```
MACRO ASSEMBLER A51 V8.00d
OBJECT MODULE PLACED IN Main.OBJ
ASSEMBLER INVOKED BY: c:\SiLabs\MCU\IDEfiles\C51\BIN\a51.exe Main.asm XR GEN DB EP NOMOD51

LOC  OBJ            LINE     SOURCE

                       1     ;=====================================================================
                       2     ;                             Pro-Tex 9000
                       3     ;
                       4     ;Revision: R.07171500  (R.MMDDHHMM)
                       5     ;
                       6     ;Project Team Members:
                       7     ; - Vince Watkins
                       8     ; - Will Smith
                       9     ; - Tyler Long
                      10     ;
                      11     ;
                      12     ;Main Code space
                      13     ;
                      14     ;
                      15     ;=====================================================================
                      16     
                      17     
                      18     
                      19     ;=====================================================================
                      20     ;   Assembler Controls
                      21     ;=====================================================================
                      22     
                      23     $debug
                      24     $print
                      25             $symbols                   ;Create Symbol table for list file
                      26             $title (MILESTONE #2)
                      27             $date (July-17-2008)
                      28             $pagewidth (132)
                      29     
                      30     ;=====================================================================
                      31     ;   Include Files
                      32     ;=====================================================================
                      33     
                      34     ;$include (C8051F020.inc) ;use with SiLabs Keil A51 compiler
                +1    35     ;-----------------------------------------------------------------------------
                +1    36     ;
                +1    37     ;
                +1    38     ;
                +1    39     ;
                +1    40     ;       FILE NAME       : C8051F020.INC
                +1    41      ;       TARGET MCUs     : C8051F020, 'F021, 'F022, 'F023
                +1    42      ;       DESCRIPTION     : Register/bit definitions for the C8051F02x product f
                +1    43      ;
                +1    44      ;       REVISION 1.0
                +1    45      ;
                +1    46      ;-----------------------------------------------------------------------------
                +1    47      ;REGISTER DEFINITIONS
                +1    48      ;
  0080          +1    49      P0       DATA  080H   ; PORT 0
  0081          +1    50      SP       DATA  081H   ; STACK POINTER
  0082          +1    51      DPL      DATA  082H   ; DATA POINTER - LOW BYTE
  0083          +1    52      DPH      DATA  083H   ; DATA POINTER - HIGH BYTE
  0084          +1    53      P4       DATA  084H   ; PORT 4
  0085          +1    54      P5       DATA  085H   ; PORT 5
  0086          +1    55      P6       DATA  086H   ; PORT 6
  0087          +1    56      PCON     DATA  087H   ; POWER CONTROL
  0088          +1    57      TCON     DATA  088H   ; TIMER CONTROL
  0089          +1    58      TMOD     DATA  089H   ; TIMER MODE
  008A          +1   59     TL0      DATA  08AH   ; TIMER 0 - LOW BYTE
  008B          +1   60     TL1      DATA  08BH   ; TIMER 1 - LOW BYTE
  008C          +1   61     TH0      DATA  08CH   ; TIMER 0 - HIGH BYTE
  008D          +1   62     TH1      DATA  08DH   ; TIMER 1 - HIGH BYTE
  008E          +1   63     CKCON    DATA  08EH   ; CLOCK CONTROL
  008F          +1   64     PSCTL    DATA  08FH   ; PROGRAM STORE R/W CONTROL
  0090          +1   65     P1       DATA  090H   ; PORT 1
  0091          +1   66     TMR3CN   DATA  091H   ; TIMER 3 CONTROL
  0092          +1   67     TMR3RLL  DATA  092H   ; TIMER 3 RELOAD REGISTER - LOW BYTE
  0093          +1   68     TMR3RLH  DATA  093H   ; TIMER 3 RELOAD REGISTER - HIGH BYTE
  0094          +1   69     TMR3L    DATA  094H   ; TIMER 3 - LOW BYTE
  0095          +1   70     TMR3H    DATA  095H   ; TIMER 3 - HIGH BYTE
  0096          +1   71     P7       DATA  096H   ; PORT 7
  0098          +1   72     SCON0    DATA  098H   ; SERIAL PORT 0 CONTROL
  0099          +1   73     SBUF0    DATA  099H   ; SERIAL PORT 0 BUFFER
  009A          +1   74     SPI0CFG  DATA  09AH   ; SERIAL PERIPHERAL INTERFACE 0 CONFIGURATION
  009B          +1   75     SPI0DAT  DATA  09BH   ; SERIAL PERIPHERAL INTERFACE 0 DATA
  009C          +1   76     ADC1     DATA  09CH   ; ADC 1 DATA
  009D          +1   77     SPI0CKR  DATA  09DH   ; SERIAL PERIPHERAL INTERFACE 0 CLOCK RATE CONTROL
  009E          +1   78     CPT0CN   DATA  09EH   ; COMPARATOR 0 CONTROL
  009F          +1   79     CPT1CN   DATA  09FH   ; COMPARATOR 1 CONTROL
  00A0          +1   80     P2       DATA  0A0H   ; PORT 2
  00A1          +1   81     EMI0TC   DATA  0A1H   ; EMIF TIMING CONTROL
  00A3          +1   82     EMI0CF   DATA  0A3H   ; EXTERNAL MEMORY INTERFACE (EMIF) CONFIGURATION
  00A4          +1   83     P0MDOUT  DATA  0A4H   ; PORT 0 OUTPUT MODE CONFIGURATION
  00A5          +1   84     P1MDOUT  DATA  0A5H   ; PORT 1 OUTPUT MODE CONFIGURATION
  00A6          +1   85     P2MDOUT  DATA  0A6H   ; PORT 2 OUTPUT MODE CONFIGURATION
  00A7          +1   86     P3MDOUT  DATA  0A7H   ; PORT 3 OUTPUT MODE CONFIGURATION
  00A8          +1   87     IE       DATA  0A8H   ; INTERRUPT ENABLE
  00A9          +1   88     SADDR0   DATA  0A9H   ; SERIAL PORT 0 SLAVE ADDRESS
  00AA          +1   89     ADC1CN  DATA  0AAH   ; ADC 1 CONTROL
  00AB          +1   90     ADC1CF   DATA  0ABH   ; ADC 1 ANALOG MUX CONFIGURATION
  00AC          +1   91     AMX1SL   DATA  0ACH   ; ADC 1 ANALOG MUX CHANNEL SELECT
  00AD          +1   92     P3IF     DATA  0ADH   ; PORT 3 EXTERNAL INTERRUPT FLAGS
  00AE          +1   93     SADEN1   DATA  0AEH   ; SERIAL PORT 1 SLAVE ADDRESS MASK
  00AF          +1   94     EMI0CN   DATA  0AFH   ; EXTERNAL MEMORY INTERFACE CONTROL
  00B0          +1   95     P3       DATA  0B0H   ; PORT 3
  00B1          +1   96     OSCXCN   DATA  0B1H   ; EXTERNAL OSCILLATOR CONTROL
  00B2          +1   97     OSCICN   DATA  0B2H   ; INTERNAL OSCILLATOR CONTROL
  00B5          +1   98     P74OUT   DATA  0B5H   ; PORTS 4 - 7 OUTPUT MODE
  00B6          +1   99     FLSCL    DATA  0B6H   ; FLASH MEMORY TIMING PRESCALER
  00B7          +1  100     FLACL    DATA  0B7H   ; FLASH ACESS LIMIT
  00B8          +1  101     IP       DATA  0B8H   ; INTERRUPT PRIORITY
  00B9          +1  102     SADEN0   DATA  0B9H   ; SERIAL PORT 0 SLAVE ADDRESS MASK
  00BA          +1  103     AMX0CF   DATA  0BAH   ; ADC 0 MUX CONFIGURATION
  00BB          +1  104     AMX0SL   DATA  0BBH   ; ADC 0 MUX CHANNEL SELECTION
  00BC          +1  105     ADC0CF   DATA  0BCH   ; ADC 0 CONFIGURATION
  00BD           +1   106     P1MDIN   DATA  0BDH   ; PORT 1 INPUT MODE
  00BE           +1   107     ADC0L    DATA  0BEH   ; ADC 0 DATA - LOW BYTE
  00BF           +1   108     ADC0H    DATA  0BFH   ; ADC 0 DATA - HIGH BYTE
  00C0           +1   109     SMB0CN   DATA  0C0H   ; SMBUS 0 CONTROL
  00C1           +1   110     SMB0STA  DATA  0C1H   ; SMBUS 0 STATUS
  00C2           +1   111     SMB0DAT  DATA  0C2H   ; SMBUS 0 DATA
  00C3           +1   112     SMB0ADR  DATA  0C3H   ; SMBUS 0 SLAVE ADDRESS
  00C4           +1   113     ADC0GTL  DATA  0C4H   ; ADC 0 GREATER-THAN REGISTER - LOW BYTE
  00C5           +1   114     ADC0GTH  DATA  0C5H   ; ADC 0 GREATER-THAN REGISTER - HIGH BYTE
  00C6           +1   115     ADC0LTL  DATA  0C6H   ; ADC 0 LESS-THAN REGISTER - LOW BYTE
  00C7           +1   116     ADC0LTH  DATA  0C7H   ; ADC 0 LESS-THAN REGISTER - HIGH BYTE
  00C8           +1   117     T2CON    DATA  0C8H   ; TIMER 2 CONTROL
  00C9           +1   118     T4CON    DATA  0C9H   ; TIMER 4 CONTROL
  00CA           +1   119     RCAP2L   DATA  0CAH   ; TIMER 2 CAPTURE REGISTER - LOW BYTE
  00CB           +1   120     RCAP2H   DATA  0CBH   ; TIMER 2 CAPTURE REGISTER - HIGH BYTE
  00CC           +1   121     TL2      DATA  0CCH   ; TIMER 2 - LOW BYTE
  00CD           +1   122     TH2      DATA  0CDH   ; TIMER 2 - HIGH BYTE
  00CF           +1   123     SMB0CR   DATA  0CFH   ; SMBUS 0 CLOCK RATE
  00D0           +1   124     PSW      DATA  0D0H   ; PROGRAM STATUS WORD
  00D1          +1   125     REF0CN   DATA  0D1H   ; VOLTAGE REFERENCE 0 CONTROL
  00D2          +1   126     DAC0L    DATA  0D2H   ; DAC 0 REGISTER - LOW BYTE
  00D3          +1   127     DAC0H    DATA  0D3H   ; DAC 0 REGISTER - HIGH BYTE
  00D4          +1   128     DAC0CN   DATA  0D4H   ; DAC 0 CONTROL
  00D5          +1   129     DAC1L    DATA  0D5H   ; DAC 1 REGISTER - LOW BYTE
  00D6          +1   130     DAC1H    DATA  0D6H   ; DAC 1 REGISTER - HIGH BYTE
  00D7          +1   131     DAC1CN   DATA  0D7H   ; DAC 1 CONTROL
  00D8          +1   132     PCA0CN   DATA  0D8H   ; PCA 0 COUNTER CONTROL
  00D9          +1   133     PCA0MD   DATA  0D9H   ; PCA 0 COUNTER MODE
  00DA          +1   134     PCA0CPM0 DATA  0DAH   ; CONTROL REGISTER FOR PCA 0 MODULE 0
  00DB          +1   135     PCA0CPM1 DATA  0DBH   ; CONTROL REGISTER FOR PCA 0 MODULE 1
  00DC          +1   136     PCA0CPM2 DATA  0DCH   ; CONTROL REGISTER FOR PCA 0 MODULE 2
  00DD          +1   137     PCA0CPM3 DATA  0DDH   ; CONTROL REGISTER FOR PCA 0 MODULE 3
  00DE          +1   138     PCA0CPM4 DATA  0DEH   ; CONTROL REGISTER FOR PCA 0 MODULE 4
  00E0          +1   139     ACC      DATA  0E0H   ; ACCUMULATOR
  00E1          +1   140     XBR0     DATA  0E1H   ; DIGITAL CROSSBAR CONFIGURATION REGISTER 0
  00E2          +1   141     XBR1     DATA  0E2H   ; DIGITAL CROSSBAR CONFIGURATION REGISTER 1
  00E3          +1   142     XBR2     DATA  0E3H   ; DIGITAL CROSSBAR CONFIGURATION REGISTER 2
  00E4          +1   143     RCAP4L   DATA  0E4H   ; TIMER 4 CAPTURE REGISTER - LOW BYTE
  00E5          +1   144     RCAP4H   DATA  0E5H   ; TIMER 4 CAPTURE REGISTER - HIGH BYTE
  00E6          +1   145     EIE1     DATA  0E6H   ; EXTERNAL INTERRUPT ENABLE 1
  00E7          +1   146     EIE2     DATA  0E7H   ; EXTERNAL INTERRUPT ENABLE 2
  00E8          +1   147     ADC0CN   DATA  0E8H   ; ADC 0 CONTROL
  00E9          +1   148     PCA0L    DATA  0E9H   ; PCA 0 TIMER - LOW BYTE
  00EA          +1   149     PCA0CPL0 DATA  0EAH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - LOW BYTE
  00EB          +1   150     PCA0CPL1 DATA  0EBH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - LOW BYTE
  00EC          +1   151     PCA0CPL2 DATA  0ECH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - LOW BYTE
  00ED          +1   152     PCA0CPL3 DATA  0EDH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - LOW BYTE
  00EE          +1   153     PCA0CPL4 DATA  0EEH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - LOW BYTE
  00EF          +1   154     RSTSRC   DATA  0EFH   ; RESET SOURCE
  00F0          +1   155     B        DATA  0F0H   ; B REGISTER
  00F1          +1   156     SCON1    DATA  0F1H   ; SERIAL PORT 1 CONTROL
  00F2          +1   157     SBUF1    DATA  0F2H   ; SERAIL PORT 1 DATA
  00F3          +1   158     SADDR1   DATA  0F3H   ; SERAIL PORT 1
  00F4          +1   159     TL4      DATA  0F4H   ; TIMER 4 DATA - LOW BYTE
  00F5          +1   160     TH4      DATA  0F5H   ; TIMER 4 DATA - HIGH BYTE
  00F6          +1   161     EIP1     DATA  0F6H   ; EXTERNAL INTERRUPT PRIORITY REGISTER 1
  00F7          +1   162     EIP2     DATA  0F7H   ; EXTERNAL INTERRUPT PRIORITY REGISTER 2
  00F8          +1   163     SPI0CN   DATA  0F8H   ; SERIAL PERIPHERAL INTERFACE 0 CONTROL
  00F9          +1   164     PCA0H    DATA  0F9H   ; PCA 0 TIMER - HIGH BYTE
  00FA          +1   165     PCA0CPH0 DATA  0FAH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 0 - HIGH BYT
  00FB          +1   166     PCA0CPH1 DATA  0FBH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 1 - HIGH BYT
  00FC          +1   167     PCA0CPH2 DATA  0FCH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 2 - HIGH BYT
  00FD          +1   168     PCA0CPH3 DATA  0FDH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 3 - HIGH BYT
  00FE          +1   169     PCA0CPH4 DATA  0FEH   ; CAPTURE/COMPARE REGISTER FOR PCA 0 MODULE 4 - HIGH BYT
  00FF          +1   170     WDTCN    DATA  0FFH   ; WATCHDOG TIMER CONTROL
                +1   171     ;
```

```
                 +1   172     ;--------------------------------------------------------------------------------
                 +1   173     ;BIT DEFINITIONS
                 +1   174     ;
                 +1   175     ; TCON 88H
  0088           +1   176     IT0      BIT   TCON.0 ; EXT. INTERRUPT 0 TYPE
  0089           +1   177     IE0      BIT   TCON.1 ; EXT. INTERRUPT 0 EDGE FLAG
  008A           +1   178     IT1      BIT   TCON.2 ; EXT. INTERRUPT 1 TYPE
  008B           +1   179     IE1      BIT   TCON.3 ; EXT. INTERRUPT 1 EDGE FLAG
  008C           +1   180     TR0      BIT   TCON.4 ; TIMER 0 ON/OFF CONTROL
  008D           +1   181     TF0      BIT   TCON.5 ; TIMER 0 OVERFLOW FLAG
  008E           +1   182     TR1      BIT   TCON.6 ; TIMER 1 ON/OFF CONTROL
  008F           +1   183     TF1      BIT   TCON.7 ; TIMER 1 OVERFLOW FLAG
                 +1   184     ;
                 +1   185     ; SCON0 98H
  0098           +1   186     RI       BIT   SCON0.0 ; RECEIVE INTERRUPT FLAG
  0099           +1   187     TI       BIT   SCON0.1 ; TRANSMIT INTERRUPT FLAG
  009A           +1   188     RB8      BIT   SCON0.2 ; RECEIVE BIT 8
  009B           +1   189     TB8      BIT   SCON0.3 ; TRANSMIT BIT 8
  009C           +1   190     REN      BIT   SCON0.4 ; RECEIVE ENABLE
  009D          +1   191     SM2      BIT   SCON0.5 ; MULTIPROCESSOR COMMUNICATION ENABLE
  009E          +1   192     SM1      BIT   SCON0.6 ; SERIAL MODE CONTROL BIT 1
  009F          +1   193     SM0      BIT   SCON0.7 ; SERIAL MODE CONTROL BIT 0
                +1   194     ;
                +1   195     ; IE A8H
  00A8          +1   196     EX0      BIT   IE.0   ; EXTERNAL INTERRUPT 0 ENABLE
  00A9          +1   197     ET0      BIT   IE.1   ; TIMER 0 INTERRUPT ENABLE
  00AA          +1   198     EX1      BIT   IE.2   ; EXTERNAL INTERRUPT 1 ENABLE
  00AB          +1   199     ET1      BIT   IE.3   ; TIMER 1 INTERRUPT ENABLE
  00AC          +1   200     ES       BIT   IE.4   ; SERIAL PORT INTERRUPT ENABLE
  00AD          +1   201     ET2      BIT   IE.5   ; TIMER 2 INTERRUPT ENABLE
  00AF          +1   202     EA       BIT   IE.7   ; GLOBAL INTERRUPT ENABLE
                +1   203     ;
                +1   204     ; IP B8H
  00B8          +1   205     PX0      BIT   IP.0   ; EXTERNAL INTERRUPT 0 PRIORITY
  00B9          +1   206     PT0      BIT   IP.1   ; TIMER 0 PRIORITY
  00BA          +1   207     PX1      BIT   IP.2   ; EXTERNAL INTERRUPT 1 PRIORITY
  00BB          +1   208     PT1      BIT   IP.3   ; TIMER 1 PRIORITY
  00BC          +1   209     PS       BIT   IP.4   ; SERIAL PORT PRIORITY
  00BD          +1   210     PT2      BIT   IP.5   ; TIMER 2 PRIORITY
                +1   211     ;
                +1   212     ; SMB0CN C0H
  00C0          +1   213     SMBTOE   BIT   SMB0CN.0 ; SMBUS 0 TIMEOUT ENABLE
  00C1          +1   214     SMBFTE   BIT   SMB0CN.1 ; SMBUS 0 FREE TIMER ENABLE
  00C2          +1   215     AA       BIT   SMB0CN.2 ; SMBUS 0 ASSERT/ACKNOWLEDGE FLAG
  00C3          +1   216     SI       BIT   SMB0CN.3 ; SMBUS 0 INTERRUPT PENDING FLAG
  00C4          +1   217     STO      BIT   SMB0CN.4 ; SMBUS 0 STOP FLAG
  00C5          +1   218     STA      BIT   SMB0CN.5 ; SMBUS 0 START FLAG
  00C6          +1   219     ENSMB    BIT   SMB0CN.6 ; SMBUS 0 ENABLE
                +1   220     ;
                +1   221     ; T2CON C8H
  00C8          +1   222     CPRL2    BIT   T2CON.0 ; CAPTURE OR RELOAD SELECT
  00C9          +1   223     CT2      BIT   T2CON.1 ; TIMER OR COUNTER SELECT
  00CA          +1   224     TR2      BIT   T2CON.2 ; TIMER 2 ON/OFF CONTROL
  00CB          +1   225     EXEN2    BIT   T2CON.3 ; TIMER 2 EXTERNAL ENABLE FLAG
  00CC          +1   226     TCLK     BIT   T2CON.4 ; TRANSMIT CLOCK FLAG
  00CD          +1   227     RCLK     BIT   T2CON.5 ; RECEIVE CLOCK FLAG
  00CE          +1   228     EXF2     BIT   T2CON.6 ; EXTERNAL FLAG
  00CF          +1   229     TF2      BIT   T2CON.7 ; TIMER 2 OVERFLOW FLAG
                +1   230     ;
                +1   231     ; PSW D0H
  00D0          +1   232     P        BIT   PSW.0  ; ACCUMULATOR PARITY FLAG
  00D1          +1   233     F1       BIT   PSW.1  ; USER FLAG 1
  00D2          +1   234     OV       BIT   PSW.2  ; OVERFLOW FLAG
  00D3          +1   235     RS0      BIT   PSW.3  ; REGISTER BANK SELECT 0
  00D4          +1   236     RS1      BIT   PSW.4  ; REGISTER BANK SELECT 1
  00D5          +1   237     F0       BIT   PSW.5  ; USER FLAG 0
```

```
 00D6        +1  238    AC       BIT   PSW.6  ; AUXILIARY CARRY FLAG
 00D7        +1  239    CY       BIT   PSW.7  ; CARRY FLAG
             +1  240    ;
             +1  241    ; PCA0CN D8H
 00D8        +1  242    CCF0     BIT   PCA0CN.0 ; PCA 0 MODULE 0 INTERRUPT FLAG
 00D9        +1  243    CCF1     BIT   PCA0CN.1 ; PCA 0 MODULE 1 INTERRUPT FLAG
 00DA        +1  244    CCF2     BIT   PCA0CN.2 ; PCA 0 MODULE 2 INTERRUPT FLAG
 00DB        +1  245    CCF3     BIT   PCA0CN.3 ; PCA 0 MODULE 3 INTERRUPT FLAG
 00DC        +1  246    CCF4     BIT   PCA0CN.4 ; PCA 0 MODULE 4 INTERRUPT FLAG
 00DE        +1  247    CR       BIT   PCA0CN.6 ; PCA 0 COUNTER RUN CONTROL BIT
 00DF        +1  248    CF       BIT   PCA0CN.7 ; PCA 0 COUNTER OVERFLOW FLAG
             +1  249    ;
             +1  250    ; ADC0CN E8H
 00E8        +1  251    AD0LJST  BIT   ADC0CN.0 ; ADC 0 RIGHT JUSTIFY DATA BIT
 00E9        +1  252    AD0WINT  BIT   ADC0CN.1 ; ADC 0 WINDOW COMPARE INTERRUPT FLAG
 00EA        +1  253    AD0STM0  BIT   ADC0CN.2 ; ADC 0 START OF CONVERSION MODE BIT 0
 00EB        +1  254    AD0STM1  BIT   ADC0CN.3 ; ADC 0 START OF CONVERSION MODE BIT 1
 00EC        +1  255    AD0BUSY  BIT   ADC0CN.4 ; ADC 0 BUSY FLAG
 00ED        +1  256    AD0INT   BIT   ADC0CN.5 ; ADC 0 CONVERISION COMPLETE INTERRUPT FLAG
```

```
  00EE          +1   257     AD0TM    BIT   ADC0CN.6 ; ADC 0 TRACK MODE
  00EF          +1   258     AD0EN    BIT   ADC0CN.7 ; ADC 0 ENABLE
                +1   259     ;
                +1   260     ; SPI0CN F8H
  00F8          +1   261     SPIEN    BIT   SPI0CN.0 ; SPI 0 SPI ENABLE
  00F9          +1   262     MSTEN    BIT   SPI0CN.1 ; SPI 0 MASTER ENABLE
  00FA          +1   263     SLVSEL   BIT   SPI0CN.2 ; SPI 0 SLAVE SELECT
  00FB          +1   264     TXBSY    BIT   SPI0CN.3 ; SPI 0 TX BUSY FLAG
  00FC          +1   265     RXOVRN   BIT   SPI0CN.4 ; SPI 0 RX OVERRUN FLAG
  00FD          +1   266     MODF     BIT   SPI0CN.5 ; SPI 0 MODE FAULT FLAG
  00FE          +1   267     WCOL     BIT   SPI0CN.6 ; SPI 0 WRITE COLLISION FLAG
  00FF          +1   268     SPIF     BIT   SPI0CN.7 ; SPI 0 INTERRUPT FLAG
                     269
                     270
                     271
                     272
                     273     ;======================================================================
                     274     ;   Variable declarations
                     275     ;======================================================================
                     276
                     277     ;LCD Commands
                     278     ;DISP CLR               EQU 00000001b ;Clears Disp & sets DDRAM addy to zero
                     279     ;DISP FUNCTION_CMD      EQU 00111000b ;Sets disp to 8-bit & 5x10 chars.
                     280     ;DISP ON                EQU 00001100b ;Turns disp ON,
                     281     ;DISP CURSOR            EQU 00001111b ;Turns disp & cursor ON, cursor flashing
                     282     ;DISP ENTRY MODE        EQU 00000110b ;Sets cursor move direction
                     283     ;DISP AUTOSHIFT_CURSOR EQU 00010100b ;Automatic move cursor right after send
                     284     ;DISP BACKSPACE         EQU 00010000b ;Shifts cursor left
                     285     ;DISP_SHIFTRT           EQU 00011100b ;Shifts entire display Right
                     286
                     287     ;LCD WRITE              EQU 1000h     ;LCD Write address RS=1 & RW=0
                     288     ;LCD READ               EQU 1100h     ;LCD Read busy address RS=0 & RW =1
                     289     ;LCD_CMD                EQU 1200h     ;LCD Command address RS=0 & RW =0
                     290
                     291     ;Keypad Commands
                     292     ;KEY_READ               EQU 4000h     ;Keypad read cmd addr. for DPTR
                     293
                     294
                     295     ;======================================================================
                     296     ;   Reset/Interrupt Vectors
                     297     ;======================================================================
                     298
0000                 299         org   0000h
0000 0216D4          300         ljmp  Main
                     301
0003                 302         org   0003h   ;/INT0 interrupt vector for Keypad
0003 02024D          303         ljmp  Key_ISR
```

```

0013                   org   0013h   ;/INT1 interrupd vector for Alarms
0013 021552            ljmp Alarm_Check

001B                   org   001Bh   ;Timer1 interrupt vector for geting acceleration
001B 020D36            ljmp  ADC_GetAcc





               ;=====================================================================
               ;   Main Routine
               ;=====================================================================


0030                   org   0030h

               ;=====================================================================
                   ;   Include Files
                   ;=====================================================================

                   ;$include (LCD.asm) ;LCD routines
            +1     ;=====================================================================
            +1     ;                               Pro-Tex 9000
            +1     ;
            +1     ;Revision: R.07171500  (R.MMDDHHMM)
            +1     ;
            +1     ;Project Team Members:
            +1     ; - Vince Watkins
            +1     ; - Will Smith
            +1     ; - Tyler Long
            +1     ;
            +1     ;LCD Subroutines
            +1     ;
            +1     ;
            +1     ;
            +1     ;=====================================================================
            +1
            +1
            +1     ;=====================================================================
            +1     ;   Variable declarations
            +1     ;=====================================================================
            +1
            +1     ;LCD Commands
  0001      +1     DISP CLR              EQU 00000001b ;Clears Disp & sets DDRAM addy to zero
  0038      +1     DISP FUNCTION_CMD     EQU 00111000b ;Sets disp to 8-bit & 5x10 chars.
  000C      +1     DISP ON               EQU 00001100b ;Turns disp ON,
  000F      +1     DISP CURSOR           EQU 00001111b ;Turns disp & cursor ON, cursor flashing
  0006      +1     DISP ENTRY MODE       EQU 00000110b ;Sets cursor move direction
  0014      +1     DISP AUTOSHIFT_CURSOR EQU 00010100b ;Automatic move cursor right after send
  0010      +1     DISP BACKSPACE        EQU 00010000b ;Shifts cursor left
  001C      +1     DISP_SHIFTRT          EQU 00011100b ;Shifts entire display Right
            +1
  1000      +1     LCD WRITE             EQU 1000h     ;LCD Write address RS=1 & RW=0
  1100      +1     LCD READ              EQU 1100h     ;LCD Read busy address RS=0 & RW =1
  1200      +1     LCD_CMD               EQU 1200h     ;LCD Command address RS=0 & RW =0
            +1
            +1
            +1     ;=====================================================================
            +1     ;   Sub routine - Initialize LCD
            +1     ;=====================================================================
            +1
0030        +1     LCD_Init:
0030 901200 +1         mov   DPTR,#LCD CMD
0033 75E038 +1         mov   ACC,#DISP_FUNCTION_CMD
```

```
0036 F0          +1   370         movx  @DPTR,A
0037 12009E      +1   371         lcall LCD_Busy
                 +1   372
003A 901200      +1   373         mov   DPTR,#LCD CMD
003D 75E00C      +1   374         mov   ACC,#DISP_ON
0040 F0          +1   375         movx  @DPTR,A
0041 12009E      +1   376         lcall LCD_Busy
                 +1   377
0044 901200      +1   378         mov   DPTR,#LCD CMD
0047 75E006      +1   379         mov   ACC,#DISP_ENTRY_MODE
004A F0          +1   380         movx  @DPTR,A
004B 12009E      +1   381         lcall LCD_Busy
                 +1   382
004E 120093      +1   383         lcall LCD_Clear
                 +1   384
0051 22          +1   385         ret
                 +1   386
                 +1   387
                 +1   388     ;=====================================================================
                +1   389     ;   Sub routine - Prints string to LCD
                +1   390     ;
                +1   391     ;Enter subroutine with cursor in correct location, DPTR pointing at
                +1   392     ;string to print, and ACC pointing to first location of string.
                +1   393     ;
                +1   394     ;====================================================================
                +1   395
                +1   396
0052            +1   397     LCD_Print:
0052 C0D0       +1   398         push  PSW
0054 C083       +1   399         push  DPH
0056 C082       +1   400         push  DPL
0058 C0E0       +1   401         push  ACC
005A C0F0       +1   402         push  B
005C 93         +1   403         movc  A,@A + DPTR
005D 6014       +1   404         jz    LCD Return      ;Null Character Reached
005F 901000     +1   405         mov   DPTR,#LCD_WRITE
0062 F0         +1   406         movx  @DPTR,A
0063 12009E     +1   407         lcall LCD_Busy
                +1   408
0066            +1   409     LCD_Restore:
0066 D0F0       +1   410         pop   B
0068 D0E0       +1   411         pop   ACC
006A D082       +1   412         pop   DPL
006C D083       +1   413         pop   DPH
006E D0D0       +1   414         pop   PSW
0070 A3         +1   415         inc   DPTR
                +1   416
0071 80DF       +1   417         jmp   LCD_Print
                +1   418
0073            +1   419     LCD_Return:
0073 D0F0       +1   420         pop   B
0075 D0E0       +1   421         pop   ACC
0077 D082       +1   422         pop   DPL
0079 D083       +1   423         pop   DPH
007B D0D0       +1   424         pop   PSW
007D A3         +1   425         inc   DPTR       ;Leave sub with DPTR at next string in db
                +1   426
007E 22         +1   427         ret
                +1   428
                +1   429     ;====================================================================
                +1   430     ;   Sub routine - 3.0 second wait delay for screen transitions
                +1   431     ;
                +1   432     ;Registers used:
                +1   433     ; - R2
                +1   434     ;
                +1   435     ;Timers used:
```

```
               +1   436     ; - Timer0
               +1   437     ;
               +1   438     ;=====================================================================
               +1   439
007F           +1   440     LCD_Wait_3sec:
007F 7A3C      +1   441         mov     R2,#60     ;15=1sec.
