheet

8.21.1.6 WriteBlock_Flash_Page2

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address is not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into this block of flash (after erasing followed by programming).

2. Necessary to write in terms of block of number of bytes of flash specified in data sheet

8.21.1.7 WriteBlock_Flash_Page3

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address is not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Necessary to write interms of block of number of bytes of flash specified in data sheet

8.21.1.8 WriteBlock_Flash_Page4

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address in not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Necessary to write interms of block of number of bytes of flash specified in data sheet

8.21.1.9 WriteBlock_Flash_Page5

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address in not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Necessary to write in terms of block of number of bytes of flash specified in data sheet

8.21.1.10 WriteBlock_Flash_Page6

```
void WriteBlockFlash(unsigned long startaddr, unsigned char num_blocks, unsigned char *flash_array)
```

The function writes flash from starting address in terms of number of bytes of flash specified in data sheet till end address or nearest multiple. If number of bytes between starting and end address is not in multiples of number of bytes of flash specified in data sheet, write begins from address that is previous nearest multiple

Input Parameters

Input Parameters	Description
startaddr	Starting address from which flash has to be written
num_blocks	Number of blocks of flash to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Necessary to write in terms of block of number of bytes of flash specified in data sheet

8.21.1.11 WriteWord_Flash_Page1

```
void WriteWordFlash(unsigned long startaddr, unsigned int data)
```

The function writes word to flash

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be written
data	Data to be written into flash

Returns

None

Remarks

- Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).
2. Starting address has to be an even address else boundary mismatch will occur

8.21.1.12 WriteBytes_Flash_Page1

```
void WriteBytesFlash(unsigned long startaddr, unsigned int num_bytes, unsigned char *flash_array)
```

The function writes flash from starting address till end address.

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be written
num_bytes	Number of bytes to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

- Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by

programming).

2. Maximum number of bytes that can be written in one call is 64K bytes

8.21.1.13 WriteBytes_Flash_Page2

```
void WriteBytesFlash(unsigned long startaddr, unsigned int num_bytes, unsigned char *flash_array)
```

The function writes flash from starting address till end address.

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be written
num_bytes	Number of bytes to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Maximum number of bytes that can be written in one call is 64K bytes

8.21.1.14 WriteBytes_Flash_Page3

```
void WriteBytesFlash(unsigned long startaddr, unsigned int num_bytes, unsigned char *flash_array)
```

The function writes flash from starting address till end address.

Input Parameters

Input Parameters	Description
startaddr	Strating address from which flash has to be written
num_bytes	Number of bytes to be written
flash_array	Pointer to array contents of which has to be written to flash

Returns

None

Remarks

Note: 1. Necessary to erase flash block exclusively in application before writing if application had written data into to this block of flash(after erasing followed by programming).

2. Maximum number of bytes that can be written in one call is 64K bytes

8.21.2 Flash Macros

8.21.3 Flash Examples

8.21.3.1 Flash Example1

```

/*****
* 2009 Microchip Technology Inc.
*
* FileName:          Flash.c
* Dependencies:      Header (.h) files if applicable, see below
* Processor:         PIC18F
* Compiler:          MCC18 v3.30 or higher
*
* SOFTWARE LICENSE AGREEMENT:
* Microchip Technology Incorporated ("Microchip") retains all
* ownership and intellectual property rights in the code accompanying
* this message and in all derivatives hereto. You may use this code,
* and any derivatives created by any person or entity by or on your
* behalf, exclusively with Microchip's proprietary products. Your
* acceptance and/or use of this code constitutes agreement to the
* terms and conditions of this notice.
*
* CODE ACCOMPANYING THIS MESSAGE IS SUPPLIED BY MICROCHIP "AS IS". NO
* WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT
* NOT LIMITED TO, IMPLIED WARRANTIES OF NON-INFRINGEMENT,
* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS
* CODE, ITS INTERACTION WITH MICROCHIP'S PRODUCTS, COMBINATION WITH
* ANY OTHER PRODUCTS, OR USE IN ANY APPLICATION.
*
* YOU ACKNOWLEDGE AND AGREE THAT, IN NO EVENT, SHALL MICROCHIP BE
* LIABLE, WHETHER IN CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE OR
* BREACH OF STATUTORY DUTY), STRICT LIABILITY, INDEMNITY,
* CONTRIBUTION, OR OTHERWISE, FOR ANY INDIRECT, SPECIAL, PUNITIVE,
* EXEMPLARY, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, FOR COST OR
* EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE CODE, HOWSOEVER
* CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE
* DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWABLE BY LAW,
* MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THIS
* CODE, SHALL NOT EXCEED THE PRICE YOU PAID DIRECTLY TO MICROCHIP
* SPECIFICALLY TO HAVE THIS CODE DEVELOPED.
*
* You agree that you are solely responsible for testing the code and
* determining its suitability. Microchip has no obligation to modify,
* test, certify, or support the code.
*
* REVISION HISTORY:
* ~~~~~

```

```

* Author                               Date               Comments on this revision
* ~~~~~
* Harsha.J.M      04/05/10      First release of source file
* ~~~~~
*
* ADDITIONAL NOTES:
* Code uses the Peripheral library support available with MCC18 Compiler
* Code Tested on:
* PicDem2+ demo board with PIC18F4685 controller
* The Processor starts with the External Crystal (8 Mhz).
* Program depicts the usage of Flash Write/Read/Erased APIs
*****/
#define USE_OR_MASKS
#include <pl8cxxx.h>
#include "flash.h"

//-----Configuration setting
-----
/**
 * Oscillator is configured as HS
 * Fail safe monitor is enabled
 * watch dog timer is disabled
 * Extended instruction mode is disabled
 * oscillator switch over is enabled
 */
#if defined(__18F4685)                //If the selected device is PIC18F4685, then apply
below settings else user will have to set
#pragma config OSC=HS, FCMEN=ON, WDT=OFF, IESO=ON, XINST=OFF, LVP=OFF
#endif

#pragma udata      WRITE_BANK=0x200
unsigned char Write_Data[250];
#pragma udata      READ_BANK=0x300
unsigned char Read_Data[250];

void main(void)
{
    unsigned char i;

    //Initialize Data to be written to flash
    for(i=0;i<250;i++)
    {
        Write_Data[i]=i+1;
        Read_Data[i]=0;
    }

    //Write data into flash from the specified location
    WriteBytesFlash((UINT32)0x6000,(UINT16)250,Write_Data);

    //Read the data to verify the writted data
    ReadFlash((UINT32)0x6000,(UINT16)250,Read_Data);

    //Erase Flash
    EraseFlash((UINT32)0x6000,(UINT32)0x7000);

    //Read the data to verify the erased data
    ReadFlash((UINT32)0x6000,(UINT16)128,Read_Data);

    //Write data into flash from the specified location
    WriteBlockFlash((UINT32)0x6000,4,Write_Data);

    //Read the data to verify the writted data
    ReadFlash((UINT32)0x6000,(UINT16)128,Read_Data);

