





```

1:
2:             MPT, MALA
3: ;*****
4: $ nomod51
5: $ include (89c52.mcu)
6: ;*****
7: ;             LCD & KEYBOARD EQUALISATION
8: ;*****
9:     lcd_port      equ     p0
10:    key_port       equ     p1
11: ;*****
12:    rs_lcd         bit     p1.0
13:    en_lcd         bit     p1.1
14: ;*****
15:    flag           bit     8
16:    alarm_flag     bit     9
17:    sec_flag       bit     10
18:    temp_flag      bit     11
19:    out_flag       bit     12
20: ;*****
21:    lcd_data       equ     48
22:    first_line     equ     80h
23:    second_line    equ     0c0h
24:    lcd_clear      equ     01h
25:    cursor_on      equ     0fh
26:    cursor_off     equ     0ch
27: ;*****
28: ;             RTC EQUALISATION
29: ;*****
30:    ds1302clk      bit     p1.6
31:    ds1302data     bit     p1.7
32:    ds1302rst      bit     p2.0
33: ;*****
34:    alarm_active   equ     32
35:    xmtdat         equ     49
36: ;*****
37:    seconds        equ     50
38:    minits         equ     51
39:    hour           equ     52
40:    date           equ     53
41:    month          equ     54
42:    days           equ     55
43:    year           equ     56
44:    control        equ     57
45: ;*****
46:    cammand        equ     58
47:    bitcnt         equ     59
48:    bytecnt        equ     60
49: ;*****
50:    alarm_hour     equ     61
51:    alarm_minits   equ     62
52:    alarm_time     equ     63
53: ;*****
54: ;             AUXILURY EQUALISATION
55: ;*****
56:    select_fun     equ     64
57:    alarm_select   equ     65
58: ;*****
59:    alarm_count    equ     66
60:    set_count      equ     67
61:    delay_timer1   equ     68
62:    delay_timer2   equ     69
63:    sec_temp       equ     70
64:    alarm_seq      equ     71
65: ;*****
66:    check_hour     equ     72
67:    check_minits   equ     73
68:    check_time     equ     74
69: ;*****
70:    ram_space      equ     75
71: ;*****
72:    relay          bit     p3.0
73: ;*****
74:    org            sint
75:    jmp            programme
76: ;*****
77: ;*****
78: ;             PROGRAMME STARTS FROM HERE
79: ;*****
80: ;*****
81: PROGRAMME:
82: ;*****
83:    mov            p0,#0
84:    mov            p1,#0ffh
85:    mov            p2,#0
86:    mov            p3,#0
87:    clr            c
88:    clr            flag

```

```

89:      clr      rs_lcd
90:      clr      en_lcd
91:      clr      relay
92:      clr      out_flag
93:      mov      sp,#128
94:      call     lcd_initialize
95:      mov      select_fun,#1
96:      mov      alarm_select,#1
97:      mov      alarm_active,#1
98: ;*****
99:      mov      a,#first_line
100:     call     cmd_write
101:     mov      dptr,#msg_03
102:     call     dis_message
103:
104:     mov      a,#second_line
105:     call     cmd_write
106:     mov      dptr,#msg_04
107:     call     dis_message
108: ;*****
109:     call     delay_long
110:     call     delay_long
111:     call     delay_long
112: ;*****
113:     mov      cammand,#8eh
114:     mov      bytecnt,#1
115:     mov      r0,#xmtdat
116:     mov      xmtdat,#0
117:     call     send_byte
118: ;*****
119:     mov      cammand,#90h
120:     mov      bytecnt,#1
121:     mov      r0,#xmtdat
122:     mov      xmtdat,#10100101b
123:     call     send_byte
124: ;*****
125:     mov      cammand,#80h
126:     mov      bytecnt,#1
127:     mov      r0,#xmtdat
128:     mov      xmtdat,#0
129:     call     send_byte
130: ;*****
131:     mov      cammand,#0fdh
132:     mov      bytecnt,#1
133:     mov      r1,#xmtdat
134:     call     receive_byte
135:     mov      alarm_seq,xmtdat
136:     call     read_alarm
137:     call     load_alarm
138: ;*****
139: ;*****
140: ;*****
141: ;      SWITCH TO SELECTION SCREEN
142: ;*****
143: ;*****
144: ;*****
145:     mov      seconds,sec_temp
146:     call     main_timer
147: START_SFUN:
148:     call     keyscan
149:     jz       start_sfun
150: ;*****
151:     cjne     a,#1,st_001
152:     call     select_function
153:     call     delay_sec
154:     jmp      start_sfun
155: st_001:
156:     cjne     a,#2,st_002
157:     call     function_up
158:     call     select_function
159:     call     delay_sec
160:     jmp      start_sfun
161: st_002:
162:     cjne     a,#3,st_003
163:     call     function_down
164:     call     select_function
165:     call     delay_sec
166:     jmp      start_sfun
167: st_003:
168:     cjne     a,#4,st_004
169: ;*****
170:     mov      a,select_fun
171:     cjne     a,#1,main_001
172:     call     main_timer
173:     call     select_function
174:     call     delay_sec
175:     jmp      start_sfun
176: main_001:

```

```

177:      cjne    a,#2,main_002
178:      call    set_select_all
179:      call    select_function
180:      call    delay_sec
181:      jmp     start_sfun
182:  main_002:
183:      cjne    a,#3,main_003
184:      call    settimedate
185:      call    select_function
186:      call    delay_sec
187:      jmp     start_sfun
188:  main_003:
189:      cjne    a,#4,main_004
190:      call    information
191:      call    select_function
192:      call    delay_sec
193:      jmp     start_sfun
194:  main_004:
195:      cjne    a,#5,st_004
196:  ;*****
197:  st_004:
198:      jmp     start_sfun
199:  ;*****
200:  ;*****
201:  ;*****
202:  ; SUBROUTINE TO DISPLAY TIME & DATE
203:  ;*****
204:  ;*****
205:  ;*****
206:  MAIN_TIMER:
207:      clr     sec_flag
208:      mov     a,#cursor_off
209:      call    cmd_write
210:      call    timer_display
211:  start_mt:
212:  ;*****
213:      mov     cammand,#0bfh
214:      mov     bytecnt,#8
215:      mov     r1,#xmtdat+1
216:      call    receive_byte
217:      call    timer_display
218:      call    load_alarm
219:  ;*****
220:      mov     a,seconds

221:      jnz     mtloop1
222:      mov     a,minits
223:      cjne    a,check_minits,mtloop1
224:      mov     a,hour
225:      anl     a,#00111111b
226:      cjne    a,check_hour,mtloop1
227:      setb    relay
228:      setb    out_flag
229:      jmp     mtloop2
230:  mtloop1:
231:      jnb     out_flag,start_mt00
232:      mov     a,seconds
233:      cjne    a,check_time,start_mt00
234:      clr     relay
235:      clr     out_flag
236:      inc     alarm_seq
237:      mov     a,alarm_seq
238:      cjne    a,#9,mtloop2
239:      mov     alarm_seq,#1
240:  mtloop2:
241:  ;*****
242:      mov     cammand,#0fch
243:      mov     bytecnt,#1
244:      mov     r0,#xmtdat
245:      mov     xmtdat,alarm_seq
246:      call    send_byte
247:  ;*****
248:  ;*****
249:  START_MT00:
250:      call    keyscan
251:      jz      start_mt
252:
253:      cjne    a,#1,mt001
254:      mov     a,#cursor_on
255:      call    cmd_write
256:      ret
257:  mt001:
258:      cjne    a,#2,mt002
259:      setb    relay
260:      jmp     start_mt
261:  mt002:
262:      cjne    a,#3,mt003
263:      clr     relay
264:  mt003:

```

265: jmp start_mt	309: call data_write
266: ;*****	310: ;*****
267: ;*****	311: mov a,#second_line
268: TIMER_DISPLAY:	312: call cmd_write
269: mov a,#first_line	313:
270: call cmd_write	314: mov a,days
271:	315: call display_days
272: mov a,hour	316: mov a,#","
273: anl a,#00011111b	317: call data_write
274: call display_data	318: mov a,#" "
275: ;*****	319: call data_write
276: mov a,seconds	320: mov a,date
277: jb acc.0,timd001	321: call display_data
278: mov a,#":"	322: mov a,#" "
279: call data_write	323: call data_write
280: jmp timd002	324: mov a,month
281: timd001:	325: call display_month
282: mov a,#" "	326: mov a,#" "
283: call data_write	327: call data_write
284: timd002:	328: mov a,#"2"
285: ;*****	329: call data_write
286: mov a,minits	330: mov a,#"0"
287: call display_data	331: call data_write
288: ;*****	332: mov a,year
289: mov a,hour	333: call display_data
290: jb acc.5,timd003	334: ret
291: mov a,#"A"	335: ;*****
292: call data_write	336: ;*****
293: jmp timd004	337: DISPLAY_DAYS:
294: timd003:	338: cjne a,#1,dds001
295: mov a,#"P"	339: mov dptr,#msg_24
296: call data_write	340: call dis_message
297: timd004:	341: ret
298: mov a,#"M"	342: dds001:
299: call data_write	343: cjne a,#2,dds002
300: ;*****	344: mov dptr,#msg_25
301: mov dptr,#msg_22	345: call dis_message
302: call dis_message	346: ret
303:	347: dds002:
304: mov a,#first_line+15	348: cjne a,#3,dds003
305: call cmd_write	349: mov dptr,#msg_26
306: mov a,alarm_seq	350: call dis_message
307: anl a,#00001111b	351: ret
308: call bcd2ascii	352: dds003:

```

353:      cjne      a,#4,dds004
354:      mov       dptr,#msg_27
355:      call      dis_message
356:      ret
357:  dds004:
358:      cjne      a,#5,dds005
359:      mov       dptr,#msg_28
360:      call      dis_message
361:      ret
362:  dds005:
363:      cjne      a,#6,dds006
364:      mov       dptr,#msg_29
365:      call      dis_message
366:      ret
367:  dds006:
368:      cjne      a,#7,dds007
369:      mov       dptr,#msg_30
370:      call      dis_message
371:  dds007:
372:      ret
373: ;*****
374: ;*****
375: DISPLAY_MONTH:
376:      cjne      a,#1,dm001
377:      mov       dptr,#msg_31
378:      call      dis_message
379:      ret
380:  dm001:
381:      cjne      a,#2,dm002
382:      mov       dptr,#msg_32
383:      call      dis_message
384:      ret
385:  dm002:
386:      cjne      a,#3,dm003
387:      mov       dptr,#msg_33
388:      call      dis_message
389:      ret
390:  dm003:
391:      cjne      a,#4,dm004
392:      mov       dptr,#msg_34
393:      call      dis_message
394:      ret
395:  dm004:
396:      cjne      a,#5,dm005
397:      mov       dptr,#msg_35
398:      call      dis_message
399:      ret
400:  dm005:
401:      cjne      a,#6,dm006
402:      mov       dptr,#msg_36
403:      call      dis_message
404:      ret
405:  dm006:
406:      cjne      a,#7,dm007
407:      mov       dptr,#msg_37
408:      call      dis_message
409:      ret
410:  dm007:
411:      cjne      a,#8,dm008
412:      mov       dptr,#msg_38
413:      call      dis_message
414:      ret
415:  dm008:
416:      cjne      a,#9,dm009
417:      mov       dptr,#msg_39
418:      call      dis_message
419:      ret
420:  dm009:
421:      cjne      a,#10h,dm010
422:      mov       dptr,#msg_40
423:      call      dis_message
424:      ret
425:  dm010:
426:      cjne      a,#11h,dm011
427:      mov       dptr,#msg_41
428:      call      dis_message
429:      ret
430:  dm011:
431:      cjne      a,#12h,dm012
432:      mov       dptr,#msg_42
433:      call      dis_message
434:  dm012:
435:      ret
436: ;*****
437: ;*****
438: LOAD_ALARM:
439:      mov       a,alarm_seq
440:      cjne      a,#1,lam001

```

```

441:      mov     check_hour,ram_space+0
442:      mov     check_minit,ram_space+1
443:      mov     check_time,ram_space+2
444:      jmp     lam008
445:  lam001:
446:      cjne    a,#2,lam002
447:      mov     check_hour,ram_space+3
448:      mov     check_minit,ram_space+4
449:      mov     check_time,ram_space+5
450:      jmp     lam008
451:  lam002:
452:      cjne    a,#3,lam003
453:      mov     check_hour,ram_space+6
454:      mov     check_minit,ram_space+7
455:      mov     check_time,ram_space+8
456:      jmp     lam008
457:  lam003:
458:      cjne    a,#4,lam004
459:      mov     check_hour,ram_space+9
460:      mov     check_minit,ram_space+10
461:      mov     check_time,ram_space+11
462:      jmp     lam008
463:  lam004:
464:      cjne    a,#5,lam005
465:      mov     check_hour,ram_space+12
466:      mov     check_minit,ram_space+13
467:      mov     check_time,ram_space+14
468:      jmp     lam008
469:  lam005:
470:      cjne    a,#6,lam006
471:      mov     check_hour,ram_space+15
472:      mov     check_minit,ram_space+16
473:      mov     check_time,ram_space+17
474:      jmp     lam008
475:  lam006:
476:      cjne    a,#7,lam007
477:      mov     check_hour,ram_space+18
478:      mov     check_minit,ram_space+19
479:      mov     check_time,ram_space+20
480:      jmp     lam008
481:  lam007:
482:      cjne    a,#8,lam008
483:      mov     check_hour,ram_space+21
484:      mov     check_minit,ram_space+22
485:      mov     check_time,ram_space+23
486:      lam008:
487:      ret
488: ;*****
489: ;*****
490: ;*****
491: ;      SUBROUTINE TO SET TIME & DATE
492: ;*****
493: ;*****
494: ;*****
495: SET_SELECT_ALL:
496:      call    set_display
497:      mov     set_count,#8
498:
499:      call    select_all
500:      call    delay_long
501: ;*****
502: SET_SELECT_KEY:
503:      call    keyscan
504:      jz      set_select_key
505: ;*****
506:      cjne    a,#1,ssk001
507:      call    delay_sec
508:      ret
509: ;*****
510:  ssk001:
511:      cjne    a,#2,ssk002
512:      call    change_select_all
513:      call    write_time
514:      call    set_display
515:      call    select_all
516:      call    delay_sec
517:      jmp     set_select_key
518:  ssk002:
519:      cjne    a,#3,ssk003
520:      call    change_select_all
521:      call    write_time
522:      call    set_display
523:      call    select_all
524:      call    delay_sec
525:      jmp     set_select_key
526:  ssk003:
527:      cjne    a,#4,ssk004
528:      call    select_set

```



```

529:      call    select_all
530:      call    delay_sec
531:      call    delay_sec
532:      call    delay_sec
533:      call    delay_sec
534:      ssk004:
535:      jmp     set_select_key
536: ;*****
537: ;*****
538: WRITE_TIME:
539:      mov     seconds,#0
540:      mov     cammand,#0beh
541:      mov     bytecnt,#8
542:      mov     r0,#xmtdat+1
543:      call    send_byte
544:      ret
545: ;*****
546: ;*****
547: SELECT_ALL:
548:      mov     a,set_count
549:      cjne    a,#8,salc002
550:
551:      mov     a,#first_line+1
552:      call    cmd_write
553:      ret
554:      salc002:
555:      cjne    a,#7,salc003
556:      mov     a,#first_line+4
557:      call    cmd_write
558:      ret
559:      salc003:
560:      cjne    a,#6,salc004
561:      mov     a,#first_line+6
562:      call    cmd_write
563:      ret
564:      salc004:
565:      cjne    a,#5,salc005
566:      mov     a,#first_line+15
567:      call    cmd_write
568:      ret
569:      salc005:
570:      cjne    a,#4,salc006
571:      mov     a,#second_line+2
572:      call    cmd_write
573:      ret
574:      salc006:
575:      cjne    a,#3,salc007
576:      mov     a,#second_line+6
577:      call    cmd_write
578:      ret
579:      salc007:
580:      cjne    a,#2,salc008
581:      mov     a,#second_line+10
582:      call    cmd_write
583:      ret
584:      salc008:
585:      cjne    a,#1,salc009
586:      mov     a,#second_line+15
587:      call    cmd_write
588:      salc009:
589:      ret
590: ;*****
591: ;*****
592: SELECT_SET:
593:      djnz    set_count,salc001
594:      mov     set_count,#8
595:      salc001:
596:      ret
597: ;*****
598: ;*****
599: CHANGE_SELECT_ALL:
600:      mov     a,set_count
601:      cjne    a,#8,csa001
602:      csa0011:
603: ;*****
604:      call    keyscan
605:      jz      csa0011
606:
607:      cjne    a,#2,csa0012
608:      mov     a,hour
609:      anl     a,#00011111b
610:      call    hour_increment
611:      orl     a,#10000000b
612:      mov     hour,a
613:      jmp     csa0013
614:      csa0012:
615:      cjne    a,#3,csa0013
616:      mov     a,hour