0081 758C00    +1   442         mov     TH0,#00h
0084 758A00    +1   443         mov     TL0,#00h
0087 D28C      +1   444         setb    TR0
               +1   445
0089           +1   446     LCD_Timer0_OV:
0089 308DFD    +1   447         jnb     TF0,LCD_Timer0_OV
008C C28D      +1   448         clr     TF0
008E DAF9      +1   449         djnz    R2,LCD_Timer0_OV
0090 C28C      +1   450         clr     TR0
0092 22        +1   451         ret
               +1   452
               +1   453
               +1   454
```

```
                  +1   455     ;====================================================================
                  +1   456     ;  Sub routine - Clear LCD
                  +1   457     ;====================================================================
                  +1   458
0093              +1   459     LCD_Clear:
0093 901200       +1   460         mov   DPTR,#LCD CMD
0096 75E001       +1   461         mov   ACC,#DISP_CLR
0099 F0           +1   462         movx  @DPTR,A
009A 12009E       +1   463         lcall LCD_Busy
009D 22           +1   464         ret
                  +1   465
                  +1   466     ;====================================================================
                  +1   467     ;  Sub routine - Wait for LCD
                  +1   468     ;====================================================================
                  +1   469
009E              +1   470     LCD_Busy:
009E 901100       +1   471         mov   DPTR,#LCD_READ
00A1 E0           +1   472         movx  A,@DPTR
00A2 20E7F9       +1   473         JB    ACC.7,LCD_Busy  ;If bit 7 high, LCD still busy
00A5 22           +1   474         ret
                  +1   475
                  +1   476
                  +1   477
                  +1   478     ;-----------------------------------
                  +1   479     ;-  LCD Screen Strings  --
                  +1   480     ;-----------------------------------
                  +1   481
00A6              +1   482     LCD_First:   ;State 00h
00A6 45434554     +1   483         db "ECET 3220 Summer 08",0
00AA 20333232
00AE 30205375
00B2 6D6D6572
00B6 20303800
00BA 54796C65     +1   484         db "Tyler Long",0
00BE 72204C6F
00C2 6E6700
00C5 57696C20     +1   485         db "Wil Smith",0
00C9 536D6974
00CD 6800
00CF 56696E63     +1   486         db "Vincent Watkins",0
00D3 656E7420
00D7 5761746B
00DB 696E7300
                  +1   487
00DF              +1   488     LCD_Pro Tex:        ;State_01h
00DF 50726F2D     +1   489         db "Pro-Tex 9000",0
00E3 54657820
```

```
00E7 39303030
00EB 00
                +1   490
00EC            +1   491     LCD_Password_Entry: ;State_02h & 0Ah
00EC 456E7465   +1   492         db "Enter PW:",0
00F0 72205057
00F4 3A00
                +1   493
00F6            +1   494     LCD_PW_Bad:          ;State_03h,04h,0Ch, & 0Dh
00F6 496E7661   +1   495         db "Invalid PW",0
00FA 6C696420
00FE 505700
0101 54727920   +1   496         db "Try Again:",0
0105 41676169
0109 6E3A00
                +1   497
010C            +1   498     LCD_SysLocked:       ;State_05h
010C 53797374   +1   499         db "System  Locked",0
0110 656D2020
```

```
0114 4C6F636B
0118 656400
                +1   500
011B            +1   501     LCD_Home:       ;State_06h
011B 486F6D65   +1   502         db "Home",0
011F 00
0120 41636365   +1   503         db "Accel:      ",0
0124 6C3A2020
0128 20202020
012C 00
012D 41636365   +1   504         db "Accel STPT:     ",0
0131 6C205354
0135 50543A20
0139 20202020
013D 00
013E 50726573   +1   505         db "Press ",22h,"ENT",22h," for Menu",0
0142 73202245
0146 4E542220
014A 666F7220
014E 4D656E75
0152 00
                +1   506
0153            +1   507     LCD_Main Menu:  ;State 07h
0153 4D656E75   +1   508         db "Menu  ",22h,"ENT",22h,"=","Home",0
0157 20202245
015B 4E54223D
015F 486F6D65
0163 00
0164 312E2041   +1   509         db "1. Arm/Dis",0
0168 726D2F44
016C 697300
016F 322E2043   +1   510         db "2. Change Acc STPT",0
0173 68616E67
0177 65204163
017B 63205354
017F 505400
0182 332E2043   +1   511         db "3. Change PW",0
0186 68616E67
018A 65205057
018E 00
                +1   512
018F            +1   513     LCD_ArmDis:     ;State_08h
018F 312E2041   +1   514         db "1. Arm",0
0193 726D00
0196 322E2044   +1   515         db "2. Disarm",0
019A 69736172
019E 6D00
```

```
01A0 50726573   +1   516         db "Press ",22h,"ENT",22h," for Menu",0
01A4 73202245
01A8 4E542220
01AC 666F7220
01B0 4D656E75
01B4 00
                +1   517
01B5            +1   518     LCD_SysArmed:   ;State 09h
01B5 53797374   +1   519         db "System Armed",0
01B9 656D2041
01BD 726D6564
01C1 00
                +1   520
01C2            +1   521     LCD_SysDisArmed:   ;State 0Bh
01C2 53797374   +1   522         db "System  Disarmed",0
01C6 656D2020
01CA 44697361
01CE 726D6564
01D2 00
```

```
                +1   523
01D3            +1   524     LCD_AccStpt:     ;State 0Eh
01D3 456E7465   +1   525         db "Enter New Acc STPT",0
01D7 72204E65
01DB 77204163
01DF 63205354
01E3 505400
01E6 20307E31   +1   526         db " 0",7Eh,"120 ","= ","0",7Eh,"+/-","1.20g ",0
01EA 3230203D
01EE 20307E2B
01F2 2F2D312E
01F6 32306720
01FA 00
01FB 7E00       +1   527         db 7Eh,0
                +1   528
01FD            +1   529     LCD_Valid_STPT:   ;State 0Fh
01FD 53657470   +1   530         db "Setpoint Changed",0
0201 6F696E74
0205 20436861
0209 6E676564
020D 00
                +1   531
020E            +1   532     LCD_Invalid_STPT:   ;State 10h
020E 496E7661   +1   533         db "Invalid Setpoint",0
0212 6C696420
0216 53657470
021A 6F696E74
021E 00
                +1   534
                +1   535
021F            +1   536     LCD_Current_PW:   ;State 11h
021F 456E7465   +1   537         db "Enter Curr PW:",0
0223 72204375
0227 72722050
022B 573A00
                +1   538
022E            +1   539     LCD_New_PW:   ;State 14h
022E 456E7465   +1   540         db "Enter New PW:",0
0232 72204E65
0236 77205057
023A 3A00
                +1   541
023C            +1   542     LCD_Changed_PW:   ;State 15h
023C 50617373   +1   543         db "Password Changed",0
0240 776F7264
0244 20436861
0248 6E676564
```

```
024C 00
                +1
                        ;$include (Key.asm) ;Keypad routines
                +1      ;======================================================================
                +1      ;                             Pro-Tex 9000
                +1      ;
                +1      ;Revision: R.07171500  (R.MMDDHHMM)
                +1      ;
                +1      ;Project Team Members:
                +1      ; - Vince Watkins
                +1      ; - Will Smith
                +1      ; - Tyler Long
                +1      ;
                +1      ;
                +1      ;Keypad Subroutines
                +1      ;
                +1      ;
                +1      ;======================================================================
                +1
                +1   562     ;======================================================================
                +1   563     ;   Variable declarations
                +1   564     ;======================================================================
                +1   565
                +1   566     ;Keypad Commands
  4000          +1   567     KEY_READ              EQU 4000h     ;Keypad read cmd addr. for DPTR
                +1   568
                +1   569
                +1   570
                +1   571
                +1   572
                +1   573     ;======================================================================
                +1   574     ;   Sub routine - Keypad ISR
                +1   575     ;
                +1   576     ;Data is left in ACC after this ISR
                +1   577     ;
                +1   578     ;Registers
                +1   579     ; - R1: Determines BS/non_Func key presses allowed
                +1   580     ;
                +1   581     ;======================================================================
                +1   582
024D            +1   583     Key_ISR:
                +1   584
024D C0D0       +1   585         push    PSW
024F C083       +1   586         push    DPH
0251 C082       +1   587         push    DPL
0253 C0E0       +1   588         push    ACC
0255 C0F0       +1   589         push    B
                +1   590
                +1   591        ; jnb     19h,test   ;Timer0 status
                +1   592         ;setb    TF0   ;Fake out program on return to TF0 in ADC.asm
                +1   593
0257 904000     +1   594         mov     DPTR,#KEY_READ
025A E0         +1   595         movx    A,@DPTR
025B 540F       +1   596         anl     A,#0Fh          ;Bit mask
                +1   597
                +1   598     ;Function key check
025D 605E       +1   599         jz      Key_Backspace   ;BS key pressed
                +1   600
                +1   601     ;Enter key check
025F C0E0       +1   602         push    ACC             ;save ACC with bit masked value
0261 C3         +1   603         clr     C
0262 9408       +1   604         subb    A,#08h          ;08h=Enter key
0264 605C       +1   605         jz      Key_Enter       ;ENT key pressed
0266 D0E0       +1   606         pop     ACC             ;Restore ACC from Bit masked value
                +1   607
                +1   608     ;Caps Lock key check
```

```
0268 C0E0       +1   609         push    ACC             ;save ACC with bit masked value
026A C3         +1   610         clr     C
026B 9404       +1   611         subb    A,#04h          ;04h=Caps lock key
026D 602B       +1   612         jz      Key_Caps        ;Caps lock key pressed
026F D0E0       +1   613         pop     ACC             ;Restore ACC from Bit masked value
                +1   614
                +1   615     ;Blue Function key check
0271 C0E0       +1   616         push    ACC             ;save ACC with bit masked value
0273 C3         +1   617         clr     C
0274 940C       +1   618         subb    A,#0Ch          ;0Ch=Blue key
0276 6029       +1   619         jz      Key_Blue        ;Blue key pressed
0278 D0E0       +1   620         pop     ACC             ;Restore ACC from Bit masked value
                +1   621
                +1   622     ;Pink Function key check
027A C0E0       +1   623         push    ACC             ;save ACC with bit masked value
027C C3         +1   624         clr     C
027D 940D       +1   625         subb    A,#0Dh          ;0Dh=Pink key
027F 6027       +1   626         jz      Key_Pink        ;Pink key pressed
0281 D0E0       +1   627         pop     ACC             ;Restore ACC from Bit masked value
```

```
                 +1   628
                 +1   629     ;Green Function key check
0283 C0E0        +1   630         push    ACC              ;save ACC with bit masked value
0285 C3          +1   631         clr     C
0286 940E        +1   632         subb    A,#0Eh           ;0Eh=Green key
0288 6025        +1   633         jz      Key_Green        ;Green key pressed
028A D0E0        +1   634         pop     ACC              ;Restore ACC from Bit masked value
                 +1   635
                 +1   636     ;Red Function key check
028C C0E0        +1   637         push    ACC              ;save ACC with bit masked value
028E C3          +1   638         clr     C
028F 940F        +1   639         subb    A,#0Fh           ;0Fh=Red key
0291 6023        +1   640         jz      Key_Red          ;Red key pressed
0293 D0E0        +1   641         pop     ACC              ;Restore ACC from Bit masked value
                 +1   642
                 +1   643
0295 1205B3      +1   644         lcall   Key_State_Chk     ;Program reaches this point if no
                 +1   645                                   ;function keys pressed
                 +1   646
0298 802F        +1   647         jmp     Key_KeyRelease
                 +1   648
029A             +1   649     Key_Caps:
029A D0E0        +1   650         pop     ACC
029C 120576      +1   651         lcall   Key Func Caps
029F 8028        +1   652         jmp     Key_KeyRelease
                 +1   653
02A1             +1   654     Key_Blue:
02A1 D0E0        +1   655         pop     ACC
02A3 12057B      +1   656         lcall   Key Func Blue
02A6 8021        +1   657         jmp     Key_KeyRelease
                 +1   658
02A8             +1   659     Key_Pink:
02A8 D0E0        +1   660         pop     ACC
02AA 120589      +1   661         lcall   Key Func Pink
02AD 801A        +1   662         jmp     Key_KeyRelease
                 +1   663
02AF             +1   664     Key_Green:
02AF D0E0        +1   665         pop     ACC
02B1 120597      +1   666         lcall   Key Func Green
02B4 8013        +1   667         jmp     Key_KeyRelease
                 +1   668
02B6             +1   669     Key_Red:
02B6 D0E0        +1   670         pop     ACC
02B8 1205A5      +1   671         lcall   Key Func Red
02BB 800C        +1   672         jmp     Key_KeyRelease
                 +1   673
02BD             +1   674     Key_Backspace:
```

```
02BD 1204F4     +1   675        lcall   Key Func BS
02C0 8007       +1   676        jmp     Key_KeyRelease
                +1   677
02C2            +1   678    Key_Enter:
02C2 D0E0       +1   679        pop     ACC               ;Restore ACC
02C4 1202D7     +1   680        lcall   Key Func Ent      ;go to check state
02C7 8000       +1   681        jmp     Key_KeyRelease
                +1   682
                +1   683
02C9            +1   684    Key_KeyRelease:
02C9 3082FD     +1   685        jnb     P0.2,$    ;Wait for release of key /INT0
                +1   686
02CC D0F0       +1   687        pop     B
02CE D0E0       +1   688        pop     ACC
02D0 D082       +1   689        pop     DPL
02D2 D083       +1   690        pop     DPH
02D4 D0D0       +1   691        pop     PSW
                +1   692
02D6 32         +1   693        reti
```

```
                 +1
                 +1     ;======================================================================
                 +1     ;   Sub routine - Enter Function Key valid state check
                 +1     ;
                 +1     ;This routine determines if the current state allows for
                 +1     ;the enter key to be pressed.
                 +1     ;
                 +1     ;Addresses:
                 +1     ; - 21h: Checks for state when Enter key ok to press
                 +1     ;
                 +1     ;Registers:
                 +1     ; - R1: Points to current state (21h)
                 +1     ;======================================================================
                 +1
02D7             +1     Key_Func_Ent:
02D7 B70208      +1         cjne    @R1,#02h,Key_Func_Ent_01      ;Check for State 02h
02DA 120737      +1         lcall   RAM Read PW                   ;Get current PW from RAM
02DD 12048B      +1         lcall   Key PW Check 02h              ;compare PW
02E0 616D        +1         jmp     Key_Func_Ent_Finish
                 +1
02E2             +1     Key_Func_Ent 01:
02E2 B70308      +1         cjne    @R1,#03h,Key_Func_Ent_02      ;Check for State 03h
02E5 120737      +1         lcall   RAM Read PW                   ;Get current PW from RAM
02E8 1204AE      +1         lcall   Key PW Check 03h              ;compare PW
02EB 616D        +1         jmp     Key_Func_Ent_Finish
                 +1
02ED             +1     Key_Func_Ent 02:
02ED B70408      +1         cjne    @R1,#04h,Key_Func_Ent_03      ;Check for State 04h
02F0 120737      +1         lcall   RAM Read PW                   ;Get current PW from RAM
02F3 1204D1      +1         lcall   Key PW Check 04h              ;compare PW
02F6 8075        +1         jmp     Key_Func_Ent_Finish
                 +1
02F8             +1     Key_Func_Ent 03:
02F8 B70607      +1         cjne    @R1,#06h,Key_Func_Ent_04      ;Check for State_06h
02FB 7407        +1         mov     A,#07h                        ;Go to State 07h
02FD 120D27      +1         lcall   State Lookup                  ;Initiate State_07h
0300 806B        +1         jmp     Key_Func_Ent_Finish
                 +1
0302             +1     Key_Func_Ent 04:
0302 B70707      +1         cjne    @R1,#07h,Key_Func_Ent_05      ;Check for State_07h
0305 7406        +1         mov     A,#06h                        ;Go to State 06h
0307 120D27      +1         lcall   State Lookup                  ;Initiate State_06h
030A 8061        +1         jmp     Key_Func_Ent_Finish
                 +1
030C             +1     Key_Func_Ent 05:
030C B70A08      +1         cjne    @R1,#0Ah,Key_Func_Ent_06      ;Check for State 0Ah
030F 120737      +1         lcall   RAM_Read_PW                   ;Get current PW from RAM
```

```
0312 120468      +1   741          lcall   Key PW Check 0Ah               ;compare PW
0315 8056        +1   742          jmp     Key_Func_Ent_Finish
                 +1   743
0317             +1   744      Key_Func_Ent 06:
0317 B70C08      +1   745          cjne    @R1,#0Ch,Key_Func_Ent_07       ;Check for State 0Ch
031A 120737      +1   746          lcall   RAM Read PW                    ;Get current PW from RAM
031D 120445      +1   747          lcall   Key PW Check 0Ch               ;compare PW
0320 804B        +1   748          jmp     Key_Func_Ent_Finish
                 +1   749
0322             +1   750      Key_Func_Ent 07:
0322 B70D08      +1   751          cjne    @R1,#0Dh,Key_Func_Ent_08       ;Check for State 0Dh
0325 120737      +1   752          lcall   RAM Read PW                    ;Get current PW from RAM
0328 120422      +1   753          lcall   Key PW Check 0Dh               ;compare PW
032B 8040        +1   754          jmp     Key_Func_Ent_Finish
                 +1   755
032D             +1   756      Key_Func_Ent 08:
032D B70807      +1   757          cjne    @R1,#08h,Key_Func_Ent_09       ;Check for State_08h
0330 7407        +1   758          mov     A,#07h                         ;Go to State 07h
0332 120D27      +1   759          lcall   State_Lookup                   ;Initiate State_07h
```

```
0335 8036       +1   760         jmp     Key_Func_Ent_Finish
                +1   761
0337            +1   762     Key_Func_Ent 09:
0337 B71108     +1   763         cjne    @R1,#11h,Key_Func_Ent_10      ;Check for State 11h
033A 120737     +1   764         lcall   RAM Read PW                   ;Get current PW from RAM
033D 1203FF     +1   765         lcall   Key PW Check 11h              ;compare PW
0340 802B       +1   766         jmp     Key_Func_Ent_Finish
                +1   767
0342            +1   768     Key_Func_Ent 10:
0342 B71208     +1   769         cjne    @R1,#12h,Key_Func_Ent_11      ;Check for State 12h
0345 120737     +1   770         lcall   RAM Read PW                   ;Get current PW from RAM
0348 1203DC     +1   771         lcall   Key PW Check 12h              ;compare PW
034B 8020       +1   772         jmp     Key_Func_Ent_Finish
                +1   773
034D            +1   774     Key_Func_Ent 11:
034D B71308     +1   775         cjne    @R1,#13h,Key_Func_Ent_12      ;Check for State 13h
0350 120737     +1   776         lcall   RAM Read PW                   ;Get current PW from RAM
0353 1203B9     +1   777         lcall   Key PW Check 13h              ;compare PW
0356 8015       +1   778         jmp     Key_Func_Ent_Finish
                +1   779
0358            +1   780     Key_Func_Ent 12:
0358 B7140A     +1   781         cjne    @R1,#14h,Key_Func_Ent_13      ;Check for State 13h
035B 12074B     +1   782         lcall   RAM Write_PW                  ;Update RAM w/ new PW
035E 7415       +1   783         mov     A,#15h                        ;Go to State_15h
0360 120D27     +1   784         lcall   State Lookup
0363 8008       +1   785         jmp     Key_Func_Ent_Finish
                +1   786
0365            +1   787     Key_Func_Ent 13:
0365 B70E05     +1   788         cjne    @R1,#0Eh,Key Func Ent_Finish  ;Check for State_0Eh
0368 12036E     +1   789         lcall   Key Accel Valid Check               ;
036B 8000       +1   790         jmp     Key_Func_Ent_Finish
                +1   791
                +1   792
036D            +1   793     Key_Func_Ent_Finish:
036D 22         +1   794         ret
                +1   795
                +1   796     ;=================================================================
                +1   797     ;   Sub routine - State 0E Valid Acceleration STPT Check
                +1   798     ;
                +1   799     ;This sub branches to either State 10h or 0Fh based on a valid
                +1   800     ;setpoint being entered.
                +1   801     ;
                +1   802     ;Addresses:
                +1   803     ; - 21h: Checks for state when Enter key ok to press
                +1   804     ; - 27h: MSB for Accel STPT
                +1   805     ; - 26h: Next byte for Accel STPT
                +1   806     ; - 25h: LSB for Accel STPT
                +1   ;
                +1   ;Registers:
                +1   ; - R1: Points to current state (21h)
                +1   ;==================================================================
                +1
036E            +1   Key_Accel_Valid_Check:
                +1
                +1
036E E527       +1       mov   A,27h                      ;MSB Accel STPT entered from RAM
0370 B43200     +1       cjne  A,#32h,$ + 3               ;1's digit greater than 1?
0373 503C       +1       jnc   Key_Accel_Invalid          ;Load invalid STPT state
                +1
0375 B43100     +1       cjne  A,#31h,$ + 3               ;1's digit 0 or 1?
0378 400E       +1       jc    Key_Accel_Easy             ;Jump if 1's digit ='s 0
                +1                                        ;else 1's digit ='s 1
                +1
037A E526       +1       mov   A,26h                      ;tenth's Accel byte
037C B43300     +1       cjne  A,#33h,$ + 3               ;Tenth's digit > 2?
037F 5030       +1       jnc   Key_Accel_Invalid          ;Tenth's digit too high
```

```
                 +1   826
0381 B43200      +1   827         cjne  A,#32h,$ + 3            ;Tenth's digit = 2?
0384 5012        +1   828         jnc   Key Accel Easy1         ;Tenth's digit = 2
0386 8019        +1   829         jmp   Key_Accel_Easy2         ;Tenth's digit < 2
                 +1   830
                 +1   831
0388             +1   832     Key_Accel_Easy:  ;1's digit ='s 0
0388 E526        +1   833         mov   A,26h
038A B44000      +1   834         cjne  A,#40h,$ + 3            ;
038D 5022        +1   835         jnc   Key_Accel_Invalid       ;Tenth's digit >=40h
                 +1   836
038F E525        +1   837         mov   A,25h
0391 B44000      +1   838         cjne  A,#40h,$ + 3
0394 501B        +1   839         jnc   Key Accel Invalid       ;100's digit >=40h
0396 8012        +1   840         jmp   Key_Accel_Valid         ;else jump to valid state
                 +1   841
0398             +1   842     Key_Accel_Easy1:  ;1's digit ='s 1 & 10's=2
0398 E525        +1   843         mov   A,25h
039A B43100      +1   844         cjne  A,#31h,$ + 3
039D 5012        +1   845         jnc   Key Accel Invalid       ;100's digit >=40h
039F 8009        +1   846         jmp   Key_Accel_Valid         ;else jump to valid state
                 +1   847
03A1             +1   848     Key_Accel_Easy2:  ;1's digit ='s 1 & 10's < 2
03A1 E525        +1   849         mov   A,25h
03A3 B44000      +1   850         cjne  A,#40h,$ + 3
03A6 5009        +1   851         jnc   Key Accel Invalid       ;100's digit >=40h
03A8 8000        +1   852         jmp   Key_Accel_Valid         ;else jump to valid state
                 +1   853
03AA             +1   854     Key_Accel_Valid:
03AA 740F        +1   855         mov   A,#0Fh                  ;State 0Fh
03AC 120D27      +1   856         lcall State Lookup
03AF 8007        +1   857         jmp   Key_Accel_Valid_Finish  ;Program returns here once the state
                 +1   858                                       ;machine is finished
                 +1   859
03B1             +1   860     Key_Accel_Invalid:
03B1 7410        +1   861         mov   A,#10h                  ;State 10h
03B3 120D27      +1   862         lcall State Lookup
03B6 8000        +1   863         jmp   Key_Accel_Valid_Finish  ;Program returns here once the state
                 +1   864                                       ;machine is finished
                 +1   865
03B8             +1   866     Key_Accel_Valid_Finish:
03B8 22          +1   867         ret                           ;ret to ENT key state check
                 +1   868
                 +1   869
                 +1   870     ;====================================================================
                 +1   871     ;   Sub routine - State 13h Password check
                 +1   872     ;
```

```
                  +1   873     ;This routine determines if the entered password matches that of the
                  +1   874     ;one stored in RAM.
                  +1   875     ;
                  +1   876     ;Addresses:
                  +1   877     ; - 21h: Checks for state when Enter key ok to press
                  +1   878     ;
                  +1   879     ;Registers:
                  +1   880     ; - R1: Points to current state (21h)
                  +1   881     ;=====================================================================
                  +1   882
03B9              +1   883     Key_PW Check 13h:
03B9 E52F         +1   884         mov   A,2Fh                   ;MSB PW from RAM
03BB B52B16       +1   885         cjne  A,2Bh,Key_PW_Bad_13h
03BE E52E         +1   886         mov   A,2Eh                   ;
03C0 B52A11       +1   887         cjne  A,2Ah,Key_PW_Bad_13h
03C3 E52D         +1   888         mov   A,2Dh                   ;
03C5 B5290C       +1   889         cjne  A,29h,Key_PW_Bad_13h
03C8 E52C         +1   890         mov   A,2Ch                   ;LSB PW from RAM
03CA B52807       +1   891         cjne  A,28h,Key_PW_Bad_13h    ;Program jumps to bad PW state if PW
```

```
                +1   892                                       ;entered was incorrect
                +1   893
03CD            +1   894     Key_PW Ok_13h:
03CD 7414       +1   895         mov   A,#14h                  ;State 14h
03CF 120D27     +1   896         lcall State Lookup
03D2 8007       +1   897         jmp   Key_PW_Check_13h_Finish ;Program returns here once the state
                +1   898                                       ;machine is finished
                +1   899
03D4            +1   900     Key_PW Bad 13h:
03D4 7405       +1   901         mov   A,#05h                  ;State 05h
03D6 120D27     +1   902         lcall State Lookup
03D9 8000       +1   903         jmp   Key_PW_Check_13h_Finish ;Program returns here once the state
                +1   904                                       ;machine is finished
                +1   905
03DB            +1   906     Key_PW Check_13h_Finish:
03DB 22         +1   907         ret                           ;ret to ENT key state check
                +1   908
                +1   909
                +1   910     ;=====================================================================
                +1   911     ;   Sub routine - State 12h Password check
                +1   912     ;
                +1   913     ;This routine determines if the entered password matches that of the
                +1   914     ;one stored in RAM.
                +1   915     ;
                +1   916     ;Addresses:
                +1   917     ; - 21h: Checks for state when Enter key ok to press
                +1   918     ;
                +1   919     ;Registers:
                +1   920     ; - R1: Points to current state (21h)
                +1   921     ;=====================================================================
                +1   922
03DC            +1   923     Key_PW Check 12h:
03DC E52F       +1   924         mov   A,2Fh                   ;MSB PW from RAM
03DE B52B16     +1   925         cjne  A,2Bh,Key_PW_Bad_12h
03E1 E52E       +1   926         mov   A,2Eh                   ;
03E3 B52A11     +1   927         cjne  A,2Ah,Key_PW_Bad_12h
03E6 E52D       +1   928         mov   A,2Dh                   ;
03E8 B5290C     +1   929         cjne  A,29h,Key_PW_Bad_12h
03EB E52C       +1   930         mov   A,2Ch                   ;LSB PW from RAM
03ED B52807     +1   931         cjne  A,28h,Key_PW_Bad_12h    ;Program jumps to bad PW state if PW
                +1   932                                       ;entered was incorrect
                +1   933
03F0            +1   934     Key_PW Ok_12h:
03F0 7414       +1   935         mov   A,#14h                  ;State 14h
03F2 120D27     +1   936         lcall State Lookup
03F5 8007       +1   937         jmp   Key_PW_Check_12h_Finish ;Program returns here once the state
                +1   938                                       ;machine is finished
                +1   939
03F7            +1   940     Key_PW Bad 12h:
03F7 7413       +1   941         mov   A,#13h                  ;State 13h
03F9 120D27     +1   942         lcall State Lookup
03FC 8000       +1   943         jmp   Key_PW_Check_12h_Finish ;Program returns here once the state
                +1   944                                       ;machine is finished
                +1   945
03FE            +1   946     Key_PW Check_12h_Finish:
03FE 22         +1   947         ret                           ;ret to ENT key state check
                +1   948
                +1   949
                +1   950     ;===================================================================
                +1   951     ;   Sub routine - State 11h Password check
                +1   952     ;
                +1   953     ;This routine determines if the entered password matches that of the
                +1   954     ;one stored in RAM.
                +1   955     ;
                +1   956     ;Addresses:
                +1   957     ; - 21h: Checks for state when Enter key ok to press
```

```
                 +1   958     ;
                 +1   959     ;Registers:
                 +1   960     ; - R1: Points to current state (21h)
                 +1   961     ;=====================================================================
                 +1   962
03FF             +1   963     Key_PW Check 11h:
03FF E52F        +1   964         mov   A,2Fh                   ;MSB PW from RAM
0401 B52B16      +1   965         cjne  A,2Bh,Key_PW_Bad_11h
0404 E52E        +1   966         mov   A,2Eh                   ;
0406 B52A11      +1   967         cjne  A,2Ah,Key_PW_Bad_11h
0409 E52D        +1   968         mov   A,2Dh                   ;
040B B5290C      +1   969         cjne  A,29h,Key_PW_Bad_11h
040E E52C        +1   970         mov   A,2Ch                   ;LSB PW from RAM
0410 B52807      +1   971         cjne  A,28h,Key_PW_Bad_11h    ;Program jumps to bad PW state if PW
                 +1   972                                       ;entered was incorrect
                 +1   973
0413             +1   974     Key_PW Ok_11h:
0413 7414        +1   975         mov   A,#14h                  ;State 14h
0415 120D27      +1   976         lcall State Lookup
0418 8007        +1   977         jmp   Key_PW_Check_11h_Finish ;Program returns here once the state
                 +1   978                                       ;machine is finished
                 +1   979
041A             +1   980     Key_PW Bad 11h:
041A 7412        +1   981         mov   A,#12h                  ;State 12h
041C 120D27      +1   982         lcall State Lookup
041F 8000        +1   983         jmp   Key_PW_Check_11h_Finish ;Program returns here once the state
                 +1   984                                       ;machine is finished
                 +1   985
0421             +1   986     Key_PW Check_11h_Finish:
0421 22          +1   987         ret                           ;ret to ENT key state check
                 +1   988
                 +1   989
                 +1   990     ;=====================================================================
                 +1   991     ;   Sub routine - State 0Dh Password check
                 +1   992     ;
                 +1   993     ;This routine determines if the entered password matches that of the
                 +1   994     ;one stored in RAM.
                 +1   995     ;
                 +1   996     ;Addresses:
                 +1   997     ; - 21h: Checks for state when Enter key ok to press
                 +1   998     ;
                 +1   999     ;Registers:
                 +1  1000     ; - R1: Points to current state (21h)
                 +1  1001     ;=====================================================================
                 +1  1002
0422             +1  1003     Key_PW Check 0Dh:
0422 E52F        +1  1004         mov   A,2Fh                   ;MSB PW from RAM
```

```
0424 B52B16      +1  1005         cjne  A,2Bh,Key_PW_Bad_0Dh
0427 E52E        +1  1006         mov   A,2Eh                    ;
0429 B52A11      +1  1007         cjne  A,2Ah,Key_PW_Bad_0Dh
042C E52D        +1  1008         mov   A,2Dh                    ;
042E B5290C      +1  1009         cjne  A,29h,Key_PW_Bad_0Dh
0431 E52C        +1  1010         mov   A,2Ch                    ;LSB PW from RAM
0433 B52807      +1  1011         cjne  A,28h,Key_PW_Bad_0Dh    ;Program jumps to bad PW state if PW
                 +1  1012                                        ;entered was incorrect
                 +1  1013
0436             +1  1014     Key_PW Ok_0Dh:
0436 740B        +1  1015         mov   A,#0Bh                   ;State 0Bh
0438 120D27      +1  1016         lcall State Lookup
043B 8007        +1  1017         jmp   Key_PW_Check_0Dh_Finish ;Program returns here once the state
                 +1  1018                                        ;machine is finished
                 +1  1019
043D             +1  1020     Key_PW Bad 0Dh:
043D 7405        +1  1021         mov   A,#05h                   ;State 05h
043F 120D27      +1  1022         lcall State Lookup
0442 8000        +1  1023         jmp   Key_PW_Check_0Dh_Finish ;Program returns here once the state
```

```
                +1  1024                                       ;machine is finished
                +1  1025
0444            +1  1026     Key_PW Check_0Dh_Finish:
0444 22         +1  1027         ret                           ;ret to ENT key state check
                +1  1028
                +1  1029
                +1  1030     ;=====================================================================
                +1  1031     ;   Sub routine - State 0Ch Password check
                +1  1032     ;
                +1  1033     ;This routine determines if the entered password matches that of the
                +1  1034     ;one stored in RAM.
                +1  1035     ;
                +1  1036     ;Addresses:
                +1  1037     ; - 21h: Checks for state when Enter key ok to press
                +1  1038     ;
                +1  1039     ;Registers:
                +1  1040     ; - R1: Points to current state (21h)
                +1  1041     ;=====================================================================
                +1  1042
0445            +1  1043     Key_PW Check 0Ch:
0445 E52F       +1  1044         mov   A,2Fh                   ;MSB PW from RAM
0447 B52B16     +1  1045         cjne  A,2Bh,Key_PW_Bad_0Ch
044A E52E       +1  1046         mov   A,2Eh                   ;
044C B52A11     +1  1047         cjne  A,2Ah,Key_PW_Bad_0Ch
044F E52D       +1  1048         mov   A,2Dh                   ;
0451 B5290C     +1  1049         cjne  A,29h,Key_PW_Bad_0Ch
0454 E52C       +1  1050         mov   A,2Ch                   ;LSB PW from RAM
0456 B52807     +1  1051         cjne  A,28h,Key_PW_Bad_0Ch    ;Program jumps to bad PW state if PW
                +1  1052                                       ;entered was incorrect
                +1  1053
0459            +1  1054     Key_PW Ok_0Ch:
0459 740B       +1  1055         mov   A,#0Bh                  ;State 0Bh
045B 120D27     +1  1056         lcall State Lookup
045E 8007       +1  1057         jmp   Key_PW_Check_0Ch_Finish ;Program returns here once the state
                +1  1058                                       ;machine is finished
                +1  1059
0460            +1  1060     Key_PW Bad 0Ch:
0460 740D       +1  1061         mov   A,#0Dh                  ;State 0Dh
0462 120D27     +1  1062         lcall State Lookup
0465 8000       +1  1063         jmp   Key_PW_Check_0Ch_Finish ;Program returns here once the state
                +1  1064                                       ;machine is finished
                +1  1065
0467            +1  1066     Key_PW Check_0Ch_Finish:
0467 22         +1  1067         ret                           ;ret to ENT key state check
                +1  1068
                +1  1069     ;=====================================================================
                +1  1070     ;   Sub routine - State 0Ah Password check
                +1  1071     ;
                +1  1072     ;This routine determines if the entered password matches that of the
                +1  1073     ;one stored in RAM.