    while(1);                //End of Program
}

```

8.22 Prototype_Page



Input Parameters

Parameters are passed to arguments of functions as a bitmask, created by performing either bitwise AND operation ('&') or bitwise OR operation ('|'), configurable either way as shown in the examples

Input Parameters	Description

Returns

Remarks

9 Symbol Reference

9.1 Files

The following table lists files in this documentation.

Files

Name	Description
adc.h (see page 1207)	
ancomp.h (see page 1207)	
capture.h (see page 1207)	
compare.h (see page 1208)	
ctmu.h (see page 1208)	
dpslp.h (see page 1208)	
EEP.h (see page 1208)	
flash.h (see page 1208)	
i2c.h (see page 1208)	
mwire.h (see page 1208)	
pcpwm.h (see page 1208)	
pmp.h (see page 1208)	
portb.h (see page 1208)	
pps.h (see page 1208)	
ProMPT.h (see page 1209)	
pwm.h (see page 1209)	
reset.h (see page 1209)	
rtcc.h (see page 1209)	
spi.h (see page 1209)	
timers.h (see page 1209)	
usart.h (see page 1209)	

9.1.1 adc.h

9.1.2 ancomp.h

9.1.3 capture.h

9.1.4 compare.h

9.1.5 ctmu.h

9.1.6 dpslp.h

9.1.7 EEP.h

9.1.8 flash.h

9.1.9 i2c.h

9.1.10 mwire.h

9.1.11 pcpwm.h

9.1.12 pmp.h

9.1.13 portb.h

9.1.14 pps.h

9.1.15 ProMPT.h

9.1.16 pwm.h

9.1.17 reset.h

9.1.18 rtcc.h

9.1.19 spi.h

9.1.20 timers.h

9.1.21 usart.h

Index

—
_BOOL 932

1

18C452 Family 761
18C801 Family 738
18C858 Family 731
18F/LF14K22 Family 465
18F/LF14K50 Family 457
18F/LF46J11 Family 431
18F/LF46J50 Family 405
18F1320 Family 784
18F1330 Family 806
18F4320 Family 713
18F4321 Family 635
18F4431 Family 811
18F4450 Family 708
18F452 Family 753
18F4539 Family 746
18F4550/23/20 Family 699
18F4553 Family 474
18F458 Family 768
18F45J10 Family 789
18F45K22 Family 827
18F4610 Family 668
18F4620 Family 683
18F46K20 Family 533
18F47J53 Family 286
18F4x80/5 Family 651
18F66K80 Family 275
18F8410 Family 626
18F8490 Family 618
18F8493 Family 482
18F84J15 Family 501
18F85J90/11 Family 580
18F8621 Family 589
18F8680 Family 599
18F8720 Family 722

18F8722 Family 607
18F8723 Family 491
18F87J10 Family 570
18F87J11 Family 522
18F87J50 Family 511
18F87J90 Family 384
18F87J93 Family 394
18F87K90/22 Family 326
18F97J60 Family 541
18F97J72 Family 315

A

A/D Converter (ADC) 840
ADC (1320 Family) 784
ADC (1330 Family) 806
ADC (14K22 Family) 466
ADC (14K50 Family) 457
ADC (2431 Family) 819
ADC (258 Family) 768
ADC (25J10 Family) 798
ADC (2610 Family) 676
ADC (2620 Family) 691
ADC (2x21 Family) 643
ADC (2x8x Family) 660
ADC (2xJ11 Family) 431
ADC (2xJ50 Family) 405
ADC (2xJ53 Family) 287
ADC (4320 Family) 713
ADC (4431 Family) 811
ADC (4450 Family) 708
ADC (4539 Family) 746
ADC (4553 Family) 474
ADC (458 Family) 776
ADC (45J10 Family) 789
ADC (45K22 Set) 827
ADC (45xx Family) 700
ADC (4610 Family) 668
ADC (4620 Family) 683
ADC (46K20 Family) 533
ADC (4x21 Family) 635
ADC (4x8x Family) 652

ADC (4xJ11 Family) 443	Analog Comparator (258 Family) 777
ADC (4xJ50 Family) 417	Analog Comparator (25J10 Family) 799
ADC (4xJ53 Family) 300	Analog Comparator (2610 Family) 676
ADC (65K90 Set) 342	Analog Comparator (2620 Family) 692
ADC (66K80 Set) 275	Analog Comparator (2x21 Family) 644
ADC (67J60 Family) 542	Analog Comparator (2xJ11 Family) 432
ADC (6xK90 Set) 326	Analog Comparator (2xJ50 Family) 406
ADC (801 Family) 739	Analog Comparator (2xJ53 Set) 287
ADC (8410 Family) 626	Analog Comparator (4320 Family) 714
ADC (8490 Family) 618	Analog Comparator (4553 Family) 475
ADC (8493 Family) 482	Analog Comparator (45J10 Family) 790
ADC (84J15 Family) 501	Analog Comparator (45K22 Family) 828
ADC (858 Family) 731	Analog Comparator (45xx Family) 700
ADC (85J90/11 Family) 581	Analog Comparator (4610 Family) 669
ADC (85K90 Set) 370	Analog Comparator (4620 Family) 684
ADC (8621 Family) 589	Analog Comparator (46K20 Family) 533
ADC (8680 Family) 599	Analog Comparator (4x21 Family) 636
ADC (8720 Family) 722	Analog Comparator (4x8x Family) 652
ADC (8722 Family) 607	Analog Comparator (4xJ11 Family) 444
ADC (8723 Family) 491	Analog Comparator (4xJ50 Family) 418
ADC (87J10 Family) 570	Analog Comparator (4xJ53 Set) 300
ADC (87J11 Family) 522	Analog Comparator (65K90 Set) 343
ADC (87J50 Family) 511	Analog Comparator (66K80 Set) 276
ADC (87J60 Family) 551	Analog Comparator (67J60 Family) 542
ADC (87J90 Family) 384	Analog Comparator (6xK50 Set) 328
ADC (87J93 Family) 394	Analog Comparator (8410 Family) 627
ADC (8xK90 Set) 356	Analog Comparator (8490 Family) 619
ADC (97J60 Family) 560	Analog Comparator (8493 Family) 483
ADC (97J72 Family) 315	Analog Comparator (84J15 Family) 502
ADC (C452 Family) 761	Analog Comparator (858 Family) 732
ADC (F452 Family) 753	Analog Comparator (85J90/11 Family) 581
ADC Example 1 898	Analog Comparator (85K90 Set) 371
ADC Examples 898	Analog Comparator (8621 Family) 590
ADC Functions 840	Analog Comparator (8680 Family) 600
ADC Macros 898	Analog Comparator (8720 Family) 722
ADC Structs,Records,Enums 898	Analog Comparator (8722 Family) 608
adc.h 1207	Analog Comparator (8723 Family) 491
Analog Comparator (87J90 Family) 384	Analog Comparator (87J10 Family) 571
Analog Comparator (97J72 Family) 316	Analog Comparator (87J11 Family) 523
Analog Comparator (14K22 Family) 466	Analog Comparator (87J50 Family) 512
Analog Comparator (14K50 Family) 458	Analog Comparator (87J60 Family) 551

Analog Comparator (87J93 Family) 395
Analog Comparator (8xK90 Set) 357
Analog Comparator (97J60 Family) 561
Analog Comparator (ANCOMP) 899
ancomp.h 1207

B

baud_1USART 1155
baud_2USART 1157
baud_USART 1158
Busy__EEP 1062
Busy_ADC 897

C

capture.h 1207
Charge Time Measurement Unit (CTMU) 925
Close_ADC 897
Close_ANCOMP 922
Close_ancomp_Page1 923
Close_ancomp1_Page1 923
Close_ancomp2_Page1 923
Close_ancomp3_Page1 923
Close_Capture 987
Close_Capture1 987
Close_Capture10 990
Close_Capture2 987
Close_Capture3 988
Close_Capture4 988
Close_Capture5 988
Close_Capture6 989
Close_Capture7 989
Close_Capture8 989
Close_Capture9 989
Close_Compare 1021
Close_Compare1 1021
Close_Compare10 1024