```

617: anl a,#00011111b	661: call cmd_write
618: call hour_decrement	662: mov a,#"P"
619: orl a,#10000000b	663: call data_write
620: mov hour,a	664: mov a,#"M"
621: csa0013:	665: call data_write
622: ret	666: ret
623: ;*****	667: ;*****
624: csa001:	668: csa003:
625: cjne a,#7,csa002	669: cjne a,#5,csa004
626: csa0021:	670: ;*****
627: ;*****	671: csa0041:
628: call keyscan	672: call keyscan
629: jz csa0021	673: jz csa0041
630:	674:
631: cjne a,#2,csa0022	675: cjne a,#2,csa0042
632: mov a,minits	676: mov a,alarm_seq
633: call minits_increment	677: call alarm_increment
634: mov minits,a	678: mov alarm_seq,a
635: ret	679: jmp csa0043
636: csa0022:	680: csa0042:
637: cjne a,#3,csa0023	681: cjne a,#3,csa0043
638: mov a,minits	682: mov a,alarm_seq
639: call minits_decrement	683: call alarm_decrement
640: mov minits,a	684: mov alarm_seq,a
641: csa0023:	685: csa0043:
642: ret	686: ;*****
643: ;*****	687: mov cammand,#0fch
644: csa002:	688: mov bytecnt,#1
645: cjne a,#6,csa003	689: mov r0,#xmtdat
646: ;*****	690: mov xmtdat,alarm_seq
647: mov a,hour	691: call send_byte
648: cpl acc.5	692: ret
649: mov hour,a	693: ;*****
650: jb acc.5,csa0031	694: csa004:
651: mov a,#first_line+5	695: cjne a,#4,csa005
652: call cmd_write	696: csa0051:
653: mov a,#"A"	697: ;*****
654: call data_write	698: call keyscan
655: mov a,#"M"	699: jz csa0051
656: call data_write	700:
657: ret	701: cjne a,#2,csa0052
658: ;*****	702: mov a,days
659: csa0031:	703: call day_increment
660: mov a,#first_line+5	704: mov days,a

705: ret	749: call month_decrement
706: csa0052:	750: mov month,a
707: cjne a,#3,csa0053	751: csa0073:
708: mov a,days	752: ret
709: call day_decrement	753: ;*****
710: mov days,a	754: csa007:
711: csa0053:	755: cjne a,#1,csa008
712: ret	756: csa0081:
713: ;*****	757: ;*****
714: csa005:	758: call keyscan
715: cjne a,#3,csa006	759: jz csa0081
716: csa0061:	760:
717: ;*****	761: cjne a,#2,csa0082
718: call keyscan	762: mov a,year
719: jz csa0061	763: call year_increment
720:	764: mov year,a
721: cjne a,#2,csa0062	765: ret
722: mov a,date	766: csa0082:
723: call date_increment	767: cjne a,#3,csa008
724: mov date,a	768: mov a,year
725: ret	769: call year_decrement
726: csa0062:	770: mov year,a
727: cjne a,#3,csa0063	771: ;*****
728: mov a,date	772: csa008:
729: call date_decrement	773: ret
730: mov date,a	774: ;*****
731: csa0063:	775: ;*****
732: ret	776: HOUR_INCREMENT:
733: ;*****	777: call bcd2hex
734: csa006:	778: cjne a,#12,hii001
735: cjne a,#2,csa007	779: call hex2bcd
736: csa0071:	780: ret
737: ;*****	781: hii001:
738: call keyscan	782: inc a
739: jz csa0071	783: call hex2bcd
740:	784: ret
741: cjne a,#2,csa0072	785: ;*****
742: mov a,month	786: ;*****
743: call month_increment	787: HOUR_DECREMENT:
744: mov month,a	788: call bcd2hex
745: ret	789: cjne a,#1,hdi001
746: csa0072:	790: call hex2bcd
747: cjne a,#3,csa0073	791: ret
748: mov a,month	792: hdi001:

```

793:      dec      a
794:      call     hex2bcd
795:      ret
796: ;*****
797: ;*****
798: MINITS_INCREMENT:
799:      call     bcd2hex
800:      cjne     a,#59,mdi001
801:      call     hex2bcd
802:      ret
803: mdi001:
804:      inc      a
805:      call     hex2bcd
806:      ret
807: ;*****
808: ;*****
809: MINITS_DECREMENT:
810:      call     bcd2hex
811:      cjne     a,#0,mdd001
812:      call     hex2bcd
813:      ret
814: mdd001:
815:      dec      a
816:      call     hex2bcd
817:      ret
818: ;*****
819: ;*****
820: DAY_INCREMENT:
821:      call     bcd2hex
822:      cjne     a,#7,Dii001
823:      call     hex2bcd
824:      ret
825: Dii001:
826:      inc      a
827:      call     hex2bcd
828:      ret
829: ;*****
830: ;*****
831: DAY_DECREMENT:
832:      call     bcd2hex
833:      cjne     a,#1,ddi001
834:      call     hex2bcd
835:      ret
836: ddi001:

```

```

837:      dec      a
838:      call     hex2bcd
839:      ret
840: ;*****
841: ;*****
842: DATE_INCREMENT:
843:      call     bcd2hex
844:      cjne     a,#31,dai001
845:      call     hex2bcd
846:      ret
847: dai001:
848:      inc      a
849:      call     hex2bcd
850:      ret
851: ;*****
852: ;*****
853: DATE_DECREMENT:
854:      call     bcd2hex
855:      cjne     a,#1,dadi001
856:      call     hex2bcd
857:      ret
858: dadi001:
859:      dec      a
860:      call     hex2bcd
861:      ret
862: ;*****
863: ;*****
864: MONTH_INCREMENT:
865:      call     bcd2hex
866:      cjne     a,#12,mit001
867:      call     hex2bcd
868:      ret
869: mit001:
870:      inc      a
871:      call     hex2bcd
872:      ret
873: ;*****
874: ;*****
875: MONTH_DECREMENT:
876:      call     bcd2hex
877:      cjne     a,#1,mdt001
878:      call     hex2bcd
879:      ret
880: mdt001:

```

881: dec a	925: dec a
882: call hex2bcd	926: call hex2bcd
883: ret	927: ret
884: ;*****	928: ;*****
885: ;*****	929: ;*****
886: YEAR_INCREMENT:	930: SET_DISPLAY:
887: call bcd2hex	931: mov a,#first_line
888: cjne a,#99,yit001	932: call cmd_write
889: call hex2bcd	933:
890: ret	934: mov a,hour
891: yit001:	935: anl a,#0001111b
892: inc a	936: call display_data
893: call hex2bcd	937: mov a,#":"
894: ret	938: call data_write
895: ;*****	939: mov a,minits
896: ;*****	940: call display_data
897: YEAR_DECREMENT:	941: mov a,hour
898: call bcd2hex	942: anl a,#00100000b
899: cjne a,#0,ydt001	943: jnz sdis001
900: call hex2bcd	944: mov a,#"A"
901: ret	945: call data_write
902: ydt001:	946: jmp sdis002
903: dec a	947: sdis001:
904: call hex2bcd	948: mov a,#"P"
905: ret	949: call data_write
906: ;*****	950: sdis002:
907: ;*****	951: mov a,#"M"
908: ALARM_INCREMENT:	952: call data_write
909: call bcd2hex	953: ;*****
910: cjne a,#8,ait001	954: mov dptr,#msg_22
911: call hex2bcd	955: call dis_message
912: ret	956:
913: ait001:	957: mov a,#first_line+15
914: inc a	958: call cmd_write
915: call hex2bcd	959: mov a,alarm_seq
916: ret	960: anl a,#00001111b
917: ;*****	961: call bcd2ascii
918: ;*****	962: call data_write
919: ALARM_DECREMENT:	963: ;*****
920: call bcd2hex	964: mov a,#second_line
921: cjne a,#1,adt001	965: call cmd_write
922: call hex2bcd	966:
923: ret	967: mov a,days
924: adt001:	968: call display_days

