

```
;      File: EEPROM INTERAFACE
;      INTERNAL OSC. RUN AT 8MHZ, 4 BIT LCD INTERFACE
;      LCD CONNECTED TO PORTD (LCD PIN D4 TO D7:CONNECT TO RD0-RD3 , RD4-EN,
;      R5-RS.), LED CONNECT TO RC0- LED 10, RC1- LED 11, RC2-LED 8 , RE1 AND RE2- LED14
;      AND LED13,RA7 AND RA6-LED12 AND LED9
```

```
list      p=16f887
#include p16f887.inc
; Set configuration bits using definitions from the include file, p16f84.inc
;      __config      __CONFIG1, _CP_OFF & _WDT_OFF & _INTOSCIO & _LVP_OFF
;      __CONFIG      __CONFIG1, _FOSC_INTOSC & _WDTE_OFF & _PWRTE_ON &
_MCLRE_OFF & _CP_OFF & _BOREN_ON & _CLKOUTEN_OFF
;      __CONFIG      __CONFIG2, _WRT_OFF & _STVREN_ON & _BORV_LO &
_LPBOR_OFF & _LVP_OFF & _DEVID1 & _IDLOC0
__CONFIG      __CONFIG1, _PWRTE_ON & _MCLRE_ON & _CP_OFF & _BOREN_OFF &
_WDT_OFF & _INTOSCIO & _LVP_OFF & _IESO_OFF & _FCMEN_OFF & _CPD_OFF
```

```
;__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _XT_OSC &
_WRT_ENABLE_OFF & _LVP_OFF & _DEBUG_OFF & _CPD_OFF
```

```
;.....PORTD BITS.....
```

```
;...PORTD BITS.....
```

```
RS      EQU      0X05
EN      EQU      0X04
D7      EQU      0X03
D6      EQU      0X02
D5      EQU      0X01
D4      EQU      0X00
```

```
;      #define      D4 PORTB,0 ;
```

```
; Set User ID Memory
```

```
CBLOCK 0X20
```

```
BIN1
```

```
TEMP
```

```
      ;DECLARATION OF VARIABLE
```

```
TEMP1
```

```
LOOPENT
```

```
TEMP2
```

```
COUNTER1
```

```
COUNTER2
```

```
    BIN
```

```
    VALUE
```

```
    VALUE1
```

```
    VALUE2
```

```
    FIRSTNO
```

```
    SECONDNO
```

```
    THIRDN0
```

```
count
```

```
count1
```

```

huns
    tens
    ones
    ENDC
    ORG 0X0000
    GOTO MAIN
    ORG 0X0005
MAIN
    BANKSEL OSCCON ;INTERNAL OSCILLATOR USED FOR 8MHZ
    MOVLW 0X71
    MOVWF OSCCON
    BANKSEL PORTE
    CLRF PORTE ; MAKE PORT D O/P
    BANKSEL TRISE
    CLRF TRISE
    BANKSEL PORTD
    CLRF PORTD ; MAKE PORT D O/P
    BANKSEL TRISD
    CLRF TRISD ;ALL PINS OF PORT D CONFIGURED AS
OUTPUT
    MOVLW 0X00
    BANKSEL ANSELH ; PORTB KEPT AT DIGITAL NOT ANALOG
    MOVWF ANSELH
    BANKSEL TRISB
    MOVLW B'00111111' ;PIN RA0 CONFIGURE AS INPUT
    MOVWF TRISB
    BANKSEL PORTB
    MOVWF PORTB
    BANKSEL PORTC
    CLRF PORTC
    BANKSEL TRISC
    CLRF TRISC
    MOVLW B'00000001' ;PIN RA0 CONFIGURE AS INPUT
    BANKSEL PORTA
    CLRF PORTA
    BANKSEL TRISA
    MOVWF TRISA
    BANKSEL ANSEL
    MOVWF ANSEL
    MOVLW H'FF'
    BANKSEL PORTC
    MOVWF PORTC
    ; MOVLW H'1C'
    ; BANKSEL BIN1
    ; MOVWF BIN1
    CALL LCDINI
    ;BANKSEL VALUE
    ; MOVLW H'1C'
    ; MOVWF VALUE
    ; MOVWF BIN
    ; CALL EDWRITE
    ; CALL EDREAD
    ;CALL CONVERSION
    ; MOVLW B'10000000'