Close_Compare2 1022
Close_Compare3 1022
Close_Compare4 1022
Close_Compare5 1022
Close_Compare6 1023

Close_Compare7 1023
Close_Compare8 1023
Close_Compare9 1024
Close_CTMU 927
Close_ECapture1 990
Close_ECapture2 990
Close_ECapture3 991
Close_ECompare1 1024
Close_ECompare2 1024
Close_ECompare3 1025
Close_EPWM1 1059
Close_EPWM2_Page1 1059
Close_EPWM3_Page1 1059
Close_I2C 1071
Close_pcpwm 1186
Close_PMP 1192
Close_PWM 1056
Close_PWM1 1056
Close_PWM10_Page1 1058
Close_PWM2 1056
Close_PWM3 1056
Close_PWM4 1057
Close_PWM5 1057
Close_PWM6_Page1 1057
Close_PWM7_Page1 1057
Close_PWM8_Page1 1058
Close_PWM9_Page1 1058
Close_SW_RTCC 1106
Close_Timer 1143
Close_Timer0 1143
Close_Timer1 1143
Close_Timer10_Page1 1146
Close_Timer12_Page1 1146
Close_Timer2 1144
Close_Timer3 1144
Close_Timer4 1144
Close_Timer5 1144
Close_Timer6_Page1 1145
Close_Timer7_Page1 1145
Close_Timer8_Page1 1145
Comparator Example1 924

Comparator Examples 924
 Comparator Functions 899
 Comparator Macros 924
 compare.h 1208
 Convert_ADC 896
 CTMU (2xJ53 Family) 288
 CTMU (45K22 Set) 829
 CTMU (4xJ53 Family) 301
 CTMU (65K90 Set) 344
 CTMU (66K80 Set) 277
 CTMU (6xK90 Set) 329
 CTMU (85K90 Set) 371
 CTMU (87J90 Family) 385
 CTMU (8xK90 Set) 357
 CTMU (97J72 Family) 316
 CTMU (2xJ11 Family) 432
 CTMU (2xJ50 Family) 406
 CTMU (4xJ11 Family) 444
 CTMU (4xJ50 Family) 418
 CTMU (87J93 Family) 395
 CTMU Example1 928
 CTMU Examples 928
 CTMU Functions 925
 CTMU Macros 928
 ctmu.h 1208
 CurrentControl_CTMU 927

D

Deep Sleep (2xJ11 Family) 433
 Deep Sleep (2xJ50 Family) 407
 Deep Sleep (2xJ53 Family) 289
 Deep Sleep (4xJ11 Family) 445
 Deep Sleep (4xJ50 Family) 419
 Deep Sleep (4xJ53 Family) 302
 Deep Sleep (66K80 Family) 286
 Deep Sleep (DPSLP) 929
 Deep Sleep Example1 932
 Deep Sleep Examples 932
 Deep Sleep Functions 929
 Deep Sleep Macros 932
 DeepSleep Structs,Records,Enums 932

DeepSleep_WakeUpSource 930
 Device Modules 275
 Devices 11
 dpslp.h 1208
 dt_assignment_pcpwm 1185
 dt_clk_source_pcpwm 1185

E

EEAckPolling_I2C 1075
 EEByteWrite_I2C 1079
 EECurrentAddRead_I2C 1077
 EEP (1320 Family) 786
 EEP (1330 Family) 808
 EEP (14K22 Family) 469
 EEP (14K50 Family) 461
 EEP (2431 Family) 824
 EEP (258 Family) 773
 EEP (2620 Family) 695
 EEP (2x21 Family) 648
 EEP (2x8x Family) 664
 EEP (4320 Family) 718
 EEP (4431 Family) 816
 EEP (4539 Family) 749
 EEP (4553 Family) 477
 EEP (458 Family) 781
 EEP (45K22 Set) 835
 EEP (45xx Family) 705
 EEP (4620 Family) 687
 EEP (46K20 Family) 536
 EEP (4x21 Family) 640
 EEP (4x8x Family) 656
 EEP (65K90 Set) 349
 EEP (66K80 Set) 281
 EEP (6xK90 Set) 334
 EEP (85K90 Set) 376
 EEP (8621 Family) 593
 EEP (8680 Family) 603
 EEP (8720 Family) 727
 EEP (8722 Family) 613
 EEP (8723 Family) 495
 EEP (8xK90 Set) 363

EEP (F452 Family) 757
EEP Example1 1062
EEP Examples 1062
EEP Functions 1061
EEP.h 1208
EEPPageWrite_I2C 1081
EEPROM (EEP) 1060
EERandomRead_I2C 1086
EESequentialRead_I2C 1083
Erase_Flash_Page1 1196
Erase_Flash_Page2 1196
Erase_Flash_Page3 1197

F

Files 1207
Flash 1196
Flash (1320 Family) 789
Flash (1330 Family) 810
Flash (14K22 Family) 473
Flash (14K50 Family) 465
Flash (2431 Family) 826
Flash (258 Family) 775
Flash (25J10 Family) 806
Flash (2610 Family) 683
Flash (2620 Family) 699
Flash (2x21 Family) 650
Flash (2x80/5 Family) 667
Flash (2xJ11 Set) 442
Flash (2xJ50 Set) 416
Flash (2xJ53 Set) 299
Flash (4320 Family) 721
Flash (4431 Family) 819
Flash (4450 Family) 712
Flash (4539 Family) 752
Flash (4553 Family) 482
Flash (458 Family) 783
Flash (45J10 Family) 798
Flash (45K22 Family) 838
Flash (45xx Family) 708
Flash (4610Family) 675
Flash (4620Family) 691
Flash (46K20 Family) 541
Flash (4x21 Family) 642
Flash (4x80/5 Family) 659
Flash (4xJ11 Set) 456
Flash (4xJ50 Set) 430
Flash (4xJ53 Set) 313
Flash (65K90 Set) 356
Flash (66K80 Family) 284
Flash (67J60 Set) 550
Flash (6xK90) 341
Flash (8410 Family) 634
Flash (8490 Family) 626
Flash (8493 Family) 490
Flash (84J15 Family) 511
Flash (85J90/11 Family) 588
Flash (85K90 Set) 383
Flash (8621Family) 598
Flash (8680 Family) 607
Flash (8720 Family) 730
Flash (8722 Family) 617
Flash (8723 Family) 501
Flash (87J10 Family) 580
Flash (87J11 Family) 532
Flash (87J50 Family) 521
Flash (87J60 Set) 559
Flash (87J90 Family) 394
Flash (87J93 Family) 404
Flash (8xK90 Set) 370
Flash (97J60 Set) 569
Flash (97J72 Family) 325
Flash (F452 Family) 760
Flash Example1 1204
Flash Examples 1204
Flash Functions 1196
Flash Macros 1204
flash.h 1208

G

gets_1USART 1159
gets_2USART 1159
gets_I2C 1070

gets_I2C1 1070	I2C (4x8x Family) 654
gets_I2C2 1071	I2C (4xJ11 Family) 447
gets_Mwire 1168	I2C (4xJ50 Family) 421
gets_Mwire1 1167	I2C (4xJ53 Family) 305
gets_Mwire2 1168	I2C (65K90 Set) 347
gets_SPI 1115	I2C (67J60 Family) 544
gets_SPI1 1115	I2C (6xK90 Set) 332
gets_SPI2 1116	I2C (801 Family) 740
gets_USART 1160	I2C (8410 Family) 629
Getting Started 9	I2C (8490 Family) 620
Goto_DeepSleep 930	I2C (8493 Family) 485
	I2C (84J15 Family) 504
	I2C (858 Family) 733
	I2C (85K90 Set) 374
I2C (66K80 Family) 279	I2C (8680 Family) 601
I2C (85J90/11 Family) 583	I2C (8720 Family) 725
I2C (8621 Family) 592	I2C (8722 Family) 610
I2C (87J90 Family) 387	I2C (8723 Family) 493
I2C (97J72 Family) 318	I2C (87J10 Family) 573
I2C (14K22 Family) 468	I2C (87J11 Family) 525
I2C (14K50 Family) 459	I2C (87J50 Family) 515
I2C (2431 Family) 822	I2C (87J60 Family) 553
I2C (258 Family) 770	I2C (87J93 Family) 397
I2C (25J10 Family) 800	I2C (8xK90 Set) 361
I2C (2610 Family) 678	I2C (97J60 Family) 563
I2C (2620 Family) 693	I2C (C452 Family) 763
I2C (2x21 Family) 645	I2C (F452 Family) 755
I2C (2x8x Family) 662	I2C EEPROM (66K80 Family) 280
I2C (2xJ11 Family) 435	I2C EEPROM (85J90/11 Family) 584
I2C (2xJ50 Family) 409	I2C EEPROM (8621 Family) 593
I2C (4320 Family) 715	I2C EEPROM (87J90 Family) 388
I2C (4431 Family) 814	I2C EEPROM (97J72 Family) 319
I2C (4539 Family) 746	I2C EEPROM (14K22 Family) 469
I2C (4553 Family) 476	I2C EEPROM (14K50 Family) 460