969:	mov	a,#" "	1013:	call	delay_sec
970:	call	data_write	1014:	jmp	set_timedate
971:	mov	a,#" "	1015:	std002:	
972:	call	data_write	1016:	cjne	a,#3,std003
973:	mov	a,date	1017:	call	alarm_fun_down
974:	call	display_data	1018:	call	display_alarm
975:	mov	a,#" "	1019:	call	delay_sec
976:	call	data_write	1020:	call	delay_sec
977:	mov	a,month	1021:	jmp	set_timedate
978:	call	display_month	1022:	std003:	
979:	mov	a,#" "	1023:	cjne	a,#4,std004
980:	call	data_write	1024:	call	setalarm
981:	mov	a,#"2"	1025:	call	display_alarm
982:	call	data_write	1026:	call	delay_sec
983:	mov	a,#"0"	1027:	call	delay_sec
984:	call	data_write	1028:	call	delay_sec
985:	mov	a,year	1029:	call	delay_sec
986:	call	display_data	1030:	std004:	
987:	ret		1031:	jmp	set_timedate
988: ;*****			1032: ;*****		
989: ;*****			1033: ;*****		
990: ;*****			1034: ;	ALARM SELECTION DISPLAY	
991: ;	SUBROUTINE TO SET ALARM TIME		1035: ;*****		
992: ;*****			1036: ;*****		
993: ;*****			1037: DISPLAY_ALARM:		
994: ;*****			1038:	mov	a,alarm_select
995: SETTIMEDATE:			1039:	cjne	a,#1,da001
996:	mov	alarm_select,#1	1040: ;*****		
997:	call	display_alarm	1041:	mov	a,#first_line
998:	call	delay_long	1042:	call	cmd_write
999: SET_TIMEDATE:			1043:	mov	dptr,#msg_10
1000:	call	keyscan	1044:	call	dis_message
1001:	jz	set_timedate	1045:		
1002:			1046:	mov	a,#" "
1003:	cjne	a,#1,std001	1047:	call	data_write
1004:	call	delay_sec	1048:		
1005:	call	delay_sec	1049:	mov	a,#second_line
1006:	ret		1050:	call	cmd_write
1007:	jmp	set_timedate	1051:	mov	dptr,#msg_11
1008:	std001:		1052:	call	dis_message
1009:	cjne	a,#2,std002	1053:		
1010:	call	alarm_fun_up	1054:	mov	a,#second_line+15
1011:	call	display_alarm	1055:	call	cmd_write
1012:	call	delay_sec	1056:	mov	a,#1

1057:	call	data_write	1101:		
1058:			1102:	mov	a,#first_line+15
1059:	mov	a,#first_line	1103:	call	cmd_write
1060:	call	cmd_write	1104:	mov	a,#0
1061:	mov	a,#cursor_on	1105:	call	data_write
1062:	call	cmd_write	1106:		
1063:	ret		1107:	mov	a,#second_line
1064:	;*****		1108:	call	cmd_write
1065:	da001:		1109:	mov	dptr,#msg_13
1066:	cjne	a,#2,da002	1110:	call	dis_message
1067:			1111:		
1068:	mov	a,#first_line	1112:	mov	a,#second_line+15
1069:	call	cmd_write	1113:	call	cmd_write
1070:	mov	dptr,#msg_11	1114:	mov	a,#1
1071:	call	dis_message	1115:	call	data_write
1072:			1116:		
1073:	mov	a,#first_line+15	1117:	mov	a,#first_line
1074:	call	cmd_write	1118:	call	cmd_write
1075:	mov	a,#0	1119:	mov	a,#cursor_on
1076:	call	data_write	1120:	call	cmd_write
1077:			1121:	ret	
1078:	mov	a,#second_line	1122:	;*****	
1079:	call	cmd_write	1123:	da003:	
1080:	mov	dptr,#msg_12	1124:	cjne	a,#4,da004
1081:	call	dis_message	1125:		
1082:			1126:	mov	a,#first_line
1083:	mov	a,#second_line+15	1127:	call	cmd_write
1084:	call	cmd_write	1128:	mov	dptr,#msg_13
1085:	mov	a,#1	1129:	call	dis_message
1086:	call	data_write	1130:		
1087:			1131:	mov	a,#first_line+15
1088:	mov	a,#first_line	1132:	call	cmd_write
1089:	call	cmd_write	1133:	mov	a,#0
1090:	mov	a,#cursor_on	1134:	call	data_write
1091:	call	cmd_write	1135:		
1092:	ret		1136:	mov	a,#second_line
1093:	;*****		1137:	call	cmd_write
1094:	da002:		1138:	mov	dptr,#msg_14
1095:	cjne	a,#3,da003	1139:	call	dis_message
1096:			1140:		
1097:	mov	a,#first_line	1141:	mov	a,#second_line+15
1098:	call	cmd_write	1142:	call	cmd_write
1099:	mov	dptr,#msg_12	1143:	mov	a,#1
1100:	call	dis_message	1144:	call	data_write

1145:			1189:	mov	a,#first_line+15
1146:	mov	a,#first_line	1190:	call	cmd_write
1147:	call	cmd_write	1191:	mov	a,#0
1148:	mov	a,#cursor_on	1192:	call	data_write
1149:	call	cmd_write	1193:		
1150:	ret		1194:	mov	a,#second_line
1151:	;*****				
1152:	da004:		1195:	call	cmd_write
1153:	cjne	a,#5,da005	1196:	mov	dptr,#msg_16
1154:			1197:	call	dis_message
1155:	mov	a,#first_line	1198:		
1156:	call	cmd_write	1199:	mov	a,#second_line+15
1157:	mov	dptr,#msg_14	1200:	call	cmd_write
1158:	call	dis_message	1201:	mov	a,#1
1159:			1202:	call	data_write
1160:	mov	a,#first_line+15	1203:		
1161:	call	cmd_write	1204:	mov	a,#first_line
1162:	mov	a,#0	1205:	call	cmd_write
1163:	call	data_write	1206:	mov	a,#cursor_on
1164:			1207:	call	cmd_write
1165:	mov	a,#second_line	1208:	ret	
1166:	call	cmd_write	1209:	;*****	
1167:	mov	dptr,#msg_15	1210:	da006:	
1168:	call	dis_message	1211:	cjne	a,#7,da007
1169:			1212:		
1170:	mov	a,#second_line+15	1213:	mov	a,#first_line
1171:	call	cmd_write	1214:	call	cmd_write
1172:	mov	a,#1	1215:	mov	dptr,#msg_16
1173:	call	data_write	1216:	call	dis_message
1174:			1217:		
1175:	mov	a,#first_line	1218:	mov	a,#first_line+15
1176:	call	cmd_write	1219:	call	cmd_write
1177:	mov	a,#cursor_on	1220:	mov	a,#0
1178:	call	cmd_write	1221:	call	data_write
1179:	ret		1222:		
1180:	;*****				
1181:	da005:		1223:	mov	a,#second_line
1182:	cjne	a,#6,da006	1224:	call	cmd_write
1183:			1225:	mov	dptr,#msg_17
1184:	mov	a,#first_line	1226:	call	dis_message
1185:	call	cmd_write	1227:		
1186:	mov	dptr,#msg_15	1228:	mov	a,#first_line
1187:	call	dis_message	1229:	call	cmd_write
1188:			1230:	mov	a,#cursor_on
			1231:	call	cmd_write
			1232:	;*****	

1233: da007:	1277: dec alarm_select
1234: cjne a,#8,da008	1278: ret
1235: ;*****	1279: ;*****
1236: mov a,#first_line	1280: ;*****
1237: call cmd_write	1281: ; SELECT & SET ALARM VALUES
1238: mov dptr,#msg_17	1282: ;*****
1239: call dis_message	1283: ;*****
1240: ;*****	1284: SETALARM:
1241: mov a,#first_line+15	1285: call read_alarm
1242: call cmd_write	1286: clr alarm_flag
1243: mov a,#0	1287: mov alarm_count,#6
1244: call data_write	1288: mov a,#lcd_clear
1245: ;*****	1289: call cmd_write
1246: mov a,#second_line	1290: ;*****
1247: call cmd_write	1291: ;*****
1248: mov dptr,#msg_50	1292: mov dptr,#msg_18
1249: call dis_message	1293: call dis_message
1250: ;*****	1294: ;*****
1251: mov a,#first_line	1295: call check_active
1252: call cmd_write	1296: jb alarm_flag,setalm001
1253: mov a,#cursor_on	1297: mov dptr,#msg_21
1254: call cmd_write	1298: call dis_message
1255: ;*****	1299: jmp setalm002
1256: da008:	1300: setalm001:
1257: ret	1301: mov dptr,#msg_20
1258: ;*****	1302: call dis_message
1259: ;*****	1303: setalm002:
1260: ; SELECT ALARM NUMBER	1304: ;*****
1261: ;*****	1305: mov a,#second_line
1262: ;*****	1306: call cmd_write
1263: ALARM_FUN_UP:	1307: mov dptr,#msg_19
1264: mov a,alarm_select	1308: call dis_message
1265: cjne a,#8,afu_001	1309: mov a,alarm_hour
1266: ret	1310: anl a,#00011111b
1267: afu_001:	1311: call display_data
1268: inc alarm_select	1312: ;*****
1269: ret	1313: mov a,#":"
1270: ;*****	1314: call data_write
1271: ;*****	1315: ;*****
1272: ALARM_FUN_DOWN:	1316: mov a,alarm_minits
1273: mov a,alarm_select	1317: call display_data
1274: cjne a,#1,afd_001	1318: ;*****
1275: ret	1319: mov a,alarm_hour
1276: afd_001:	1320: jb acc.5,tim_001