```

```

        ; CALL COMMAND
        ; CALL DELAY
        ; CALL DISPLAY
        ;CALL EDREAD
;SLEEP
BACK
        CALL EDWRITE
        CALL EDREAD
        BANKSEL PORTB
        BTFSC PORTB,RB3
        CALL DELAY
        BTFSC PORTB,RB3
        GOTO MAIN1
        GOTO MAIN2
MAIN2:
        ;CALL EDREAD
        INCF VALUE
        MOVF VALUE,0
        MOVWF BIN
        CALL CONVERSION
        ; MOVLW B'10000000'
        ; CALL COMMAND
        CALL DELAY
        CALL DISPLAY
        CALL EDWRITE
        CALL EDREAD
        BANKSEL PORTE
        BCF PORTE,2
        GOTO BACK
MAIN1:
        ;CALL EDREAD
        ;CALL EDWRITE
        ; CALL EDREAD
        ;MOVF VALUE1,0
        ; MOVWF BIN1
        CALL CONVERSION1
        CALL DISPLAY1
        BANKSEL PORTE
        BSF PORTE,2
        ;CALL EDWRITE
        ; CALL EDREAD
        GOTO BACK
        ;sleep

EDWRITE:
        BANKSEL EECON1
        BTFSC EECON1,WR      ;Wait for
        GOTO $-1
        ; BANKSEL VALUE
        ; MOVF VALUE,0
        ; BANKSEL EEDATA
        ; MOVWF EEDATA
        BANKSEL EEADR
        MOVLW H'05'

```

```

        MOVWF EEADR
        BANKSEL VALUE
        MOVF VALUE,0
NOP
        BANKSEL EEDATA
        MOVWF EEDATA
        BANKSEL EECON1
        BCF EECON1,EEPGD      ; ACCESS DATA MEMORY
        BSF EECON1,WREN
        BCF INTCON,GIE        ;Disable INTs
        BANKSEL EECON2
        MOVLW 0X55             ; Unlock-write voodoo
        MOVWF EECON2
        MOVLW 0XAA
        MOVWF EECON2
        BANKSEL EECON1
        BSF EECON1,WR
        NOP
        NOP      ;INITIATE WRITE CYCLE
JUMP
        BTFSC EECON1,WR      ;
        GOTO JUMP
        BCF EECON1,WREN
        BCF EECON1,WR
        ; BSF EECON1,EEPGD
        RETURN
EDREAD:
        BANKSEL EECON1
        ; BTFSC EECON1,WR
        ; GOTO EDREAD
        BCF EECON1,EEPGD
        BANKSEL EEADR
        MOVLW H'05'           ; load eeaddress
        MOVWF EEADR           ; store address
        BSF EECON1,RD
        BANKSEL EEDATA
        MOVF EEDATA,W         ; read data to w
        BANKSEL BIN1
        MOVWF BIN1
        BANKSEL VALUE1
        MOVWF VALUE1
        BANKSEL VALUE
        MOVWF VALUE
        BANKSEL EECON1
        BCF EECON1,RD
        BSF EECON1,EEPGD

;BANKSEL BIN1
; MOVWF BIN1
; BANKSEL VALUE1
; MOVWF VALUE1
;BANKSEL VALUE
; MOVWF VALUE
CALL DELAY

```

RETURN

DISPLAY

MOVLW B'10000000'

CALL COMMAND

MOVF huns,0

CALL WRITE

MOVLW B'10000001'

CALL COMMAND

MOVF tens,0

CALL WRITE

MOVLW B'10000010'

CALL COMMAND

MOVF ones,0

CALL WRITE

RETURN

LCDINI

CALL DELAY

MOVLW B'00101000'; FUNCTION SET(28H) TO

CALL COMMAND ; COMMAND TO TRANSFER SAME ON LCD

MOVLW B'00101000'; FUNCTION SET(28H) TO

CALL COMMAND ; COMMAND TO TRANSFER SAME ON LCD

MOVLW B'00101000'; FUNCTION SET(28H) TO

CALL COMMAND ; COMMAND TO TRANSFER SAME ON LCD

MOVLW B'00000001' ; CLEAR SCREEN

CALL COMMAND

MOVLW B'00001100' ;0C FOR DISPLAY SHIFT

CALL COMMAND

MOVLW B'00000110' ;06 H FOR ENTRY MODE SET

CALL COMMAND

RETURN

COMMAND:

BANKSEL TEMP2

MOVWF TEMP2

ANDLW B'11110000' ; AND WITH 0F0 TO KEEP MSB

NOP

BANKSEL TEMP1

MOVWF TEMP1 ; SAFE SAME IN TEMP1

SWAPF TEMP1,0 ;SWAP DATA OF TEMP1

MOVF TEMP1 ; MOVE CONTANT OF TEMP1 INTO W

ADDLW B'00010000' ;MAKE EN=1

NOP

BANKSEL PORTD

MOVWF PORTD

NOP

NOP

ANDLW B'00001111'

BANKSEL PORTD

MOVWF PORTD

NOP

NOP

BANKSEL TEMP2

MOVF TEMP2,W

```

ANDLW B'00001111' ;AND WITH 0FH
ADDLW B'00010000' ;EN=1
BANKSEL PORTD
MOVWF PORTD
NOP
NOP
ANDLW B'00001111' ;EN=0
BANKSEL PORTD
MOVWF PORTD
CALL DELAY
RETURN

```

#### WRITE

```

BANKSEL TEMP2
MOVWF TEMP2 ;
ANDLW B'11110000' ;
MOVWF TEMP1 ; SAFE SAME IN TEMP1
SWAPF TEMP1 ;SWAP DATA OF TEMP1
MOVF TEMP1,0 ; MOVE CONTANT OF TEMP1 INTO W
ADDLW B'00110000' ;MAKE EN=1,RS=1
NOP
BCF STATUS,RP0
MOVWF PORTD
NOP
NOP
ANDLW B'00001111'
ADDLW B'00100000' ;EN=0, RS=1
BCF STATUS,RP0
MOVWF PORTD
NOP
BANKSEL TEMP2
MOVF TEMP2,W ;MOVE CONTANT OF TEMP INTO W
ANDLW B'00001111' ;AND WITH 0FH
ADDLW B'00110000' ;EN=1,RS=1
BANKSEL PORTD
MOVWF PORTD
NOP
ANDLW B'00001111' ;EN=0
ADDLW B'00100000' ;EN=0,RS=1
BCF STATUS,RP0
MOVWF PORTD
CALL DELAY
RETURN

```

#### DELAY

```

MOVLW H'64'
MOVWF COUNTER2

```

#### LOOP2

```

MOVLW H'FF'
MOVWF COUNTER1

```

#### LOOP4

```

DECFSZ COUNTER1 ;2
GOTO LOOP4 ;2

```

```

DECFSZ COUNTER2      ;2
GOTO LOOP2           ;2
RETURN

```

ADCINI:

```

    BANKSEL ANSELH
    CLRF ANSELH
    BANKSEL ADCON1
    BCF ADCON1,ADFM    ;CONVERSION RESULT LEFT JUSTIFIED
    BCF ADCON1,VCFG1    ;VSS IS USED AS NEGATIVE
    BCF ADCON1,VCFG0    ;VDD USED AS VREF. INPUT
    BANKSEL ADCON0      ;
    MOVLW B'00000001'
    MOVWF ADCON0        ; A/D CONVERTER ENABLE
LOOP
    BANKSEL ADCON0
    BTFSC ADCON0,1      ; IF BIT B1 OF ADCON0 IS ONE, NEXT INSTRUCTION
EXECUTED, IF ZERO,NEXT INSTRUCTION IS DISCARD (ZERO CONFIRM AD CONVERSION
COMPLETED)
    GOTO LOOP
    BANKSEL ADRESH
    MOVF ADRESH,W
    BANKSEL BIN
    MOVWF BIN
    BANKSEL VALUE
    MOVWF VALUE
    BSF ADCON0,1        ;SET BIT 1
    RETURN