I2C (458 Family) 778	I2C EEPROM (258 Family) 771
I2C (45J10 Family) 791	I2C EEPROM (25J10 Family) 801
I2C (45K22 Family) 832	I2C EEPROM (2610 Family) 678
I2C (45xx Family) 702	I2C EEPROM (2620 Family) 694
I2C (4610 Family) 670	I2C EEPROM (2x21 Family) 646
I2C (4620 Family) 685	I2C EEPROM (2x8x Family) 663
I2C (46K20 Family) 535	I2C EEPROM (2xJ11 Family) 436
I2C (4x21 Family) 637	

I2C EEPROM (2xJ50 Family) 410	I2C Example1_Master 1072
I2C EEPROM (2xJ53 Family) 292	I2C Example1_Slave 1074
I2C EEPROM (4320 Family) 716	I2C Examples 1072
I2C EEPROM (44K22 Family) 833	I2C Functions 1063
I2C EEPROM (4539 Family) 747	I2C Macros 1072
I2C EEPROM (4553 Family) 477	i2c.h 1208
I2C EEPROM (458 Family) 779	I2C_EEAckPolling 1075
I2C EEPROM (45J10 Family) 793	I2C_EEAckPolling1 1076
I2C EEPROM (45xx Family) 703	I2C_EEAckPolling2 1076
I2C EEPROM (4610 Family) 671	I2C_EEByteWrite 1079
I2C EEPROM (4620 Family) 686	I2C_EEByteWrite1 1080
I2C EEPROM (46K20 Family) 536	I2C_EEByteWrite2 1080
I2C EEPROM (4x21 Family) 638	I2C_EECurrentAddRead 1077
I2C EEPROM (4x8x Family) 655	I2C_EECurrentAddRead1 1078
I2C EEPROM (4xJ11 Family) 448	I2C_EECurrentAddRead2 1078
I2C EEPROM (4xJ50 Family) 422	I2C_EEP Example1 1088
I2C EEPROM (4xJ53 Family) 306	I2C_EEP Examples 1088
I2C EEPROM (65K90 Set) 348	I2C_EEP Functions 1075
I2C EEPROM (67J60 Family) 545	I2C_EEPPageWrite 1081
I2C EEPROM (6xK90 Set) 334	I2C_EEPPageWrite1 1082
I2C EEPROM (801 Family) 741	I2C_EEPPageWrite2 1083
I2C EEPROM (8410 Family) 630	I2C_EERandomRead 1086
I2C EEPROM (8490 Family) 621	I2C_EERandomRead1 1086
I2C EEPROM (8493 Family) 485	I2C_EERandomRead2 1087
I2C EEPROM (84J15 Family) 505	I2C_EESequentialRead 1083
I2C EEPROM (858 Family) 734	I2C_EESequentialRead1 1084
I2C EEPROM (85K90 Set) 376	I2C_EESequentialRead2 1085
I2C EEPROM (8680 Family) 602	I2C_gets 1070
I2C EEPROM (8720 Family) 725	I2C_Open 1063
I2C EEPROM (8722 Family) 612	I2C_puts 1068
I2C EEPROM (8723 Family) 495	I2C_Read 1067
I2C EEPROM (87J10 Family) 574	I2C_Write 1065
I2C EEPROM (87J11 Family) 526	Idle_I2C 1072
I2C EEPROM (87J50 Family) 516	Input Capture (1320 Family) 785
I2C EEPROM (87J60 Family) 554	Input Capture (14K22 Family) 467
I2C EEPROM (87J93 Family) 398	Input Capture (14K50 Family) 458
I2C EEPROM (8xK90 Set) 362	Input Capture (2431 Family) 820
I2C EEPROM (97J60 Family) 564	Input Capture (258 Family) 769
I2C EEPROM (C452 Family) 764	Input Capture (25J10 Family) 799
I2C EEPROM (F452 Family) 755	Input Capture (2610 Family) 676
I2C EEPROM (I2C_EEP) 1075	Input Capture (2620 Family) 692

Input Capture (2x21 Family) 644	Input Capture (87J60 Family) 552
Input Capture (2x8x Family) 661	Input Capture (87J90 Family) 385
Input Capture (2xJ11 Family) 434	Input Capture (87J93 Family) 396
Input Capture (2xJ50 Family) 408	Input Capture (8xK90 Set) 358
Input Capture (2xJ53 Set) 289	Input Capture (97J60 Family) 561
Input Capture (4320 Family) 714	Input Capture (97J72 Family) 317
Input Capture (4431 Family) 812	Input Capture (C452 Family) 762
Input Capture (4450 Family) 709	Input Capture (F452 Family) 753
Input Capture (4553 Family) 475	Input Capture (INCAP) 954
Input Capture (458 Family) 777	Input Capture Example1 991
Input Capture (45J10 Family) 790	Input Capture Examples 991
Input Capture (45K22 Set) 830	Input Capture Function 954
Input Capture (45xx Family) 701	Input Capture Macros 991
Input Capture (4610 Family) 669	Input Capture Structs, Records, Enums 991
Input Capture (4620 Family) 684	Inter Integrated Circuit Communication (I2C) 1063
Input Capture (46K20 Family) 534	Introduction 2
Input Capture (4x21 Family) 636	IO Ports 1088
Input Capture (4x8x Family) 653	IO Ports (1320 Family) 786
Input Capture (4xJ11 Family) 446	IO Ports (1330 Family) 808
Input Capture (4xJ50 Family) 420	IO Ports (14K22 Family) 469
Input Capture (4xJ53 Set) 302	IO Ports (14K50 Family) 461
Input Capture (65K90 Set) 345	IO Ports (2431 Family) 823
Input Capture (66K80 Set) 277	IO Ports (258 Family) 772
Input Capture (67J60 Family) 543	IO Ports (25J10 Family) 803
Input Capture (6xK90 Set) 330	IO Ports (2610 Family) 680
Input Capture (801 Family) 739	IO Ports (2620 Family) 696
Input Capture (8410 Family) 627	IO Ports (2x21 Family) 647
Input Capture (8490 Family) 619	IO Ports (2x8x Family) 664
Input Capture (8493 Family) 483	IO Ports (4320 Family) 718
Input Capture (84J15 Family) 502	IO Ports (4431 Family) 815
Input Capture (858 Family) 732	IO Ports (4450 Family) 710
Input Capture (85J90/11 Family) 582	IO Ports (4539 Family) 749
Input Capture (85K90 Set) 372	IO Ports (4553 Family) 478
Input Capture (8621 Family) 590	IO Ports (458 Family) 780
Input Capture (8680 Family) 600	IO Ports (45J10 Family) 795
Input Capture (8720 Family) 723	IO Ports (45K22 Family) 835
Input Capture (8722 Family) 608	IO Ports (45xx Family) 704
Input Capture (8723 Family) 492	IO Ports (4610 Family) 672
Input Capture (87J10 Family) 571	IO Ports (4620 Family) 688
Input Capture (87J11 Family) 523	IO Ports (46K20 Family) 537
Input Capture (87J50 Family) 513	IO Ports (4x21 Family) 639

IO Ports (4x8x Family) 657
 IO Ports (65K90 Set) 349
 IO Ports (66K80 Set) 281
 IO Ports (67J60 Family) 546
 IO Ports (6xK90 Set) 335
 IO Ports (801 Family) 743
 IO Ports (8410 Family) 631
 IO Ports (8490 Family) 623
 IO Ports (8493 Family) 486
 IO Ports (84J15 Family) 506
 IO Ports (858 Family) 735
 IO Ports (85J90/11 Family) 584
 IO Ports (85K90 Set) 377
 IO Ports (8621 Family) 594
 IO Ports (8680 Family) 604
 IO Ports (8720 Family) 727
 IO Ports (8722 Family) 614
 IO Ports (8723 Family) 496
 IO Ports (87J10 Family) 575
 IO Ports (87J11 Family) 527
 IO Ports (87J50 Family) 517
 IO Ports (87J60 Family) 555
 IO Ports (87J90 Family) 388
 IO Ports (8xK90 Set) 363
 IO Ports (97J60 Family) 565
 IO Ports (97J72 Family) 320
 IO Ports (C452 Family) 765
 IO Ports (F452 Family) 757
 IO Ports (2xJ11 Family) 436
 IO Ports (2xJ50 Family) 412
 IO Ports (2xJ53 Family) 294
 IO Ports (4xJ11 Family) 449
 IO Ports (4xJ50 Family) 423
 IO Ports (4xJ53 Family) 308