                +1  1074     ;
                +1  1075     ;Addresses:
                +1  1076     ; - 21h: Checks for state when Enter key ok to press
                +1  1077     ;
                +1  1078     ;Registers:
                +1  1079     ; - R1: Points to current state (21h)
                +1  1080     ;======================================================================
                +1  1081
0468            +1  1082     Key_PW Check 0Ah:
0468 E52F       +1  1083         mov   A,2Fh                  ;MSB PW from RAM
046A B52B16     +1  1084         cjne  A,2Bh,Key_PW_Bad_0Ah
046D E52E       +1  1085         mov   A,2Eh                  ;
046F B52A11     +1  1086         cjne  A,2Ah,Key_PW_Bad_0Ah
0472 E52D       +1  1087         mov   A,2Dh                  ;
0474 B5290C     +1  1088         cjne  A,29h,Key_PW_Bad_0Ah
0477 E52C       +1  1089         mov   A,2Ch                  ;LSB PW from RAM
```

```
0479 B52807     +1  1090         cjne  A,28h,Key_PW_Bad_0Ah    ;Program jumps to bad PW state if PW
                +1  1091                                       ;entered was incorrect
                +1  1092
047C            +1  1093     Key_PW Ok_0Ah:
047C 740B       +1  1094         mov   A,#0Bh                  ;State 0Bh
047E 120D27     +1  1095         lcall State Lookup
0481 8007       +1  1096         jmp   Key_PW_Check_0Ah_Finish ;Program returns here once the state
                +1  1097                                       ;machine is finished
                +1  1098
0483            +1  1099     Key_PW Bad 0Ah:
0483 740C       +1  1100         mov   A,#0Ch                  ;State 0Ch
0485 120D27     +1  1101         lcall State Lookup
0488 8000       +1  1102         jmp   Key_PW_Check_0Ah_Finish ;Program returns here once the state
                +1  1103                                       ;machine is finished
                +1  1104
048A            +1  1105     Key_PW Check_0Ah_Finish:
048A 22         +1  1106         ret                           ;ret to ENT key state check
                +1  1107
                +1  1108     ;=====================================================================
                +1  1109     ;   Sub routine - State 02h Password check
                +1  1110     ;
                +1  1111     ;This routine determines if the entered password matches that of the
                +1  1112     ;one stored in RAM.
                +1  1113     ;
                +1  1114     ;Addresses:
                +1  1115     ; - 21h: Checks for state when Enter key ok to press
                +1  1116     ;
                +1  1117     ;Registers:
                +1  1118     ; - R1: Points to current state (21h)
                +1  1119     ;=====================================================================
                +1  1120
048B            +1  1121     Key_PW Check 02h:
048B E52F       +1  1122         mov   A,2Fh                   ;MSB PW from RAM
048D B52B16     +1  1123         cjne  A,2Bh,Key_PW_Bad_02h
0490 E52E       +1  1124         mov   A,2Eh                   ;
0492 B52A11     +1  1125         cjne  A,2Ah,Key_PW_Bad_02h
0495 E52D       +1  1126         mov   A,2Dh                   ;
0497 B5290C     +1  1127         cjne  A,29h,Key_PW_Bad_02h
049A E52C       +1  1128         mov   A,2Ch                   ;LSB PW from RAM
049C B52807     +1  1129         cjne  A,28h,Key_PW_Bad_02h    ;Program jumps to bad PW state if PW
                +1  1130                                       ;entered was incorrect
                +1  1131
049F            +1  1132     Key_PW Ok_02h:
049F 7406       +1  1133         mov   A,#06h                  ;State 06h
04A1 120D27     +1  1134         lcall State Lookup
04A4 8007       +1  1135         jmp   Key_PW_Check_02h_Finish ;Program returns here once the state
                +1  1136                                       ;machine is finished
                +1
04A6            +1    Key_PW Bad 02h:
04A6 7403       +1        mov   A,#03h                  ;State 03h
04A8 120D27     +1        lcall State Lookup
04AB 8000       +1        jmp   Key_PW_Check_02h_Finish ;Program returns here once the state
                +1                                      ;machine is finished
                +1
04AD            +1    Key_PW Check_02h_Finish:
04AD 22         +1        ret                           ;ret to ENT key state check
                +1
                +1
                +1    ;=================================================================
                +1    ;   Sub routine - State 03h Password check
                +1    ;
                +1    ;This routine determines if the entered password matches that of the
                +1    ;one stored in RAM.
                +1    ;
                +1    ;Addresses:
                +1    ; - 21h: Checks for state when Enter key ok to press
                +1  1156     ;
                +1  1157     ;Registers:
                +1  1158     ; - R1: Points to current state (21h)
                +1  1159     ;=====================================================================
                +1  1160
04AE            +1  1161     Key_PW_Check_03h:
04AE E52F       +1  1162         mov   A,2Fh                     ;MSB PW from RAM
04B0 B52B16     +1  1163         cjne  A,2Bh,Key_PW_Bad_03h
04B3 E52E       +1  1164         mov   A,2Eh                     ;
04B5 B52A11     +1  1165         cjne  A,2Ah,Key_PW_Bad_03h
04B8 E52D       +1  1166         mov   A,2Dh                     ;
04BA B5290C     +1  1167         cjne  A,29h,Key_PW_Bad_03h
04BD E52C       +1  1168         mov   A,2Ch                     ;LSB PW from RAM
04BF B52807     +1  1169         cjne  A,28h,Key_PW_Bad_03h      ;Program jumps to bad PW state if PW
                +1  1170                                         ;entered was incorrect
                +1  1171
04C2            +1  1172     Key_PW_Ok_03h:
04C2 7406       +1  1173         mov   A,#06h                    ;State 06h
04C4 120D27     +1  1174         lcall State_Lookup
04C7 8007       +1  1175         jmp   Key_PW_Check_03h_Finish ;Program returns here once the state
                +1  1176                                         ;machine is finished
                +1  1177
04C9            +1  1178     Key_PW_Bad_03h:
04C9 7404       +1  1179         mov   A,#04h                    ;State 04h
04CB 120D27     +1  1180         lcall State_Lookup
04CE 8000       +1  1181         jmp   Key_PW_Check_03h_Finish ;Program returns here once the state
                +1  1182                                         ;machine is finished
                +1  1183
04D0            +1  1184     Key_PW_Check_03h_Finish:
04D0 22         +1  1185         ret                             ;ret to ENT key state check
                +1  1186
                +1  1187     ;=====================================================================
                +1  1188     ;   Sub routine - State 04h Password check
                +1  1189     ;
                +1  1190     ;This routine determines if the entered password matches that of the
                +1  1191     ;one stored in RAM.
                +1  1192     ;
                +1  1193     ;Addresses:
                +1  1194     ; - 21h: Checks for state when Enter key ok to press
                +1  1195     ;
                +1  1196     ;Registers:
                +1  1197     ; - R1: Points to current state (21h)
                +1  1198     ;=====================================================================
                +1  1199
04D1            +1  1200     Key_PW_Check_04h:
04D1 E52F       +1  1201         mov   A,2Fh                     ;MSB PW from RAM
04D3 B52B16     +1  1202         cjne  A,2Bh,Key_PW_Bad_04h
```

```
04D6 E52E       +1  1203         mov   A,2Eh                      ;
04D8 B52A11     +1  1204         cjne  A,2Ah,Key_PW_Bad_04h
04DB E52D       +1  1205         mov   A,2Dh                      ;
04DD B5290C     +1  1206         cjne  A,29h,Key_PW_Bad_04h
04E0 E52C       +1  1207         mov   A,2Ch                      ;LSB PW from RAM
04E2 B52807     +1  1208         cjne  A,28h,Key_PW_Bad_04h       ;Program jumps to bad PW state if PW
                +1  1209                                          ;entered was incorrect
                +1  1210
04E5            +1  1211     Key_PW Ok_04h:
04E5 7406       +1  1212         mov   A,#06h                     ;State 06h
04E7 120D27     +1  1213         lcall State Lookup
04EA 8007       +1  1214         jmp   Key_PW_Check_04h_Finish ;Program returns here once the state
                +1  1215                                          ;machine is finished
                +1  1216
04EC            +1  1217     Key_PW Bad 04h:
04EC 7405       +1  1218         mov   A,#05h                     ;State 05h
04EE 120D27     +1  1219         lcall State Lookup
04F1 8000       +1  1220         jmp   Key_PW_Check_04h_Finish ;Program returns here once the state
                +1  1221                                          ;machine is finished
```

```
                 +1  1222
04F3             +1  1223     Key_PW Check_04h_Finish:
04F3 22          +1  1224         ret                            ;ret to ENT key state check
                 +1  1225
                 +1  1226
                 +1  1227     ;======================================================================
                 +1  1228     ;   Sub routine - Backspace Function Key valid state check
                 +1  1229     ;
                 +1  1230     ;This routine determines if the current state allows for
                 +1  1231     ;the backspace key to be pressed.
                 +1  1232     ;
                 +1  1233     ;Addresses:
                 +1  1234     ; - 21h: Checks for state when BS key ok to press
                 +1  1235     ;
                 +1  1236     ;Registers:
                 +1  1237     ; - R1: Points to current state (21h)
                 +1  1238     ;======================================================================
                 +1  1239
04F4             +1  1240     Key_Func BS:
                 +1  1241         ;State 02h is true? then continue else next
04F4 B70205      +1  1242         cjne    @R1,#02h,Key Func_BS_01
04F7 12054D      +1  1243         lcall   Key BS Resolve
04FA 8050        +1  1244         jmp     Key_Func_BS_Finish
                 +1  1245
04FC             +1  1246     Key_Func BS 01:
                 +1  1247         ;State 03h is true? then continue else next
04FC B70305      +1  1248         cjne    @R1,#03h,Key Func_BS_02
04FF 12054D      +1  1249         lcall   Key BS Resolve
0502 8048        +1  1250         jmp     Key_Func_BS_Finish
                 +1  1251
0504             +1  1252     Key_Func BS 02:
                 +1  1253         ;State 04h is true? then continue else next
0504 B70405      +1  1254         cjne    @R1,#04h,Key Func_BS_03
0507 12054D      +1  1255         lcall   Key BS Resolve
050A 8040        +1  1256         jmp     Key_Func_BS_Finish
                 +1  1257
050C             +1  1258     Key_Func BS 03:
                 +1  1259         ;State 0Ah is true? then continue else next
050C B70A05      +1  1260         cjne    @R1,#0Ah,Key Func_BS_04
050F 12054D      +1  1261         lcall   Key BS Resolve
0512 8038        +1  1262         jmp     Key_Func_BS_Finish
                 +1  1263
0514             +1  1264     Key_Func BS 04:
                 +1  1265         ;State 0Ch is true? then continue else next
0514 B70C05      +1  1266         cjne    @R1,#0Ch,Key Func_BS_05
0517 12054D      +1  1267         lcall   Key BS Resolve
051A 8030        +1  1268         jmp     Key_Func_BS_Finish
```

```
                +1  1269
051C            +1  1270     Key_Func BS 05:
                +1  1271         ;State 0Dh is true? then continue else next
051C B70D05     +1  1272         cjne    @R1,#0Dh,Key Func_BS_06
051F 12054D     +1  1273         lcall   Key BS Resolve
0522 8028       +1  1274         jmp     Key_Func_BS_Finish
                +1  1275
0524            +1  1276     Key_Func BS 06:
                +1  1277         ;State 11h is true? then continue else next
0524 B71105     +1  1278         cjne    @R1,#11h,Key Func_BS_07
0527 12054D     +1  1279         lcall   Key BS Resolve
052A 8020       +1  1280         jmp     Key_Func_BS_Finish
                +1  1281
052C            +1  1282     Key_Func BS 07:
                +1  1283         ;State 12h is true? then continue else next
052C B71205     +1  1284         cjne    @R1,#12h,Key Func_BS_08
052F 12054D     +1  1285         lcall   Key BS Resolve
0532 8018       +1  1286         jmp     Key_Func_BS_Finish
                +1  1287
```

```
0534            +1  1288     Key_Func BS 08:
                +1  1289         ;State 13h is true? then continue else next
0534 B71305     +1  1290         cjne    @R1,#13h,Key Func_BS_09
0537 12054D     +1  1291         lcall   Key BS Resolve
053A 8010       +1  1292         jmp     Key_Func_BS_Finish
                +1  1293
053C            +1  1294     Key_Func BS 09:
                +1  1295         ;State 14h is true? then continue else finish
053C B71405     +1  1296         cjne    @R1,#14h,Key Func_BS_10
053F 12054D     +1  1297         lcall   Key BS Resolve
0542 8008       +1  1298         jmp     Key_Func_BS_Finish
                +1  1299
0544            +1  1300     Key_Func BS 10:
                +1  1301         ;State 0Eh is true? then continue else finish
0544 B70E05     +1  1302         cjne    @R1,#0Eh,Key Func_BS_Finish
0547 12054D     +1  1303         lcall   Key BS Resolve
054A 8000       +1  1304         jmp     Key_Func_BS_Finish
                +1  1305
                +1  1306
054C            +1  1307     Key_Func_BS_Finish:
054C 22         +1  1308         ret
                +1  1309
                +1  1310     ;====================================================================
                +1  1311     ;   Sub routine - Backspace Resolve
                +1  1312     ;
                +1  1313     ;This routine determines if backspace key can delete a character.
                +1  1314     ;
                +1  1315     ;
                +1  1316     ;
                +1  1317     ;Registers:
                +1  1318     ; - R0: Points to MSB of password entered (2Bh)
                +1  1319     ; - R2: Determines BS/non Func key presses allowed
                +1  1320     ;====================================================================
054D            +1  1321     Key_BS Resolve:
054D C3         +1  1322         clr     C
054E BA0100     +1  1323         cjne    R2,#01h,$ + 3
0551 4022       +1  1324         jc      Key_BS_Resolve_Finish
                +1  1325
0553 901200     +1  1326         mov     DPTR,#LCD CMD
0556 75E010     +1  1327         mov     ACC,#DISP_BACKSPACE
0559 F0         +1  1328         movx    @DPTR,A
055A 12009E     +1  1329         lcall   LCD_Busy
                +1  1330
055D 901000     +1  1331         mov     DPTR,#LCD_WRITE
0560 75E020     +1  1332         mov     ACC,#20h          ;space character
0563 F0         +1  1333         movx    @DPTR,A
0564 12009E     +1  1334         lcall   LCD_Busy
```

```
                +1  1335
0567 901200     +1  1336         mov     DPTR,#LCD CMD
056A 75E010     +1  1337         mov     ACC,#DISP_BACKSPACE
056D F0         +1  1338         movx    @DPTR,A
056E 12009E     +1  1339         lcall   LCD_Busy
                +1  1340
0571 1A         +1  1341         dec     R2
0572 08         +1  1342         inc     R0
0573 8000       +1  1343         jmp     Key_BS_Resolve_Finish
                +1  1344
                +1  1345
0575            +1  1346     Key_BS Resolve_Finish:
0575 22         +1  1347         ret
                +1  1348
                +1  1349
                +1  1350     ;=====================================================================
                +1  1351     ;   Sub routine - Caps lock Function Key
                +1  1352     ;
                +1  1353     ;Addresses:
```

```
                +1  1354     ; - 07h: Bit Used to determine caps lock function key pressed
                +1  1355     ; - 20h: General address location for function keys: caps lock & colors
                +1  1356     ;
                +1  1357     ;Registers:
                +1  1358     ; - ACC enters as 00h
                +1  1359     ;=====================================================================
                +1  1360
0576            +1  1361     Key_Func_Caps:
0576 B207       +1  1362         cpl   07h
0578 B293       +1  1363         cpl   P1.3
057A 22         +1  1364         ret
                +1  1365
                +1  1366
                +1  1367     ;=====================================================================
                +1  1368     ;   Sub routine - Blue Function Key
                +1  1369     ;
                +1  1370     ;Addresses:
                +1  1371     ; - 00h: Bit Used to determine blue function key pressed
                +1  1372     ; - 20h: General address location for function keys: caps lock & colors
                +1  1373     ;=====================================================================
                +1  1374
057B            +1  1375     Key_Func_Blue:
057B 5320F0     +1  1376         anl     20h,#0F0h   ;Bit mask to erase lower nibble for other
                +1  1377                             ;function keys previously pressed
                +1  1378                             ;This also keeps the value of caps lock in tact.
                +1  1379
057E D200       +1  1380         setb    00h         ;Sets LSB in addy 20h for blue func. key
0580 C294       +1  1381         clr     P1.4        ;Turns off Red LED
0582 C295       +1  1382         clr     P1.5        ;Turns off Green LED
0584 C296       +1  1383         clr     P1.6        ;Turns off Pink LED
0586 D297       +1  1384         setb    P1.7        ;Turns on Blue LED
0588 22         +1  1385         ret
                +1  1386
                +1  1387
                +1  1388
                +1  1389     ;=====================================================================
                +1  1390     ;   Sub routine - Pink Function Key
                +1  1391     ;
                +1  1392     ;Addresses:
                +1  1393     ; - 01h: Bit Used to determine pink function key pressed
                +1  1394     ; - 20h: General address location for function keys: caps lock & colors
                +1  1395     ;=====================================================================
                +1  1396
0589            +1  1397     Key_Func_Pink:
0589 5320F0     +1  1398         anl     20h,#0F0h   ;Bit mask to erase lower nibble for other
                +1  1399                             ;function keys previously pressed
                +1  1400                             ;This also keeps the value of caps lock in tact.
                +1  1401
058C D201       +1  1402        setb    01h         ;Sets bit 1 in addy 20h for pink func. key
058E C294       +1  1403        clr     P1.4        ;Turns off Red LED
0590 C295       +1  1404        clr     P1.5        ;Turns off Green LED
0592 D296       +1  1405        setb    P1.6        ;Turns on Pink LED
0594 C297       +1  1406        clr     P1.7        ;Turns off Blue LED
0596 22         +1  1407        ret
                +1  1408
                +1  1409    ;=====================================================================
                +1  1410    ;   Sub routine - Green Function Key
                +1  1411    ;
                +1  1412    ;Addresses:
                +1  1413    ; - 02h: Bit Used to determine pink function key pressed
                +1  1414    ; - 20h: General address location for function keys: caps lock & colors
                +1  1415    ;=====================================================================
                +1  1416
0597            +1  1417    Key_Func_Green:
0597 5320F0     +1  1418        anl     20h,#0F0h   ;Bit mask to erase lower nibble for other
                +1  1419                            ;function keys previously pressed
```

```
                 +1  1420                             ;This also keeps the value of caps lock in tact.
                 +1  1421
059A D202        +1  1422        setb    02h          ;Sets bit 2 in addy 20h for green func. key
059C C294        +1  1423        clr     P1.4         ;Turns off Red LED
059E D295        +1  1424        setb    P1.5         ;Turns on Green LED
05A0 C296        +1  1425        clr     P1.6         ;Turns off Pink LED
05A2 C297        +1  1426        clr     P1.7         ;Turns off Blue LED
05A4 22          +1  1427        ret
                 +1  1428
                 +1  1429    ;=========================================================================
                 +1  1430    ;  Sub routine - Red Function Key
                 +1  1431    ;
                 +1  1432    ;Addresses:
                 +1  1433    ; - 03h: Bit Used to determine pink function key pressed
                 +1  1434    ; - 20h: General address location for function keys: caps lock & colors
                 +1  1435    ;=========================================================================
                 +1  1436
05A5             +1  1437    Key_Func_Red:
05A5 5320F0      +1  1438        anl     20h,#0F0h    ;Bit mask to erase lower nibble for other
                 +1  1439                             ;function keys previously pressed
                 +1  1440                             ;This also keeps the value of caps lock in tact.
                 +1  1441
05A8 D203        +1  1442        setb    03h          ;Sets bit 3 in addy 20h for red func. key
05AA D294        +1  1443        setb    P1.4         ;Turns on Red LED
05AC C295        +1  1444        clr     P1.5         ;Turns off Green LED
05AE C296        +1  1445        clr     P1.6         ;Turns off Pink LED
05B0 C297        +1  1446        clr     P1.7         ;Turns off Blue LED
05B2 22          +1  1447        ret
                 +1  1448
                 +1  1449    ;=========================================================================
                 +1  1450    ;   Sub routine - Key valid state check
                 +1  1451    ;
                 +1  1452    ;Addresses:
                 +1  1453    ; - 20h: Evaluates this addresed to determine funct. key & caps lock
                 +1  1454    ;status.  Leave this sub with correct lookup table in DPTR.  When this
                 +1  1455    ;sub is initially called, the ACC has the value of the key (according
                 +1  1456    ;to the actual value as determined by the keypad schematic) pressed.
                 +1  1457    ;
                 +1  1458    ;Registers:
                 +1  1459    ; - R0: Points to MSB of password entered (2Bh)
                 +1  1460    ; - R1: Points to current state (21h)
                 +1  1461    ; - R2: Determines BS/non_Func key presses allowed
                 +1  1462    ;
                 +1  1463    ;=========================================================================
                 +1  1464
05B3             +1  1465    Key_State_Chk:
                 +1  1466
```

```
                +1  1467          ;State 02h is true? then continue else finish
05B3 B70205     +1  1468          cjne    @R1,#02h,Key_State_Chk_01
05B6 12067F     +1  1469          lcall   Key Func PW
05B9 8060       +1  1470          jmp     Key_State_Chk_Finish
                +1  1471
05BB            +1  1472      Key_State Chk 01:
                +1  1473          ;State 03h is true? then continue else finish
05BB B70305     +1  1474          cjne    @R1,#03h,Key_State_Chk_02
05BE 12067F     +1  1475          lcall   Key Func PW
05C1 8058       +1  1476          jmp     Key_State_Chk_Finish
                +1  1477
05C3            +1  1478      Key_State Chk 02:
                +1  1479          ;State 04h is true? then continue else finish
05C3 B70405     +1  1480          cjne    @R1,#04h,Key_State_Chk_03
05C6 12067F     +1  1481          lcall   Key Func PW
05C9 8050       +1  1482          jmp     Key_State_Chk_Finish
                +1  1483
05CB            +1  1484      Key_State Chk 03:
                +1  1485          ;State 07h is true? then continue else finish
```

```
05CB B70705     +1  1486         cjne    @R1,#07h,Key State_Chk_04
05CE 120631     +1  1487         lcall   Key State07h Menu
05D1 8048       +1  1488         jmp     Key_State_Chk_Finish
                +1  1489
05D3            +1  1490     Key_State Chk 04:
                +1  1491         ;State 08h is true? then continue else finish
05D3 B70805     +1  1492         cjne    @R1,#08h,Key State_Chk_05
05D6 12061C     +1  1493         lcall   Key State08h Menu
05D9 8040       +1  1494         jmp     Key_State_Chk_Finish
                +1  1495
05DB            +1  1496     Key_State Chk 05:
                +1  1497         ;State 0Ah is true? then continue else finish
05DB B70A05     +1  1498         cjne    @R1,#0Ah,Key_State_Chk_06
05DE 12067F     +1  1499         lcall   Key Func PW
05E1 8038       +1  1500         jmp     Key_State_Chk_Finish
                +1  1501
05E3            +1  1502     Key_State Chk 06:
                +1  1503         ;State 0Ch is true? then continue else finish
05E3 B70C05     +1  1504         cjne    @R1,#0Ch,Key_State_Chk_07
05E6 12067F     +1  1505         lcall   Key Func PW
05E9 8030       +1  1506         jmp     Key_State_Chk_Finish
                +1  1507
05EB            +1  1508     Key_State Chk 07:
                +1  1509         ;State 0Dh is true? then continue else finish
05EB B70D05     +1  1510         cjne    @R1,#0Dh,Key_State_Chk_08
05EE 12067F     +1  1511         lcall   Key Func PW
05F1 8028       +1  1512         jmp     Key_State_Chk_Finish
                +1  1513
05F3            +1  1514     Key_State Chk 08:
                +1  1515         ;State 11h is true? then continue else finish
05F3 B71105     +1  1516         cjne    @R1,#11h,Key_State_Chk_09
05F6 12067F     +1  1517         lcall   Key Func PW
05F9 8020       +1  1518         jmp     Key_State_Chk_Finish
                +1  1519
05FB            +1  1520     Key_State Chk 09:
                +1  1521         ;State 12h is true? then continue else finish
05FB B71205     +1  1522         cjne    @R1,#12h,Key_State_Chk_10
05FE 12067F     +1  1523         lcall   Key Func PW
0601 8018       +1  1524         jmp     Key_State_Chk_Finish
                +1  1525
0603            +1  1526     Key_State Chk 10:
                +1  1527         ;State 13h is true? then continue else finish
0603 B71305     +1  1528         cjne    @R1,#13h,Key_State_Chk_11
0606 12067F     +1  1529         lcall   Key Func PW
0609 8010       +1  1530         jmp     Key_State_Chk_Finish
                +1  1531
060B            +1  1532     Key_State_Chk_11:
```

```
                +1          ;State 14h is true? then continue else finish
060B B71405     +1          cjne    @R1,#14h,Key_State_Chk_12
060E 12067F     +1          lcall   Key Func PW
0611 8008       +1          jmp     Key_State_Chk_Finish
                +1
0613            +1      Key_State Chk 12:
                +1          ;State 0Eh is true? then continue else finish
0613 B70E05     +1          cjne    @R1,#0Eh,Key State_Chk_Finish
0616 120650     +1          lcall   Key Func Accel
0619 8000       +1          jmp     Key_State_Chk_Finish
                +1
061B            +1      Key_State_Chk_Finish:
061B 22         +1          ret
                +1
                +1      ;====================================================================
                +1      ;   Sub routine - State_08h non-function key menu selection
                +1      ;
                +1      ;Registers:
                +1      ;
                +1  1552     ;=====================================================================
                +1  1553
061C            +1  1554     Key_State08h_Menu:
                +1  1555
                +1  1556         ;Option 1? then continue else finish
061C B40307     +1  1557         cjne    A,#03h,Key_State08h_Menu_01
061F 7409       +1  1558         mov     A,#09h                      ;State_09h
0621 120D27     +1  1559         lcall   State Lookup                ;Go to state
0624 800A       +1  1560         jmp     Key_State08h_Menu_Finish
                +1  1561
0626            +1  1562     Key_State08h Menu 01:
                +1  1563         ;Option 2? then continue else finish
0626 B40207     +1  1564         cjne    A,#02h,Key_State08h_Menu_Finish
0629 740A       +1  1565         mov     A,#0Ah                      ;State_0Ah
062B 120D27     +1  1566         lcall   State Lookup                ;Go to state
062E 8000       +1  1567         jmp     Key_State08h_Menu_Finish
                +1  1568
                +1  1569
0630            +1  1570     Key_State08h_Menu_Finish:
                +1  1571
0630 22         +1  1572         ret
                +1  1573
                +1  1574     ;=====================================================================
                +1  1575     ;   Sub routine - State_07h non-function key menu selection
                +1  1576     ;
                +1  1577     ;Registers:
                +1  1578     ;
                +1  1579     ;=====================================================================
                +1  1580
0631            +1  1581     Key_State07h_Menu:
                +1  1582
                +1  1583         ;Option 1? then continue else finish
0631 B40307     +1  1584         cjne    A,#03h,Key_State07h_Menu_01
0634 7408       +1  1585         mov     A,#08h                      ;State_08h
0636 120D27     +1  1586         lcall   State Lookup                ;Go to state
0639 8014       +1  1587         jmp     Key_State07h_Menu_Finish
                +1  1588
063B            +1  1589     Key_State07h Menu 01:
                +1  1590         ;Option 2? then continue else finish
063B B40207     +1  1591         cjne    A,#02h,Key_State07h_Menu_02
063E 740E       +1  1592         mov     A,#0Eh                      ;State_0Eh
0640 120D27     +1  1593         lcall   State Lookup                ;Go to state
0643 800A       +1  1594         jmp     Key_State07h_Menu_Finish
                +1  1595
0645            +1  1596     Key_State07h Menu 02:
                +1  1597         ;Option 3? then continue else finish
0645 B40107     +1  1598         cjne    A,#01h,Key_State07h_Menu_Finish
```

```
0648 7411       +1               mov     A,#11h                      ;State_11h
064A 120D27     +1               lcall   State Lookup                ;Go to state
064D 8000       +1               jmp     Key_State07h_Menu_Finish
                +1
                +1
064F            +1           Key_State07h_Menu_Finish:
                +1
064F 22         +1               ret
                +1
                +1
                +1           ;=====================================================================
                +1           ;   Sub routine - Key input resolution for Acceleration Setpoint
                +1           ;
                +1           ;This sub is to be used for entering in the new acceleration
                +1           ;setpoint into State_0Eh.  It will also write the ascii to
                +1           ;scratch pad RAM for setpoint analysis.
                +1           ;
                +1           ;Addresses:
                +1           ; - 20h: Evaluates this address to determine funct. key & caps lock
                +1  1618     ;status.  Leave this sub with correct lookup table in DPTR.  When this
                +1  1619     ;sub is initially called, the ACC has the value of the key (according
                +1  1620     ;to the actual value as determined by the keypad schematic) pressed.
                +1  1621     ;
                +1  1622     ; - 27h,26h,& 25h: Location of stored Acceleration STPT
                +1  1623     ;
                +1  1624     ;Registers:
                +1  1625     ; - R0: Points to MSB of Accel STPT (27h)
                +1  1626     ; - R1: Points to current state (21h)
                +1  1627     ; - R2: Determines BS/non_Func key presses allowed
                +1  1628     ;
                +1  1629     ;==========================================================================
                +1  1630
0650            +1  1631     Key_Func_Accel:
0650 C0D0       +1  1632         push    PSW
0652 C0E0       +1  1633         push    ACC
0654 C0F0       +1  1634         push    B
                +1  1635
0656 C3         +1  1636         clr     C
0657 BA0300     +1  1637         cjne    R2,#03h,$ + 3        ;jmp to end if 3 char's entered
065A 501C       +1  1638         jnc     Key_Func_Accel_Restore     ;
                +1  1639
                +1  1640
                +1  1641         ;Check Bit 00h for Blue function key, Numbers only
065C 20000B     +1  1642         jb      00h,Key Func Accel Bluekey
                +1  1643         ;Check Bit 01h for Pink function key
065F 200102     +1  1644         jb      01h,Key_Func_Accel_Pinkkey
                +1  1645
0662 8014       +1  1646         jmp     Key_Func_Accel_Restore
                +1  1647
                +1  1648
                +1  1649
0664            +1  1650     Key_Func_Accel_Pinkkey:
0664 0A         +1  1651         inc     R2                   ;Prog reaches this point if R2<3
0665 9006EF     +1  1652         mov     DPTR,#Key Pink LC   ;Pink lowercase lookup table
0668 8004       +1  1653         jmp     Key_Func_Accel_Finish
                +1  1654
066A            +1  1655     Key_Func_Accel_Bluekey:
066A 0A         +1  1656         inc     R2                   ;Prog reaches this point if R2<3
066B 9006E3     +1  1657         mov     DPTR,#Key_Blue_Num
                +1  1658
066E            +1  1659     Key_Func_Accel Finish:
066E 93         +1  1660         movc    A,@A + DPTR         ;Updates ACC w/ corresponding
                +1  1661                                     ;char. in lookup table
066F F6         +1  1662         mov     @R0,A               ;ascii in ACC to R0 pointer
0670 18         +1  1663         dec     R0                  ;Next PW location
                +1  1664
```

```
0671            +1  1665     Key_Func_Accel_Char:
                +1  1666
0671 901000     +1  1667         mov     DPTR,#LCD_WRITE    ;Writes the actual character to LCD
0674 F0         +1  1668         movx    @DPTR,A            ;
0675 12009E     +1  1669         lcall   LCD_Busy
                +1  1670
0678            +1  1671     Key_Func_Accel_Restore:
0678 D0F0       +1  1672         pop     B
067A D0E0       +1  1673         pop     ACC
067C D0D0       +1  1674         pop     PSW
                +1  1675
067E 22         +1  1676         ret
                +1  1677
                +1  1678
                +1  1679
                +1  1680     ;======================================================================
                +1  1681     ;   Sub routine - Key input resolution for Password states
                +1  1682     ;
                +1  1683     ;This sub is to be used for states that require password entry before
                +1  1684     ;the next state can be achieved.  This sub will write the password
                +1  1685     ;entered into on chip RAM and allow for a maximum of a 4 char.