```

CONVERSION:

```

    movlw 8
    movwf count
    clrf huns
    clrf tens
    clrf ones

BCDADD3

    movlw 5
    subwf huns, 0
    btfsc STATUS, C
    CALL ADD3HUNS
    movlw 5
    subwf tens, 0
    btfsc STATUS, C
    CALL ADD3TENS
    movlw 5
    subwf ones, 0
    btfsc STATUS, C
    CALL ADD3ONES
    decf count, 1
    bcf STATUS, C
    rlf BIN, 1
    rlf ones, 1
    btfsc ones,4 ;
    CALL CARRYONES

```

```

    rlf tens, 1
    btfsc tens, 4 ;
    CALL CARRYTENS
    rlf huns, 1
    bcf STATUS, C
    movf count, 0
    btfss STATUS, Z
    GOTO BCDADD3
    movf huns, 0 ; add ASCII Offset
    addlw h'30'
    movwf huns
    movf tens, 0 ; add ASCII Offset
    addlw h'30'
    movwf tens
    movf ones, 0 ; add ASCII Offset
    addlw h'30'
    movwf ones
    RETURN

```

ADD3HUNS

```

    movlw 3
    addwf huns, 1
    RETURN

```

ADD3TENS

```

    movlw 3
    addwf tens, 1
    RETURN
    ADD3ONES
    movlw 3
    addwf ones, 1
    RETURN

```

CARRYONES

```

    bcf ones, 4
    bsf STATUS, C
    RETURN

```

CARRYTENS

```

    bcf tens, 4
    bsf STATUS, C
    RETURN

```

CONVERSION1:

```

;banksel manoj
    movlw 8

```

SECONDNO, ONES THIRDN0

```

    movwf count1
    clrf FIRSTNO ; HUNS FIRSTNO, TENS
    clrf SECONDNO
    clrf THIRDN0

```

BCDADD4

```

    movlw 5
    subwf FIRSTNO, 0
    btfsc STATUS, C
    CALL ADD3HUNS1
    movlw 5

```



```

        subwf SECONDNO,0
        btfsc STATUS,C
CALL ADD3TENS1
        movlw 5
        subwf THIRDN0,0
        btfsc STATUS,C
        CALL ADD3ONES1
        decf count1,1
        bcf STATUS, C
        rlf BIN1,1
        rlf THIRDN0,1
        btfsc THIRDN0,4 ;
CALL CARRYONES1
        rlf SECONDNO,1
        btfsc SECONDNO,4 ;
        CALL CARRYTENS1
        rlf FIRSTNO,1
        bcf STATUS,C
        movf count1,0
        btfss STATUS, Z
        GOTO BCDADD4
        movf FIRSTNO,0 ; add ASCII Offset
        addlw h'30'
        movwf FIRSTNO
        movf SECONDNO,0 ; add ASCII Offset
        addlw h'30'
        movwf SECONDNO
        movf THIRDN0,0 ; add ASCII Offset
        addlw h'30'
        movwf THIRDN0
        RETURN

```

ADD3HUNS1

```

        movlw 3
        addwf FIRSTNO,1
        RETURN

```

ADD3TENS1

```

        movlw 3
        addwf SECONDNO,1
        RETURN

```

ADD3ONES1

```

        movlw 3
        addwf THIRDN0,1
        RETURN

```

CARRYONES1

```

        bcf THIRDN0,4
        bsf STATUS, C
        RETURN

```

CARRYTENS1

```

        bcf SECONDNO,4
        bsf STATUS,C
        RETURN

```

```

DISPLAY1
MOVLW H'8C'
CALL COMMAND
;MOVF huns,0
MOVF FIRSTNO,0
    CALL WRITE
MOVLW H'8D'
CALL COMMAND
;MOVF tens ,0
MOVF SECONDDNO,0
    CALL WRITE
MOVLW H'8E'
CALL COMMAND
;MOVF ones,0
    MOVF THIRDDNO,0
        CALL WRITE
    RETURN
END

```

```

;DelayOneSecond
; movlw D'6'
; movwf CounterC
; movlw D'24'
; movwf CounterB
; movlw D'168'
; movwf CounterA
;loop
; decfsz CounterA,1
;goto loop
; decfsz CounterB,1
; goto loop
;decfsz CounterC,1
; goto loop
;retlw

```