 IO Ports (87J93 Family) 400
 iPPS_Input_Page1/2 936
 iPPS_Input_Page1/2_1 938
 iPPS_Input_Page3 947
 iPPS_Output_Page1 944
 iPPS_Output_Page2 945
 iPPS_Output_Page3 950

IsBufferNEmpy_PMP 1189
 IsBufferNFull_PMP 1189
 IsResetFrom_DeepSleep 931

M

MasterRead_PMP 1188
 MasterWrite_PMP 1189
 MicroWire (MWIRE) 1165
 Modules 840
 MWIRE (14K22 Family) 473
 MWIRE (14K50 Family) 464
 MWIRE (2431 Family) 826
 MWIRE (258 Family) 775
 MWIRE (25J10 Family) 805
 MWIRE (2610 Family) 682
 MWIRE (2620 Family) 698
 MWIRE (2x21 Family) 650
 MWIRE (2x8x Family) 667
 MWIRE (2xJ11 Family) 442
 MWIRE (2xJ50 Family) 416
 MWIRE (2xJ53 Family) 298
 MWIRE (4320 Family) 720
 MWIRE (4431 Family) 818
 MWIRE (4539 Family) 751
 MWIRE (4553 Family) 481
 MWIRE (458 Family) 783
 MWIRE (45J10 Family) 797
 MWIRE (45K22 Family) 836
 MWIRE (45xx Family) 707
 MWIRE (4610 Family) 674
 MWIRE (4620 Family) 690
 MWIRE (46K20 Family) 540
 MWIRE (4x21 Family) 642
 MWIRE (4x8x Family) 659
 MWIRE (4xJ11 Family) 455
 MWIRE (4xJ50 Family) 430
 MWIRE (4xJ53 Family) 312
 MWIRE (65K90 Set) 350
 MWIRE (66K80 Family) 282
 MWIRE (67J60 Family) 549
 MWIRE (6xK90 Set) 335

O

Open_Compare4_Page2 998	Open_EPWM3_Page3 1041
Open_Compare4_Page3 999	Open_I2C 1063
Open_Compare4_Page4 1019	Open_I2C1 1064
Open_Compare5 1000	Open_I2C2 1065
Open_Compare5_Page2 1001	Open_Mwire 1167
Open_Compare5_Page3 1002	Open_Mwire1 1166
Open_Compare5_Page4 1020	Open_Mwire2 1166
Open_Compare6_Page1 1010	Open_pcpwm_Page1 1174
Open_Compare7_Page1 1011	Open_pcpwm_Page2 1178
Open_Compare8_Page1 1012	Open_PMP 1187
Open_Compare8_Page2 1013	Open_PORTB 1089
Open_Compare9_Page1 1014	Open_PWM 1026
Open_CTMU 925	Open_PWM1 1026
Open_ECapture1 966	Open_PWM10_Page1 1035
Open_ECapture1_Page2 967	Open_PWM2 1027
Open_ECapture1_Page3 968	Open_PWM2_Page2 1030
Open_ECapture1_Page4 969	Open_PWM3 1027
Open_ECapture1_Page5 979	Open_PWM3_Page2 1028
Open_ECapture2_Page1 970	Open_PWM4 1028
Open_ECapture2_Page2 971	Open_PWM4_Page2 1029
Open_ECapture2_Page4 980	Open_PWM4_Page3 1029
Open_ECapture3_Page1 972	Open_PWM4_Page4 1041
Open_ECapture3_Page4 981	Open_PWM5 1030
Open_ECompare1 1003	Open_PWM5_Page2 1031
Open_ECompare1_Page2 1004	Open_PWM5_Page3 1032
Open_ECompare1_Page3 1005	Open_PWM5_Page4 1042
Open_ECompare1_Page4 1006	Open_PWM6_Page1 1032
Open_ECompare1_Page5 1016	Open_PWM7_Page1 1033
Open_ECompare2_Page1 1007	Open_PWM8_Page1 1033
Open_ECompare2_Page2 1008	Open_PWM8_Page2 1034
Open_ECompare2_Page4 1017	Open_PWM9_Page1 1035
Open_ECompare3_Page1 1009	Open_RB0INT 1089
Open_ECompare3_Page4 1018	Open_RB1INT 1090
Open_EPWM1 1036	Open_RB2INT 1091
Open_EPWM1_Page2 1036	Open_RB3INT 1092
Open_EPWM1_Page3 1037	Open_SPI 1108
Open_EPWM1_Page4 1040	Open_SPI1 1109
Open_EPWM2_Page1 1038	Open_SPI2 1110
Open_EPWM2_Page2 1038	Open_SW_RTCC 1106
Open_EPWM2_Page3 1040	Open_Timer 1119
Open_EPWM3_Page1 1039	Open_Timer0 1119

Open_Timer1_Page1	1120	Output Compare (2xJ11 Family)	434
Open_Timer1_Page2	1121	Output Compare (2xJ50 Family)	408
Open_Timer10_Page1	1135	Output Compare (2xJ53 Set)	290
Open_Timer12_Page1	1136	Output Compare (4320 Family)	715
Open_Timer2	1123	Output Compare (4431 Family)	812
Open_Timer3_Page1	1124	Output Compare (4450 Family)	709
Open_Timer3_Page2	1125	Output Compare (4553 Family)	475
Open_Timer4	1130	Output Compare (458 Family)	777
Open_Timer5	1127	Output Compare (45J10 Family)	791
Open_Timer5_Page2	1129	Output Compare (45K22 Set)	830
Open_Timer6_Page1	1131	Output Compare (45xx Family)	701
Open_Timer7_Page1	1132	Output Compare (4610 Family)	669
Open_Timer8_Page1	1134	Output Compare (4620 Family)	685
Open_USART	1152	Output Compare (46K20 Family)	534
OpenADC_Page1	840	Output Compare (4x21 Family)	636
OpenADC_Page10	860	Output Compare (4x8x Family)	653
OpenADC_Page11	862	Output Compare (4xJ11 Family)	446
OpenADC_Page12	865	Output Compare (4xJ50 Family)	420
OpenADC_Page13	867	Output Compare (4xJ53 Set)	303
OpenADC_Page14	870	Output Compare (65K90 Set)	345
OpenADC_Page15	873	Output Compare (66K80 Set)	278
OpenADC_Page16	876	Output Compare (67J60 Family)	543
OpenADC_Page17	879	Output Compare (6xK90 Set)	331
OpenADC_Page2	842	Output Compare (801 Family)	740
OpenADC_Page3	844	Output Compare (8410 Family)	628
OpenADC_Page4	846	Output Compare (8490 Family)	619
OpenADC_Page5	848	Output Compare (8493 Family)	484
OpenADC_Page6	851	Output Compare (84J15 Family)	503
OpenADC_Page7	854	Output Compare (858 Family)	732
OpenADC_Page8	855	Output Compare (85J90/11 Family)	582
OpenADC_Page9	858	Output Compare (85K90 Set)	373
Output Compare (1320 Family)	785	Output Compare (8621 Family)	591
Output Compare (14K22 Family)	467	Output Compare (8680 Family)	600
Output Compare (14K50 Family)	458	Output Compare (8720 Family)	723
Output Compare (2431 Family)	820	Output Compare (8722 Family)	609
Output Compare (258 Family)	769	Output Compare (8723 Family)	492
Output Compare (25J10 Family)	799	Output Compare (87J10 Family)	572
Output Compare (2610 Family)	677	Output Compare (87J11 Family)	524
Output Compare (2620 Family)	693	Output Compare (87J50 Family)	513
Output Compare (2x21 Family)	644	Output Compare (87J60 Family)	552
Output Compare (2x8x Family)	661	Output Compare (87J90 Family)	386

Output Compare (87J93 Family) 397	PIC18F1320 232
Output Compare (8xK90 Set) 359	PIC18F1320 Family 232
Output Compare (97J60 Family) 562	PIC18F1330 230
Output Compare (97J72 Family) 317	PIC18F1330 Family 230
Output Compare (C452 Family) 762	PIC18F13K22 112
Output Compare (F452 Family) 754	PIC18F13K50 109
Output Compare (OCMP) 992	PIC18F14K22 111
Output Compare Example1 1025	PIC18F14K50 108
Output Compare Examples 1025	PIC18F1xK22 Family 111
Output Compare Functions 992	PIC18F1xK50 Family 108
Output Compare Macros 1025	PIC18F2220 230
OVD_CTRL_pcpwm 1184	PIC18F2221 194
OVD_IO_STA_pcpwm 1184	PIC18F2320 229

P

Parallel Master Port (PMP) 1186	PIC18F2321 193
PCPWM (1330 Family) 807	PIC18F2331 142
PCPWM (2431 Family) 821	PIC18F23K20 138
PCPWM (4431 Family) 813	PIC18F23K22 273