                +1  1686     ;password.
                +1  1687     ;
                +1  1688     ;Addresses:
                +1  1689     ; - 20h: Evaluates this address to determine funct. key & caps lock
                +1  1690     ;status.  Leave this sub with correct lookup table in DPTR.  When this
                +1  1691     ;sub is initially called, the ACC has the value of the key (according
                +1  1692     ;to the actual value as determined by the keypad schematic) pressed.
                +1  1693     ;
                +1  1694     ;Registers:
                +1  1695     ; - R0: Points to MSB of password entered (2Bh)
                +1  1696     ; - R1: Points to current state (21h)
                +1  1697     ; - R2: Determines BS/non_Func key presses allowed
                +1  1698     ;
                +1  1699     ;=====================================================================
                +1  1700
067F            +1  1701     Key_Func_PW:
067F C0D0       +1  1702         push    PSW
0681 C0E0       +1  1703         push    ACC
0683 C0F0       +1  1704         push    B
                +1  1705
0685 C3         +1  1706         clr     C
0686 BA0400     +1  1707         cjne    R2,#04h,$ + 3         ;jmp to end if 4 char's entered
0689 5051       +1  1708         jnc     Key_Func_Restore      ;
068B 0A         +1  1709         inc     R2                    ;Prog reaches this point if R2<4
                +1  1710
                +1  1711
                +1  1712         ;Check Bit 00h for Blue function key, Numbers only
068C 200032     +1  1713         jb      00h,Key Func Bluekey
                +1  1714         ;Check Bit 01h for Pink function key
068F 200122     +1  1715         jb      01h,Key Func Pinkkey
                +1  1716         ;Check Bit 02h for Green function key
0692 200212     +1  1717         jb      02h,Key Func Greenkey
                +1  1718         ;Check Bit 03h for Red function key
0695 200302     +1  1719         jb      03h,Key_Func_Redkey
                +1  1720
0698 802A       +1  1721         jmp     Key_Func_Finish
                +1  1722
069A            +1  1723     Key_Func_Redkey:
069A 300705     +1  1724         jnb     07h,$ + 8            ;07h=caps lock, jump to
                +1  1725                                      ;'Red LC' if not caps lock
069D 90072B     +1  1726         mov     DPTR,#Key Red UpC    ;Red uppercase lookup table
06A0 8022       +1  1727         sjmp    Key_Func_Finish
                +1  1728
06A2 90071F     +1  1729         mov     DPTR,#Key Red LC     ;Red lowercase lookup table
06A5 801D       +1  1730         jmp     Key_Func_Finish
```

```
                +1  1731
06A7            +1  1732    Key_Func_Greenkey:
06A7 300705     +1  1733        jnb     07h,$ + 8           ;07h=caps lock, jump to
                +1  1734                                    ;'Green LC' if not caps lock
06AA 900713     +1  1735        mov     DPTR,#Key Green_UpC ;Green uppercase lookup table
06AD 8015       +1  1736        sjmp    Key_Func_Finish
                +1  1737
06AF 900707     +1  1738        mov     DPTR,#Key Green_LC  ;Green lowercase lookup table
06B2 8010       +1  1739        jmp     Key_Func_Finish
                +1  1740
                +1  1741
06B4            +1  1742    Key_Func_Pinkkey:
06B4 300705     +1  1743        jnb     07h,$ + 8           ;07h=caps lock, jump to
                +1  1744                                    ;'Pink LC' if not caps lock
06B7 9006FB     +1  1745        mov     DPTR,#Key Pink UpC  ;Pink uppercase lookup table
06BA 8008       +1  1746        sjmp    Key_Func_Finish
                +1  1747
06BC 9006EF     +1  1748        mov     DPTR,#Key Pink LC   ;Pink lowercase lookup table
06BF 8003       +1  1749        jmp     Key_Func_Finish
```

```
                +1  1750
06C1            +1  1751     Key_Func_Bluekey:
06C1 9006E3     +1  1752         mov     DPTR,#Key_Blue_Num
                +1  1753
                +1  1754
06C4            +1  1755     Key_Func_Finish:
06C4 93         +1  1756         movc    A,@A + DPTR         ;Updates ACC w/ corresponding
                +1  1757                                     ;char. in lookup table
06C5 F6         +1  1758         mov     @R0,A               ;ascii in ACC to R0 pointer
06C6 18         +1  1759         dec     R0                  ;Next PW location
                +1  1760
                +1  1761
06C7            +1  1762     Key_Func_PW_StateCk:
06C7 B71409     +1  1763         cjne    @R1,#14h,Key_Func_Star  ;Checks state to either print '*'
                +1  1764                                         ;or the char. during a PW change
                +1  1765                                         ;state
                +1  1766
06CA            +1  1767     Key_Func_Char:
                +1  1768
06CA 901000     +1  1769         mov     DPTR,#LCD_WRITE    ;Writes the actual character instead
06CD F0         +1  1770         movx    @DPTR,A            ;of '*' when PW change state=true
06CE 12009E     +1  1771         lcall   LCD Busy
06D1 8009       +1  1772         jmp     Key_Func_Restore
                +1  1773
06D3            +1  1774     Key_Func_Star:
06D3 901000     +1  1775         mov     DPTR,#LCD_WRITE     ;write '*' on each char. entry
06D6 742A       +1  1776         mov     A,#'*'              ;ascii value for '*'
06D8 F0         +1  1777         movx    @DPTR,A
06D9 12009E     +1  1778         lcall   LCD_Busy
                +1  1779
06DC            +1  1780     Key_Func_Restore:
06DC D0F0       +1  1781         pop     B
06DE D0E0       +1  1782         pop     ACC
06E0 D0D0       +1  1783         pop     PSW
                +1  1784
06E2 22         +1  1785         ret
                +1  1786
                +1  1787
                +1  1788     ;-------------------------------------
                +1  1789     ;-  Keypad Lookup Tables  --
                +1  1790     ;-------------------------------------
                +1  1791
06E3            +1  1792     Key_Blue Num:
06E3 00333231   +1  1793         db 0,"3","2","1"
06E7 00363534   +1  1794         db 0,"6","5","4"
06EB 00393837   +1  1795         db 0,"9","8","7"
                +1  1796
```

```
06EF              +1  1797     Key_Pink LC:        ;Lower Case
06EF 00757473     +1  1798         db 0,"u","t","s"
06F3 00787776     +1  1799         db 0,"x","w","v"
06F7 00307A79     +1  1800         db 0,"0","z","y"
                  +1  1801
06FB              +1  1802     Key_Pink UpC:       ;Upper Case
06FB 00555453     +1  1803         db 0,"U","T","S"
06FF 00585756     +1  1804         db 0,"X","W","V"
0703 00305A59     +1  1805         db 0,"0","Z","Y"
                  +1  1806
0707              +1  1807     Key_Green LC:       ;Lower Case
0707 006C6B6A     +1  1808         db 0,"l","k","j"
070B 006F6E6D     +1  1809         db 0,"o","n","m"
070F 00727170     +1  1810         db 0,"r","q","p"
                  +1  1811
0713              +1  1812     Key_Green UpC:      ;Upper Case
0713 004C4B4A     +1  1813         db 0,"L","K","J"
0717 004F4E4D     +1  1814         db 0,"O","N","M"
071B 00525150     +1  1815         db 0,"R","Q","P"
```

```
                 +1  1816
071F             +1  1817     Key_Red LC:          ;Lower Case
071F 00636261    +1  1818         db 0,"c","b","a"
0723 00666564    +1  1819         db 0,"f","e","d"
0727 00696867    +1  1820         db 0,"i","h","g"
                 +1  1821
072B             +1  1822     Key_Red UpC:         ;Upper Case
072B 00434241    +1  1823         db 0,"C","B","A"
072F 00464544    +1  1824         db 0,"F","E","D"
0733 00494847    +1  1825         db 0,"I","H","G"
                 +1  1826
                     1827     ;$include (RAM.asm) ;RAM routines
                 +1  1828     ;======================================================================
                 +1  1829     ;                              Pro-Tex 9000
                 +1  1830     ;
                 +1  1831     ;Revision: R.07171500  (R.MMDDHHMM)
                 +1  1832     ;
                 +1  1833     ;Project Team Members:
                 +1  1834     ; - Vince Watkins
                 +1  1835     ; - Will Smith
                 +1  1836     ; - Tyler Long
                 +1  1837     ;
                 +1  1838     ;=RAM Subroutines=
                 +1  1839     ;
                 +1  1840     ;'RAM Read' & 'RAM Write' subroutines will be called from the Main.asm.
                 +1  1841     ;The DPTR shall have the proper location to either read from
                 +1  1842     ;or write to.
                 +1  1843     ;
                 +1  1844     ;
                 +1  1845     ;Registers Used:
                 +1  1846     ; - R2: Contains value to write to RAM or value read from RAM upon
                 +1  1847     ;       exit of subroutine.
                 +1  1848     ; - ACC: Used to transfer from/to RAM
                 +1  1849     ;======================================================================
                 +1  1850
                 +1  1851
                 +1  1852     ;======================================================================
                 +1  1853     ;   Variable declarations
                 +1  1854     ;======================================================================
                 +1  1855
                 +1  1856     ;RAM Commands
  2000           +1  1857     RAM_RdWr              EQU 2000h     ;RAM read/write cmd addr. for DPTR
                 +1  1858
                 +1  1859
                 +1  1860
                 +1  1861
                 +1  1862     ;======================================================================
                +1     ;   Sub routine - Read from RAM
                +1     ;
                +1     ; - R0: Points to LSB of password from RAM (2Ch)
                +1     ;============================================================================
                +1
0737            +1     RAM_Read_PW:
0737 782C       +1         mov   R0,#2Ch          ;Pointer to LSB in scratch pad RAM
0739 902000     +1         mov   DPTR,#RAM_RdWr  ;LSB of PW in RAM
                +1
073C E0         +1         movx  A,@DPTR          ;Save LSB of PW from RAM=>ACC
073D F6         +1         mov   @R0,A            ;Save LSB of PW to scratch pad RAM '2Ch'
073E A3         +1         inc   DPTR             ;next PW character in RAM
073F 08         +1         inc   R0               ;next scratch pad RAM location
                +1                                ;next character
                +1
0740 E0         +1         movx  A,@DPTR          ;Save next PW char from RAM=>ACC
0741 F6         +1         mov   @R0,A            ;Save next PW char to scratch pad RAM '2Dh'
0742 A3         +1         inc   DPTR             ;next PW character in RAM
0743 08         +1         inc   R0               ;next scratch pad RAM location
```

```
                +1  1882                                ;next character
                +1  1883
0744 E0         +1  1884          movx  A,@DPTR         ;Save next PW char from RAM=>ACC
0745 F6         +1  1885          mov   @R0,A           ;Save next PW char to scratch pad RAM '2Eh'
0746 A3         +1  1886          inc   DPTR            ;next PW character in RAM
0747 08         +1  1887          inc   R0              ;next scratch pad RAM location
                +1  1888                                ;next character
                +1  1889
0748 E0         +1  1890          movx  A,@DPTR         ;Save MSB of PW from RAM=>ACC
0749 F6         +1  1891          mov   @R0,A           ;Save MSB of PW to scratch pad RAM '2Fh'
                +1  1892                                ;Last char of PW saved
                +1  1893
                +1  1894
074A 22         +1  1895          ret
                +1  1896
                +1  1897      ;=====================================================================
                +1  1898      ;   Sub routine - Write to RAM
                +1  1899      ;
                +1  1900      ; - R0: Points to LSB of password entered into scratch RAM (28h)
                +1  1901      ;=====================================================================
                +1  1902
074B            +1  1903      RAM_Write_PW:
074B 7828       +1  1904          mov   R0,#28h         ;Pointer to LSB in scratch pad RAM
074D 902000     +1  1905          mov   DPTR,#RAM_RdWr  ;LSB of PW in RAM
                +1  1906
0750 E6         +1  1907          mov   A,@R0           ;Save LSB of PW to ACC
0751 F0         +1  1908          movx  @DPTR,A         ;mov ACC=>RAM
0752 A3         +1  1909          inc   DPTR            ;next PW character in RAM
0753 08         +1  1910          inc   R0              ;next scratch pad RAM location
                +1  1911                                ;next character
                +1  1912
0754 E6         +1  1913          mov   A,@R0           ;Save next PW char to ACC
0755 F0         +1  1914          movx  @DPTR,A         ;mov ACC=>RAM
0756 A3         +1  1915          inc   DPTR            ;next PW character in RAM
0757 08         +1  1916          inc   R0              ;next scratch pad RAM location
                +1  1917                                ;next character
                +1  1918
0758 E6         +1  1919          mov   A,@R0           ;Save next PW char to ACC
0759 F0         +1  1920          movx  @DPTR,A         ;mov ACC=>RAM
075A A3         +1  1921          inc   DPTR            ;next PW character in RAM
075B 08         +1  1922          inc   R0              ;next scratch pad RAM location
                +1  1923                                ;next character
                +1  1924
075C E6         +1  1925          mov   A,@R0           ;Save MSB PW char to ACC
075D F0         +1  1926          movx  @DPTR,A         ;mov ACC=>RAM
                +1  1927
                +1  1928
```

```
075E 22         +1        ret
                +1
                +1
                +1
                +1       ;======================================================================
                +1       ;   Sub routine - Write to ADC string to RAM
                +1       ;
                +1       ; - R0: Points to LSB of password entered into scratch RAM (28h)
                +1       ;======================================================================
                +1
                +1
075F            +1       RAM_Write_ADC:
075F C083       +1           push  DPH                   ;save DPTR for table
0761 C082       +1           push  DPL                   ;save DPTR for table
0763 752404     +1           mov   24h, #04h             ;Initial DPL value for ext RAM
                +1
0766            +1       RAM_Write_Loop:
0766 7400       +1                       mov             A,#00h
0768 93         +1           movc  A,@A + DPTR           ;ascii char from table
```

```
0769 6016       +1  1948        jz    RAM_Write_ADC_Return
                +1  1949
076B C083       +1  1950        push  DPH                     ;save DPTR for table
076D C082       +1  1951        push  DPL                     ;save DPTR for table
                +1  1952
076F 758320     +1  1953        mov   DPH,#20h                ;Masked 'RAM RdWr' address to help
0772 852482     +1  1954        mov   DPL,24h                 ;w/ loading new DPTR addy for RAM
                +1  1955
0775 F0         +1  1956        movx  @DPTR,A
0776 A3         +1  1957        inc   DPTR                    ;DPTR inc for RAM location
0777 858224     +1  1958        mov   24h,DPL                 ;Save low byte of DPTR to scratch
                +1  1959
077A D082       +1  1960        pop   DPL                     ;Restore table DPTR
077C D083       +1  1961        pop   DPH                     ;
                +1  1962
077E A3         +1  1963        inc   DPTR                    ;Next character in table
077F 80E5       +1  1964        sjmp  RAM_Write_Loop          ;rinse and repeat
                +1  1965
0781            +1  1966    RAM_Write_ADC_Return:
0781 D082       +1  1967        pop   DPL                     ;This will restore the DPTR for the
0783 D083       +1  1968        pop   DPH                     ;initial charcter in table for printing
                +1  1969                                      ;as long as current State=06h
                +1  1970
                +1  1971
0785 22         +1  1972        ret
                +1  1973
                +1  1974
                +1  1975
                +1  1976
                +1  1977
                +1  1978
                +1  1979
                +1  1980
                +1  1981    ;======================================================================
                +1  1982    ;   Sub routine - Initialize RAM w/Default password
                +1  1983    ;======================================================================
                +1  1984
0786            +1  1985    RAM_Init:
0786 902000     +1  1986        mov   DPTR,#RAM_RdWr  ;
                +1  1987
0789 7434       +1  1988        mov   A,#'4'          ;Fourth PW Char
078B F0         +1  1989        movx  @DPTR,A         ;
                +1  1990
078C A3         +1  1991        inc   DPTR            ;
078D 7433       +1  1992        mov   A,#'3'          ;Third PW Char
078F F0         +1  1993        movx  @DPTR,A         ;
                +1  1994
```

```
0790 A3         +1  1995         inc   DPTR            ;
0791 7432       +1  1996         mov   A,#'2'          ;Second PW Char
0793 F0         +1  1997         movx  @DPTR,A         ;
                +1  1998
0794 A3         +1  1999         inc   DPTR            ;
0795 7431       +1  2000         mov   A,#'1'          ;First PW Char
0797 F0         +1  2001         movx  @DPTR,A         ;
                +1  2002
0798 90200A     +1  2003         mov   DPTR,#200Ah     ;Sets location of null character string
079B 740D       +1  2004         mov   A,#0Dh          ;Carriage Return Char
079D F0         +1  2005         movx  @DPTR,A
079E A3         +1  2006                     inc     DPTR
                +1  2007
079F 740A       +1  2008                     mov   A,#0Ah          ;Line Feed Char
07A1 F0         +1  2009         movx  @DPTR,A
                +1  2010
07A2 A3         +1  2011                     inc     DPTR
                +1  2012
07A3 7400       +1  2013                     mov   A,#00h          ;Null Character
```

```
07A5 F0          +1  2014         movx  @DPTR,A
                 +1  2015
07A6 22          +1  2016         ret
                 +1  2017
                     2018     ;$include (Init.asm) ;Initialization routines
                 +1  2019     ;======================================================================
                 +1  2020     ;                              Pro-Tex 9000
                 +1  2021     ;
                 +1  2022     ;Revision: R.07171500  (R.MMDDHHMM)
                 +1  2023     ;
                 +1  2024     ;Project Team Members:
                 +1  2025     ; - Vince Watkins
                 +1  2026     ; - Will Smith
                 +1  2027     ; - Tyler Long
                 +1  2028     ;
                 +1  2029     ;
                 +1  2030     ;Initialization Routines
                 +1  2031     ;
                 +1  2032     ;
                 +1  2033     ;======================================================================
                 +1  2034
                 +1  2035
                 +1  2036     ;-------------------------------------
                 +1  2037     ;-  Generated Initialization File  --
                 +1  2038     ;-------------------------------------
                 +1  2039
                 +1  2040
                 +1  2041     ; Peripheral specific initialization functions,
                 +1  2042     ; Called from the Init_Device label
07A7             +1  2043     Reset Sources Init:
07A7 75FFDE      +1  2044         mov   WDTCN,#0DEh
07AA 75FFAD      +1  2045         mov   WDTCN,#0ADh
07AD 22          +1  2046         ret
                 +1  2047
07AE             +1  2048     Timer Init:
07AE 758E40      +1  2049         mov  CKCON,#040h
07B1 758911      +1  2050         mov  TMOD,#011h
07B4 75C934      +1  2051         mov  T4CON,#034h
07B7 75E4FC      +1  2052         mov  RCAP4L,#0FCh
07BA 75E5FF      +1  2053         mov  RCAP4H,#0FFh
07BD 22          +1  2054         ret
                 +1  2055
07BE             +1  2056     UART Init:
07BE 758710      +1  2057         mov  PCON,#010h
07C1 75F140      +1  2058         mov  SCON1,#040h
07C4 22          +1  2059         ret
                 +1  2060
```

```
07C5            +1  2061     EMI_Init:
07C5 75A32F     +1  2062         mov  EMI0CF,#02Fh
07C8 75A1EF     +1  2063         mov  EMI0TC,#0EFh  ;/WR & /RD = 12 SYSCLK cycles
07CB 22         +1  2064         ret
                +1  2065
                +1  2066
07CC            +1  2067     Port IO Init:
                +1  2068         ; P0.0  -  TX1 (UART1), Push-Pull,  Digital
                +1  2069         ; P0.1  -  RX1 (UART1), Push-Pull,  Digital
                +1  2070         ; P0.2  -  INT0 (Tmr0), Push-Pull,  Digital, Keypad Int.
                +1  2071         ; P0.3  -  INT1 (Tmr1), Push-Pull,  Digital, Alm Int.
                +1  2072         ; P0.4  -  Unassigned,  Push-Pull,  Digital
                +1  2073         ; P0.5  -  Unassigned,  Push-Pull,  Digital
                +1  2074         ; P0.6  -  Unassigned,  Push-Pull,  Digital
                +1  2075         ; P0.7  -  Unassigned,  Push-Pull,  Digital
                +1  2076
                +1  2077         ; P1.0  -  Unassigned,  Open-Drain, Digital
                +1  2078         ; P1.1  -  Unassigned,  Open-Drain, Digital
                +1  2079         ; P1.2  -  Unassigned,  Open-Drain, Digital
                +1  2080         ; P1.3  -  Unassigned,  Open-Drain, Digital
                +1  2081         ; P1.4  -  Unassigned,  Open-Drain, Digital
                +1  2082         ; P1.5  -  Unassigned,  Open-Drain, Digital
                +1  2083         ; P1.6  -  Unassigned,  Open-Drain, Digital
                +1  2084         ; P1.7  -  Unassigned,  Open-Drain, Digital
                +1  2085
                +1  2086         ; P2.0  -  Unassigned,  Push-Pull,  Digital
                +1  2087         ; P2.1  -  Unassigned,  Push-Pull,  Digital
                +1  2088         ; P2.2  -  Unassigned,  Push-Pull,  Digital
                +1  2089         ; P2.3  -  Unassigned,  Push-Pull,  Digital
                +1  2090         ; P2.4  -  Unassigned,  Push-Pull,  Digital
                +1  2091         ; P2.5  -  Unassigned,  Push-Pull,  Digital
                +1  2092         ; P2.6  -  Unassigned,  Push-Pull,  Digital
                +1  2093         ; P2.7  -  Unassigned,  Push-Pull,  Digital
                +1  2094
                +1  2095         ; P3.0  -  Unassigned,  Push-Pull,  Digital
                +1  2096         ; P3.1  -  Unassigned,  Push-Pull,  Digital
                +1  2097         ; P3.2  -  Unassigned,  Push-Pull,  Digital
                +1  2098         ; P3.3  -  Unassigned,  Push-Pull,  Digital
                +1  2099         ; P3.4  -  Unassigned,  Push-Pull,  Digital
                +1  2100         ; P3.5  -  Unassigned,  Push-Pull,  Digital
                +1  2101         ; P3.6  -  Unassigned,  Push-Pull,  Digital
                +1  2102         ; P3.7  -  Unassigned,  Push-Pull,  Digital
                +1  2103
07CC 75A4FF     +1  2104         mov  P0MDOUT,   #0FFh
07CF 75A6FF     +1  2105         mov  P2MDOUT,   #0FFh
07D2 75A7FF     +1  2106         mov  P3MDOUT,   #0FFh
07D5 75B5FF     +1  2107         mov  P74OUT,    #0FFh
07D8 75E214     +1  2108         mov  XBR1,      #014h
07DB 75E344     +1  2109         mov  XBR2,      #044h
07DE 22         +1  2110         ret
                +1  2111
                +1  2112
07DF            +1  2113     Oscillator Init:
07DF 75B167     +1  2114         mov   OSCXCN,#067h
07E2 781E       +1  2115         mov   R0,#030       ; Wait 1ms for initialization
07E4            +1  2116     Osc_Wait1:
07E4 E4         +1  2117         clr   A
07E5 D5E0FD     +1  2118         djnz  ACC,$
07E8 D8FA       +1  2119         djnz  R0,Osc_Wait1
07EA            +1  2120     Osc_Wait2:
07EA E5B1       +1  2121         mov   A,OSCXCN
07EC 30E7FB     +1  2122         jnb   ACC.7,Osc Wait2
07EF 75B208     +1  2123         mov   OSCICN,#008h
07F2 22         +1  2124         ret
                +1  2125
07F3            +1  2126     Interrupts_Init:
```

```
07F3 75A880     +1  2127         mov  IE,#080h
07F6 75B808     +1  2128         mov  IP,#008h
07F9 22         +1  2129         ret
                +1  2130
                +1  2131
                +1  2132     ; Initialization function for device,
                +1  2133     ; Call Init Device from your main program
07FA            +1  2134     Init Device:
07FA 1207A7     +1  2135         lcall Reset Sources_Init
07FD 1207AE     +1  2136         lcall Timer Init
0800 1207BE     +1  2137         lcall UART Init
0803 1207C5     +1  2138         lcall EMI Init
0806 1207CC     +1  2139         lcall Port IO Init
0809 1207DF     +1  2140         lcall Oscillator Init
080C 1207F3     +1  2141         lcall Interrupts_Init
080F 22         +1  2142         ret
                +1  2143
                +1  2144
                +1  2145
```

```
                        2146     ;$include (State.asm) ;State Machine routines
                   +1   2147     ;======================================================================
                   +1   2148     ;                              Pro-Tex 9000
                   +1   2149     ;
                   +1   2150     ;Revision: R.07171500  (R.MMDDHHMM)
                   +1   2151     ;
                   +1   2152     ;Project Team Members:
                   +1   2153     ; - Vince Watkins
                   +1   2154     ; - Will Smith
                   +1   2155     ; - Tyler Long
                   +1   2156     ;
                   +1   2157     ;=State Machine Routines=
                   +1   2158     ;
                   +1   2159     ;
                   +1   2160     ;
                   +1   2161     ;
                   +1   2162     ;Registers Used:
                   +1   2163     ; -
                   +1   2164     ;======================================================================
                   +1   2165
                   +1   2166
                   +1   2167     ;======================================================================
                   +1   2168     ;   Variable declarations
                   +1   2169     ;======================================================================
                   +1   2170
  00B0             +1   2171     A 7447    bit    P3.0  ;These two bits control the 7-seg to show
  00B1             +1   2172     B_7447    bit    P3.1  ;how many PW attempts are left. 3=>0
                   +1   2173
                   +1   2174
                   +1   2175
                   +1   2176     ;======================================================================
                   +1   2177     ;   Table for State machine
                   +1   2178     ;======================================================================
                   +1   2179
0810               +1   2180     State Table:
0810 083C          +1   2181         dw    State 00    ;pointer to State 00
0812 08A6          +1   2182         dw    State 01    ;pointer to State 01
0814 08C3          +1   2183         dw    State 02    ;pointer to State 02
0816 08F2          +1   2184         dw    State 03    ;pointer to State 03
0818 0926          +1   2185         dw    State 04    ;pointer to State 04
081A 095A          +1   2186         dw    State 05    ;pointer to State 05
081C 097E          +1   2187         dw    State 06    ;pointer to State 06
081E 0A23          +1   2188         dw    State 07    ;pointer to State 07
0820 0A7F          +1   2189         dw    State 08    ;pointer to State 08
0822 0AC3          +1   2190         dw    State 09    ;pointer to State 09
0824 0AEB          +1   2191         dw    State 0A    ;pointer to State 0A
0826 0B0F          +1   2192         dw    State_0B    ;pointer to State 0B
```

```
0828 0B48       +1        dw    State 0C    ;pointer to State 0C
082A 0B7A       +1        dw    State 0D    ;pointer to State 0D
082C 0BAC       +1        dw    State 0E    ;pointer to State 0E
082E 0BFE       +1        dw    State 0F    ;pointer to State 0F
0830 0C2B       +1        dw    State 10    ;pointer to State 10
0832 0C58       +1        dw    State 11    ;pointer to State 11
0834 0C7C       +1        dw    State 12    ;pointer to State 12
0836 0CAE       +1        dw    State 13    ;pointer to State 13
0838 0CE0       +1        dw    State 14    ;pointer to State 14
083A 0CFA       +1        dw    State_15    ;pointer to State 15
                +1
                +1    ;======================================================================
                +1    ;   State_00
                +1    ;
                +1    ;This state shows the 'Initialization' screen.  Keypad /INT0 is
                +1    ;disabled in this state.
                +1    ;
                +1    ;
                +1    ;======================================================================
                +1  2212
083C            +1  2213     State 00:                        ;Initialization screen
083C 752100     +1  2214         mov     21h,#00h             ;Current State
083F C219       +1  2215         clr     19h                  ;Enables Acc Alarm
0841 C2B1       +1  2216         clr     B 7447               ;No password prompts
0843 C2B0       +1  2217         clr     A_7447               ;here
                +1  2218
                +1  2219
                +1  2220
0845 9000A6     +1  2221         mov     DPTR,#LCD_First      ;State 00 Screen pointer
0848 75E000     +1  2222         mov     ACC,#00h             ;Points to first char. in string
084B 120052     +1  2223         lcall   LCD_Print            ;Display State_06 Screen; 1st line
                +1  2224
                +1  2225         ;State_00 Screen - 2nd line
                +1  2226
084E C083       +1  2227         push    DPH
0850 C082       +1  2228         push    DPL                  ;Saves DPTR for next line in screen
                +1  2229
0852 901200     +1  2230         mov     DPTR,#LCD CMD        ;Locates Cursor
0855 75E0C0     +1  2231         mov     ACC,#80h + 40h
0858 F0         +1  2232         movx    @DPTR,A
0859 12009E     +1  2233         lcall   LCD_Busy
                +1  2234
085C D082       +1  2235         pop     DPL
085E D083       +1  2236         pop     DPH                  ;Restore next line for screen
                +1  2237
0860 75E000     +1  2238         mov     ACC,#00h             ;offset for char in string
                +1  2239                                      ;00h=>1st char in string
0863 120052     +1  2240         lcall   LCD_Print            ;Display State_00 Screen; 2nd line
                +1  2241
                +1  2242         ;State_00 Screen - 3rd line
                +1  2243
0866 C083       +1  2244         push    DPH
0868 C082       +1  2245         push    DPL                  ;Saves DPTR for next line in screen
                +1  2246
086A 901200     +1  2247         mov     DPTR,#LCD CMD        ;Locates Cursor
086D 75E094     +1  2248         mov     ACC,#80h + 14h
0870 F0         +1  2249         movx    @DPTR,A
0871 12009E     +1  2250         lcall   LCD_Busy
                +1  2251
0874 D082       +1  2252         pop     DPL
0876 D083       +1  2253         pop     DPH                  ;Restore next line for screen
                +1  2254
0878 75E000     +1  2255         mov     ACC,#00h             ;offset for char in string
                +1  2256                                      ;00h=>1st char in string
087B 120052     +1  2257         lcall   LCD_Print            ;Display State_00 Screen; 3rd line
                +1  2258
                +1        ;State_00 Screen - 4th line
                +1
087E C083       +1        push    DPH
0880 C082       +1        push    DPL                ;Saves DPTR for next line in screen
                +1
0882 901200     +1        mov     DPTR,#LCD_CMD      ;Locates Cursor
0885 75E0D4     +1        mov     ACC,#80h + 54h
0888 F0         +1        movx    @DPTR,A
0889 12009E     +1        lcall   LCD_Busy
                +1
088C D082       +1        pop     DPL
088E D083       +1        pop     DPH                ;Restore next line for screen
                +1
0890 75E000     +1        mov     ACC,#00h           ;offset for char in string
                +1                                   ;00h=>1st char in string
0893 120052     +1        lcall   LCD_Print          ;Display State_06 Screen; 4th line
                +1
0896 901200     +1        mov     DPTR,#LCD_CMD         ;Turns of cursor
0899 740C       +1        mov     A,#DISP_ON            ;and stops
```

```
089B F0          +1  2278         movx    @DPTR,A               ;blinking
089C 12009E      +1  2279         lcall   LCD_Busy              ;
                 +1  2280
                 +1  2281
089F 12007F      +1  2282         lcall   LCD Wait 3sec
08A2 120093      +1  2283         lcall   LCD_Clear
08A5 22          +1  2284         ret
                 +1  2285
                 +1  2286     ;=====================================================================
                 +1  2287     ;   State_01
                 +1  2288     ;
                 +1  2289     ;This state shows the 'Pro-Tex 9000' screen.  Keypad /INT0 is
                 +1  2290     ;disabled in this state.
                 +1  2291     ;
                 +1  2292     ;
                 +1  2293     ;=====================================================================
                 +1  2294
08A6             +1  2295     State 01:
08A6 752101      +1  2296         mov     21h,#01h             ;Current State
08A9 901200      +1  2297         mov     DPTR,#LCD CMD        ;Locates Cursor
08AC 75E084      +1  2298         mov     ACC,#80h + 04h
08AF F0          +1  2299         movx    @DPTR,A
08B0 12009E      +1  2300         lcall   LCD_Busy
                 +1  2301
08B3 9000DF      +1  2302         mov     DPTR,#LCD_Pro_Tex    ;2st Screen pointer
08B6 75E000      +1  2303         mov     ACC,#00h             ;Points to first char. in string
08B9 120052      +1  2304         lcall   LCD_Print            ;Display Screen
                 +1  2305
08BC 12007F      +1  2306         lcall   LCD Wait 3sec
08BF 120093      +1  2307         lcall   LCD_Clear
08C2 22          +1  2308         ret
                 +1  2309
                 +1  2310     ;=====================================================================
                 +1  2311     ;   State_02
                 +1  2312     ;
                 +1  2313     ;This state shows the 'Enter PW:' screen.  Keypad /INT0 is
                 +1  2314     ;enabled in this state.