1321: mov a,#"A"	1365: cjne a,#4,set004
1322: call data_write	1366: call set_alarm_count
1323: jmp tim_002	1367: call delay_sec
1324: tim_001:	1368: call delay_sec
1325: mov a,#"P"	1369: call delay_sec
1326: call data_write	1370: call delay_sec
1327: tim_002:	1371: call delay_sec
1328: mov a,#"M"	1372: set004:
1329: call data_write	1373: jmp setalarmwait
1330: ;*****	1374: ;*****
1331: mov a,#">"	1375: ;*****
1332: call data_write	1376: READ_ALARM:
1333:	1377: mov cammand,#0ffh
1334: mov a,alarm_time	1378: mov bytecnt,#31
1335: call display_data	1379: mov r1,#ram_space
1336:	1380: call receive_byte
1337: mov a,#"s"	1381: ;*****
1338: call data_write	1382: mov a,alarm_select
1339: ;*****	1383: cjne a,#1,ram001
1340: call set_alarm_count	1384: mov alarm_hour,ram_space
1341: ;*****	1385: mov alarm_minits,ram_space+1
1342: ;*****	1386: mov alarm_time,ram_space+2
1343: ;*****	1387: jmp ram008
1344: SETALARMWAIT:	1388: ram001:
1345: call keyscan	1389: cjne a,#2,ram002
1346: jz setalarmwait	1390: mov alarm_hour,ram_space+3
1347:	1391: mov alarm_minits,ram_space+4
1348: cjne a,#1,set001	1392: mov alarm_time,ram_space+5
1349: ret	1393: jmp ram008
1350: set001:	1394: ram002:
1351: cjne a,#2,set002	1395: cjne a,#3,ram003
1352: call change_time	1396: mov alarm_hour,ram_space+6
1353: call write_alarm	1397: mov alarm_minits,ram_space+7
1354: call delay_sec	1398: mov alarm_time,ram_space+8
1355: call delay_sec	1399: jmp ram008
1356: jmp setalarmwait	1400: ram003:
1357: set002:	1401: cjne a,#4,ram004
1358: cjne a,#3,set003	1402: mov alarm_hour,ram_space+9
1359: call change_time	1403: mov alarm_minits,ram_space+10
1360: call write_alarm	1404: mov alarm_time,ram_space+11
1361: call delay_sec	1405: jmp ram008
1362: call delay_sec	1406: ram004:
1363: jmp setalarmwait	1407: cjne a,#5,ram005
1364: set003:	1408: mov alarm_hour,ram_space+12

1409:	mov	alarm_minits,ram_space+13	1453:	wam003:	
1410:	mov	alarm_time,ram_space+14	1454:	cjne	a,#4,wam004
1411:	jmp	ram008	1455:	mov	ram_space+9,alarm_hour
1412:	ram005:		1456:	mov	ram_space+10,alarm_minits
1413:	cjne	a,#6,ram006	1457:	mov	ram_space+11,alarm_time
1414:	mov	alarm_hour,ram_space+15	1458:	jmp	wam008
1415:	mov	alarm_minits,ram_space+16	1459:	wam004:	
1416:	mov	alarm_time,ram_space+17	1460:	cjne	a,#5,wam005
1417:	jmp	ram008	1461:	mov	ram_space+12,alarm_hour
1418:	ram006:		1462:	mov	ram_space+13,alarm_minits
1419:	cjne	a,#7,ram007	1463:	mov	ram_space+14,alarm_time
1420:	mov	alarm_hour,ram_space+18	1464:	jmp	wam008
1421:	mov	alarm_minits,ram_space+19	1465:	wam005:	
1422:	mov	alarm_time,ram_space+20	1466:	cjne	a,#6,wam006
1423:	jmp	ram008	1467:	mov	ram_space+15,alarm_hour
1424:	ram007:		1468:	mov	ram_space+16,alarm_minits
1425:	cjne	a,#8,ram008	1469:	mov	ram_space+17,alarm_time
1426:	mov	alarm_hour,ram_space+21	1470:	jmp	wam008
1427:	mov	alarm_minits,ram_space+22	1471:	wam006:	
1428:	mov	alarm_time,ram_space+23	1472:	cjne	a,#7,wam007
1429:	ram008:		1473:	mov	ram_space+18,alarm_hour
1430:	mov	alarm_active,ram_space+24	1474:	mov	ram_space+19,alarm_minits
1431:	ret		1475:	mov	ram_space+20,alarm_time
1432:	;*****				
1433:	;*****				
1434:	WRITE_ALARM:				
1435:	mov	a,alarm_select	1476:	jmp	wam008
1436:	cjne	a,#1,wam001	1477:	wam007:	
1437:	mov	ram_space,alarm_hour	1478:	cjne	a,#8,wam008
1438:	mov	ram_space+1,alarm_minits	1479:	mov	ram_space+21,alarm_hour
1439:	mov	ram_space+2,alarm_time	1480:	mov	ram_space+22,alarm_minits
1440:	jmp	wam008	1481:	mov	ram_space+23,alarm_time
1441:	wam001:		1482:	wam008:	
1442:	cjne	a,#2,wam002	1483:	mov	ram_space+24,alarm_active
1443:	mov	ram_space+3,alarm_hour	1484:	mov	cammand,#0feh
1444:	mov	ram_space+4,alarm_minits	1485:	mov	bytecnt,#31
1445:	mov	ram_space+5,alarm_time	1486:	mov	r0,#ram_space
1446:	jmp	wam008	1487:	call	send_byte
1447:	wam002:		1488:	ret	
1448:	cjne	a,#3,wam003	1489:	;*****	
1449:	mov	ram_space+6,alarm_hour	1490:	;*****	
1450:	mov	ram_space+7,alarm_minits	1491:	SET_ALARM_COUNT:	
1451:	mov	ram_space+8,alarm_time	1492:	djnz	alarm_count,sac001
1452:	jmp	wam008	1493:	mov	alarm_count,#5
			1494:	sac001:	
			1495:	mov	a,alarm_count
			1496:	cjne	a,#5,sac002

1497:	mov	a,#first_line+15	1541: ;*****
1498:	call	cmd_write	1542: jmp cht005
1499:	ret		1543: cht001:
1500: sac002:			1544: cjne a,#4,cht002
1501: cjne a,#4,sac003			1545: ;*****
1502: mov a,#second_line+6			1546: cht002wait:
1503: call cmd_write			1547: call keyscan
1504: ret			1548: jz cht002wait
1505: sac003:			1549:
1506: cjne a,#3,sac004			1550: cjne a,#2,cht0021
1507: mov a,#second_line+9			1551: mov a,alarm_hour
1508: call cmd_write			1552: push acc
1509: ret			1553: anl a,#00011111b
1510: sac004:			1554: call halarm_increment
1511: cjne a,#2,sac005			1555: mov alarm_hour,a
1512: mov a,#second_line+11			1556: pop acc
1513: call cmd_write			1557: anl a,#00100000b
1514: ret			1558: orl a,alarm_hour
1515: sac005:			1559: mov a,#second_line+5
1516: cjne a,#1,sac006			1560: call cmd_write
1517: mov a,#second_line+14			1561: mov a,alarm_hour
1518: call cmd_write			1562: anl a,#00011111b
1519: sac006:			1563: call display_data
1520: ret			1564: jmp cht0022
1521: ;*****			1565: cht0021:
1522: ;*****			1566: cjne a,#3,cht0022
1523: ;*****			1567: mov a,alarm_hour
1524: CHANGE_TIME:			1568: push acc
1525: mov a,alarm_count			1569: anl a,#00011111b
1526: cjne a,#5,cht001			1570: call halarm_decrement
1527: ;*****			1571: mov alarm_hour,a
1528: call toggle_active			1572: pop acc
1529: mov a,#first_line+8			1573: anl a,#00100000b
1530: call cmd_write			1574: orl a,alarm_hour
1531: jb alarm_flag,cht0011			1575: mov a,#second_line+5
1532: mov dptr,#msg_21			1576: call cmd_write
1533: call dis_message			1577: mov a,alarm_hour
1534: jmp cht0012			1578: anl a,#00011111b
1535: cht0011:			1579: call display_data
1536: mov dptr,#msg_20			1580: cht0022:
1537: call dis_message			1581: mov a,#second_line+6
1538: cht0012:			1582: call cmd_write
1539: mov a,#first_line+15			1583: ;*****
1540: call cmd_write			1584: jmp cht005