PCPWM Functions 1174	PIC18F2410 210
PCPWM Macros 1186	PIC18F242 254
pcpwm.h 1208	PIC18F2420 227
Peripheral Pin Select (PPS) 934	PIC18F2423 223
PIC18 Peripheral Library 1	PIC18F2431 141
PIC18C242 258	PIC18F2431 Set 819
PIC18C252 257	PIC18F2439 247
PIC18C442 256	PIC18F2450 219
PIC18C452 255	PIC18F2455 217
PIC18C452 Family 254	PIC18F2458 116
PIC18C601 242	PIC18F248 250
PIC18C658 240	PIC18F2480 213
PIC18C801 241	PIC18F24J10 171
PIC18C801 Family 241	PIC18F24J11 99
PIC18C858 239	PIC18F24J50 84
PIC18C858 Family 239	PIC18F24K20 138
PIC18F/LF26J11 Set 431	PIC18F24K22 271
PIC18F/LF26J50 Set 405	PIC18F2510 209
PIC18F/LF46J11 Set 443	PIC18F2515 208
PIC18F/LF46J50 Set 417	PIC18F252 253
PIC18F1220 232	PIC18F2520 226
PIC18F1230 231	PIC18F2523 222
	PIC18F2525 203
	PIC18F2539 246

PIC18F2550 216	PIC18F4410 207
PIC18F2553 116	PIC18F442 252
PIC18F258 249	PIC18F4420 225
PIC18F258 Set 768	PIC18F4423 221
PIC18F2580 213	PIC18F4431 139
PIC18F2585 200	PIC18F4431 Family 139
PIC18F25J10 170	PIC18F4431 Set 811
PIC18F25J10 Set 798	PIC18F4439 245
PIC18F25J11 98	PIC18F4450 218
PIC18F25J50 83	PIC18F4450 Family 218
PIC18F25K20 137	PIC18F4455 215
PIC18F25K22 269	PIC18F4458 115
PIC18F25K80 27	PIC18F448 248
PIC18F2610 208	PIC18F4480 212
PIC18F2610 Set 675	PIC18F44J10 170
PIC18F2620 202	PIC18F44J11 96
PIC18F2620 Set 691	PIC18F44J50 80
PIC18F2680 199	PIC18F44K20 135
PIC18F2682 197	PIC18F44K22 261
PIC18F2685 196	PIC18F4510 206
PIC18F26J11 97	PIC18F4515 205
PIC18F26J13 41	PIC18F452 251
PIC18F26J50 82	PIC18F452 Family 251
PIC18F26J53 37	PIC18F4520 224
PIC18F26K20 136	PIC18F4520 Family 224
PIC18F26K22 267	PIC18F4523 220
PIC18F26K80 26	PIC18F4523 Family 220
PIC18F27J13 42	PIC18F4525 201
PIC18F27J53 36	PIC18F4539 244
PIC18F2x21 Set 643	PIC18F4539 Family 243
PIC18F2x80/5 Set 660	PIC18F4550 214
PIC18F2xJ53 Set 287	PIC18F4550 Family 214
PIC18F4220 228	PIC18F4553 115
PIC18F4221 192	PIC18F4553 Family 115
PIC18F4320 228	PIC18F458 248
PIC18F4320 Family 228	PIC18F458 Family 248
PIC18F4321 191	PIC18F458 Set 776
PIC18F4321 Family 191	PIC18F4580 211
PIC18F4331 140	PIC18F4580 Family 211
PIC18F43K20 135	PIC18F4585 198
PIC18F43K22 263	PIC18F45J10 169

PIC18F45J10 Family 169	PIC18F6490 187
PIC18F45J10 Set 789	PIC18F6493 118
PIC18F45J11 94	PIC18F64J11 167
PIC18F45J50 79	PIC18F64J15 122
PIC18F45K20 134	PIC18F64J90 162
PIC18F45K22 259	PIC18F6520 238
PIC18F45K22 Family 258	PIC18F6525 175
PIC18F45K80 25	PIC18F6527 185
PIC18F4610 204	PIC18F6585 179
PIC18F4610 Family 204	PIC18F65J10 158
PIC18F4610 Set 668	PIC18F65J11 166
PIC18F4620 201	PIC18F65J15 157
PIC18F4620 Family 200	PIC18F65J50 128
PIC18F4620 Set 683	PIC18F65J90 161
PIC18F4680 198	PIC18F65K22 51
PIC18F4680 Family 197	PIC18F65K80 23
PIC18F4682 195	PIC18F65K90 65
PIC18F4685 195	PIC18F6620 237
PIC18F4685 Family 194	PIC18F6621 174
PIC18F46J11 93	PIC18F6622 184
PIC18F46J11 Family 93	PIC18F6627 184
PIC18F46J13 40	PIC18F6628 121
PIC18F46J50 78	PIC18F6680 178
PIC18F46J50 Family 78	PIC18F66J10 156
PIC18F46J53 35	PIC18F66J11 132
PIC18F46K20 133	PIC18F66J15 155
PIC18F46K20 Family 133	PIC18F66J16 131
PIC18F46K22 265	PIC18F66J50 127
PIC18F46K80 24	PIC18F66J55 126
PIC18F47J13 39	PIC18F66J60 150
PIC18F47J53 34	PIC18F66J65 149
PIC18F47Jx3 Family 33	PIC18F66J90 76
PIC18F4x21 Set 635	PIC18F66J93 70
PIC18F4x80/5 Set 652	PIC18F66K22 49
PIC18F4xJ53 Set 299	PIC18F66K27 50
PIC18F6310 190	PIC18F66K80 22
PIC18F6390 188	PIC18F66K80 Family 21
PIC18F6393 119	PIC18F66K90 64
PIC18F63J11 168	PIC18F66K95 63
PIC18F63J90 163	PIC18F6720 236
PIC18F6410 190	PIC18F6722 183

PIC18F6723 120	PIC18F8620 234
PIC18F67J10 154	PIC18F8621 172
PIC18F67J11 131	PIC18F8621 Family 172
PIC18F67J50 125	PIC18F8622 182
PIC18F67J60 148	PIC18F8627 181
PIC18F67J60 Set 542	PIC18F8628 120
PIC18F67J90 75	PIC18F8680 176
PIC18F67J93 69	PIC18F8680 Family 176
PIC18F67K22 48	PIC18F86J10 152
PIC18F67K90 62	PIC18F86J11 130
PIC18F6xK90 Set 326	PIC18F86J15 151
PIC18F6xK90_32 342	PIC18F86J16 129
PIC18F6xK90_non_32 326	PIC18F86J50 124
PIC18F8310 189	PIC18F86J55 123
PIC18F8390 186	PIC18F86J60 147
PIC18F8393 117	PIC18F86J65 146
PIC18F83J11 165	PIC18F86J72 56
PIC18F83J90 160	PIC18F86J90 73
PIC18F8410 188	PIC18F86J93 67
PIC18F8410 Family 188	PIC18F86K22 45
PIC18F8490 186	PIC18F86K27 46
PIC18F8490 Family 186	PIC18F86K90 60
PIC18F8493 117	PIC18F86K95 59
PIC18F8493 Family 117	PIC18F8720 233
PIC18F84J11 164	PIC18F8720 Family 233
PIC18F84J15 121	PIC18F8722 180
PIC18F84J15 Family 121	PIC18F8722 Family 180
PIC18F84J90 159	PIC18F8723 119
PIC18F8520 235	PIC18F8723 Family 119
PIC18F8525 173	PIC18F87J10 151
PIC18F8527 182	PIC18F87J10 Family 150
PIC18F8585 177	PIC18F87J11 128
PIC18F85J10 154	PIC18F87J11 Family 128
PIC18F85J11 164	PIC18F87J50 122
PIC18F85J11 Family 163	PIC18F87J50 Family 122
PIC18F85J15 153	PIC18F87J60 146
PIC18F85J50 125	PIC18F87J60 Set 551
PIC18F85J90 158	PIC18F87J72 55
PIC18F85J90 Family 158	PIC18F87J90 72
PIC18F85K22 47	PIC18F87J90 Family 72
PIC18F85K90 61	PIC18F87J93 66

PIC18F87J93 Family 66	PIC18LF46J50 85
PIC18F87K22 44	PIC18LF46K22 266
PIC18F87K22 Family 44	PIC18LF46K80 30
PIC18F87K90 58	PIC18LF65K80 29
PIC18F87K90 Family 58	PIC18LF66K80 28
PIC18F8xK90 Set 356	PMP (4xJ11 Family) 454
PIC18F8xK90_32 370	PMP (4xJ50 Family) 428
PIC18F8xK90_non_32 356	PMP (4xJ53 Family) 314
PIC18F96J60 145	PMP Example1 1192
PIC18F96J65 144	PMP Examples 1192
PIC18F96J72 53	PMP Functions 1187
PIC18F97J60 143	PMP Macros 1192
PIC18F97J60 Family 143	PMP Structs,Records,Enums 1192
PIC18F97J60 Set 560	pmp.h 1208
PIC18F97J72 52	portb.h 1208
PIC18F97J72 Family 52	Ports Example1 1093
PIC18LF13K22 114	Ports Examples 1093
PIC18LF13K50 111	Ports Functions 1088
PIC18LF14K22 113	Ports Macros 1093
PIC18LF14K50 110	Power Control PWM (PCPWM) 1173
PIC18LF23K22 274	PPS (2xJ11 Family) 434
PIC18LF24J11 107	PPS (2xJ50 Family) 408
PIC18LF24J50 92	PPS (2xJ53 Family) 289
PIC18LF24K22 272	PPS (4xJ11 Family) 445
PIC18LF25J11 106	PPS (4xJ50 Family) 419
PIC18LF25J50 90	PPS (4xJ53 Family) 301
PIC18LF25K22 270	PPS Example1 953
PIC18LF25K80 33	PPS Examples 953