                 +1  2315     ;
                 +1  2316     ;Registers:
                 +1  2317     ; - R0: Points to MSB of password entered (2Bh)
                 +1  2318     ; - R1: Points to current state (21h)
                 +1  2319     ; - R2: Determines BS/non Func key presses allowed
                 +1  2320     ;=====================================================================
                 +1  2321
08C3             +1  2322     State 02:
08C3 782B        +1  2323         mov     R0,#2Bh              ;Pointer for MSB of psswd
08C5 7921        +1  2324         mov     R1,#21h              ;Pointer for current state
```

```
08C7 7A00       +1  2325         mov     R2,#00h             ;BS/Non Func key presses
08C9 752102     +1  2326         mov     21h,#02h            ;Current State
                +1  2327
08CC D2B1       +1  2328         setb    B 7447              ;Three PW attempts left
08CE D2B0       +1  2329         setb    A_7447              ;
                +1  2330
                +1  2331         ;mov     DPTR,#LCD CMD        ;Locates Cursor
                +1  2332         ;mov     ACC,#80h + 00h
                +1  2333         ;movx    @DPTR,A
                +1  2334         ;lcall   LCD_Busy
                +1  2335
08D0 9000EC     +1  2336         mov     DPTR,#LCD_Password_Entry;Screen pointer
08D3 75E000     +1  2337         mov     ACC,#00h                ;Points to first char. in string
08D6 120052     +1  2338         lcall   LCD_Print               ;Display Screen
                +1  2339
08D9 752000     +1  2340         mov   20h,#00h          ;Address 20h used for determining
                +1  2341                                 ;Caps and function keys pressed
                +1  2342
08DC 12057B     +1  2343         lcall Key_Func_Blue     ;Starts with numbers as default, Blue LED
```

```



08DF 901200            mov   DPTR,#LCD CMD     ;LCD Command
08E2 75E00F            mov   ACC,#DISP_CURSOR  ;Shows & blinks cursor
08E5 F0                movx  @DPTR,A
08E6 12009E            lcall LCD_Busy

08E9 D2A8              setb  EX0                ;/INT0 Keypad interrupts enabled

08EB D2AA              setb  EX1                ;/INT1 Alarm interrupts enabled
                                                ;

08ED D2AB              setb  ET1                ;Timer 1 Interrupt enabled
08EF D28E              setb  TR1                ;Start Timer 1

08F1 22                ret

                   ;======================================================================
                   ;   State_03
                   ;
                   ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                   ;enabled in this state.
                   ;
                   ;Registers:
                   ; - R0: Points to MSB of password entered (2Bh)
                   ; - R1: Points to current state (21h)
                   ; - R2: Determines BS/non Func key presses allowed
                   ;======================================================================

08F2               State 03:
08F2 782B              mov     R0,#2Bh             ;Pointer for MSB of psswd
08F4 7921              mov     R1,#21h             ;Pointer for current state
08F6 7A00              mov     R2,#00h             ;BS/Non Func key presses
08F8 752103            mov     21h,#03h            ;Current State

08FB D2B1              setb    B 7447              ;Two PW attempts left
08FD C2B0              clr     A_7447              ;

08FF 120093            lcall   LCD_Clear

0902 9000F6            mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0905 75E000            mov     ACC,#00h            ;Points to first char. in string
0908 120052            lcall   LCD_Print           ;Display Screen

090B C083              push    DPH
090D C082              push    DPL               ;Saves DPTR for next line in screen

```

```
090F 901200     +1  2391         mov     DPTR,#LCD CMD     ;Locates Cursor
0912 75E0C0     +1  2392         mov     ACC,#80h + 40h
0915 F0         +1  2393         movx    @DPTR,A
0916 12009E     +1  2394         lcall   LCD_Busy
                +1  2395
0919 D082       +1  2396         pop     DPL
091B D083       +1  2397         pop     DPH               ;Restore next line for screen
                +1  2398
091D 75E000     +1  2399         mov     ACC,#00h          ;offset for char in string
                +1  2400                                   ;00h=>1st char in string
0920 120052     +1  2401         lcall   LCD_Print         ;Display Screen
                +1  2402
0923 D2A8       +1  2403         setb  EX0                ;/INT0 Keypad interrupts enabled
                +1  2404                                  ;on 6th screen
0925 22         +1  2405         ret
                +1  2406
                +1  2407     ;======================================================================
                +1  2408     ;   State_04
                +1  2409     ;
                +1  2410     ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                +1  2411     ;enabled in this state.
                +1  2412     ;
                +1  2413     ;Registers:
                +1  2414     ; - R0: Points to MSB of password entered (2Bh)
                +1  2415     ; - R1: Points to current state (21h)
                +1  2416     ; - R2: Determines BS/non Func key presses allowed
                +1  2417     ;=======================================================================
                +1  2418
0926            +1  2419     State 04:
0926 782B       +1  2420         mov     R0,#2Bh             ;Pointer for MSB of psswd
0928 7921       +1  2421         mov     R1,#21h             ;Pointer for current state
092A 7A00       +1  2422         mov     R2,#00h             ;BS/Non Func key presses
092C 752104     +1  2423         mov     21h,#04h            ;Current State
                +1  2424
092F C2B1       +1  2425         clr     B 7447              ;One PW attempt left
0931 D2B0       +1  2426         setb    A_7447              ;
                +1  2427
0933 120093     +1  2428         lcall   LCD_Clear
                +1  2429
0936 9000F6     +1  2430         mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0939 75E000     +1  2431         mov     ACC,#00h            ;Points to first char. in string
093C 120052     +1  2432         lcall   LCD_Print           ;Display Screen
                +1  2433
093F C083       +1  2434         push    DPH
0941 C082       +1  2435         push    DPL               ;Saves DPTR for next line in screen
                +1  2436
0943 901200     +1  2437         mov     DPTR,#LCD CMD     ;Locates Cursor
0946 75E0C0     +1  2438         mov     ACC,#80h + 40h
0949 F0         +1  2439         movx    @DPTR,A
094A 12009E     +1  2440         lcall   LCD_Busy
                +1  2441
094D D082       +1  2442         pop     DPL
094F D083       +1  2443         pop     DPH               ;Restore next line for screen
                +1  2444
0951 75E000     +1  2445         mov     ACC,#00h          ;offset for char in string
                +1  2446                                   ;00h=>1st char in string
0954 120052     +1  2447         lcall   LCD_Print         ;Display Screen
                +1  2448
0957 D2A8       +1  2449         setb  EX0                ;/INT0 Keypad interrupts enabled
                +1  2450                                  ;on 6th screen
0959 22         +1  2451         ret
                +1  2452
                +1  2453     ;=======================================================================
                +1  2454     ;   State_05
                +1  2455     ;
                +1  2456     ;This state shows the 'System Locked' screen.  Keypad /INT0 is
                +1   ;disabled in this state.
                +1   ;
                +1   ;Registers:
                +1   ; - R0: Points to MSB of password entered (2Bh)
                +1   ; - R1: Points to current state (21h)
                +1   ; - R2: Determines BS/non Func key presses allowed
                +1   ;=====================================================================
                +1
095A            +1   State 05:
095A C2AF       +1       clr     EA                     ;Disable all interrupts
095C 120093     +1       lcall   LCD_Clear
                +1
095F C2B1       +1       clr     B 7447                 ;No PW attempts left
0961 C2B0       +1       clr     A_7447                 ;
                +1
0963 901200     +1       mov     DPTR,#LCD CMD          ;Locates Cursor
0966 75E083     +1       mov     ACC,#80h + 03h
0969 F0         +1       movx    @DPTR,A
096A 12009E     +1       lcall   LCD_Busy
```

```
                +1  2476
096D 90010C     +1  2477        mov     DPTR,#LCD_SysLocked  ;Screen pointer
0970 75E000     +1  2478        mov     ACC,#00h             ;Points to first char. in string
0973 120052     +1  2479        lcall   LCD_Print            ;Display Screen
                +1  2480
0976 901200     +1  2481        mov     DPTR,#LCD CMD        ;Turns of cursor
0979 740C       +1  2482        mov     A,#DISP_ON           ;and stops
097B F0         +1  2483        movx    @DPTR,A              ;blinking
                +1  2484
                +1  2485
097C 80FE       +1  2486        sjmp    $                    ;lock up program because of too many
                +1  2487                                     ;password attempts
                +1  2488
                +1  2489    ;==========================================================================
                +1  2490    ;   State_06
                +1  2491    ;
                +1  2492    ;This state shows the 'Home' screen.  Keypad /INT0 is
                +1  2493    ;enabled in this state.
                +1  2494    ;
                +1  2495    ;Registers:
                +1  2496    ; - R1: Points to current state (21h)
                +1  2497    ;==========================================================================
                +1  2498
097E            +1  2499    State 06:
097E 7921       +1  2500        mov     R1,#21h             ;Pointer for current state
                +1  2501
                +1  2502
0980 C2B1       +1  2503        clr     B 7447               ;No password prompts
0982 C2B0       +1  2504        clr     A_7447               ;here
                +1  2505
0984 120093     +1  2506        lcall   LCD_Clear
                +1  2507
0987 90011B     +1  2508        mov     DPTR,#LCD_Home     ;State 06 Screen pointer
098A 75E000     +1  2509        mov     ACC,#00h           ;Points to first char. in string
098D 120052     +1  2510        lcall   LCD_Print          ;Display State_06 Screen; 1st line
                +1  2511
                +1  2512        ;State_06 Screen - 2nd line
                +1  2513
0990 C083       +1  2514        push    DPH
0992 C082       +1  2515        push    DPL                ;Saves DPTR for next line in screen
                +1  2516
0994 901200     +1  2517        mov     DPTR,#LCD CMD      ;Locates Cursor
0997 75E0C0     +1  2518        mov     ACC,#80h + 40h
099A F0         +1  2519        movx    @DPTR,A
099B 12009E     +1  2520        lcall   LCD_Busy
                +1  2521
099E D082       +1  2522        pop     DPL
```

```
09A0 D083        +1  2523         pop     DPH                 ;Restore next line for screen
                 +1  2524
09A2 75E000      +1  2525         mov     ACC,#00h            ;offset for char in string
                 +1  2526                                     ;00h=>1st char in string
09A5 120052      +1  2527         lcall   LCD_Print           ;Display State_06 Screen; 2nd line
                 +1  2528
                 +1  2529         ;State_06 Screen - 3rd line
                 +1  2530
09A8 C083        +1  2531         push    DPH
09AA C082        +1  2532         push    DPL                 ;Saves DPTR for next line in screen
                 +1  2533
09AC 901200      +1  2534         mov     DPTR,#LCD_CMD       ;Locates Cursor
09AF 75E094      +1  2535         mov     ACC,#80h + 14h
09B2 F0          +1  2536         movx    @DPTR,A
09B3 12009E      +1  2537         lcall   LCD_Busy
                 +1  2538
09B6 D082        +1  2539         pop     DPL
09B8 D083        +1  2540         pop     DPH                 ;Restore next line for screen
                 +1  2541
```

```
09BA 75E000      +1  2542         mov     ACC,#00h          ;offset for char in string
                 +1  2543                                   ;00h=>1st char in string
09BD 120052      +1  2544         lcall   LCD_Print         ;Display State_06 Screen; 3rd line
                 +1  2545
                 +1  2546         ;State_06 Screen - 4th line
                 +1  2547
09C0 C083        +1  2548         push    DPH
09C2 C082        +1  2549         push    DPL               ;Saves DPTR for next line in screen
                 +1  2550
09C4 901200      +1  2551         mov     DPTR,#LCD_CMD     ;Locates Cursor
09C7 75E0D4      +1  2552         mov     ACC,#80h + 54h
09CA F0          +1  2553         movx    @DPTR,A
09CB 12009E      +1  2554         lcall   LCD_Busy
                 +1  2555
09CE D082        +1  2556         pop     DPL
09D0 D083        +1  2557         pop     DPH               ;Restore next line for screen
                 +1  2558
09D2 75E000      +1  2559         mov     ACC,#00h          ;offset for char in string
                 +1  2560                                   ;00h=>1st char in string
09D5 120052      +1  2561         lcall   LCD_Print         ;Display State_06 Screen; 4th line
                 +1  2562
09D8 901200      +1  2563         mov     DPTR,#LCD_CMD        ;Turns of cursor
09DB 740C        +1  2564         mov     A,#DISP_ON           ;and stops
09DD F0          +1  2565         movx    @DPTR,A              ;blinking
09DE 12009E      +1  2566         lcall   LCD_Busy             ;
                 +1  2567
09E1 901200      +1  2568         mov     DPTR,#LCD_CMD     ;Locates Cursor for Accel STPT
09E4 75E0A0      +1  2569         mov     ACC,#80h + 20h
09E7 F0          +1  2570         movx    @DPTR,A
09E8 12009E      +1  2571         lcall   LCD_Busy
                 +1  2572
09EB 901000      +1  2573         mov     DPTR,#LCD_WRITE   ;Writes the default Acceleration STPT
09EE 8527E0      +1  2574         mov     ACC,27h           ;of 0.75g to LCD
09F1 F0          +1  2575         movx    @DPTR,A
09F2 12009E      +1  2576         lcall   LCD_BUSY
                 +1  2577
09F5 901000      +1  2578         mov     DPTR,#LCD_WRITE
09F8 75E02E      +1  2579         mov     ACC,#'.'
09FB F0          +1  2580         movx    @DPTR,A
09FC 12009E      +1  2581         lcall   LCD_BUSY
                 +1  2582
09FF 901000      +1  2583         mov     DPTR,#LCD_WRITE
0A02 8526E0      +1  2584         mov     ACC,26h
0A05 F0          +1  2585         movx    @DPTR,A
0A06 12009E      +1  2586         lcall   LCD_BUSY
                 +1  2587
0A09 901000      +1  2588         mov     DPTR,#LCD_WRITE
```

```
0A0C 8525E0     +1  2589        mov     ACC,25h
0A0F F0         +1  2590        movx    @DPTR,A
0A10 12009E     +1  2591        lcall   LCD_BUSY
                +1  2592
0A13 901000     +1  2593        mov     DPTR,#LCD_WRITE
0A16 75E067     +1  2594        mov     ACC,#'g'
0A19 F0         +1  2595        movx    @DPTR,A
0A1A 12009E     +1  2596        lcall   LCD_BUSY         ;Last character of Acceleration STPT
                +1  2597
0A1D 752106     +1  2598        mov     21h,#06h            ;Current State
                +1  2599
                +1  2600
0A20 D2A8       +1  2601        setb  EX0                ;/INT0 Keypad interrupts enabled
                +1  2602                                 ;on State_06 screen
                +1  2603
0A22 22         +1  2604        ret
                +1  2605
                +1  2606
                +1  2607    ;======================================================================
                +1  2608     ;   State_07
                +1  2609     ;
                +1  2610     ;This state shows the 'Menu' screen.  Keypad /INT0 is
                +1  2611     ;enabled in this state.
                +1  2612     ;
                +1  2613     ;Registers:
                +1  2614     ; - R1: Points to current state (21h)
                +1  2615     ;======================================================================
                +1  2616
0A23            +1  2617     State 07:
0A23 7921       +1  2618         mov     R1,#21h             ;Pointer for current state
0A25 752107     +1  2619         mov     21h,#07h           ;Current State
                +1  2620
0A28 120093     +1  2621         lcall   LCD_Clear
                +1  2622
0A2B 900153     +1  2623         mov     DPTR,#LCD_Main_Menu ;State 07 Screen pointer
0A2E 75E000     +1  2624         mov     ACC,#00h            ;Points to first char. in string
0A31 120052     +1  2625         lcall   LCD_Print           ;Display State_07 Screen; 1st line
                +1  2626
                +1  2627         ;State_07 Screen - 2nd line
                +1  2628
0A34 C083       +1  2629         push    DPH
0A36 C082       +1  2630         push    DPL               ;Saves DPTR for next line in screen
                +1  2631
0A38 901200     +1  2632         mov     DPTR,#LCD_CMD     ;Locates Cursor
0A3B 75E0C0     +1  2633         mov     ACC,#80h + 40h
0A3E F0         +1  2634         movx    @DPTR,A
0A3F 12009E     +1  2635         lcall   LCD_Busy
                +1  2636
0A42 D082       +1  2637         pop     DPL
0A44 D083       +1  2638         pop     DPH               ;Restore next line for screen
                +1  2639
0A46 75E000     +1  2640         mov     ACC,#00h          ;offset for char in string
                +1  2641                                   ;00h=>1st char in string
0A49 120052     +1  2642         lcall   LCD_Print         ;Display State_07 Screen; 2nd line
                +1  2643
                +1  2644         ;State_07 Screen - 3rd line
                +1  2645
0A4C C083       +1  2646         push    DPH
0A4E C082       +1  2647         push    DPL              ;Saves DPTR for next line in screen
                +1  2648
0A50 901200     +1  2649         mov     DPTR,#LCD_CMD    ;Locates Cursor
0A53 75E094     +1  2650         mov     ACC,#80h + 14h
0A56 F0         +1  2651         movx    @DPTR,A
0A57 12009E     +1  2652         lcall   LCD_Busy
                +1  2653
0A5A D082       +1  2654         pop     DPL
```

```
0A5C D083        +1  2655         pop     DPH               ;Restore next line for screen
                 +1  2656
0A5E 75E000      +1  2657         mov     ACC,#00h          ;offset for char in string
                 +1  2658                                   ;00h=>1st char in string
0A61 120052      +1  2659         lcall   LCD_Print         ;Display State_07 Screen; 3rd line
                 +1  2660
                 +1  2661         ;State_07 Screen - 4th line
                 +1  2662
0A64 C083        +1  2663         push    DPH
0A66 C082        +1  2664         push    DPL               ;Saves DPTR for next line in screen
                 +1  2665
0A68 901200      +1  2666         mov     DPTR,#LCD_CMD     ;Locates Cursor
0A6B 75E0D4      +1  2667         mov     ACC,#80h + 54h
0A6E F0          +1  2668         movx    @DPTR,A
0A6F 12009E      +1  2669         lcall   LCD_Busy
                 +1  2670
0A72 D082        +1  2671         pop     DPL
0A74 D083        +1  2672         pop     DPH               ;Restore next line for screen
                 +1  2673
```

```
0A76 75E000     +1  2674         mov     ACC,#00h        ;offset for char in string
                +1  2675                                 ;00h=>1st char in string
0A79 120052     +1  2676         lcall   LCD_Print       ;Display State_07 Screen; 4th line
                +1  2677
0A7C D2A8       +1  2678         setb  EX0                ;/INT0 Keypad interrupts enabled
                +1  2679                                  ;on State_07 screen
                +1  2680
0A7E 22         +1  2681         ret
                +1  2682
                +1  2683     ;=====================================================================
                +1  2684     ;   State_08
                +1  2685     ;
                +1  2686     ;This state shows the 'Arm/Disarm' screen.  Keypad /INT0 is
                +1  2687     ;enabled in this state.
                +1  2688     ;
                +1  2689     ;Registers:
                +1  2690     ; - R1: Points to current state (21h)
                +1  2691     ;=====================================================================
                +1  2692
0A7F            +1  2693     State 08:
0A7F 7921       +1  2694         mov     R1,#21h             ;Pointer for current state
0A81 752108     +1  2695         mov     21h,#08h            ;Current State
                +1  2696
0A84 120093     +1  2697         lcall   LCD_Clear
                +1  2698
0A87 90018F     +1  2699         mov     DPTR,#LCD_ArmDis    ;State 08 Screen pointer
0A8A 75E000     +1  2700         mov     ACC,#00h            ;Points to first char. in string
0A8D 120052     +1  2701         lcall   LCD_Print           ;Display State_08 Screen; 1st line
                +1  2702
                +1  2703         ;State_08 Screen - 2nd line
                +1  2704
0A90 C083       +1  2705         push    DPH
0A92 C082       +1  2706         push    DPL              ;Saves DPTR for next line in screen
                +1  2707
0A94 901200     +1  2708         mov     DPTR,#LCD CMD    ;Locates Cursor
0A97 75E0C0     +1  2709         mov     ACC,#80h + 40h
0A9A F0         +1  2710         movx    @DPTR,A
0A9B 12009E     +1  2711         lcall   LCD_Busy
                +1  2712
0A9E D082       +1  2713         pop     DPL
0AA0 D083       +1  2714         pop     DPH              ;Restore next line for screen
                +1  2715
0AA2 75E000     +1  2716         mov     ACC,#00h         ;offset for char in string
                +1  2717                                  ;00h=>1st char in string
0AA5 120052     +1  2718         lcall   LCD_Print        ;Display State_08 Screen; 2nd line
                +1  2719
                +1  2720         ;State_08 Screen - 3rd line
                +1  2721
0AA8 C083       +1  2722         push    DPH
0AAA C082       +1  2723         push    DPL                ;Saves DPTR for next line in screen
                +1  2724
0AAC 901200     +1  2725         mov     DPTR,#LCD CMD      ;Locates Cursor
0AAF 75E0D4     +1  2726         mov     ACC,#80h + 54h
0AB2 F0         +1  2727         movx    @DPTR,A
0AB3 12009E     +1  2728         lcall   LCD_Busy
                +1  2729
0AB6 D082       +1  2730         pop     DPL
0AB8 D083       +1  2731         pop     DPH                ;Restore next line for screen
                +1  2732
0ABA 75E000     +1  2733         mov     ACC,#00h           ;offset for char in string
                +1  2734                                    ;00h=>1st char in string
0ABD 120052     +1  2735         lcall   LCD_Print          ;Display State_08 Screen; 3rd line
                +1  2736
                +1  2737
0AC0 D2A8       +1  2738         setb  EX0                  ;/INT0 Keypad interrupts enabled
                +1  2739                                    ;on State_08 screen
                +1  2740
0AC2 22         +1  2741        ret
                +1  2742
                +1  2743    ;=============================================================
                +1  2744    ;   State_09
                +1  2745    ;
                +1  2746    ;This state shows the 'System Armed' screen.  Keypad /INT0 is
                +1  2747    ;enabled in this state.
                +1  2748    ;
                +1  2749    ;Registers:
                +1  2750    ; - R1: Points to current state (21h)
                +1  2751    ;=============================================================
                +1  2752
0AC3            +1  2753    State 09:
0AC3 7921       +1  2754        mov     R1,#21h             ;Pointer for current state
0AC5 752109     +1  2755        mov     21h,#09h            ;Current State
0AC8 D218       +1  2756        setb    18h                 ;Arming system
0ACA D291       +1  2757        setb    P1.1                ;Red LED indicator
                +1  2758
0ACC 120093     +1  2759        lcall   LCD_Clear
                +1  2760
0ACF 901200     +1  2761        mov     DPTR,#LCD CMD        ;Locates Cursor
0AD2 75E084     +1  2762        mov     ACC,#80h + 04h
0AD5 F0         +1  2763        movx    @DPTR,A
0AD6 12009E     +1  2764        lcall   LCD_Busy
                +1  2765
0AD9 9001B5     +1  2766        mov     DPTR,#LCD_SysArmed  ;State 09 Screen pointer
0ADC 75E000     +1  2767        mov     ACC,#00h            ;Points to first char. in string
0ADF 120052     +1  2768        lcall   LCD_Print           ;Display State_09 Screen; 1st line
                +1  2769
0AE2 12007F     +1  2770        lcall   LCD_Wait_3sec       ;Screen Delay
                +1  2771
0AE5 7406       +1  2772        mov     A,#06h              ;Load in State 06
0AE7 120D27     +1  2773        lcall   State_Lookup        ;Return to State_06
0AEA 22         +1  2774        ret
                +1  2775
                +1  2776
                +1  2777    ;=============================================================
                +1  2778    ;   State_0A
                +1  2779    ;
                +1  2780    ;This state shows the 'Enter PW:' screen.  Keypad /INT0 is
                +1  2781    ;enabled in this state.
                +1  2782    ;
                +1  2783    ;Registers:
                +1  2784    ; - R0: Points to MSB of password entered (2Bh)
                +1  2785    ; - R1: Points to current state (21h)
                +1  2786    ; - R2: Determines BS/non_Func key presses allowed
                +1  2787     ;======================================================================
                +1  2788
0AEB            +1  2789     State 0A:
0AEB 782B       +1  2790         mov     R0,#2Bh             ;Pointer for MSB of psswd
0AED 7921       +1  2791         mov     R1,#21h             ;Pointer for current state
0AEF 7A00       +1  2792         mov     R2,#00h             ;BS/Non Func key presses
0AF1 75210A     +1  2793         mov     21h,#0Ah            ;Current State
                +1  2794
0AF4 D2B1       +1  2795         setb    B 7447              ;Three PW attempts left
0AF6 D2B0       +1  2796         setb    A_7447              ;
                +1  2797
0AF8 120093     +1  2798         lcall   LCD_Clear
                +1  2799
0AFB 9000EC     +1  2800         mov     DPTR,#LCD_Password_Entry;Screen pointer
0AFE 75E000     +1  2801         mov     ACC,#00h                ;Points to first char. in string
0B01 120052     +1  2802         lcall   LCD_Print               ;Display Screen
                +1  2803
                +1  2804
0B04 901200     +1  2805         mov   DPTR,#LCD_CMD     ;LCD Command
```

```
0B07 75E00F     +1  2806         mov   ACC,#DISP_CURSOR  ;Shows & blinks cursor
0B0A F0         +1  2807         movx  @DPTR,A
0B0B 12009E     +1  2808         lcall LCD_Busy
                +1  2809
                +1  2810
0B0E 22         +1  2811         ret
                +1  2812
                +1  2813     ;=====================================================================
                +1  2814     ;   State_0B
                +1  2815     ;
                +1  2816     ;This state shows the 'System  Disarmed' screen.  Keypad /INT0 is
                +1  2817     ;enabled in this state.
                +1  2818     ;
                +1  2819     ;Registers:
                +1  2820     ; - R1: Points to current state (21h)
                +1  2821     ;=====================================================================
                +1  2822
0B0F            +1  2823     State 0B:
0B0F 7921       +1  2824         mov     R1,#21h             ;Pointer for current state
0B11 75210B     +1  2825         mov     21h,#0Bh            ;Current State
0B14 C219       +1  2826         clr     19h
0B16 C218       +1  2827         clr     18h                 ;Disarming system
0B18 C291       +1  2828         clr     P1.1                ;Red LED indicator
0B1A C290       +1  2829         clr     P1.0                ;Disables flashing Alarm LED
0B1C C292       +1  2830         clr     P1.2                ;Disables Tamper LED
                +1  2831
0B1E 120093     +1  2832         lcall   LCD_Clear
                +1  2833
0B21 901200     +1  2834         mov     DPTR,#LCD CMD        ;Turns of cursor
0B24 740C       +1  2835         mov     A,#DISP_ON           ;and stops
0B26 F0         +1  2836         movx    @DPTR,A              ;blinking
0B27 12009E     +1  2837         lcall   LCD_Busy
                +1  2838
0B2A 901200     +1  2839         mov     DPTR,#LCD CMD        ;Locates Cursor
0B2D 75E082     +1  2840         mov     ACC,#80h + 02h
0B30 F0         +1  2841         movx    @DPTR,A
0B31 12009E     +1  2842         lcall   LCD_Busy
                +1  2843
0B34 9001C2     +1  2844         mov     DPTR,#LCD_SysDisArmed  ;State 0B Screen pointer
0B37 75E000     +1  2845         mov     ACC,#00h            ;Points to first char. in string
0B3A 120052     +1  2846         lcall   LCD_Print           ;Display State_0B Screen; 1st line
                +1  2847
0B3D 12007F     +1  2848         lcall   LCD_Wait_3sec       ;Screen Delay
                +1  2849
0B40 7406       +1  2850         mov     A,#06h              ;Load in State 06
0B42 120D27     +1  2851         lcall   State_Lookup        ;Return to State_06
                +1  2852
```

```
0B45 D2AA       +1  2853        setb    EX1                 ;Re-enable /INT1 Alarms
                +1  2854
0B47 22         +1  2855        ret
                +1  2856
                +1  2857    ;=====================================================================
                +1  2858    ;   State_0C
                +1  2859    ;
                +1  2860    ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                +1  2861    ;enabled in this state.
                +1  2862    ;
                +1  2863    ;Registers:
                +1  2864    ; - R0: Points to MSB of password entered (2Bh)
                +1  2865    ; - R1: Points to current state (21h)
                +1  2866    ; - R2: Determines BS/non Func key presses allowed
                +1  2867    ;=====================================================================
                +1  2868
0B48            +1  2869    State 0C:
0B48 782B       +1  2870        mov     R0,#2Bh             ;Pointer for MSB of psswd
0B4A 7921       +1  2871        mov     R1,#21h             ;Pointer for current state
```

```
0B4C 7A00       +1  2872         mov     R2,#00h             ;BS/Non Func key presses
0B4E 75210C     +1  2873         mov     21h,#0Ch            ;Current State
                +1  2874
0B51 D2B1       +1  2875         setb    B 7447              ;Two PW attempts left
0B53 C2B0       +1  2876         clr     A_7447              ;
                +1  2877
0B55 120093     +1  2878         lcall   LCD_Clear
                +1  2879
0B58 9000F6     +1  2880         mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0B5B 75E000     +1  2881         mov     ACC,#00h            ;Points to first char. in string
0B5E 120052     +1  2882         lcall   LCD_Print           ;Display Screen
                +1  2883
0B61 C083       +1  2884         push    DPH
0B63 C082       +1  2885         push    DPL               ;Saves DPTR for next line in screen
                +1  2886
0B65 901200     +1  2887         mov     DPTR,#LCD CMD     ;Locates Cursor
0B68 75E0C0     +1  2888         mov     ACC,#80h + 40h
0B6B F0         +1  2889         movx    @DPTR,A
0B6C 12009E     +1  2890         lcall   LCD_Busy
                +1  2891
0B6F D082       +1  2892         pop     DPL
0B71 D083       +1  2893         pop     DPH               ;Restore next line for screen
                +1  2894
0B73 75E000     +1  2895         mov     ACC,#00h          ;offset for char in string
                +1  2896                                   ;00h=>1st char in string
0B76 120052     +1  2897         lcall   LCD_Print         ;Display Screen
                +1  2898
0B79 22         +1  2899         ret
                +1  2900
                +1  2901     ;=====================================================================
                +1  2902     ;   State_0D
                +1  2903     ;
                +1  2904     ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                +1  2905     ;enabled in this state.
                +1  2906     ;
                +1  2907     ;Registers:
                +1  2908     ; - R0: Points to MSB of password entered (2Bh)
                +1  2909     ; - R1: Points to current state (21h)
                +1  2910     ; - R2: Determines BS/non Func key presses allowed
                +1  2911     ;=====================================================================
                +1  2912
0B7A            +1  2913     State 0D:
0B7A 782B       +1  2914         mov     R0,#2Bh             ;Pointer for MSB of psswd
0B7C 7921       +1  2915         mov     R1,#21h             ;Pointer for current state
0B7E 7A00       +1  2916         mov     R2,#00h             ;BS/Non Func key presses
0B80 75210D     +1  2917         mov     21h,#0Dh            ;Current State
                +1  2918
```

```
0B83 C2B1       +1  2919         clr     B 7447              ;One PW attempt left
0B85 D2B0       +1  2920         setb    A_7447              ;
                +1  2921
0B87 120093     +1  2922         lcall   LCD_Clear
                +1  2923
0B8A 9000F6     +1  2924         mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0B8D 75E000     +1  2925         mov     ACC,#00h            ;Points to first char. in string
0B90 120052     +1  2926         lcall   LCD_Print           ;Display Screen
                +1  2927
0B93 C083       +1  2928         push    DPH
0B95 C082       +1  2929         push    DPL               ;Saves DPTR for next line in screen
                +1  2930
0B97 901200     +1  2931         mov     DPTR,#LCD CMD     ;Locates Cursor
0B9A 75E0C0     +1  2932         mov     ACC,#80h + 40h
0B9D F0         +1  2933         movx    @DPTR,A
0B9E 12009E     +1  2934         lcall   LCD_Busy
                +1  2935
0BA1 D082       +1  2936         pop     DPL
0BA3 D083       +1  2937         pop     DPH               ;Restore next line for screen
```

```
                +1  2938
0BA5 75E000     +1  2939         mov     ACC,#00h          ;offset for char in string
                +1  2940                                   ;00h=>1st char in string
0BA8 120052     +1  2941         lcall   LCD_Print         ;Display Screen
                +1  2942
0BAB 22         +1  2943         ret
                +1  2944
                +1  2945
                +1  2946
                +1  2947     ;=================================================================
                +1  2948     ;   State_0E
                +1  2949     ;
                +1  2950     ;This state shows the 'New Acc STPT:' screen.  Keypad /INT0 is
                +1  2951     ;enabled in this state.