1585:	cht002:	1629:	call	cmd_write	
1586:	cjne	a,#3,cht003	1630:	mov	a,#"P"
1587:	;*****				
1588:	cht003wait:	1632:	cht0042:		
1589:	call	keyscan	1633:	mov	a,#"M"
1590:	jz	cht003wait	1634:	call	data_write
1591:			1635:	mov	a,#second_line+11
1592:	cjne	a,#2,cht0031	1636:	call	cmd_write
1593:	mov	a,alarm_minits	1637:	call	delay_sec
1594:	call	malarm_increment	1638:	;*****	
1595:	mov	alarm_minits,a	1639:	jmp	cht005
1596:	mov	a,#second_line+8	1640:	cht004:	
1597:	call	cmd_write	1641:	cjne	a,#1,cht005
1598:	mov	a,alarm_minits	1642:	;*****	
1599:	call	display_data	1643:	cht005wait:	
1600:	jmp	cht0032	1644:	call	keyscan
1601:	cht0031:		1645:	jz	cht005wait
1602:	cjne	a,#3,cht0032	1646:		
1603:	mov	a,alarm_minits	1647:	cjne	a,#2,cht0051
1604:	call	malarm_decrement	1648:	mov	a,alarm_time
1605:	mov	alarm_minits,a	1649:	call	talarm_increment
1606:	mov	a,#second_line+8	1650:	mov	alarm_time,a
1607:	call	cmd_write	1651:	mov	a,#second_line+13
1608:	mov	a,alarm_minits	1652:	call	cmd_write
1609:	call	display_data	1653:	mov	a,alarm_time
1610:	cht0032:		1654:	call	display_data
1611:	mov	a,#second_line+9	1655:	jmp	cht0052
1612:	call	cmd_write	1656:	cht0051:	
1613:	;*****				
1614:	jmp	cht005	1657:	cjne	a,#3,cht0052
1615:	cht003:		1658:	mov	a,alarm_time
1616:	cjne	a,#2,cht004	1659:	call	talarm_decrement
1617:	;*****				
1618:	mov	a,alarm_hour	1660:	mov	alarm_time,a
1619:	cpl	acc.5	1661:	mov	a,#second_line+13
1620:	mov	alarm_hour,a	1662:	call	cmd_write
1621:	jb	acc.5,cht0041	1663:	mov	a,alarm_time
1622:	mov	a,#second_line+10	1664:	call	display_data
1623:	call	cmd_write	1665:	cht0052:	
1624:	mov	a,#"A"	1666:	mov	a,#second_line+14
1625:	call	data_write	1667:	call	cmd_write
1626:	jmp	cht0042	1668:	;*****	
1627:	cht0041:		1669:	cht005:	
1628:	mov	a,#second_line+10	1670:	ret	
			1671:	;*****	
			1672:	;*****	

1673: HALARM_INCREMENT:

```

1674:      call    bcd2hex
1675:      cjne     a,#12,hai001
1676:      call    hex2bcd
1677:      ret
1678:  hai001:
1679:      inc      a
1680:      call    hex2bcd
1681:      ret

```

1682: ;*****

1683: ;*****

1684: HALARM_DECREMENT:

```

1685:      call    bcd2hex
1686:      cjne     a,#1,had001
1687:      call    hex2bcd
1688:      ret
1689:  had001:
1690:      dec      a
1691:      call    hex2bcd
1692:      ret

```

1693: ;*****

1694: ;*****

1695: MALARM_INCREMENT:

```

1696:      call    bcd2hex
1697:      cjne     a,#59,mai001
1698:      call    hex2bcd
1699:      ret
1700:  mai001:
1701:      inc      a
1702:      call    hex2bcd
1703:      ret

```

1704: ;*****

1705: ;*****

1706: MALARM_DECREMENT:

```

1707:      call    bcd2hex
1708:      cjne     a,#0,mad001
1709:      call    hex2bcd
1710:      ret
1711:  mad001:
1712:      dec      a
1713:      call    hex2bcd
1714:      ret

```

1715: ;*****

1716: ;*****

1717: TALARM_INCREMENT:

```

1718:      call    bcd2hex
1719:      cjne     a,#59,tai001
1720:      call    hex2bcd
1721:      ret
1722:  tai001:
1723:      inc      a
1724:      call    hex2bcd
1725:      ret

```

1726: ;*****

1727: ;*****

1728: TALARM_DECREMENT:

```

1729:      call    bcd2hex
1730:      cjne     a,#1,tad001
1731:      call    hex2bcd
1732:      ret
1733:  tad001:
1734:      dec      a
1735:      call    hex2bcd
1736:      ret

```

1737: ;*****

1738: ;*****

1739: TOGGLE_ACTIVE:

```

1740:      mov      a,alarm_select
1741:      cjne     a,#1,cav001
1742:      cpl      0
1743:      mov      c,0
1744:      mov      alarm_flag,c
1745:      ret

```

1746: cav001:

1747: cjne a,#2,cav002

1748: cpl 1

1749: mov c,1

1750: mov alarm_flag,c

1751: ret

1752: cav002:

1753: cjne a,#3,cav003

1754: cpl 2

1755: mov c,2

1756: mov alarm_flag,c

1757: ret

1758: cav003:

1759: cjne a,#4,cav004

1760: cpl 3

1761:	mov	c,3	1805:	ret
1762:	mov	alarm_flag,c	1806:	chav003:
1763:	ret		1807:	cjne a,#4,chav004
1764:	cav004:		1808:	mov c,3
1765:	cjne	a,#5,cav005	1809:	mov alarm_flag,c
1766:	cpl	4	1810:	ret
1767:	mov	c,4	1811:	chav004:
1768:	mov	alarm_flag,c	1812:	cjne a,#5,chav005
1769:	ret		1813:	mov c,4
1770:	cav005:		1814:	mov alarm_flag,c
1771:	cjne	a,#6,cav006	1815:	ret
1772:	cpl	5	1816:	chav005:
1773:	mov	c,5	1817:	cjne a,#6,chav006
1774:	mov	alarm_flag,c	1818:	mov c,5
1775:	ret		1819:	mov alarm_flag,c
1776:	cav006:		1820:	ret
1777:	cjne	a,#7,cav007	1821:	chav006:
1778:	cpl	6	1822:	cjne a,#7,chav007
1779:	mov	c,6	1823:	mov c,6
1780:	mov	alarm_flag,c	1824:	mov alarm_flag,c
1781:	cav007:		1825:	chav007:
1782:	cjne	a,#8,cav008	1826:	cjne a,#8,chav008
1783:	cpl	7	1827:	mov c,7
1784:	mov	c,7	1828:	mov alarm_flag,c
1785:	mov	alarm_flag,c	1829:	chav008:
1786:	cav008:		1830:	ret
1787:	ret		1831:	;*****
1788:	;*****		1832:	;*****
1789:	;*****		1833:	; SUBROUTINE DISPLAY INFORMATION
1790:	CHECK_ACTIVE:		1834:	;*****
1791:	mov	a,alarm_select	1835:	;*****
1792:	cjne	a,#1,chav001	1836:	INFORMATION:
1793:	mov	c,0	1837:	mov a,#first_line
1794:	mov	alarm_flag,c	1838:	call cmd_write
1795:	ret		1839:	mov dptr,#msg_01
1796:	chav001:		1840:	call dis_message
1797:	cjne	a,#2,chav002	1841:	
1798:	mov	c,1	1842:	mov a,#second_line
1799:	mov	alarm_flag,c	1843:	call cmd_write
1800:	ret		1844:	mov dptr,#msg_02
1801:	chav002:		1845:	call dis_message
1802:	cjne	a,#3,chav003	1846:	start_inform:
1803:	mov	c,2	1847:	call keyscan
1804:	mov	alarm_flag,c	1848:	jz start_inform