PIC18LF26J11 104	PPS Macros 935
PIC18LF26J50 89	pps.h 1208
PIC18LF26K22 268	PPS_Input_Page1/2 935
PIC18LF26K80 32	PPS_Input_Page1/2_1 939
PIC18LF43K22 264	PPS_Input_Page3 948
PIC18LF44J11 103	PPS_Output_Page1 941
PIC18LF44J50 88	PPS_Output_Page2 942
PIC18LF44K22 262	PPS_Output_Page3 951
PIC18LF45J11 102	ProMPT.h 1209
PIC18LF45J50 87	Prototype_Page 1206
PIC18LF45K22 260	Pulse Width Modulation (PWM) 1026
PIC18LF45K80 31	putrs_1USART 1160
PIC18LF46J11 101	putrs_2USART 1161

puts_USART 1161	PWM (4610 Family) 670
puts_1USART 1162	PWM (4620 Family) 685
puts_2USART 1162	PWM (46K20 Family) 535
puts_I2C 1068	PWM (4x21 Family) 637
puts_I2C1 1069	PWM (4x8x Family) 654
puts_I2C2 1069	PWM (4xJ11 Family) 446
puts_SPI 1114	PWM (4xJ50 Family) 420
puts_SPI1 1114	PWM (67J60 Family) 544
puts_SPI2 1114	PWM (801 Family) 740
puts_USART 1163	PWM (8410 Family) 628
PWM (2610 Family) 677	PWM (8493 Family) 484
PWM (2620 Family) 695	PWM (84J15 Family) 503
PWM (2x21 Family) 645	PWM (858 Family) 733
PWM (2xJ53 Set) 291	PWM (8621 Family) 591
PWM (4431 Family) 813	PWM (8680 Family) 601
PWM (45K22 Set) 831	PWM (8720 Family) 724
PWM (4xJ53 Set) 304	PWM (8722 Family) 610
PWM (65K90 Set) 346	PWM (8723 Family) 493
PWM (66K80 Set) 279	PWM (87J10 Family) 572
PWM (6xK90 Set) 331	PWM (87J11 Family) 524
PWM (8490 Family) 620	PWM (87J50 Family) 514
PWM (85J90/11 Family) 582	PWM (87J60 Family) 553
PWM (85K90 Set) 374	PWM (87J93 Family) 397
PWM (87J90 Family) 386	PWM (97J60 Family) 562
PWM (8xK90 Set) 360	PWM (C452 Family) 762
PWM (97J72 Family) 318	PWM (F452 Family) 754
PWM (1320 Family) 785	PWM Example1 1060
PWM (14K22 Family) 467	PWM Examples 1060
PWM (14K50 Family) 459	PWM Functions 1026
PWM (2431 Family) 821	PWM Macros 1059
PWM (258 Family) 770	PWM Structs,Records,Enums 1059
PWM (25J10 Family) 800	pwm.h 1209
PWM (2x8x Family) 661	
PWM (2xJ11 Family) 435	R
PWM (2xJ50 Family) 409	Read_1USART 1163
PWM (4320 Family) 715	Read_2USART 1163
PWM (4450 Family) 710	Read_ADC 897
PWM (4553 Family) 476	Read_Byte_EEP 1061
PWM (458 Family) 778	Read_Capture 983
PWM (45J10 Family) 791	Read_Capture1 983
PWM (45xx Family) 702	Read_Capture10 986

Read_Capture2 984	RTCC (4xJ11 Family) 449
Read_Capture3 984	RTCC (4xJ50 Family) 424
Read_Capture4 984	RTCC (4xJ53 Family) 308
Read_Capture5 984	RTCC (87J93 Family) 400
Read_Capture6 985	RTCC Example1 1104
Read_Capture7 985	RTCC Examples 1104
Read_Capture8 985	RTCC Functions 1095
Read_Capture9 986	RTCC Macros 1104
Read_DSGPR 931	RTCC Structs,Records,Enums 1104
Read_ECapture1 986	rtcc.h 1209
Read_ECapture2 986	RTCC_InitClock 1095
Read_ECapture3 987	RTCC_ReadAlrmDate 1095
Read_Flash_Page1 1198	RTCC_ReadAlrmTime 1096
Read_I2C 1067	RTCC_ReadAlrmTimeDate 1096
Read_I2C1 1067	RTCC_ReadDate 1096
Read_I2C2 1068	RTCC_ReadTime 1097
Read_Mwire 1171	RTCC_ReadTimeDate 1097
Read_Mwire1 1170	RTCC_SetAlarmRpt 1098
Read_Mwire2 1171	RTCC_SetAlarmRptCount 1104
Read_SPI 1113	RTCC_SetCalibration 1103
Read_SPI1 1113	RTCC_SetChimeEnable 1103
Read_SPI2 1113	RTCC_WriteAlrmDate 1102
Read_Timer 1137	RTCC_WriteAlrmTime 1101
Read_Timer0 1137	RTCC_WriteAlrmTimeDate 1101
Read_Timer1 1138	RTCC_WriteDate 1100
Read_Timer3 1138	RTCC_WriteTime 1099
Read_Timer5 1139	RTCC_WriteTimeDate 1099
Read_Timer7_Page1 1139	RTCC_WrOn 1098
Read_USART 1164	
Real Time Clock & Calender (RTCC) 1095	
Release Notes 5	
reset.h 1209	
RTCC (65K90 Set) 351	
RTCC (6xK90 Set) 336	
RTCC (85K90 Set) 378	
RTCC (87J90 Family) 389	
RTCC (8xK90 Set) 365	
RTCC (97J72 Family) 320	
RTCC (2xJ11 Family) 437	
RTCC (2xJ50 Family) 412	
RTCC (2xJ53 Family) 294	
	S
	SelChanConv_ADC 889
	SelChanConvADC_Page1 889
	SelChanConvADC_Page2 890
	SelChanConvADC_Page3 891
	SelChanConvADC_Page4 892
	SelChanConvADC_Page5 893
	SelChanConvADC_Page6 894
	SelChanConvADC_Page7 895
	SelChanConvADC_Page8 896
	Serial Peripheral Interface (SPI) 1108
	Set_TmrCCPSrc 1141

Set_TmrCCPSrc_Page1 1141	SlaveWriteBuffers_PMP 1191
Set_TmrCCPSrc_Page2 1142	Software RTCC (SW_RTCC) 1106
Set_TmrCCPSrc_Page3 1142	SPI (66K80 Family) 282
SetAddress_PMP 1187	SPI (85J90/11 Family) 585
SetChan_ADC 882	SPI (87J90 Family) 390
SetChanADC_Page1 882	SPI (97J72 Family) 322
SetChanADC_Page2 883	SPI (14K22 Family) 470
SetChanADC_Page3 884	SPI (14K50 Family) 462
SetChanADC_Page4 885	SPI (2431 Family) 822
SetChanADC_Page5 886	SPI (258 Family) 771
SetChanADC_Page6 887	SPI (25J10 Family) 802
SetChanADC_Page7 887	SPI (2610 Family) 679
SetChanADC_Page8 889	SPI (2620 Family) 694
SetDC_EPWM1 1054	SPI (2x21 Family) 646
SetDC_EPWM2_Page1 1054	SPI (2x8x Family) 663
SetDC_EPWM3_Page1 1055	SPI (2xJ11 Family) 439
SetDC_PWM 1047	SPI (2xJ50 Family) 410
SetDC_PWM1 1047	SPI (2xJ53 Family) 293
SetDC_PWM10_Page1 1053	SPI (4320 Family) 717
SetDC_PWM2 1048	SPI (4431 Family) 815
SetDC_PWM3 1048	SPI (4539 Family) 748
SetDC_PWM4 1049	SPI (4553 Family) 479
SetDC_PWM5 1050	SPI (458 Family) 779
SetDC_PWM6_Page1 1050	SPI (45J10 Family) 793
SetDC_PWM7_Page1 1051	SPI (45K22 Family) 834
SetDC_PWM8_Page1 1052	SPI (45xx Family) 703
SetDC_PWM9_Page1 1052	SPI (4610 Family) 671
Setdc0_pcpwm 1182	SPI (4620 Family) 687
Setdc1_pcpwm 1182	SPI (46K20 Family) 538
Setdc2_pcpwm 1183	SPI (4x21 Family) 638
Setdc3_pcpwm 1183	SPI (4x8x Family) 655
SetOutput_EPWM1 1045	SPI (4xJ11 Family) 451
SetOutput_EPWM2_Page1 1046	SPI (4xJ50 Family) 425
SetOutput_EPWM3_Page1 1046	SPI (4xJ53 Family) 307
SetOutput_PWM 1043	SPI (65K90 Set) 352
SetOutput_PWM1 1043	SPI (67J60 Family) 547
SetOutput_PWM2 1043	SPI (6xK90 Set) 338
SetOutput_PWM3 1044	SPI (801 Family) 742
SlaveReadBufferN_PMP 1190	SPI (8410 Family) 630
SlaveReadBuffers_PMP 1190	SPI (8490 Family) 622
SlaveWriteBufferN_PMP 1191	SPI (8493 Family) 487

SPI (84J15 Family) 507	SW_RTCC (14K22 Family) 470
SPI (853K90 Set) 380	SW_RTCC (14K50 Family) 462
SPI (858 Family) 735	SW_RTCC (2431 Family) 824
SPI (8621 Family) 595	SW_RTCC (258 Family) 773
SPI (8680 Family) 602	SW_RTCC (25J10 Family) 803
SPI (8720 Family) 726	SW_RTCC (2610 Family) 680
SPI (8722 Family) 612	SW_RTCC (2620 Family) 696