                +1  2952     ;
                +1  2953     ;Registers:
                +1  2954     ; - R0: Points to MSB of Accel STPT (27h)
                +1  2955     ; - R1: Points to current state (21h)
                +1  2956     ; - R2: Determines BS/non Func key presses allowed
                +1  2957     ;=================================================================
                +1  2958
0BAC            +1  2959     State 0E:
0BAC 7827       +1  2960         mov     R0,#27h             ;Pointer for MSB of Accel STPT
0BAE 7921       +1  2961         mov     R1,#21h             ;Pointer for current state
0BB0 7A00       +1  2962         mov     R2,#00h             ;
0BB2 75210E     +1  2963         mov     21h,#0Eh            ;Current State
                +1  2964
0BB5 120093     +1  2965         lcall   LCD_Clear
                +1  2966
0BB8 9001D3     +1  2967         mov     DPTR,#LCD_AccStpt   ;State 0E Screen pointer
0BBB 75E000     +1  2968         mov     ACC,#00h            ;Points to first char. in string
0BBE 120052     +1  2969         lcall   LCD_Print           ;Display State_08 Screen; 1st line
                +1  2970
                +1  2971
                +1  2972         ;State_0E Screen - 2nd line
                +1  2973
0BC1 C083       +1  2974         push    DPH
0BC3 C082       +1  2975         push    DPL              ;Saves DPTR for next line in screen
                +1  2976
0BC5 901200     +1  2977         mov     DPTR,#LCD CMD    ;Locates Cursor
0BC8 75E0C0     +1  2978         mov     ACC,#80h + 40h
0BCB F0         +1  2979         movx    @DPTR,A
0BCC 12009E     +1  2980         lcall   LCD_Busy
                +1  2981
0BCF D082       +1  2982         pop     DPL
0BD1 D083       +1  2983         pop     DPH              ;Restore next line for screen
                +1  2984
```

```
0BD3 75E000     +1  2985          mov     ACC,#00h          ;offset for char in string
                +1  2986                                    ;00h=>1st char in string
0BD6 120052     +1  2987          lcall   LCD_Print         ;Display State_0E Screen; 2nd line
                +1  2988
                +1  2989          ;State_0E Screen - 3rd line
                +1  2990
0BD9 C083       +1  2991          push    DPH
0BDB C082       +1  2992          push    DPL               ;Saves DPTR for next line in screen
                +1  2993
0BDD 901200     +1  2994          mov     DPTR,#LCD_CMD     ;Locates Cursor
0BE0 75E09A     +1  2995          mov     ACC,#80h + 1Ah
0BE3 F0         +1  2996          movx    @DPTR,A
0BE4 12009E     +1  2997          lcall   LCD_Busy
                +1  2998
0BE7 D082       +1  2999          pop     DPL
0BE9 D083       +1  3000          pop     DPH               ;Restore next line for screen
                +1  3001
0BEB 75E000     +1  3002          mov     ACC,#00h          ;offset for char in string
                +1  3003                                    ;00h=>1st char in string
```

```
0BEE 120052     +1  3004         lcall   LCD_Print        ;Display State_0E Screen; 3rd line
                +1  3005
                +1  3006
0BF1 901200     +1  3007         mov   DPTR,#LCD_CMD     ;LCD Command
0BF4 75E00F     +1  3008         mov   ACC,#DISP_CURSOR  ;Shows & blinks cursor
0BF7 F0         +1  3009         movx  @DPTR,A
0BF8 12009E     +1  3010         lcall LCD_Busy
                +1  3011
                +1  3012
0BFB D2A8       +1  3013         setb  EX0                ;/INT0 Keypad interrupts enabled
                +1  3014                                  ;on State_0E screen
                +1  3015
0BFD 22         +1  3016         ret
                +1  3017
                +1  3018     ;==============================================================
                +1  3019     ;   State_0F
                +1  3020     ;
                +1  3021     ;This state shows the 'Valid Setpoint' screen.  Keypad /INT0 is
                +1  3022     ;enabled in this state.
                +1  3023     ;
                +1  3024     ;Registers:
                +1  3025     ; - R1: Points to current state (21h)
                +1  3026     ;==============================================================
                +1  3027
0BFE            +1  3028     State_0F:
0BFE 7921       +1  3029         mov     R1,#21h              ;Pointer for current state
0C00 75210F     +1  3030         mov     21h,#0Fh             ;Current State
                +1  3031
0C03 120093     +1  3032         lcall   LCD_Clear
                +1  3033
0C06 901200     +1  3034         mov     DPTR,#LCD_CMD         ;Turns of cursor
0C09 740C       +1  3035         mov     A,#DISP_ON            ;and stops
0C0B F0         +1  3036         movx    @DPTR,A               ;blinking
0C0C 12009E     +1  3037         lcall   LCD_Busy
                +1  3038
0C0F 901200     +1  3039         mov     DPTR,#LCD_CMD         ;Locates Cursor
0C12 75E082     +1  3040         mov     ACC,#80h + 02h
0C15 F0         +1  3041         movx    @DPTR,A
0C16 12009E     +1  3042         lcall   LCD_Busy
                +1  3043
0C19 9001FD     +1  3044         mov     DPTR,#LCD_Valid_STPT;State_0B Screen pointer
0C1C 75E000     +1  3045         mov     ACC,#00h             ;Points to first char. in string
0C1F 120052     +1  3046         lcall   LCD_Print            ;Display State_0B Screen; 1st line
                +1  3047
0C22 12007F     +1  3048         lcall   LCD_Wait_3sec        ;Screen Delay
                +1  3049
0C25 7406       +1  3050         mov     A,#06h               ;Load in State_06
```

```
0C27 120D27     +1  3051         lcall   State_Lookup        ;Return to State_06
                +1  3052
0C2A 22         +1  3053         ret
                +1  3054
                +1  3055     ;===========================================================================
                +1  3056     ;   State_10
                +1  3057     ;
                +1  3058     ;This state shows the 'Invalid Accel STPT' screen.  Keypad /INT0 is
                +1  3059     ;enabled in this state.
                +1  3060     ;
                +1  3061     ;Registers:
                +1  3062     ; - R1: Points to current state (21h)
                +1  3063     ;===========================================================================
                +1  3064
0C2B            +1  3065     State 10:
0C2B 7921       +1  3066         mov     R1,#21h             ;Pointer for current state
0C2D 752110     +1  3067         mov     21h,#10h            ;Current State
                +1  3068
0C30 120093     +1  3069         lcall   LCD_Clear
```

```
                +1  3070
0C33 901200     +1  3071         mov     DPTR,#LCD CMD        ;Turns of cursor
0C36 740C       +1  3072         mov     A,#DISP_ON           ;and stops
0C38 F0         +1  3073         movx    @DPTR,A              ;blinking
0C39 12009E     +1  3074         lcall   LCD_Busy
                +1  3075
0C3C 901200     +1  3076         mov     DPTR,#LCD CMD        ;Locates Cursor
0C3F 75E082     +1  3077         mov     ACC,#80h + 02h
0C42 F0         +1  3078         movx    @DPTR,A
0C43 12009E     +1  3079         lcall   LCD_Busy
                +1  3080
0C46 90020E     +1  3081         mov     DPTR,#LCD_Invalid_STPT;State 0B Screen pointer
0C49 75E000     +1  3082         mov     ACC,#00h              ;Points to first char. in string
0C4C 120052     +1  3083         lcall   LCD_Print             ;Display State_0B Screen; 1st line
                +1  3084
0C4F 12007F     +1  3085         lcall   LCD_Wait_3sec       ;Screen Delay
                +1  3086
0C52 740E       +1  3087         mov     A,#0Eh              ;Load in State 0E
0C54 120D27     +1  3088         lcall   State_Lookup        ;Return to State_0E
                +1  3089
0C57 22         +1  3090         ret
                +1  3091
                +1  3092     ;=====================================================================
                +1  3093     ;   State_11
                +1  3094     ;
                +1  3095     ;This state shows the 'Enter Curr PW:' screen.  Keypad /INT0 is
                +1  3096     ;enabled in this state.
                +1  3097     ;
                +1  3098     ;Registers:
                +1  3099     ; - R0: Points to MSB of password entered (2Bh)
                +1  3100     ; - R1: Points to current state (21h)
                +1  3101     ; - R2: Determines BS/non Func key presses allowed
                +1  3102     ;=====================================================================
                +1  3103
0C58            +1  3104     State 11:
0C58 782B       +1  3105         mov     R0,#2Bh             ;Pointer for MSB of psswd
0C5A 7921       +1  3106         mov     R1,#21h             ;Pointer for current state
0C5C 7A00       +1  3107         mov     R2,#00h             ;BS/Non Func key presses
0C5E 752111     +1  3108         mov     21h,#11h            ;Current State
                +1  3109
0C61 D2B1       +1  3110         setb    B 7447              ;Three PW attempts left
0C63 D2B0       +1  3111         setb    A_7447              ;
                +1  3112
0C65 120093     +1  3113         lcall   LCD_Clear
                +1  3114
0C68 90021F     +1  3115         mov     DPTR,#LCD_Current_PW    ;Screen pointer
0C6B 75E000     +1  3116         mov     ACC,#00h                ;Points to first char. in string
```

```
0C6E 120052     +1        lcall   LCD_Print               ;Display Screen
                +1
                +1
0C71 901200     +1        mov   DPTR,#LCD CMD     ;LCD Command
0C74 75E00F     +1        mov   ACC,#DISP_CURSOR  ;Shows & blinks cursor
0C77 F0         +1        movx  @DPTR,A
0C78 12009E     +1        lcall LCD_Busy
                +1
0C7B 22         +1        ret
                +1
                +1    ;=====================================================================
                +1    ;   State_12
                +1    ;
                +1    ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                +1    ;enabled in this state.
                +1    ;
                +1    ;Registers:
                +1    ; - R0: Points to MSB of password entered (2Bh)
                +1    ; - R1: Points to current state (21h)
                +1  3136     ; - R2: Determines BS/non Func key presses allowed
                +1  3137     ;======================================================================
                +1  3138
0C7C            +1  3139     State 12:
0C7C 782B       +1  3140         mov     R0,#2Bh             ;Pointer for MSB of psswd
0C7E 7921       +1  3141         mov     R1,#21h             ;Pointer for current state
0C80 7A00       +1  3142         mov     R2,#00h             ;BS/Non Func key presses
0C82 752112     +1  3143         mov     21h,#12h            ;Current State
                +1  3144
0C85 D2B1       +1  3145         setb    B 7447              ;Two PW attempts left
0C87 C2B0       +1  3146         clr     A_7447              ;
                +1  3147
0C89 120093     +1  3148         lcall   LCD_Clear
                +1  3149
0C8C 9000F6     +1  3150         mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0C8F 75E000     +1  3151         mov     ACC,#00h            ;Points to first char. in string
0C92 120052     +1  3152         lcall   LCD_Print           ;Display Screen
                +1  3153
0C95 C083       +1  3154         push    DPH
0C97 C082       +1  3155         push    DPL               ;Saves DPTR for next line in screen
                +1  3156
0C99 901200     +1  3157         mov     DPTR,#LCD CMD     ;Locates Cursor
0C9C 75E0C0     +1  3158         mov     ACC,#80h + 40h
0C9F F0         +1  3159         movx    @DPTR,A
0CA0 12009E     +1  3160         lcall   LCD_Busy
                +1  3161
0CA3 D082       +1  3162         pop     DPL
0CA5 D083       +1  3163         pop     DPH               ;Restore next line for screen
                +1  3164
0CA7 75E000     +1  3165         mov     ACC,#00h          ;offset for char in string
                +1  3166                                   ;00h=>1st char in string
0CAA 120052     +1  3167         lcall   LCD_Print         ;Display Screen
                +1  3168
0CAD 22         +1  3169         ret
                +1  3170
                +1  3171     ;======================================================================
                +1  3172     ;   State_13
                +1  3173     ;
                +1  3174     ;This state shows the 'Invalid PW' screen.  Keypad /INT0 is
                +1  3175     ;enabled in this state.
                +1  3176     ;
                +1  3177     ;Registers:
                +1  3178     ; - R0: Points to MSB of password entered (2Bh)
                +1  3179     ; - R1: Points to current state (21h)
                +1  3180     ; - R2: Determines BS/non Func key presses allowed
                +1  3181     ;======================================================================
                +1  3182
```

```
0CAE            +1  3183     State 13:
0CAE 782B       +1  3184         mov     R0,#2Bh             ;Pointer for MSB of psswd
0CB0 7921       +1  3185         mov     R1,#21h             ;Pointer for current state
0CB2 7A00       +1  3186         mov     R2,#00h             ;BS/Non Func key presses
0CB4 752113     +1  3187         mov     21h,#13h            ;Current State
                +1  3188
0CB7 C2B1       +1  3189         clr     B 7447              ;One PW attempt left
0CB9 D2B0       +1  3190         setb    A_7447              ;
                +1  3191
0CBB 120093     +1  3192         lcall   LCD_Clear
                +1  3193
0CBE 9000F6     +1  3194         mov     DPTR,#LCD_PW_Bad    ;Screen pointer
0CC1 75E000     +1  3195         mov     ACC,#00h            ;Points to first char. in string
0CC4 120052     +1  3196         lcall   LCD_Print           ;Display Screen
                +1  3197
0CC7 C083       +1  3198         push    DPH
0CC9 C082       +1  3199         push    DPL               ;Saves DPTR for next line in screen
                +1  3200
0CCB 901200     +1  3201         mov     DPTR,#LCD_CMD     ;Locates Cursor
```

```
0CCE 75E0C0     +1  3202        mov     ACC,#80h + 40h
0CD1 F0         +1  3203        movx    @DPTR,A
0CD2 12009E     +1  3204        lcall   LCD_Busy
                +1  3205
0CD5 D082       +1  3206        pop     DPL
0CD7 D083       +1  3207        pop     DPH                 ;Restore next line for screen
                +1  3208
0CD9 75E000     +1  3209        mov     ACC,#00h            ;offset for char in string
                +1  3210                                    ;00h=>1st char in string
0CDC 120052     +1  3211        lcall   LCD_Print           ;Display Screen
                +1  3212
0CDF 22         +1  3213        ret
                +1  3214
                +1  3215    ;==========================================================================
                +1  3216    ;   State_14
                +1  3217    ;
                +1  3218    ;This state shows the 'Enter New PW:' screen.  Keypad /INT0 is
                +1  3219    ;enabled in this state.
                +1  3220    ;
                +1  3221    ;Registers:
                +1  3222    ; - R1: Points to current state (21h)
                +1  3223    ;==========================================================================
                +1  3224
0CE0            +1  3225    State 14:
0CE0 782B       +1  3226        mov     R0,#2Bh               ;Pointer for MSB of psswd
0CE2 7921       +1  3227        mov     R1,#21h               ;Pointer for current state
0CE4 7A00       +1  3228        mov     R2,#00h               ;BS/Non Func key presses
0CE6 752114     +1  3229        mov     21h,#14h              ;Current State
                +1  3230
0CE9 C2B1       +1  3231        clr     B 7447                 ;No password prompts
0CEB C2B0       +1  3232        clr     A_7447                 ;here
                +1  3233
0CED 120093     +1  3234        lcall   LCD_Clear
                +1  3235
0CF0 90022E     +1  3236        mov     DPTR,#LCD_New_PW        ;Screen pointer
0CF3 75E000     +1  3237        mov     ACC,#00h                ;Points to first char. in string
0CF6 120052     +1  3238        lcall   LCD_Print               ;Display Screen
                +1  3239
                +1  3240
0CF9 22         +1  3241        ret
                +1  3242
                +1  3243
                +1  3244    ;==========================================================================
                +1  3245    ;   State_15
                +1  3246    ;
                +1  3247    ;This state shows the 'Password Changed' screen.  Keypad /INT0 is
                +1  3248    ;enabled in this state.
                +1  3249     ;
                +1  3250     ;Registers:
                +1  3251     ; - R1: Points to current state (21h)
                +1  3252     ;======================================================================
                +1  3253
0CFA            +1  3254     State 15:
0CFA 7921       +1  3255         mov     R1,#21h              ;Pointer for current state
0CFC 752115     +1  3256         mov     21h,#15h             ;Current State
                +1  3257
0CFF 120093     +1  3258         lcall   LCD_Clear
                +1  3259
0D02 901200     +1  3260         mov     DPTR,#LCD CMD         ;Turns of cursor
0D05 740C       +1  3261         mov     A,#DISP_ON            ;and stops
0D07 F0         +1  3262         movx    @DPTR,A               ;blinking
0D08 12009E     +1  3263         lcall   LCD_Busy
                +1  3264
0D0B 901200     +1  3265         mov     DPTR,#LCD CMD         ;Locates Cursor
0D0E 75E082     +1  3266         mov     ACC,#80h + 02h
0D11 F0         +1  3267         movx    @DPTR,A
```

```
0D12 12009E     +1          lcall   LCD_Busy
                +1
0D15 90023C     +1          mov     DPTR,#LCD_Changed_PW  ;State 09 Screen pointer
0D18 75E000     +1          mov     ACC,#00h              ;Points to first char. in string
0D1B 120052     +1          lcall   LCD_Print             ;Display State_09 Screen; 1st line
                +1
0D1E 12007F     +1          lcall   LCD_Wait_3sec       ;Screen Delay
                +1
0D21 7406       +1          mov     A,#06h              ;Load in State 06
0D23 120D27     +1          lcall   State_Lookup        ;Return to State_06
                +1
0D26 22         +1          ret
                +1
                +1
                +1      ;===================================================================
                +1      ;   State Table Lookup
                +1      ;
                +1      ;
                +1      ;Call this routine with the ACC=> 0-n of state needed and the DPTR
                +1      ;pointing to the tag of the table of states.
                +1      ;
                +1      ;
                +1      ;===================================================================
                +1
0D27            +1      State Lookup:
0D27 900810     +1          mov   DPTR,#State_Table
0D2A C3         +1          clr   C
0D2B 33         +1          rlc   A              ;Multiply * 2 for word access
0D2C F8         +1          mov   R0, A          ;Save a copy of index
0D2D 04         +1          inc   A              ;Increment index to the low byte
0D2E 93         +1          movc  A, @A+DPTR     ;Get Low byte
0D2F C0E0       +1          push  ACC            ;Save low byte onto stack
0D31 E8         +1          mov   A, R0          ;Restore high byte
0D32 93         +1          movc  A, @A+DPTR     ;Get high byte
0D33 C0E0       +1          push  ACC            ;save high byte onto stack
0D35 22         +1          ret                  ;Direct branch to the subroutine
                        ;$include (ADC.asm) ;ADC routines
                +1      ;===================================================================
                +1      ;                             Pro-Tex 9000
                +1      ;
                +1      ;Revision: R.07171500  (R.MMDDHHMM)
                +1      ;
                +1      ;Project Team Members:
                +1      ; - Vince Watkins
                +1      ; - Will Smith
                +1      ; - Tyler Long
                +1      ;
```

```
                        +1  3315     ;=ADC Subroutines=
                        +1  3316     ;
                        +1  3317     ;'ADC Read' & 'ADC Kick'subroutines will be called from the Main.asm.
                        +1  3318     ;The DPTR shall have the proper location to either read from
                        +1  3319     ;or kick start the ADC.  Call 'ADC_Init' routine to kick start the
                        +1  3320     ;ADC.
                        +1  3321     ;
                        +1  3322     ;Addresses Used:
                        +1  3323     ; - 22h: 8051 scratch pad RAM location used to store the ADC value on
                        +1  3324     ;        exit of
                        +1  3325     ;
                        +1  3326     ;Registers Used:
                        +1  3327     ;
                        +1  3328     ;======================================================================
                        +1  3329
                        +1  3330
                        +1  3331     ;======================================================================
                        +1  3332     ;   Variable declarations
                        +1  3333     ;======================================================================
                +1  3334
                +1  3335     ;ADC Commands
  3800          +1  3336     ADC KICK      EQU 3800h     ;Kick start address for ADC
  3000          +1  3337     ADC_READ      EQU 3000h     ;Read address for ADC
                +1  3338
                +1  3339
                +1  3340
                +1  3341
                +1  3342     ;==================================================================
                +1  3343     ;   Sub routine - ADC gets current accleration
                +1  3344     ;
                +1  3345     ;
                +1  3346     ;This sub gets the current accleration and writes that value to the
                +1  3347     ;LCD if in State 06, else it just keeps the current value from the
                +1  3348     ;ADC stored in address 22h.  This sub will be called based upon an
                +1  3349     ;overflow of
                +1  3350     ;
                +1  3351     ;
                +1  3352     ;Registers:
                +1  3353     ; - R1: Points to current state (21h)
                +1  3354     ;==================================================================
                +1  3355
0D36            +1  3356     ADC_GetAcc:
0D36 C0D0       +1  3357         push    PSW
0D38 C083       +1  3358         push    DPH
0D3A C082       +1  3359         push    DPL
0D3C C0E0       +1  3360         push    ACC
0D3E C0F0       +1  3361         push    B
                +1  3362
                +1  3363
0D40 903800     +1  3364         mov   DPTR,#ADC_KICK    ;
0D43 F0         +1  3365         movx  @DPTR,A           ;Kick starts ADC to convert
                +1  3366
0D44            +1  3367     ADC_Delay_Init:
0D44 7BF5       +1  3368         mov   R3,#245           ;Wait loop for 116uS delay
0D46            +1  3369     ADC_Delay_Loop:
0D46 00         +1  3370         nop
0D47 00         +1  3371         nop
0D48 00         +1  3372         nop
0D49 00         +1  3373         nop
0D4A 00         +1  3374         nop
0D4B 00         +1  3375         nop
0D4C 00         +1  3376         nop
0D4D DBF7       +1  3377         djnz    R3,ADC_Delay_Loop
                +1  3378
                +1  3379
0D4F 903000     +1  3380         mov   DPTR,#ADC_READ    ;
```

```
0D52 E0         +1  3381         movx  A,@DPTR           ;Get current Acceleration
0D53 F522       +1  3382         mov   22h,A             ;Store ADC value in stratch RAM
                +1  3383
                +1  3384
0D55            +1  3385     ADC_State_Chk:
                +1  3386
                +1  3387
                +1  3388
0D55 120D84     +1  3389         lcall   ADC_Convert     ;Sub which converts 0-255 value from accel
                +1  3390                                 ;into correct DPTR for lookup table
                +1  3391
                +1  3392
                +1  3393
0D58 12075F     +1  3394         lcall   RAM_Write_ADC   ;Write string to ext RAM starting
                +1  3395                                 ;at address loc. 2004h
                +1  3396
0D5B 120DA0     +1  3397         lcall   ADC_Compare     ;compare here for alarm condition vs. STPT
                +1  3398
                +1  3399
                +1  3400
0D5E B70618     +1  3401        cjne    @R1,#06,ADC_Finished  ;If State=06 then convert and print
                +1  3402                                      ;ADC value to LCD
                +1  3403
                +1  3404
0D61 C083       +1  3405        push    DPH
0D63 C082       +1  3406        push    DPL
                +1  3407
0D65 901200     +1  3408        mov     DPTR,#LCD_CMD        ;Locates Cursor
0D68 75E0C7     +1  3409        mov     ACC,#80h + 47h
0D6B F0         +1  3410        movx    @DPTR,A
0D6C 12009E     +1  3411        lcall   LCD_Busy
                +1  3412
0D6F D082       +1  3413        pop     DPL
0D71 D083       +1  3414        pop     DPH
                +1  3415
0D73 75E000     +1  3416        mov     ACC,#00h            ;First char in string
0D76 120052     +1  3417        lcall LCD_Print
                +1  3418
0D79            +1  3419    ADC_Finished:
                +1  3420
0D79 D0F0       +1  3421        pop     B
0D7B D0E0       +1  3422        pop     ACC
0D7D D082       +1  3423        pop     DPL
0D7F D083       +1  3424        pop     DPH
0D81 D0D0       +1  3425        pop     PSW
                +1  3426
0D83 32         +1  3427        reti
                +1  3428
                +1  3429
                +1  3430    ;=========================================================================
                +1  3431    ;   Sub routine - ADC Convert
                +1  3432    ;
                +1  3433    ;DPTR exits this routine pointing to the first letter in string
                +1  3434    ;assuming the ACC is @ zero when using the 'movc A,@A + DPTR' command.
                +1  3435    ;
                +1  3436    ;Registers:
                +1  3437    ; - ACC: The ACC on entry into sub ='s the value from the ADC: 0-255
                +1  3438    ; - B: This ='s the number of characters in each string (07h)
                +1  3439    ; - R3: Used to manipulate the value of the B register
                +1  3440    ;
                +1  3441    ;=========================================================================
                +1  3442
0D84            +1  3443    ADC_Convert:
                +1  3444
0D84 900E52     +1  3445        mov   DPTR,#ADC_AccelTable    ;Acceleration lookup table
0D87 75F007     +1  3446        mov   B,#07h                  ;Num of Char. per string
```

```
0D8A A4          +1  3447         mul   AB                             ;Get num of times to increment DPTR
                 +1  3448                                              ;for correct location in table
0D8B ABF0        +1  3449         mov   R3,B
                 +1  3450
0D8D             +1  3451     ADC_Convert_CheckACC:
0D8D 6005        +1  3452         jz      ADC_Convert_CheckR3   ;if ACC=00h then check B
0D8F 15E0        +1  3453         dec     ACC
0D91 A3          +1  3454         inc     DPTR
0D92 80F9        +1  3455         jmp     ADC_Convert_CheckACC
                 +1  3456
0D94             +1  3457     ADC_Convert_CheckR3:
0D94 BB0100      +1  3458         cjne    R3,#01h,$ + 3
0D97 4006        +1  3459         jc      ADC_Convert_Finish    ;if B=00h then exit
0D99 1B          +1  3460         dec     R3
0D9A 15E0        +1  3461         dec     ACC
0D9C A3          +1  3462         inc     DPTR
0D9D 80EE        +1  3463         jmp     ADC_Convert_CheckACC
                 +1  3464
0D9F             +1  3465     ADC_Convert_Finish:
```

```
0D9F 22         +1  3466        ret
                +1  3467
                +1  3468    ;======================================================================
                +1  3469    ;   Sub routine - ADC Compare
                +1  3470    ;
                +1  3471    ;DPTR exits this routine pointing to the first letter in string
                +1  3472    ;assuming the ACC is @ zero when using the 'movc A,@A + DPTR' command.
                +1  3473    ;
                +1  3474    ;Registers:
                +1  3475    ; - ACC: Returns with the LSB of the actual Acceleration.
                +1  3476    ; - 27h: MSB of the Setpoint in Ascii
                +1  3477    ; - 26h: Next Byte of the Setpoint in Ascii
                +1  3478    ; - 25h: LSB of the Setpoint in Ascii
                +1  3479    ;
                +1  3480    ;======================================================================
                +1  3481
0DA0            +1  3482    ADC_Compare:
                +1  3483
0DA0 C083       +1  3484        push  DPH
0DA2 C082       +1  3485        push  DPL
                +1  3486
0DA4 902005     +1  3487        mov         DPTR,#2005h
0DA7 E0         +1  3488                    movx    A,@DPTR
0DA8 B52713     +1  3489                    cjne    A,27h,ADC_Carry1
                +1  3490
0DAB 902007     +1  3491                    mov             DPTR,#2007h
0DAE E0         +1  3492                    movx    A,@DPTR
0DAF B52613     +1  3493                    cjne    A,26h,ADC_Carry2
                +1  3494
0DB2 902008     +1  3495                    mov             DPTR,#2008h
0DB5 E0         +1  3496                    movx    A,@DPTR
0DB6 B52513     +1  3497                    cjne    A,25h,ADC_Carry3
                +1  3498
                +1  3499
0DB9            +1  3500    ADC_Compare_Finish:
                +1  3501
0DB9 D082       +1  3502        pop  DPL
0DBB D083       +1  3503        pop  DPH
                +1  3504
0DBD 22         +1  3505        ret
                +1  3506
0DBE            +1  3507    ADC_Carry1:
                +1  3508
0DBE 40F9       +1  3509                    jc              ADC Compare_Finish
0DC0 120DD3     +1  3510                    lcall   ADC Serial Print
0DC3 80F4       +1  3511                    sjmp    ADC_Compare_Finish
                +1  3512
```

```
0DC5            +1  3513     ADC_Carry2:
                +1  3514
0DC5 40F2       +1  3515                     jc              ADC Compare_Finish
0DC7 120DD3     +1  3516                     lcall   ADC Serial Print
0DCA 80ED       +1  3517                     sjmp    ADC_Compare_Finish
                +1  3518
0DCC            +1  3519     ADC_Carry3:
                +1  3520
0DCC 40EB       +1  3521                     jc              ADC Compare_Finish
0DCE 120DD3     +1  3522                     lcall   ADC Serial Print
0DD1 80E6       +1  3523                     sjmp    ADC_Compare_Finish
                +1  3524
                +1  3525
                +1  3526
                +1  3527     ;=======================================================================
                +1  3528     ;   Sub routine - ADC Alarm: This routine will print an Alarm string
                +1  3529     ;   if the Acceleration Setpoint is less than the actual Acceleration
                +1  3530     ;
                +1  3531     ;
```

```
                  +1  3532     ;
                  +1  3533     ;===================================================================
                  +1  3534
0DD3              +1  3535     ADC_Serial Print:
0DD3 20192A       +1  3536         jb    19h,ADC Serial Finish   ;Only allows for one String to print
0DD6 900E03       +1  3537         mov   DPTR,#ADC_Alarm_Table
0DD9 7400         +1  3538         mov   A,#00h
0DDB              +1  3539     ADC_Ser_Loop:
0DDB 7400         +1  3540         mov   A,#00h
0DDD 93           +1  3541         movc  A,@A + DPTR
0DDE 600D         +1  3542         jz    ADC Print_Accel         ;Print the Acceleration if Null ACC
0DE0 F5F2         +1  3543         mov   SBUF1,A                 ;Moves first Character to serial port
                  +1  3544
0DE2 E5F1         +1  3545         mov   A,SCON1
0DE4 30E1FB       +1  3546         jnb   ACC.1,$ - 2             ;Poll for Transmit flag
0DE7 75F140       +1  3547         mov   SCON1,#40h              ;Clear Transmit Flag
0DEA A3           +1  3548         inc   DPTR                    ;Point to next character in string
                  +1  3549
0DEB 80EE         +1  3550         sjmp  ADC_Ser_Loop
                  +1  3551
0DED              +1  3552     ADC_Print_Accel:
                  +1  3553
0DED 902004       +1  3554         mov   DPTR, #2004h            ;Load the location of the first Char.