1849:		1893:	mov	dptr,#msg_06
1850:	cjne	1894:	call	dis_message
1851:	ret	1895:	mov	a,#0
1852:	sif_001:	1896:	call	data_write
1853:	jmp	1897:		
1854:	*****	1898:	mov	a,#first_line+15
1855:	*****	1899:	call	cmd_write
1856:	SELECTION SUBROUTINE	1900:	mov	a,#0
1857:	*****	1901:	call	data_write
1858:	*****	1902:		
1859:	SELECT_FUNCTION:	1903:	mov	a,#second_line
1860:	mov	1904:	call	cmd_write
1861:	cjne	1905:	mov	dptr,#msg_07
1862:	*****	1906:	call	dis_message
1863:	mov	1907:	mov	a,#1
1864:	call	1908:	call	data_write
1865:	mov	1909:		
1866:	call	1910:	mov	a,#second_line+15
1867:	mov	1911:	call	cmd_write
1868:	call	1912:	mov	a,#1
1869:		1913:	call	data_write
1870:	mov	1914:		
1871:	call	1915:	mov	a,#first_line
1872:	mov	1916:	call	cmd_write
1873:	call	1917:	mov	a,#cursor_on
1874:	mov	1918:	call	cmd_write
1875:	call	1919:	*****	
1876:		1920:	ret	
1877:	mov	1921:	mt_002:	
1878:	call	1922:	cjne	a,#3,mt_003
1879:	mov	1923:	*****	
1880:	call	1924:	mov	a,#first_line
1881:		1925:	call	cmd_write
1882:	mov	1926:	mov	dptr,#msg_07
1883:	call	1927:	call	dis_message
1884:	mov	1928:	mov	a,#0
1885:	call	1929:	call	data_write
1886:	*****	1930:		
1887:	ret	1931:	mov	a,#first_line+15
1888:	mt_001:	1932:	call	cmd_write
1889:	cjne	1933:	mov	a,#0
1890:	*****	1934:	call	data_write
1891:	mov	1935:		
1892:	call	1936:	mov	a,#second_line


```

1937:      call      cmd_write
1938:      mov       dptr,#msg_08
1939:      call      dis_message
1940:      mov       a,#1
1941:      call      data_write
1942:
1943:      mov       a,#second_line+15
1944:      call      cmd_write
1945:      mov       a,#1
1946:      call      data_write
1947:
1948:      mov       a,#first_line
1949:      call      cmd_write
1950:      mov       a,#cursor_on
1951:      call      cmd_write
1952: ;*****
1953:      ret
1954:      mt_003:
1955:      cjne      a,#4,mt_004
1956: ;*****
1957:      mov       a,#first_line
1958:      call      cmd_write
1959:      mov       dptr,#msg_08
1960:      call      dis_message
1961:      mov       a,#0
1962:      call      data_write
1963:
1964:      mov       a,#first_line+15
1965:      call      cmd_write
1966:      mov       a,#0
1967:      call      data_write
1968:
1969:      mov       a,#second_line
1970:      call      cmd_write
1971:      mov       dptr,#msg_50
1972:      call      dis_message
1973:
1974:      mov       a,#first_line
1975:      call      cmd_write
1976:      mov       a,#cursor_on
1977:      call      cmd_write
1978: ;*****
1979:      mt_004:
1980:      ret
1981: ;*****
1982: ;*****
1983: ;          SELECT FUNCTIONS
1984: ;*****
1985: ;*****
1986: FUNCTION_UP:
1987:      mov       a,select_fun
1988:      cjne      a,#4,fu_001
1989:      ret
1990:      fu_001:
1991:      inc       select_fun
1992:      ret
1993: ;*****
1994: ;*****
1995: FUNCTION_DOWN:
1996:      mov       a,select_fun
1997:      cjne      a,#1,fd_001
1998:      ret
1999:      fd_001:
2000:      dec       select_fun
2001:      ret
2002: ;*****
2003: ;*****
2004: ;*****
2005: ;          KEYBOARD ROUTINES
2006: ;*****
2007: ;*****
2008: ;*****
2009: KEYSKAN:
2010:      push      b
2011:      call      keydown
2012:      jz        keyscanlast
2013:      call      keyconvert
2014:      jbc       flag,keyscanlast
2015:      mov       b,a
2016:
2017:      call      delay_msec
2018:      call      delay_msec
2019:      call      delay_msec
2020:
2021:      call      keydown
2022:      jz        keyscanlast
2023:      call      keyconvert
2024:      jbc       flag,keyscanlast

```

```

2025:      cjne      a,b,key_scan
2026:      keyscanlast:
2027:      pop       b
2028:      ret
2029:      key_scan:
2030:      clr       a
2031:      pop       b
2032:      ret
2033: ;*****
2034: ;*****
2035: KEYDOWN:
2036:      mov       a,key_port
2037:      orl       a,#11111000b
2038:      cpl       a
2039:      ret
2040: ;*****
2041: ;*****
2042: KEYCONVERT:
2043:      mov       a,key_port
2044:      anl       a,#00111100b
2045:      cjne      a,#00111000b,next_one
2046:      mov       a,#1
2047:      jmp       last_con
2048:      next_one:
2049:      cjne      a,#00110100b,next_two
2050:      mov       a,#2
2051:      jmp       last_con
2052:      next_two:
2053:      cjne      a,#00101100b,next_three
2054:      mov       a,#3
2055:      jmp       last_con
2056:      next_three:
2057:      cjne      a,#00011100b,next_four
2058:      mov       a,#4
2059:      jmp       last_con
2060:      next_four:
2061:      setb      flag
2062:      last_con:
2063:      ret
2064: ;*****
2065: ;*****
2066: ;*****
2067: ;          LCD FUNCTIONS
2068: ;*****
2069: ;*****
2070: ;*****
2071: DATA_WRITE:
2072:      mov       lcd_port,a; post acc at lcd data input
2073:      setb      rs_lcd    ; set rs high (data)
2074:      call      delay_usec
2075:      setb      en_lcd    ; receive data byte
2076:      call      delay_usec
2077:      clr       en_lcd    ; drop the selection line
2078:      call      delay_msec
2079:      ret              ; return to caller
2080: ;*****
2081: ;*****
2082: CMD_WRITE:
2083:      mov       lcd_port,a; post acc at lcd data input
2084:      clr       rs_lcd    ; set rs low (instruction)
2085:      call      delay_usec
2086:      setb      en_lcd    ; receive instruction byte
2087:      call      delay_usec
2088:      clr       en_lcd    ; drop the selection line
2089:      call      delay_msec
2090:      ret              ; return to caller
2091: ;*****
2092: ;*****
2093: LCD_INITIALIZE:
2094:      mov       a,#03ch  ; lenght(8 bits),2lines,char5x7
2095:      call      cmd_write; needs 3 times this byte (1st)
2096:      call      delay_msec
2097:      call      cmd_write; (second)
2098:      call      delay_msec
2099:      call      cmd_write; (third)
2100:      call      delay_msec
2101:      mov       a,#00001100b
2102:      call      cmd_write;
2103:      call      delay_msec
2104:      mov       a,#06h    ; move direction & shift display
2105:      call      cmd_write;
2106:      call      delay_msec
2107:      mov       a,#lcd_clear; lcd clear
2108:      call      cmd_write; clear display
2109:      call      delay_msec
2110:      call      lcd_symbols
2111:      ret
2112: ;*****

```

```

2113: ;*****
2114: DIS_MESSAGE:
2115:     clr     a
2116:     movc    a,@a+dptr; get byte pointed by dptr
2117:     inc     dptr      ; point to the next byte
2118:     jz      dis_msg
2119:     call    data_write; it was data, write it to lcd
2120:     jmp     dis_message
2121: dis_msg:
2122:     ret
2123: ;*****
2124: ;*****
2125: ;           LCD SYMBOLS
2126: ;*****
2127: ;*****
2128: LCD_SYMBOLS:
2129:     mov     a,#40h
2130:     call    cmd_write
2131: ;*****
2132: UP_ARROW:
2133:     mov     a,#4
2134:     call    data_write
2135:     mov     a,#0eh
2136:     call    data_write
2137:     mov     a,#1fh
2138:     call    data_write
2139:     mov     a,#0
2140:     call    data_write
2141:     mov     a,#0
2142:     call    data_write
2143:     mov     a,#0
2144:     call    data_write
2145:     mov     a,#0
2146:     call    data_write
2147:     mov     a,#0
2148:     call    data_write
2149: ;*****
2150: DOWN_ARROW:
2151:     mov     a,#0
2152:     call    data_write
2153:     mov     a,#0
2154:     call    data_write
2155:     mov     a,#0
2156:     call    data_write

2157:     mov     a,#0
2158:     call    data_write
2159:     mov     a,#0
2160:     call    data_write
2161:     mov     a,#1fh
2162:     call    data_write
2163:     mov     a,#0eh
2164:     call    data_write
2165:     mov     a,#4
2166:     call    data_write
2167:     ret
2168: ;*****
2169: ;*****
2170: ;*****
2171: ;           DS1302 SUBROUTINES
2172: ;*****
2173: ;*****
2174: ;*****
2175: SEND_BYTE:
2176:     clr     c
2177:     clr     ds1302rst
2178:     nop
2179:     clr     ds1302clk
2180:     nop
2181:     setb    ds1302rst
2182:     nop
2183:     mov     a,cammand
2184:     mov     bitcnt,#8
2185: s_byte0:
2186:     rrc     a
2187:     mov     ds1302data,c
2188:     nop
2189:     setb    ds1302clk
2190:     nop
2191:     clr     ds1302clk
2192:     djnz    bitcnt,s_byte0
2193:     nop
2194: ;*****
2195: s_byte1:
2196:     mov     a,@r0
2197:     mov     bitcnt,#8
2198: s_byte2:
2199:     rrc     a
2200:     mov     ds1302data,c