SPI (8723 Family) 497	SW_RTCC (2x21 Family) 648
SPI (87J10 Family) 576	SW_RTCC (2x8x Family) 665
SPI (87J11 Family) 528	SW_RTCC (2xJ11 Family) 438
SPI (87J50 Family) 518	SW_RTCC (2xJ50 Family) 414
SPI (87J60 Family) 556	SW_RTCC (2xJ53 Family) 296
SPI (87J93 Family) 399	SW_RTCC (4320 Family) 719
SPI (8xK90 Set) 366	SW_RTCC (4431 Family) 817
SPI (97J60 Family) 565	SW_RTCC (4450 Family) 711
SPI (C452 Family) 764	SW_RTCC (4539 Family) 750
SPI (F452 Family) 756	SW_RTCC (4553 Family) 478
SPI Example1_Master 1116	SW_RTCC (458 Family) 781
SPI Example1_Slave 1117	SW_RTCC (45J10 Family) 795
SPI Examples 1116	SW_RTCC (45xx Family) 705
SPI Functions 1108	SW_RTCC (4610 Family) 673
SPI Macros 1116	SW_RTCC (4620 Family) 688
spi.h 1209	SW_RTCC (46K20 Family) 537
SPI_gets 1115	SW_RTCC (4x21 Family) 640
SPI_Open 1108	SW_RTCC (4x8x Family) 656
SPI_puts 1113	SW_RTCC (4xJ11 Family) 451
SPI_Read 1113	SW_RTCC (4xJ50 Family) 425
SPI_Write 1111	SW_RTCC (4xJ53 Family) 310
SW Licence Agreement 4	SW_RTCC (67J60 Family) 546
SW_RTCC (45K22 Family) 837	SW_RTCC (801 Family) 743
SW_RTCC (65K90 Set) 352	SW_RTCC (8410 Family) 632
SW_RTCC (66K80 Family) 283	SW_RTCC (8490 Family) 622
SW_RTCC (6xK90 Set) 337	SW_RTCC (8493 Family) 487
SW_RTCC (85J90/11 Family) 585	SW_RTCC (84J15 Family) 507
SW_RTCC (85K90 Set) 379	SW_RTCC (858 Family) 736
SW_RTCC (8621 Family) 594	SW_RTCC (8680 Family) 604
SW_RTCC (87J90 Family) 390	SW_RTCC (8720 Family) 728
SW_RTCC (8xK90 Set) 366	SW_RTCC (8722 Family) 614
SW_RTCC (97J72 Family) 322	SW_RTCC (8723 Family) 496
SW_RTCC (1320 Family) 787	SW_RTCC (87J10 Family) 576
SW_RTCC (1330 Family) 809	SW_RTCC (87J11 Family) 528

SW_RTCC (87J50 Family) 517
 SW_RTCC (87J60 Family) 555
 SW_RTCC (87J93 Family) 402
 SW_RTCC (97J60 Family) 565
 SW_RTCC (C452 Family) 766
 SW_RTCC (F452 Family) 758
 SW_RTCC Example1 1107
 SW_RTCC Examples 1107
 SW_RTCC Functions 1106

T

TIMERS 1118

Timers (1320 Family) 787
 Timers (1330 Family) 809
 Timers (14K22 Family) 471
 Timers (14K50 Family) 463
 Timers (2431 Family) 824
 Timers (258 Family) 773
 Timers (25J10 Family) 804
 Timers (2610 Family) 681
 Timers (2620 Family) 697
 Timers (2x21 Family) 648
 Timers (2x8x Family) 665
 Timers (2xJ11 Family) 440
 Timers (2xJ50 Family) 414
 Timers (2xJ53 Set) 296
 Timers (4320 Family) 719
 Timers (4431 Family) 816
 Timers (4450 Family) 711
 Timers (4539 Family) 750
 Timers (4553 Family) 480
 Timers (458 Family) 781
 Timers (45J10 Family) 795
 Timers (45K22 Set) 837
 Timers (45xx Family) 706
 Timers (4610 Family) 673
 Timers (4620 Family) 689
 Timers (46K20 Family) 539
 Timers (4x21 Family) 640
 Timers (4x8x Family) 657
 Timers (4xJ11 Family) 452

Timers (4xJ50 Family) 426
 Timers (4xJ53 Set) 310
 Timers (65K90 Set) 354
 Timers (66K80 Set) 283
 Timers (67J60 Family) 548
 Timers (6xK90 Set) 340
 Timers (801 Family) 744
 Timers (8410 Family) 632
 Timers (8490 Family) 623
 Timers (8493 Family) 488
 Timers (84J15 Family) 508
 Timers (858 Family) 736
 Timers (85J90/11 Family) 586
 Timers (85K90 Set) 382
 Timers (8621 Family) 596
 Timers (8680 Family) 605
 Timers (8720 Family) 728
 Timers (8722 Family) 615
 Timers (8723 Family) 498
 Timers (87J10 Family) 577
 Timers (87J11 Family) 529
 Timers (87J50 Family) 519
 Timers (87J60 Family) 557
 Timers (87J90 Family) 391
 Timers (87J93 Family) 402
 Timers (8xK90 Set) 368
 Timers (97J60 Family) 567
 Timers (97J72 Family) 323
 Timers (C452 Family) 766
 Timers (F452 Family) 758
 TIMERS Example1 1146
 TIMERS Examples 1146
 TIMERS Functions 1119
 TIMERS Macros 1146
 TIMERS Structs,Records,Enums 1146
 timers.h 1209

U

ULPWakeUp_Enable 931
 Universal Asynchronous/Synchronous Reciever Transmitter (USART) 1148

update_SW_RTCC 1106	USART (8493 Family) 489
USART (87J90 Family) 392	USART (84J15 Family) 509
USART (97J72 Family) 324	USART (858 Family) 737
USART (1320 Family) 788	USART (85J90/11 Family) 587
USART (1330 Family) 810	USART (85K90 Set) 381
USART (14K22 Family) 472	USART (8621 Family) 597
USART (14K50 Family) 464	USART (8680 Family) 605
USART (2431 Family) 825	USART (8720 Family) 729
USART (258 Family) 774	USART (8722 Family) 616
USART (25J10 Family) 804	USART (8723 Family) 499
USART (2610 Family) 681	USART (87J10 Family) 578
USART (2620 Family) 698	USART (87J11 Family) 530
USART (2x21 Family) 649	USART (87J50 Family) 520
USART (2x8x Family) 666	USART (87J60 Family) 558
USART (2xJ11 Family) 441	USART (87J93 Family) 403
USART (2xJ50 Family) 415	USART (8xK90 Set) 367
USART (2xJ53 Family) 297	USART (97J60 Family) 567
USART (4320 Family) 720	USART (C452 Family) 767
USART (4431 Family) 817	USART (F452 Family) 759
USART (4450 Family) 712	USART Example1 1164
USART (4539 Family) 751	USART Examples 1164
USART (4553 Family) 480	USART Functions 1148
USART (458 Family) 782	USART Macros 1164
USART (45J10 Family) 796	USART Structs,Records,Enums 1164
USART (45K22 Set) 839	usart.h 1209
USART (45xx Family) 706	USART_baud 1155
USART (4610 Family) 674	USART_gets 1159
USART (4620 Family) 689	USART_Open 1148
USART (46K20 Family) 539	USART_putrs 1160
USART (4x21 Family) 641	USART_puts 1162
USART (4x8x Family) 658	USART_Read 1163
USART (4xJ11 Family) 453	USART_Write 1154
USART (4xJ50 Family) 427	
USART (4xJ53 Family) 311	W
USART (65K90 Set) 353	Write_1USART 1154
USART (66K80 Set) 285	Write_2USART 1155
USART (67J60 Family) 549	Write_Byte_EEP 1061
USART (6xK90 Set) 339	Write_I2C 1066
USART (801 Family) 744	Write_I2C1 1066
USART (8410 Family) 633	Write_I2C2 1067
USART (8490 Family) 624	Write_Mwire 1170

Write_Mwire1 1169
Write_Mwire2 1169
Write_SPI 1111
Write_SPI1 1112
Write_SPI2 1112
Write_Timer 1139
Write_Timer0 1139
Write_Timer1 1140
Write_Timer3 1140
Write_Timer5 1140
Write_Timer7_Page1 1141
Write_USART 1155
WriteBlock_Flash_Page1 1198
WriteBlock_Flash_Page2 1199
WriteBlock_Flash_Page3 1199
WriteBlock_Flash_Page4 1200
WriteBlock_Flash_Page5 1200
WriteBlock_Flash_Page6 1201
WriteBytes_Flash_Page1 1202
WriteBytes_Flash_Page2 1203
WriteBytes_Flash_Page3 1203
WriteWord_Flash_Page1 1202