0DF0              +1  3555     ADC_Ser Loop2:
0DF0 E0           +1  3556         movx  A,@DPTR                 ;Bring in Char from RAM
0DF1 600D         +1  3557         jz    ADC Serial_Finish       ;Jump to end if Null hit in RAM
0DF3 F5F2         +1  3558         mov   SBUF1, A                ;Start Serial Transmission
0DF5 E5F1         +1  3559         mov   A,SCON1
0DF7 30E1FB       +1  3560         jnb   ACC.1,$ - 2             ;Poll for Transmit Flag
0DFA 75F140       +1  3561         mov   SCON1,#40h              ;Clear Transmit Flag
0DFD A3           +1  3562         inc   DPTR                    ;Point to next Location in RAM
0DFE 80F0         +1  3563         sjmp  ADC_Ser_Loop2
                  +1  3564
0E00              +1  3565     ADC_Serial Finish:
0E00 D219         +1  3566         setb  19h
0E02 22           +1  3567         ret
                  +1  3568
0E03              +1  3569     ADC_Alarm Table:
0E03 2A2A2A2A     +1  3570         db "************************",0Dh,0Ah
0E07 2A2A2A2A
0E0B 2A2A2A2A
0E0F 2A2A2A2A
0E13 2A2A2A2A
0E17 2A2A2A2A
0E1B 0D0A
0E1D 2A202041     +1  3571         db "*  Acceleration Alarm  *",0Dh,0Ah
0E21 6363656C
```

```
0E25 65726174
0E29 696F6E20
0E2D 416C6172
0E31 6D20202A
0E35 0D0A
0E37 2A2A2A2A   +1  3572         db "************************",0Dh,0Ah,0
0E3B 2A2A2A2A
0E3F 2A2A2A2A
0E43 2A2A2A2A
0E47 2A2A2A2A
0E4B 2A2A2A2A
0E4F 0D0A00
                +1  3573
0E52            +1  3574     ADC_AccelTable:
0E52 2D312E32   +1  3575       db "-1.20g",0
0E56 306700
0E59 2D312E31   +1  3576       db "-1.19g",0
0E5D 396700
0E60 2D312E31   +1  3577       db "-1.18g",0
```

```
0E64 386700
0E67 2D312E31   +1  3578       db "-1.17g",0
0E6B 376700
0E6E 2D312E31   +1  3579       db "-1.16g",0
0E72 366700
0E75 2D312E31   +1  3580       db "-1.15g",0
0E79 356700
0E7C 2D312E31   +1  3581       db "-1.14g",0
0E80 346700
0E83 2D312E31   +1  3582       db "-1.13g",0
0E87 336700
0E8A 2D312E31   +1  3583       db "-1.12g",0
0E8E 326700
0E91 2D312E31   +1  3584       db "-1.11g",0
0E95 316700
0E98 2D312E31   +1  3585       db "-1.10g",0
0E9C 306700
0E9F 2D312E30   +1  3586       db "-1.09g",0
0EA3 396700
0EA6 2D312E30   +1  3587       db "-1.08g",0
0EAA 386700
0EAD 2D312E30   +1  3588       db "-1.07g",0
0EB1 376700
0EB4 2D312E30   +1  3589       db "-1.06g",0
0EB8 366700
0EBB 2D312E30   +1  3590       db "-1.05g",0
0EBF 356700
0EC2 2D312E30   +1  3591       db "-1.04g",0
0EC6 346700
0EC9 2D312E30   +1  3592       db "-1.03g",0
0ECD 336700
0ED0 2D312E30   +1  3593       db "-1.02g",0
0ED4 326700
0ED7 2D312E30   +1  3594       db "-1.01g",0
0EDB 316700
0EDE 2D312E30   +1  3595       db "-1.00g",0
0EE2 306700
0EE5 2D302E39   +1  3596       db "-0.99g",0
0EE9 396700
0EEC 2D302E39   +1  3597       db "-0.98g",0
0EF0 386700
0EF3 2D302E39   +1  3598       db "-0.97g",0
0EF7 376700
0EFA 2D302E39   +1  3599       db "-0.96g",0
0EFE 366700
0F01 2D302E39   +1  3600       db "-0.95g",0
0F05 356700
```

```
0F08 2D302E39   +1  3601       db "-0.94g",0
0F0C 346700
0F0F 2D302E39   +1  3602       db "-0.93g",0
0F13 336700
0F16 2D302E39   +1  3603       db "-0.92g",0
0F1A 326700
0F1D 2D302E39   +1  3604       db "-0.91g",0
0F21 316700
0F24 2D302E39   +1  3605       db "-0.90g",0
0F28 306700
0F2B 2D302E38   +1  3606       db "-0.89g",0
0F2F 396700
0F32 2D302E38   +1  3607       db "-0.88g",0
0F36 386700
0F39 2D302E38   +1  3608       db "-0.87g",0
0F3D 376700
0F40 2D302E38   +1  3609       db "-0.86g",0
0F44 366700
0F47 2D302E38   +1  3610       db "-0.85g",0
```

```
0F4B 356700
0F4E 2D302E38   +1  3611       db "-0.84g",0
0F52 346700
0F55 2D302E38   +1  3612       db "-0.83g",0
0F59 336700
0F5C 2D302E38   +1  3613       db "-0.82g",0
0F60 326700
0F63 2D302E38   +1  3614       db "-0.81g",0
0F67 316700
0F6A 2D302E38   +1  3615       db "-0.80g",0
0F6E 306700
0F71 2D302E37   +1  3616       db "-0.79g",0
0F75 396700
0F78 2D302E37   +1  3617       db "-0.78g",0
0F7C 386700
0F7F 2D302E37   +1  3618       db "-0.77g",0
0F83 376700
0F86 2D302E37   +1  3619       db "-0.76g",0
0F8A 366700
0F8D 2D302E37   +1  3620       db "-0.75g",0
0F91 356700
0F94 2D302E37   +1  3621       db "-0.74g",0
0F98 346700
0F9B 2D302E37   +1  3622       db "-0.73g",0
0F9F 336700
0FA2 2D302E37   +1  3623       db "-0.72g",0
0FA6 326700
0FA9 2D302E37   +1  3624       db "-0.71g",0
0FAD 316700
0FB0 2D302E37   +1  3625       db "-0.70g",0
0FB4 306700
0FB7 2D302E36   +1  3626       db "-0.69g",0
0FBB 396700
0FBE 2D302E36   +1  3627       db "-0.68g",0
0FC2 386700
0FC5 2D302E36   +1  3628       db "-0.67g",0
0FC9 376700
0FCC 2D302E36   +1  3629       db "-0.66g",0
0FD0 366700
0FD3 2D302E36   +1  3630       db "-0.65g",0
0FD7 356700
0FDA 2D302E36   +1  3631       db "-0.64g",0
0FDE 346700
0FE1 2D302E36   +1  3632       db "-0.63g",0
0FE5 336700
0FE8 2D302E36   +1  3633       db "-0.62g",0
0FEC 326700
```

```
0FEF 2D302E36   +1  3634        db "-0.61g",0
0FF3 316700
0FF6 2D302E36   +1  3635        db "-0.60g",0
0FFA 306700
0FFD 2D302E35   +1  3636        db "-0.59g",0
1001 396700
1004 2D302E35   +1  3637        db "-0.58g",0
1008 386700
100B 2D302E35   +1  3638        db "-0.57g",0
100F 376700
1012 2D302E35   +1  3639        db "-0.56g",0
1016 366700
1019 2D302E35   +1  3640        db "-0.55g",0
101D 356700
1020 2D302E35   +1  3641        db "-0.54g",0
1024 346700
1027 2D302E35   +1  3642        db "-0.53g",0
102B 336700
102E 2D302E35   +1  3643        db "-0.52g",0
```

```
1032 326700
1035 2D302E35   +1  3644       db "-0.51g",0
1039 316700
103C 2D302E35   +1  3645       db "-0.50g",0
1040 306700
1043 2D302E34   +1  3646       db "-0.49g",0
1047 396700
104A 2D302E34   +1  3647       db "-0.48g",0
104E 386700
1051 2D302E34   +1  3648       db "-0.47g",0
1055 376700
1058 2D302E34   +1  3649       db "-0.46g",0
105C 366700
105F 2D302E34   +1  3650       db "-0.45g",0
1063 356700
1066 2D302E34   +1  3651       db "-0.44g",0
106A 346700
106D 2D302E34   +1  3652       db "-0.43g",0
1071 336700
1074 2D302E34   +1  3653       db "-0.42g",0
1078 326700
107B 2D302E34   +1  3654       db "-0.41g",0
107F 316700
1082 2D302E34   +1  3655       db "-0.40g",0
1086 306700
1089 2D302E33   +1  3656       db "-0.39g",0
108D 396700
1090 2D302E33   +1  3657       db "-0.38g",0
1094 386700
1097 2D302E33   +1  3658       db "-0.37g",0
109B 376700
109E 2D302E33   +1  3659       db "-0.36g",0
10A2 366700
10A5 2D302E33   +1  3660       db "-0.35g",0
10A9 356700
10AC 2D302E33   +1  3661       db "-0.34g",0
10B0 346700
10B3 2D302E33   +1  3662       db "-0.33g",0
10B7 336700
10BA 2D302E33   +1  3663       db "-0.32g",0
10BE 326700
10C1 2D302E33   +1  3664       db "-0.31g",0
10C5 316700
10C8 2D302E33   +1  3665       db "-0.30g",0
10CC 306700
10CF 2D302E32   +1  3666       db "-0.29g",0
10D3 396700
```

```
10D6 2D302E32   +1  3667       db "-0.28g",0
10DA 386700
10DD 2D302E32   +1  3668       db "-0.27g",0
10E1 376700
10E4 2D302E32   +1  3669       db "-0.26g",0
10E8 366700
10EB 2D302E32   +1  3670       db "-0.25g",0
10EF 356700
10F2 2D302E32   +1  3671       db "-0.24g",0
10F6 346700
10F9 2D302E32   +1  3672       db "-0.23g",0
10FD 336700
1100 2D302E32   +1  3673       db "-0.22g",0
1104 326700
1107 2D302E32   +1  3674       db "-0.21g",0
110B 316700
110E 2D302E32   +1  3675       db "-0.20g",0
1112 306700
1115 2D302E31   +1  3676       db "-0.19g",0
```

```
1119 396700
111C 2D302E31   +1  3677       db "-0.18g",0
1120 386700
1123 2D302E31   +1  3678       db "-0.17g",0
1127 376700
112A 2D302E31   +1  3679       db "-0.16g",0
112E 366700
1131 2D302E31   +1  3680       db "-0.15g",0
1135 356700
1138 2D302E31   +1  3681       db "-0.14g",0
113C 346700
113F 2D302E31   +1  3682       db "-0.13g",0
1143 336700
1146 2D302E31   +1  3683       db "-0.12g",0
114A 326700
114D 2D302E31   +1  3684       db "-0.11g",0
1151 316700
1154 2D302E31   +1  3685       db "-0.10g",0
1158 306700
115B 2D302E30   +1  3686       db "-0.09g",0
115F 396700
1162 2D302E30   +1  3687       db "-0.08g",0
1166 386700
1169 2D302E30   +1  3688       db "-0.07g",0
116D 376700
1170 2D302E30   +1  3689       db "-0.06g",0
1174 366700
1177 2D302E30   +1  3690       db "-0.05g",0
117B 356700
117E 2D302E30   +1  3691       db "-0.04g",0
1182 346700
1185 2D302E30   +1  3692       db "-0.03g",0
1189 336700
118C 2D302E30   +1  3693       db "-0.02g",0
1190 326700
1193 2D302E30   +1  3694       db "-0.01g",0
1197 316700
119A 20302E30   +1  3695       db " 0.00g",0
119E 306700
11A1 20302E30   +1  3696       db " 0.00g",0
11A5 306700
11A8 20302E30   +1  3697       db " 0.00g",0
11AC 306700
11AF 20302E30   +1  3698       db " 0.00g",0
11B3 306700
11B6 20302E30   +1  3699       db " 0.00g",0
11BA 306700
```

```
11BD 20302E30   +1  3700       db " 0.00g",0
11C1 306700
11C4 20302E30   +1  3701       db " 0.00g",0
11C8 306700
11CB 20302E30   +1  3702       db " 0.00g",0
11CF 306700
11D2 20302E30   +1  3703       db " 0.00g",0
11D6 306700
11D9 20302E30   +1  3704       db " 0.00g",0
11DD 306700
11E0 20302E30   +1  3705       db " 0.00g",0
11E4 306700
11E7 20302E30   +1  3706       db " 0.00g",0
11EB 306700
11EE 20302E30   +1  3707       db " 0.00g",0
11F2 306700
11F5 20302E30   +1  3708       db " 0.00g",0
11F9 306700
11FC 20302E30   +1  3709       db " 0.00g",0
```

```
1200 306700
1203 20302E30   +1  3710       db " 0.00g",0
1207 306700
120A 2B302E30   +1  3711       db "+0.01g",0
120E 316700
1211 2B302E30   +1  3712       db "+0.02g",0
1215 326700
1218 2B302E30   +1  3713       db "+0.03g",0
121C 336700
121F 2B302E30   +1  3714       db "+0.04g",0
1223 346700
1226 2B302E30   +1  3715       db "+0.05g",0
122A 356700
122D 2B302E30   +1  3716       db "+0.06g",0
1231 366700
1234 2B302E30   +1  3717       db "+0.07g",0
1238 376700
123B 2B302E30   +1  3718       db "+0.08g",0
123F 386700
1242 2B302E30   +1  3719       db "+0.09g",0
1246 396700
1249 2B302E31   +1  3720       db "+0.10g",0
124D 306700
1250 2B302E31   +1  3721       db "+0.11g",0
1254 316700
1257 2B302E31   +1  3722       db "+0.12g",0
125B 326700
125E 2B302E31   +1  3723       db "+0.13g",0
1262 336700
1265 2B302E31   +1  3724       db "+0.14g",0
1269 346700
126C 2B302E31   +1  3725       db "+0.15g",0
1270 356700
1273 2B302E31   +1  3726       db "+0.16g",0
1277 366700
127A 2B302E31   +1  3727       db "+0.17g",0
127E 376700
1281 2B302E31   +1  3728       db "+0.18g",0
1285 386700
1288 2B302E31   +1  3729       db "+0.19g",0
128C 396700
128F 2B302E32   +1  3730       db "+0.20g",0
1293 306700
1296 2B302E32   +1  3731       db "+0.21g",0
129A 316700
129D 2B302E32   +1  3732       db "+0.22g",0
12A1 326700
```

```
12A4 2B302E32   +1  3733       db "+0.23g",0
12A8 336700
12AB 2B302E32   +1  3734       db "+0.24g",0
12AF 346700
12B2 2B302E32   +1  3735       db "+0.25g",0
12B6 356700
12B9 2B302E32   +1  3736       db "+0.26g",0
12BD 366700
12C0 2B302E32   +1  3737       db "+0.27g",0
12C4 376700
12C7 2B302E32   +1  3738       db "+0.28g",0
12CB 386700
12CE 2B302E32   +1  3739       db "+0.29g",0
12D2 396700
12D5 2B302E33   +1  3740       db "+0.30g",0
12D9 306700
12DC 2B302E33   +1  3741       db "+0.31g",0
12E0 316700
12E3 2B302E33   +1  3742       db "+0.32g",0
```

```
12E7 326700
12EA 2B302E33   +1  3743       db "+0.33g",0
12EE 336700
12F1 2B302E33   +1  3744       db "+0.34g",0
12F5 346700
12F8 2B302E33   +1  3745       db "+0.35g",0
12FC 356700
12FF 2B302E33   +1  3746       db "+0.36g",0
1303 366700
1306 2B302E33   +1  3747       db "+0.37g",0
130A 376700
130D 2B302E33   +1  3748       db "+0.38g",0
1311 386700
1314 2B302E33   +1  3749       db "+0.39g",0
1318 396700
131B 2B302E34   +1  3750       db "+0.40g",0
131F 306700
1322 2B302E34   +1  3751       db "+0.41g",0
1326 316700
1329 2B302E34   +1  3752       db "+0.42g",0
132D 326700
1330 2B302E34   +1  3753       db "+0.43g",0
1334 336700
1337 2B302E34   +1  3754       db "+0.44g",0
133B 346700
133E 2B302E34   +1  3755       db "+0.45g",0
1342 356700
1345 2B302E34   +1  3756       db "+0.46g",0
1349 366700
134C 2B302E34   +1  3757       db "+0.47g",0
1350 376700
1353 2B302E34   +1  3758       db "+0.48g",0
1357 386700
135A 2B302E34   +1  3759       db "+0.49g",0
135E 396700
1361 2B302E35   +1  3760       db "+0.50g",0
1365 306700
1368 2B302E35   +1  3761       db "+0.51g",0
136C 316700
136F 2B302E35   +1  3762       db "+0.52g",0
1373 326700
1376 2B302E35   +1  3763       db "+0.53g",0
137A 336700
137D 2B302E35   +1  3764       db "+0.54g",0
1381 346700
1384 2B302E35   +1  3765       db "+0.55g",0
1388 356700
```

```
138B 2B302E35   +1  3766        db "+0.56g",0
138F 366700
1392 2B302E35   +1  3767        db "+0.57g",0
1396 376700
1399 2B302E35   +1  3768        db "+0.58g",0
139D 386700
13A0 2B302E35   +1  3769        db "+0.59g",0
13A4 396700
13A7 2B302E36   +1  3770        db "+0.60g",0
13AB 306700
13AE 2B302E36   +1  3771        db "+0.61g",0
13B2 316700
13B5 2B302E36   +1  3772        db "+0.62g",0
13B9 326700
13BC 2B302E36   +1  3773        db "+0.63g",0
13C0 336700
13C3 2B302E36   +1  3774        db "+0.64g",0
13C7 346700
13CA 2B302E36   +1  3775        db "+0.65g",0
```

```
13CE 356700
13D1 2B302E36   +1  3776       db "+0.66g",0
13D5 366700
13D8 2B302E36   +1  3777       db "+0.67g",0
13DC 376700
13DF 2B302E36   +1  3778       db "+0.68g",0
13E3 386700
13E6 2B302E36   +1  3779       db "+0.69g",0
13EA 396700
13ED 2B302E37   +1  3780       db "+0.70g",0
13F1 306700
13F4 2B302E37   +1  3781       db "+0.71g",0
13F8 316700
13FB 2B302E37   +1  3782       db "+0.72g",0
13FF 326700
1402 2B302E37   +1  3783       db "+0.73g",0
1406 336700
1409 2B302E37   +1  3784       db "+0.74g",0
140D 346700
1410 2B302E37   +1  3785       db "+0.75g",0
1414 356700
1417 2B302E37   +1  3786       db "+0.76g",0
141B 366700
141E 2B302E37   +1  3787       db "+0.77g",0
1422 376700
1425 2B302E37   +1  3788       db "+0.78g",0
1429 386700
142C 2B302E37   +1  3789       db "+0.79g",0
1430 396700
1433 2B302E38   +1  3790       db "+0.80g",0
1437 306700
143A 2B302E38   +1  3791       db "+0.81g",0
143E 316700
1441 2B302E38   +1  3792       db "+0.82g",0
1445 326700
1448 2B302E38   +1  3793       db "+0.83g",0
144C 336700
144F 2B302E38   +1  3794       db "+0.84g",0
1453 346700
1456 2B302E38   +1  3795       db "+0.85g",0
145A 356700
145D 2B302E38   +1  3796       db "+0.86g",0
1461 366700
1464 2B302E38   +1  3797       db "+0.87g",0
1468 376700
146B 2B302E38   +1  3798       db "+0.88g",0
146F 386700
```

```
1472 2B302E38   +1  3799       db "+0.89g",0
1476 396700
1479 2B302E39   +1  3800       db "+0.90g",0
147D 306700
1480 2B302E39   +1  3801       db "+0.91g",0
1484 316700
1487 2B302E39   +1  3802       db "+0.92g",0
148B 326700
148E 2B302E39   +1  3803       db "+0.93g",0
1492 336700
1495 2B302E39   +1  3804       db "+0.94g",0
1499 346700
149C 2B302E39   +1  3805       db "+0.95g",0
14A0 356700
14A3 2B302E39   +1  3806       db "+0.96g",0
14A7 366700
14AA 2B302E39   +1  3807       db "+0.97g",0
14AE 376700
14B1 2B302E39   +1  3808       db "+0.98g",0
```

```
14B5 386700
14B8 2B302E39   +1  3809       db "+0.99g",0
14BC 396700
14BF 2B312E30   +1  3810       db "+1.00g",0
14C3 306700
14C6 2B312E30   +1  3811       db "+1.01g",0
14CA 316700
14CD 2B312E30   +1  3812       db "+1.02g",0
14D1 326700
14D4 2B312E30   +1  3813       db "+1.03g",0
14D8 336700
14DB 2B312E30   +1  3814       db "+1.04g",0
14DF 346700
14E2 2B312E30   +1  3815       db "+1.05g",0
14E6 356700
14E9 2B312E30   +1  3816       db "+1.06g",0
14ED 366700
14F0 2B312E30   +1  3817       db "+1.07g",0
14F4 376700
14F7 2B312E30   +1  3818       db "+1.08g",0
14FB 386700
14FE 2B312E30   +1  3819       db "+1.09g",0
1502 396700
1505 2B312E31   +1  3820       db "+1.10g",0
1509 306700
150C 2B312E31   +1  3821       db "+1.11g",0
1510 316700
1513 2B312E31   +1  3822       db "+1.12g",0
1517 326700
151A 2B312E31   +1  3823       db "+1.13g",0
151E 336700
1521 2B312E31   +1  3824       db "+1.14g",0
1525 346700
1528 2B312E31   +1  3825       db "+1.15g",0
152C 356700
152F 2B312E31   +1  3826       db "+1.16g",0
1533 366700
1536 2B312E31   +1  3827       db "+1.17g",0
153A 376700
153D 2B312E31   +1  3828       db "+1.18g",0
1541 386700
1544 2B312E31   +1  3829       db "+1.19g",0
1548 396700
154B 2B312E32   +1  3830       db "+1.20g",0
154F 306700
                    3831     ;$include (Alarm.asm) ;Alarm routines
                +1  3832     ;=================================================================
;                              Pro-Tex 9000
;
;Revision: R.07171500  (R.MMDDHHMM)
;
;Project Team Members:
; - Vince Watkins
; - Will Smith
; - Tyler Long
;
;=Alarm Subroutines=
;
;
;
;Registers Used:
;
;======================================================================


;======================================================================
                +1     ;   Variable declarations
                +1     ;=====================================================================
                +1
                +1
                +1
                +1
                +1
                +1
                +1     ;=====================================================================
                +1     ;   Sub routine - Check Alarm Status and Switches
                +1     ;
                +1     ;The end of this subroutine will disable its own external interrupt
                +1     ;before the 'reti' and also re-enable the Timer 1 count for getting
                +1     ;the current acceleration.  /INT1 will only be re-enabled once the
                +1     ;system has been sucessifully disarmed witht the correct password.
                +1     ;
                +1     ;Alarms:
                +1     ; - P3.2: Tamper Alarm
                +1     ; - P3.3: Door Ajar Alarm
                +1     ; - P3.4: Panic Alarm
                +1     ;
                +1     ;=====================================================================
                +1
1552            +1     Alarm Check:
1552 C28E       +1         clr   TR1           ;Stop Timer 1 from interrupting
                +1
1554 C0D0       +1         push    PSW
1556 C083       +1         push    DPH
1558 C082       +1         push    DPL
155A C0E0       +1         push    ACC
155C C0F0       +1         push    B
                +1
155E            +1     Alarm Panic:
155E 30B409     +1         jnb   P3.4,Alarm_Check_Armed
1561 D290       +1         setb  P1.0
1563 C2AA       +1         clr   EX1         ;Disable /INT1
1565 121596     +1         lcall Alarm Ser Panic
1568 801F       +1         jmp   Alarm_Check_Done
                +1
156A            +1     Alarm Check Armed:
156A 30181C     +1         jnb   18h,Alarm Check Done
156D 20B205     +1         jb    P3.2,Alarm Tamper      ;Tamper Alarm
1570 20B30D     +1         jb    P3.3,Alarm Door        ;Door Ajar Alarm
1573 8014       +1         jmp   Alarm_Check_Done
                +1
1575            +1     Alarm Tamper:
1575 D292       +1         setb  P1.2    ;Illuminates Yellow Tamper LED
```

```
1577 D290       +1  3899         setb  P1.0    ;Illuminates alternating flashing LEDs
1579 C2AA       +1  3900         clr   EX1         ;Disable /INT1
157B 1215AE     +1  3901         lcall Alarm Ser Tamper
157E 8009       +1  3902         jmp   Alarm_Check_Done
                +1  3903
1580            +1  3904     Alarm Door:
1580 D290       +1  3905         setb  P1.0
1582 C2AA       +1  3906         clr   EX1         ;Disable /INT1
1584 1215C6     +1  3907         lcall Alarm Ser Door
1587 8000       +1  3908         jmp   Alarm_Check_Done
                +1  3909
1589            +1  3910     Alarm_Check_Done:
                +1  3911
1589 D0F0       +1  3912         pop     B
158B D0E0       +1  3913         pop     ACC
158D D082       +1  3914         pop     DPL
158F D083       +1  3915         pop     DPH
1591 D0D0       +1  3916         pop     PSW
                +1  3917
```

```
1593 D28E       +1  3918          setb  TR1         ;Start Timer 1
                +1  3919
1595 32         +1  3920          reti
                +1  3921
                +1  3922      ;======================================================================
                +1  3923      ;   Sub routine - Serial routine for Panic Alarm
                +1  3924      ;
                +1  3925      ;======================================================================
                +1  3926
1596            +1  3927      Alarm Ser_Panic:
1596 9015DE     +1  3928          mov   DPTR,#Alarm_Panic_Serial
1599 7400       +1  3929          mov   A,#00h
159B            +1  3930      Alarm Ser_Panic Loop:
159B 7400       +1  3931          mov   A,#00h
159D 93         +1  3932          movc  A,@A + DPTR
159E 600D       +1  3933          jz    Alarm Ser_Panic_Finish
15A0 F5F2       +1  3934          mov   SBUF1,A
                +1  3935
15A2 E5F1       +1  3936          mov   A,SCON1
15A4 30E1FB     +1  3937          jnb   ACC.1,$ - 2
15A7 75F140     +1  3938          mov   SCON1,#40h
15AA A3         +1  3939          inc   DPTR
                +1  3940
15AB 80EE       +1  3941          sjmp  Alarm_Ser_Panic_Loop
                +1  3942
15AD            +1  3943      Alarm Ser_Panic_Finish:
15AD 22         +1  3944          ret
                +1  3945
                +1  3946      ;======================================================================
                +1  3947      ;   Sub routine - Serial routine for Tamper Alarm
                +1  3948      ;
                +1  3949      ;======================================================================
                +1  3950
15AE            +1  3951      Alarm Ser_Tamper:
15AE 90162D     +1  3952          mov   DPTR,#Alarm_Tamper_Serial
15B1 7400       +1  3953          mov   A,#00h
15B3            +1  3954      Alarm Ser_Tamper_Loop:
15B3 7400       +1  3955          mov   A,#00h
15B5 93         +1  3956          movc  A,@A + DPTR
15B6 600D       +1  3957          jz    Alarm Ser_Tamper_Finish
15B8 F5F2       +1  3958          mov   SBUF1,A
                +1  3959
15BA E5F1       +1  3960          mov   A,SCON1
15BC 30E1FB     +1  3961          jnb   ACC.1,$ - 2
15BF 75F140     +1  3962          mov   SCON1,#40h
15C2 A3         +1  3963          inc   DPTR
                +1  3964
```

```
15C3 80EE       +1  3965          sjmp  Alarm_Ser_Tamper_Loop
                +1  3966
15C5            +1  3967      Alarm Ser_Tamper_Finish:
15C5 22         +1  3968          ret
                +1  3969
                +1  3970      ;=====================================================================
                +1  3971      ;   Sub routine - Serial routine for Door Alarm
                +1  3972      ;
                +1  3973      ;=====================================================================
                +1  3974
15C6            +1  3975      Alarm Ser_Door:
15C6 90168B     +1  3976          mov   DPTR,#Alarm_Door_Serial
15C9 7400       +1  3977          mov   A,#00h
15CB            +1  3978      Alarm Ser_Door Loop:
15CB 7400       +1  3979          mov   A,#00h
15CD 93         +1  3980          movc  A,@A + DPTR
15CE 600D       +1  3981          jz    Alarm Ser_Door_Finish
15D0 F5F2       +1  3982          mov   SBUF1,A
                +1  3983
```

```
15D2 E5F1       +1  3984         mov   A,SCON1
15D4 30E1FB     +1  3985         jnb   ACC.1,$ - 2
15D7 75F140     +1  3986         mov   SCON1,#40h
15DA A3         +1  3987         inc   DPTR
                +1  3988
15DB 80EE       +1  3989         sjmp  Alarm_Ser_Door_Loop
                +1  3990
15DD            +1  3991     Alarm Ser_Door_Finish:
15DD 22         +1  3992         ret
                +1  3993
                +1  3994
                +1  3995
15DE            +1  3996     Alarm Panic Serial:
15DE 2A2A2A2A   +1  3997         db "************************",0Dh,0Ah
15E2 2A2A2A2A
15E6 2A2A2A2A
15EA 2A2A2A2A
15EE 2A2A2A2A
15F2 2A2A2A2A
15F6 0D0A
15F8 2A202050   +1  3998         db "*  Panic Activated!!!  *",0Dh,0Ah
15FC 616E6963
1600 20416374
1604 69766174
1608 65642121
160C 2120202A
1610 0D0A
1612 2A2A2A2A   +1  3999         db "************************",0Dh,0Ah,0
1616 2A2A2A2A
161A 2A2A2A2A
161E 2A2A2A2A
1622 2A2A2A2A
1626 2A2A2A2A
162A 0D0A00
                +1  4000
162D            +1  4001     Alarm Tamper Serial:
162D 2A2A2A2A   +1  4002         db "*****************************",0Dh,0Ah
1631 2A2A2A2A
1635 2A2A2A2A
1639 2A2A2A2A
163D 2A2A2A2A
1641 2A2A2A2A
1645 2A2A2A2A
1649 2A0D0A
164C 2A202041   +1  4003         db "*  Alarm Tamper Tripped!!!  *",0Dh,0Ah
1650 6C61726D
1654 2054616D
```

```
1658 70657220
165C 54726970
1660 70656421
1664 21212020
1668 2A0D0A
166B 2A2A2A2A   +1  4004         db "*****************************",0Dh,0Ah,0
166F 2A2A2A2A
1673 2A2A2A2A
1677 2A2A2A2A
167B 2A2A2A2A
167F 2A2A2A2A
1683 2A2A2A2A
1687 2A0D0A00
                +1  4005
168B            +1  4006     Alarm Door Serial:
168B 2A2A2A2A   +1  4007         db "**********************",0Dh,0Ah
168F 2A2A2A2A
1693 2A2A2A2A
1697 2A2A2A2A
```

```
169B 2A2A2A2A
169F 2A2A0D0A
16A3 2A202043   +1  4008         db "*  Car Door Open!!!  *",0Dh,0Ah
16A7 61722044
16AB 6F6F7220
16AF 4F70656E
16B3 21212120
16B7 202A0D0A
16BB 2A2A2A2A   +1  4009         db "**********************",0Dh,0Ah,0
16BF 2A2A2A2A
16C3 2A2A2A2A
16C7 2A2A2A2A
16CB 2A2A2A2A
16CF 2A2A0D0A
16D3 00
                    4010
                    4011
16D4                4012     Main:
16D4 758130         4013         mov     SP,#30h          ;Initialize stack pointer
16D7 1207FA         4014         lcall   Init_Device
16DA C218           4015         clr     18h             ;System Armed status
                    4016
16DC 120030         4017         lcall   LCD_Init        ;Initialize LCD
16DF 120786         4018                     lcall   RAM_Init        ;Sets default password in RAM
                    4019         ;lcall   ADC Init         ;Kick starts ADC to begin convert
16E2 759000         4020         mov     P1,#00h         ;Turns off all LEDs
                    4021
16E5 752B00         4022         mov     2Bh,#00h        ;This section will clear
16E8 752A00         4023         mov     2Ah,#00h        ;the scratch pad RAM
16EB 752900         4024         mov     29h,#00h        ;in 8051 which stores the
16EE 752800         4025         mov     28h,#00h        ;user entered PW on reset
                    4026
16F1 752730         4027         mov     27h,#30h        ;This sets the default setpoint