```

2201:	nop		2245:	djnz	bytecnt,r_byte1
2202:	setb	ds1302clk	2246:	nop	
2203:	nop		2247:	clr	ds1302rst
2204:	clr	ds1302clk	2248:	ret	
2205:	djnz	bitcnt,s_byte2	2249:	;*****	
2206:	inc	r0	2250:	;*****	
2207:	djnz	bytecnt,s_byte1	2251:	;*****	
2208:	nop		2252:	; DATA CONVERSIONS	
2209:	clr	ds1302rst	2253:	;*****	
2210:	ret		2254:	;*****	
2211:	;*****		2255:	;*****	
2212:	;*****		2256:	HEX2BCD:	
2213:	RECEIVE_BYTE:		2257:	mov	b,#10 ; divide by 10
2214:	clr	ds1302rst	2258:	div	ab ; do divide
2215:	nop		2259:	swap	a ; move result to high of a
2216:	clr	ds1302clk	2260:	orl	a,b ; or in remainder
2217:	nop		2261:	ret	; return to caller
2218:	setb	ds1302rst	2262:	;*****	
2219:	mov	a,cammand	2263:	;*****	
2220:	mov	bitcnt,#8	2264:	BCD2HEX:	
2221:	r_byte0:		2265:	push	acc ; save value
2222:	rrc	a	2266:	anl	a,#0f0h ; keep high bits
2223:	mov	ds1302data,c	2267:	swap	a ; get to low
2224:	nop		2268:	mov	b,#10 ; 10 decimal
2225:	setb	ds1302clk	2269:	mul	ab ; multiply
2226:	nop		2270:	mov	b,a ; store in b
2227:	clr	ds1302clk	2271:	pop	acc ; recover bcd value
2228:	djnz	bitcnt,r_byte0	2272:	anl	a,#0fh ; keep low nybble
2229:	nop		2273:	add	a,b ; add in high
2230:	;*****		2274:	ret	; return to caller
2231:	r_byte1:		2275:	;*****	
2232:	clr	a	2276:	;*****	
2233:	clr	c	2277:	BCD2ASCII:	
2234:	mov	bitcnt,#8	2278:	inc	a
2235:	r_byte2:		2279:	movc	a,@a+pc
2236:	nop		2280:	ret	
2237:	mov	c,ds1302data	2281:	db	"0123456789"
2238:	rrc	a	2282:	;*****	
2239:	setb	ds1302clk	2283:	;*****	
2240:	nop		2284:	DISPLAY_DATA:	
2241:	clr	ds1302clk	2285:	push	acc
2242:	djnz	bitcnt,r_byte2	2286:	anl	a,#0f0h
2243:	mov	@r1,a	2287:	swap	a
2244:	inc	r1	2288:	call	bcd2ascii

2289:	call	data_write	2333: ;*****
2290:	pop	acc	2334: ;*****
2291:	anl	a,#0fh	2335: ;*****
2292:	call	bcd2ascii	2336: ; MESSAGES
2293:	call	data_write	2337: ;*****
2294:	ret		2338: ;*****
2295: ;*****			2339: ;*****
2296: ;*****			2340: MSG_01:
2297: ;*****			2341: DB "APPLIED ELECTRO-",0
2298: ; DELAY FUNCTIONS			2342: MSG_02:
2299: ;*****			2343: DB "NICS**S-6**BATCH",0
2300: ;*****			2344: MSG_03:
2301: ;*****			2345: DB "PROJECT DONE BY",0
2302: DELAY_uSEC:			2346: MSG_04:
2303: mov r7,#8 ; 100 microseconds			2347: DB "MODEL POLY ,MALA",0
2304: djnz r7,\$			2348: ;*****
2305: ret			2349: MSG_05:
2306: ;*****			2350: DB "CLOCK DISPLAY ",0
2307: ;*****			2351: MSG_06:
2308: DELAY_MSEC:			2352: DB "CLOCK SET ",0
2309: mov r7,#16 ; 10 milliseconds			2353: MSG_07:
2310: d_mswait:			2354: DB "ALARM SET ",0
2311: mov r6,#32 ; 500 microseconds			2355: MSG_08:
2312: djnz r6,\$			2356: DB "HELP TOPICS ",0
2313: djnz r7,d_mswait			2357: ;*****
2314: ret			2358: MSG_10:
2315: ;*****			2359: DB "SET ALARM-1 ",0
2316: ;*****			2360: MSG_11:
2317: DELAY_SEC:			2361: DB "SET ALARM-2 ",0
2318: mov r6,#80 ; 100milliseconds			2362: MSG_12:
2319: d_swait1:			2363: DB "SET ALARM-3 ",0
2320: mov r5,#160 ; 1 milliseconds			2364: MSG_13:
2321: djnz r5,\$			2365: DB "SET ALARM-4 ",0
2322: djnz r6,d_swait1			2366: MSG_14:
2323: ret			2367: DB "SET ALARM-5 ",0
2324: ;*****			2368: MSG_15:
2325: ;*****			2369: DB "SET ALARM-6 ",0
2326: DELAY_LONG:			2370: MSG_16:
2327: call delay_sec			2371: DB "SET ALARM-7 ",0
2328: call delay_sec			2372: MSG_17:
2329: call delay_sec			2373: DB "SET ALARM-8 ",0
2330: call delay_sec			2374: ;*****
2331: call delay_sec			2375: MSG_18:
2332: ret			2376: DB "STATUS>>>",0

2377: MSG_19:			2421: DB "Oct",0
2378: DB "TIME",0			2422: MSG_41:
2379: ;*****			2423: DB "Nov",0
2380: MSG_20:			2424: MSG_42:
2381: DB " ACTIVE",0			2425: DB "Dec",0
2382: MSG_21:			2426: ;*****
2383: DB "INACTIVE",0			2427: MSG_43:
2384: MSG_22:			2428: DB "Nidheesh.K.J",0
2385: DB " AL",0			2429: MSG_44:
2386: ;*****			2430: DB "Nidheeshkumar",0
2387: MSG_24:			2431: MSG_45:
2388: DB "Sun",0			2432: DB "Nitheeshkumar",0
2389: MSG_25:			2433: MSG_46:
2390: DB "Mon",0			2434: DB "Nithya.J",0
2391: MSG_26:			2435: MSG_47:
2392: DB "Tue",0			2436: DB "Nivya",0
2393: MSG_27:			2437: ;*****
2394: DB "Wed",0			2438: MSG_50:
2395: MSG_28:			2439: DB "* PRESS UP KEY ",0
2396: DB "Thr",0			2440: ;*****
2397: MSG_29:			2441: ;*****
2398: DB "Fri",0			2442: ; END OF THE PROGRAMME
2399: MSG_30:			2443: ;*****
2400: DB "Sat",0			2444: ;*****
2401: ;*****			2445: END
2402: MSG_31:			2446: ;*****
2403: DB "Jan",0			2447: ;////////// GOOD BYE \\\\\\\\\\\\\\\\\\\\\\\
2404: MSG_32:			2448: ;MATTIOM EMBEDDED TECHNOLOGIES,SNEHAPURAM
2405: DB "Feb",0			
2406: MSG_33:			
2407: DB "Mar",0			
2408: MSG_34:			
2409: DB "Apr",0			
2410: MSG_35:			
2411: DB "May",0			
2412: MSG_36:			
2413: DB "Jun",0			
2414: MSG_37:			
2415: DB "Jul",0			
2416: MSG_38:			
2417: DB "Aug",0			
2418: MSG_39:			
2419: DB "Sep",0			
2420: MSG_40:			