16F4 752637         4028         mov     26h,#37h        ;for the Acceleration
16F7 752535         4029         mov     25h,#35h        ;to +- 0.75g
                    4030
                    4031
                    4032     ;===========================================================================
                    4033     ;   Screen #1
                    4034     ;===========================================================================
16FA 7400           4035         mov     A,#00h          ;State index 00
16FC 120D27         4036                     lcall   State_Lookup
                    4037
                    4038     ;===========================================================================
                    4039     ;   Screen #2
                    4040     ;===========================================================================
16FF 7401           4041         mov     A,#01h          ;State index 01
```

```
1701 120D27           lcall   State_Lookup


                  ;=====================================================================
                  ;  Screen #3
                  ;=====================================================================
1704 7402             mov     A,#02h         ;State index 02
1706 120D27           lcall   State_Lookup


1709 80FE             sjmp    $                ;Wait for interrupt from Keypad (/INT0)
                                               ;or ADC Timer


                  end
```

```
XREF SYMBOL TABLE LISTING
---- ------ ----- -------


N A M E                   T Y P E  V A L U E   ATTRIBUTES / REFERENCES

AA . . . . . . . . . . .  B ADDR   00C0H.2 A      215#
AC . . . . . . . . . . .  B ADDR   00D0H.6 A      238#
ACC. . . . . . . . . . .  D ADDR   00E0H   A      139# 369 374 379 401 411 421 461 473 588 602 606 609 613
                                                 634 637 641 650 655 660 665 670 679 688 1327 1332 1337 163
                                                 2122 2222 2231 2238 2248 2255 2265 2272 2298 2303 2337 234
                                                 2438 2445 2473 2478 2509 2518 2525 2535 2542 2552 2559 256
                                                 2594 2624 2633 2640 2650 2657 2667 2674 2700 2709 2716 272
                                                 2806 2840 2845 2881 2888 2895 2925 2932 2939 2968 2978 298
                                                 3045 3077 3082 3116 3121 3151 3158 3165 3195 3202 3209 323
                                                 3360 3409 3416 3422 3453 3461 3546 3560 3881 3913 3937 396
AD0BUSY. . . . . . . . .  B ADDR   00E8H.4 A      255#
AD0EN. . . . . . . . . .  B ADDR   00E8H.7 A      258#
AD0INT . . . . . . . . .  B ADDR   00E8H.5 A      256#
AD0LJST. . . . . . . . .  B ADDR   00E8H.0 A      251#
AD0STM0. . . . . . . . .  B ADDR   00E8H.2 A      253#
AD0STM1. . . . . . . . .  B ADDR   00E8H.3 A      254#
AD0TM. . . . . . . . . .  B ADDR   00E8H.6 A      257#
AD0WINT. . . . . . . . .  B ADDR   00E8H.1 A      252#
ADC0CF . . . . . . . . .  D ADDR   00BCH   A      105#
ADC0CN . . . . . . . . .  D ADDR   00E8H   A      147# 251 252 253 254 255 256 257 258
ADC0GTH. . . . . . . . .  D ADDR   00C5H   A      114#
ADC0GTL. . . . . . . . .  D ADDR   00C4H   A      113#
ADC0H. . . . . . . . . .  D ADDR   00BFH   A      108#
ADC0L. . . . . . . . . .  D ADDR   00BEH   A      107#
ADC0LTH. . . . . . . . .  D ADDR   00C7H   A      116#
ADC0LTL. . . . . . . . .  D ADDR   00C6H   A      115#
ADC1 . . . . . . . . . .  D ADDR   009CH   A      76#
ADC1CF . . . . . . . . .  D ADDR   00ABH   A      90#
ADC1CN . . . . . . . . .  D ADDR   00AAH   A      89#
ADC ACCELTABLE . . . . .  C ADDR   0E52H   A      3445 3574#
ADC ALARM TABLE. . . . .  C ADDR   0E03H   A      3537 3569#
ADC CARRY1 . . . . . . .  C ADDR   0DBEH   A      3489 3507#
ADC CARRY2 . . . . . . .  C ADDR   0DC5H   A      3493 3513#
ADC CARRY3 . . . . . . .  C ADDR   0DCCH   A      3497 3519#
ADC COMPARE. . . . . . .  C ADDR   0DA0H   A      3397 3482#
ADC COMPARE FINISH . . .  C ADDR   0DB9H   A      3500# 3509 3511 3515 3517 3521 3523
ADC CONVERT. . . . . . .  C ADDR   0D84H   A      3389 3443#
ADC CONVERT CHECKACC . .  C ADDR   0D8DH   A      3451# 3455 3463
ADC CONVERT CHECKR3. . .  C ADDR   0D94H   A      3452 3457#
ADC CONVERT FINISH . . .  C ADDR   0D9FH   A      3459 3465#
ADC_DELAY_INIT . . . . .  C ADDR   0D44H   A      3367#
```

```
ADC DELAY LOOP . . . . .  C ADDR   0D46H   A      3369# 3377
ADC FINISHED . . . . . .  C ADDR   0D79H   A      3401 3419#
ADC GETACC . . . . . . .  C ADDR   0D36H   A      309 3356#
ADC KICK . . . . . . . .  N NUMB   3800H   A      3336# 3364
ADC PRINT ACCEL. . . . .  C ADDR   0DEDH   A      3542 3552#
ADC READ . . . . . . . .  N NUMB   3000H   A      3337# 3380
ADC SERIAL FINISH. . . .  C ADDR   0E00H   A      3536 3557 3565#
ADC SERIAL PRINT . . . .  C ADDR   0DD3H   A      3510 3516 3522 3535#
ADC SER LOOP . . . . . .  C ADDR   0DDBH   A      3539# 3550
ADC SER LOOP2. . . . . .  C ADDR   0DF0H   A      3555# 3563
ADC STATE CHK. . . . . .  C ADDR   0D55H   A      3385#
ALARM CHECK. . . . . . .  C ADDR   1552H   A      306 3875#
ALARM CHECK ARMED. . . .  C ADDR   156AH   A      3885 3891#
ALARM CHECK DONE . . . .  C ADDR   1589H   A      3889 3892 3895 3902 3908 3910#
ALARM DOOR . . . . . . .  C ADDR   1580H   A      3894 3904#
ALARM DOOR SERIAL. . . .  C ADDR   168BH   A      3976 4006#
ALARM PANIC. . . . . . .  C ADDR   155EH   A      3884#
ALARM PANIC SERIAL . . .  C ADDR   15DEH   A      3928 3996#
ALARM_SER_DOOR . . . . .  C ADDR   15C6H   A      3907 3975#
```

```
ALARM SER DOOR FINISH. .  C ADDR   15DDH   A       3981 3991#
ALARM SER DOOR LOOP. . .  C ADDR   15CBH   A       3978# 3989
ALARM SER PANIC. . . . .  C ADDR   1596H   A       3888 3927#
ALARM SER PANIC FINISH .  C ADDR   15ADH   A       3933 3943#
ALARM SER PANIC LOOP . .  C ADDR   159BH   A       3930# 3941
ALARM SER TAMPER . . . .  C ADDR   15AEH   A       3901 3951#
ALARM SER TAMPER FINISH.  C ADDR   15C5H   A       3957 3967#
ALARM SER TAMPER_LOOP. .  C ADDR   15B3H   A       3954# 3965
ALARM TAMPER . . . . . .  C ADDR   1575H   A       3893 3897#
ALARM TAMPER_SERIAL. . .  C ADDR   162DH   A       3952 4001#
AMX0CF . . . . . . . . .  D ADDR   00BAH   A       103#
AMX0SL . . . . . . . . .  D ADDR   00BBH   A       104#
AMX1SL . . . . . . . . .  D ADDR   00ACH   A       91#
A 7447 . . . . . . . . .  B ADDR   00B0H.0 A       2171# 2217 2329 2380 2426 2470 2504 2796 2876 2920 3111 3
B. . . . . . . . . . . .  D ADDR   00F0H   A       155# 402 410 420 589 687 1634 1672 1704 1781 3361 3421 34
B 7447 . . . . . . . . .  B ADDR   00B0H.1 A       2172# 2216 2328 2379 2425 2469 2503 2795 2875 2919 3110 3
CCF0 . . . . . . . . . .  B ADDR   00D8H.0 A       242#
CCF1 . . . . . . . . . .  B ADDR   00D8H.1 A       243#
CCF2 . . . . . . . . . .  B ADDR   00D8H.2 A       244#
CCF3 . . . . . . . . . .  B ADDR   00D8H.3 A       245#
CCF4 . . . . . . . . . .  B ADDR   00D8H.4 A       246#
CF . . . . . . . . . . .  B ADDR   00D8H.7 A       248#
CKCON. . . . . . . . . .  D ADDR   008EH   A       63# 2049
CPRL2. . . . . . . . . .  B ADDR   00C8H.0 A       222#
CPT0CN . . . . . . . . .  D ADDR   009EH   A       78#
CPT1CN . . . . . . . . .  D ADDR   009FH   A       79#
CR . . . . . . . . . . .  B ADDR   00D8H.6 A       247#
CT2. . . . . . . . . . .  B ADDR   00C8H.1 A       223#
CY . . . . . . . . . . .  B ADDR   00D0H.7 A       239#
DAC0CN . . . . . . . . .  D ADDR   00D4H   A       128#
DAC0H. . . . . . . . . .  D ADDR   00D3H   A       127#
DAC0L. . . . . . . . . .  D ADDR   00D2H   A       126#
DAC1CN . . . . . . . . .  D ADDR   00D7H   A       131#
DAC1H. . . . . . . . . .  D ADDR   00D6H   A       130#
DAC1L. . . . . . . . . .  D ADDR   00D5H   A       129#
DISP AUTOSHIFT_CURSOR. .  N NUMB   0014H   A       354#
DISP BACKSPACE . . . . .  N NUMB   0010H   A       355# 1327 1337
DISP CLR . . . . . . . .  N NUMB   0001H   A       349# 461
DISP CURSOR. . . . . . .  N NUMB   000FH   A       352# 2347 2806 3008 3121
DISP ENTRY MODE. . . . .  N NUMB   0006H   A       353# 379
DISP FUNCTION CMD. . . .  N NUMB   0038H   A       350# 369
DISP ON. . . . . . . . .  N NUMB   000CH   A       351# 374 2277 2482 2564 2835 3035 3072 3261
DISP_SHIFTRT . . . . . .  N NUMB   001CH   A       356#
DPH. . . . . . . . . . .  D ADDR   0083H   A       52# 399 413 423 586 690 1941 1950 1953 1961 1968 2227 223
                                                  2388 2397 2434 2443 2514 2523 2531 2540 2548 2557 2629 263
                                                  2705 2714 2722 2731 2884 2893 2928 2937 2974 2983 2991 300
                                                  3358 3405 3414 3424 3484 3503 3879 3915
```

```
DPL. . . . . . . . . . . D ADDR   0082H   A      51# 400 412 422 587 689 1942 1951 1954 1958 1960 1967 222
                                                2269 2389 2396 2435 2442 2515 2522 2532 2539 2549 2556 263
                                                2671 2706 2713 2723 2730 2885 2892 2929 2936 2975 2982 299
                                                3206 3359 3406 3413 3423 3485 3502 3880 3914
EA . . . . . . . . . . . B ADDR   00A8H.7 A      202# 2466
EIE1 . . . . . . . . . . D ADDR   00E6H   A      145#
EIE2 . . . . . . . . . . D ADDR   00E7H   A      146#
EIP1 . . . . . . . . . . D ADDR   00F6H   A      161#
EIP2 . . . . . . . . . . D ADDR   00F7H   A      162#
EMI0CF . . . . . . . . . D ADDR   00A3H   A      82# 2062
EMI0CN . . . . . . . . . D ADDR   00AFH   A      94#
EMI0TC . . . . . . . . . D ADDR   00A1H   A      81# 2063
EMI INIT . . . . . . . . C ADDR   07C5H   A      2061# 2138
ENSMB. . . . . . . . . . B ADDR   00C0H.6 A      219#
ES . . . . . . . . . . . B ADDR   00A8H.4 A      200#
ET0. . . . . . . . . . . B ADDR   00A8H.1 A      197#
ET1. . . . . . . . . . . B ADDR   00A8H.3 A      199# 2356
ET2. . . . . . . . . . . B ADDR   00A8H.5 A      201#
EX0. . . . . . . . . . . B ADDR   00A8H.0 A      196# 2351 2403 2449 2601 2678 2738 3013
```

```
EX1. . . . . . . . . . . .  B ADDR   00A8H.2 A      198# 2353 2853 3887 3900 3906
EXEN2. . . . . . . . . . .  B ADDR   00C8H.3 A      225#
EXF2 . . . . . . . . . . .  B ADDR   00C8H.6 A      228#
F0 . . . . . . . . . . . .  B ADDR   00D0H.5 A      237#
F1 . . . . . . . . . . . .  B ADDR   00D0H.1 A      233#
FLACL. . . . . . . . . . .  D ADDR   00B7H   A      100#
FLSCL. . . . . . . . . . .  D ADDR   00B6H   A      99#
IE . . . . . . . . . . . .  D ADDR   00A8H   A      87# 196 197 198 199 200 201 202 2127
IE0. . . . . . . . . . . .  B ADDR   0088H.1 A      177#
IE1. . . . . . . . . . . .  B ADDR   0088H.3 A      179#
INIT DEVICE. . . . . . . .  C ADDR   07FAH   A      2134# 4014
INTERRUPTS_INIT. . . . . .  C ADDR   07F3H   A      2126# 2141
IP . . . . . . . . . . . .  D ADDR   00B8H   A      101# 205 206 207 208 209 210 2128
IT0. . . . . . . . . . . .  B ADDR   0088H.0 A      176#
IT1. . . . . . . . . . . .  B ADDR   0088H.2 A      178#
KEY ACCEL EASY . . . . . .  C ADDR   0388H   A      820 832#
KEY ACCEL EASY1. . . . . .  C ADDR   0398H   A      828 842#
KEY ACCEL EASY2. . . . . .  C ADDR   03A1H   A      829 848#
KEY ACCEL INVALID. . . . .  C ADDR   03B1H   A      817 825 835 839 845 851 860#
KEY ACCEL VALID. . . . . .  C ADDR   03AAH   A      840 846 852 854#
KEY ACCEL VALID CHECK. . .  C ADDR   036EH   A      789 812#
KEY ACCEL VALID FINISH . .  C ADDR   03B8H   A      857 863 866#
KEY BACKSPACE. . . . . . .  C ADDR   02BDH   A      599 674#
KEY BLUE . . . . . . . . .  C ADDR   02A1H   A      619 654#
KEY BLUE NUM . . . . . . .  C ADDR   06E3H   A      1657 1752 1792#
KEY BS RESOLVE . . . . . .  C ADDR   054DH   A      1243 1249 1255 1261 1267 1273 1279 1285 1291 1297 1303 13
KEY BS RESOLVE_FINISH. . .  C ADDR   0575H   A      1324 1343 1346#
KEY CAPS . . . . . . . . .  C ADDR   029AH   A      612 649#
KEY ENTER. . . . . . . . .  C ADDR   02C2H   A      605 678#
KEY FUNC ACCEL . . . . . .  C ADDR   0650H   A      1541 1631#
KEY FUNC ACCEL BLUEKEY . .  C ADDR   066AH   A      1642 1655#
KEY FUNC ACCEL CHAR. . . .  C ADDR   0671H   A      1665#
KEY FUNC ACCEL FINISH. . .  C ADDR   066EH   A      1653 1659#
KEY FUNC ACCEL PINKKEY . .  C ADDR   0664H   A      1644 1650#
KEY FUNC ACCEL_RESTORE . .  C ADDR   0678H   A      1638 1646 1671#
KEY FUNC BLUE. . . . . . .  C ADDR   057BH   A      656 1375# 2343
KEY FUNC BLUEKEY . . . . .  C ADDR   06C1H   A      1713 1751#
KEY FUNC BS. . . . . . . .  C ADDR   04F4H   A      675 1240#
KEY FUNC BS 01 . . . . . .  C ADDR   04FCH   A      1242 1246#
KEY FUNC BS 02 . . . . . .  C ADDR   0504H   A      1248 1252#
KEY FUNC BS 03 . . . . . .  C ADDR   050CH   A      1254 1258#
KEY FUNC BS 04 . . . . . .  C ADDR   0514H   A      1260 1264#
KEY FUNC BS 05 . . . . . .  C ADDR   051CH   A      1266 1270#
KEY FUNC BS 06 . . . . . .  C ADDR   0524H   A      1272 1276#
KEY FUNC BS 07 . . . . . .  C ADDR   052CH   A      1278 1282#
KEY FUNC BS 08 . . . . . .  C ADDR   0534H   A      1284 1288#
KEY_FUNC_BS_09 . . . . . .  C ADDR   053CH   A      1290 1294#
```

```
KEY FUNC BS 10 . . . . .  C ADDR   0544H   A       1296 1300#
KEY FUNC BS FINISH . . .  C ADDR   054CH   A       1244 1250 1256 1262 1268 1274 1280 1286 1292 1298 1302 13
KEY FUNC CAPS. . . . . .  C ADDR   0576H   A       651 1361#
KEY FUNC CHAR. . . . . .  C ADDR   06CAH   A       1767#
KEY FUNC ENT . . . . . .  C ADDR   02D7H   A       680 708#
KEY FUNC ENT 01. . . . .  C ADDR   02E2H   A       709 714#
KEY FUNC ENT 02. . . . .  C ADDR   02EDH   A       715 720#
KEY FUNC ENT 03. . . . .  C ADDR   02F8H   A       721 726#
KEY FUNC ENT 04. . . . .  C ADDR   0302H   A       727 732#
KEY FUNC ENT 05. . . . .  C ADDR   030CH   A       733 738#
KEY FUNC ENT 06. . . . .  C ADDR   0317H   A       739 744#
KEY FUNC ENT 07. . . . .  C ADDR   0322H   A       745 750#
KEY FUNC ENT 08. . . . .  C ADDR   032DH   A       751 756#
KEY FUNC ENT 09. . . . .  C ADDR   0337H   A       757 762#
KEY FUNC ENT 10. . . . .  C ADDR   0342H   A       763 768#
KEY FUNC ENT 11. . . . .  C ADDR   034DH   A       769 774#
KEY FUNC ENT 12. . . . .  C ADDR   0358H   A       775 780#
KEY FUNC ENT 13. . . . .  C ADDR   0365H   A       781 787#
KEY_FUNC_ENT_FINISH. . .  C ADDR   036DH   A       712 718 724 730 736 742 748 754 760 766 772 778 785 788 7
```

```
KEY FUNC FINISH. . . . .  C ADDR   06C4H   A      1721 1727 1730 1736 1739 1746 1749 1755#
KEY FUNC GREEN . . . . .  C ADDR   0597H   A      666 1417#
KEY FUNC GREENKEY. . . .  C ADDR   06A7H   A      1717 1732#
KEY FUNC PINK. . . . . .  C ADDR   0589H   A      661 1397#
KEY FUNC PINKKEY . . . .  C ADDR   06B4H   A      1715 1742#
KEY FUNC PW. . . . . . .  C ADDR   067FH   A      1469 1475 1481 1499 1505 1511 1517 1523 1529 1535 1701#
KEY FUNC PW STATECK. . .  C ADDR   06C7H   A      1762#
KEY FUNC RED . . . . . .  C ADDR   05A5H   A      671 1437#
KEY FUNC REDKEY. . . . .  C ADDR   069AH   A      1719 1723#
KEY FUNC RESTORE . . . .  C ADDR   06DCH   A      1708 1772 1780#
KEY FUNC STAR. . . . . .  C ADDR   06D3H   A      1763 1774#
KEY GREEN. . . . . . . .  C ADDR   02AFH   A      633 664#
KEY GREEN LC . . . . . .  C ADDR   0707H   A      1738 1807#
KEY GREEN UPC. . . . . .  C ADDR   0713H   A      1735 1812#
KEY ISR. . . . . . . . .  C ADDR   024DH   A      303 583#
KEY KEYRELEASE . . . . .  C ADDR   02C9H   A      647 652 657 662 667 672 676 681 684#
KEY PINK . . . . . . . .  C ADDR   02A8H   A      626 659#
KEY PINK LC. . . . . . .  C ADDR   06EFH   A      1652 1748 1797#
KEY PINK UPC . . . . . .  C ADDR   06FBH   A      1745 1802#
KEY PW BAD 02H . . . . .  C ADDR   04A6H   A      1123 1125 1127 1129 1138#
KEY PW BAD 03H . . . . .  C ADDR   04C9H   A      1163 1165 1167 1169 1178#
KEY PW BAD 04H . . . . .  C ADDR   04ECH   A      1202 1204 1206 1208 1217#
KEY PW BAD 0AH . . . . .  C ADDR   0483H   A      1084 1086 1088 1090 1099#
KEY PW BAD 0CH . . . . .  C ADDR   0460H   A      1045 1047 1049 1051 1060#
KEY PW BAD 0DH . . . . .  C ADDR   043DH   A      1005 1007 1009 1011 1020#
KEY PW BAD 11H . . . . .  C ADDR   041AH   A      965 967 969 971 980#
KEY PW BAD 12H . . . . .  C ADDR   03F7H   A      925 927 929 931 940#
KEY PW BAD 13H . . . . .  C ADDR   03D4H   A      885 887 889 891 900#
KEY PW CHECK 02H . . . .  C ADDR   048BH   A      711 1121#
KEY PW CHECK 02H_FINISH.  C ADDR   04ADH   A      1135 1141 1144#
KEY PW CHECK 03H . . . .  C ADDR   04AEH   A      717 1161#
KEY PW CHECK 03H_FINISH.  C ADDR   04D0H   A      1175 1181 1184#
KEY PW CHECK 04H . . . .  C ADDR   04D1H   A      723 1200#
KEY PW CHECK 04H_FINISH.  C ADDR   04F3H   A      1214 1220 1223#
KEY PW CHECK 0AH . . . .  C ADDR   0468H   A      741 1082#
KEY PW CHECK 0AH_FINISH.  C ADDR   048AH   A      1096 1102 1105#
KEY PW CHECK 0CH . . . .  C ADDR   0445H   A      747 1043#
KEY PW CHECK 0CH_FINISH.  C ADDR   0467H   A      1057 1063 1066#
KEY PW CHECK 0DH . . . .  C ADDR   0422H   A      753 1003#
KEY PW CHECK 0DH_FINISH.  C ADDR   0444H   A      1017 1023 1026#
KEY PW CHECK 11H . . . .  C ADDR   03FFH   A      765 963#
KEY PW CHECK 11H_FINISH.  C ADDR   0421H   A      977 983 986#
KEY PW CHECK 12H . . . .  C ADDR   03DCH   A      771 923#
KEY PW CHECK 12H_FINISH.  C ADDR   03FEH   A      937 943 946#
KEY PW CHECK 13H . . . .  C ADDR   03B9H   A      777 883#
KEY PW CHECK 13H_FINISH.  C ADDR   03DBH   A      897 903 906#
KEY_PW_OK_02H. . . . . .  C ADDR   049FH   A      1132#
```

```
KEY PW OK 03H. . . . . .  C ADDR   04C2H   A      1172#
KEY PW OK 04H. . . . . .  C ADDR   04E5H   A      1211#
KEY PW OK 0AH. . . . . .  C ADDR   047CH   A      1093#
KEY PW OK 0CH. . . . . .  C ADDR   0459H   A      1054#
KEY PW OK 0DH. . . . . .  C ADDR   0436H   A      1014#
KEY PW OK 11H. . . . . .  C ADDR   0413H   A      974#
KEY PW OK 12H. . . . . .  C ADDR   03F0H   A      934#
KEY PW OK 13H. . . . . .  C ADDR   03CDH   A      894#
KEY READ . . . . . . . .  N NUMB   4000H   A      567# 594
KEY RED. . . . . . . . .  C ADDR   02B6H   A      640 669#
KEY RED LC . . . . . . .  C ADDR   071FH   A      1729 1817#
KEY RED UPC. . . . . . .  C ADDR   072BH   A      1726 1822#
KEY STATE07H MENU. . . .  C ADDR   0631H   A      1487 1581#
KEY STATE07H MENU 01 . .  C ADDR   063BH   A      1584 1589#
KEY STATE07H MENU 02 . .  C ADDR   0645H   A      1591 1596#
KEY STATE07H MENU FINISH  C ADDR   064FH   A      1587 1594 1598 1601 1604#
KEY STATE08H MENU. . . .  C ADDR   061CH   A      1493 1554#
KEY STATE08H MENU 01 . .  C ADDR   0626H   A      1557 1562#
KEY_STATE08H_MENU_FINISH  C ADDR   0630H   A      1560 1564 1567 1570#
```

```
KEY STATE CHK. . . . . .  C ADDR   05B3H   A      644 1465#
KEY STATE CHK 01 . . . .  C ADDR   05BBH   A      1468 1472#
KEY STATE CHK 02 . . . .  C ADDR   05C3H   A      1474 1478#
KEY STATE CHK 03 . . . .  C ADDR   05CBH   A      1480 1484#
KEY STATE CHK 04 . . . .  C ADDR   05D3H   A      1486 1490#
KEY STATE CHK 05 . . . .  C ADDR   05DBH   A      1492 1496#
KEY STATE CHK 06 . . . .  C ADDR   05E3H   A      1498 1502#
KEY STATE CHK 07 . . . .  C ADDR   05EBH   A      1504 1508#
KEY STATE CHK 08 . . . .  C ADDR   05F3H   A      1510 1514#
KEY STATE CHK 09 . . . .  C ADDR   05FBH   A      1516 1520#
KEY STATE CHK 10 . . . .  C ADDR   0603H   A      1522 1526#
KEY STATE CHK 11 . . . .  C ADDR   060BH   A      1528 1532#
KEY STATE CHK 12 . . . .  C ADDR   0613H   A      1534 1538#
KEY STATE CHK FINISH . .  C ADDR   061BH   A      1470 1476 1482 1488 1494 1500 1506 1512 1518 1524 1530 15
LCD ACCSTPT. . . . . . .  C ADDR   01D3H   A      524# 2967
LCD ARMDIS . . . . . . .  C ADDR   018FH   A      513# 2699
LCD_BUSY . . . . . . . .  C ADDR   009EH   A      371 376 381 407 463 470# 473 1329 1334 1339 1669 1771 177
                                                 2300 2349 2394 2440 2475 2520 2537 2554 2566 2571 2576 258
                                                 2652 2669 2711 2728 2764 2808 2837 2842 2890 2934 2980 299
                                                 3079 3123 3160 3204 3263 3268 3411
LCD CHANGED PW . . . . .  C ADDR   023CH   A      542# 3270
LCD_CLEAR. . . . . . . .  C ADDR   0093H   A      383 459# 2283 2307 2382 2428 2467 2506 2621 2697 2759 279
                                                 3032 3069 3113 3148 3192 3234 3258
LCD_CMD. . . . . . . . .  N NUMB   1200H   A      360# 368 373 378 460 1326 1336 2230 2247 2264 2276 2297 2
                                                 2481 2517 2534 2551 2563 2568 2632 2649 2666 2708 2725 276
                                                 2931 2977 2994 3007 3034 3039 3071 3076 3120 3157 3201 326
LCD CURRENT PW . . . . .  C ADDR   021FH   A      536# 3115
LCD FIRST. . . . . . . .  C ADDR   00A6H   A      482# 2221
LCD HOME . . . . . . . .  C ADDR   011BH   A      501# 2508
LCD INIT . . . . . . . .  C ADDR   0030H   A      367# 4017
LCD INVALID STPT . . . .  C ADDR   020EH   A      532# 3081
LCD MAIN MENU. . . . . .  C ADDR   0153H   A      507# 2623
LCD NEW PW . . . . . . .  C ADDR   022EH   A      539# 3236
LCD PASSWORD_ENTRY . . .  C ADDR   00ECH   A      491# 2336 2800
LCD_PRINT. . . . . . . .  C ADDR   0052H   A      397# 417 2223 2240 2257 2274 2304 2338 2386 2401 2432 244
                                                 2561 2625 2642 2659 2676 2701 2718 2735 2768 2802 2846 288
                                                 2987 3004 3046 3083 3117 3152 3167 3196 3211 3238 3272 341
LCD PRO TEX. . . . . . .  C ADDR   00DFH   A      488# 2302
LCD PW BAD . . . . . . .  C ADDR   00F6H   A      494# 2384 2430 2880 2924 3150 3194
LCD READ . . . . . . . .  N NUMB   1100H   A      359# 471
LCD RESTORE. . . . . . .  C ADDR   0066H   A      409#
LCD RETURN . . . . . . .  C ADDR   0073H   A      404 419#
LCD SYSARMED . . . . . .  C ADDR   01B5H   A      518# 2766
LCD SYSDISARMED. . . . .  C ADDR   01C2H   A      521# 2844
LCD SYSLOCKED. . . . . .  C ADDR   010CH   A      498# 2477
LCD TIMER0 OV. . . . . .  C ADDR   0089H   A      446# 447 449
LCD_VALID_STPT . . . . .  C ADDR   01FDH   A      529# 3044
```

```
LCD WAIT 3SEC. . . . . .  C ADDR   007FH    A       440# 2282 2306 2770 2848 3048 3085 3274
LCD WRITE. . . . . . . .  N NUMB   1000H    A       358# 405 1331 1667 1769 1775 2573 2578 2583 2588 2593
MAIN . . . . . . . . . .  C ADDR   16D4H    A       300 4012#
MODF . . . . . . . . . .  B ADDR   00F8H.5  A       266#
MSTEN. . . . . . . . . .  B ADDR   00F8H.1  A       262#
OSCICN . . . . . . . . .  D ADDR   00B2H    A       97# 2123
OSCILLATOR_INIT. . . . .  C ADDR   07DFH    A       2113# 2140
OSCXCN . . . . . . . . .  D ADDR   00B1H    A       96# 2114 2121
OSC WAIT1. . . . . . . .  C ADDR   07E4H    A       2116# 2119
OSC WAIT2. . . . . . . .  C ADDR   07EAH    A       2120# 2122
OV . . . . . . . . . . .  B ADDR   00D0H.2  A       234#
P. . . . . . . . . . . .  B ADDR   00D0H.0  A       232#
P0 . . . . . . . . . . .  D ADDR   0080H    A       49# 685
P0MDOUT. . . . . . . . .  D ADDR   00A4H    A       83# 2104
P1 . . . . . . . . . . .  D ADDR   0090H    A       65# 1363 1381 1382 1383 1384 1403 1404 1405 1406 1423 142
                                                   1445 1446 2757 2828 2829 2830 3886 3898 3899 3905 4020
P1MDIN . . . . . . . . .  D ADDR   00BDH    A       106#
P1MDOUT. . . . . . . . .  D ADDR   00A5H    A       84#
P2 . . . . . . . . . . .  D ADDR   00A0H    A       80#
```

```
P2MDOUT. . . . . . . . . .  D ADDR   00A6H   A      85# 2105
P3 . . . . . . . . . . . .  D ADDR   00B0H   A      95# 2171 2172 3885 3893 3894
P3IF . . . . . . . . . . .  D ADDR   00ADH   A      92#
P3MDOUT. . . . . . . . . .  D ADDR   00A7H   A      86# 2106
P4 . . . . . . . . . . . .  D ADDR   0084H   A      53#
P5 . . . . . . . . . . . .  D ADDR   0085H   A      54#
P6 . . . . . . . . . . . .  D ADDR   0086H   A      55#
P7 . . . . . . . . . . . .  D ADDR   0096H   A      71#
P74OUT . . . . . . . . . .  D ADDR   00B5H   A      98# 2107
PCA0CN . . . . . . . . . .  D ADDR   00D8H   A      132# 242 243 244 245 246 247 248
PCA0CPH0 . . . . . . . . .  D ADDR   00FAH   A      165#
PCA0CPH1 . . . . . . . . .  D ADDR   00FBH   A      166#
PCA0CPH2 . . . . . . . . .  D ADDR   00FCH   A      167#
PCA0CPH3 . . . . . . . . .  D ADDR   00FDH   A      168#
PCA0CPH4 . . . . . . . . .  D ADDR   00FEH   A      169#
PCA0CPL0 . . . . . . . . .  D ADDR   00EAH   A      149#
PCA0CPL1 . . . . . . . . .  D ADDR   00EBH   A      150#
PCA0CPL2 . . . . . . . . .  D ADDR   00ECH   A      151#
PCA0CPL3 . . . . . . . . .  D ADDR   00EDH   A      152#
PCA0CPL4 . . . . . . . . .  D ADDR   00EEH   A      153#
PCA0CPM0 . . . . . . . . .  D ADDR   00DAH   A      134#
PCA0CPM1 . . . . . . . . .  D ADDR   00DBH   A      135#
PCA0CPM2 . . . . . . . . .  D ADDR   00DCH   A      136#
PCA0CPM3 . . . . . . . . .  D ADDR   00DDH   A      137#
PCA0CPM4 . . . . . . . . .  D ADDR   00DEH   A      138#
PCA0H. . . . . . . . . . .  D ADDR   00F9H   A      164#
PCA0L. . . . . . . . . . .  D ADDR   00E9H   A      148#
PCA0MD . . . . . . . . . .  D ADDR   00D9H   A      133#
PCON . . . . . . . . . . .  D ADDR   0087H   A      56# 2057
PORT_IO INIT . . . . . . .  C ADDR   07CCH   A      2067# 2139
PS . . . . . . . . . . . .  B ADDR   00B8H.4 A      209#
PSCTL. . . . . . . . . . .  D ADDR   008FH   A      64#
PSW. . . . . . . . . . . .  D ADDR   00D0H   A      124# 232 233 234 235 236 237 238 239 398 414 424 585 691
                                                   3357 3425 3878 3916
PT0. . . . . . . . . . . .  B ADDR   00B8H.1 A      206#
PT1. . . . . . . . . . . .  B ADDR   00B8H.3 A      208#
PT2. . . . . . . . . . . .  B ADDR   00B8H.5 A      210#
PX0. . . . . . . . . . . .  B ADDR   00B8H.0 A      205#
PX1. . . . . . . . . . . .  B ADDR   00B8H.2 A      207#
RAM INIT . . . . . . . . .  C ADDR   0786H   A      1985# 4018
RAM RDWR . . . . . . . . .  N NUMB   2000H   A      1857# 1870 1905 1986
RAM READ PW. . . . . . . .  C ADDR   0737H   A      710 716 722 740 746 752 764 770 776 1868#
RAM WRITE ADC. . . . . . .  C ADDR   075FH   A      1940# 3394
RAM WRITE ADC RETURN . . .  C ADDR   0781H   A      1948 1966#
RAM WRITE LOOP . . . . . .  C ADDR   0766H   A      1945# 1964
RAM WRITE PW . . . . . . .  C ADDR   074BH   A      782 1903#
RB8. . . . . . . . . . . .  B ADDR   0098H.2 A      188#
```

```
RCAP2H . . . . . . . . . .  D ADDR   00CBH   A      120#
RCAP2L . . . . . . . . . .  D ADDR   00CAH   A      119#
RCAP4H . . . . . . . . . .  D ADDR   00E5H   A      144# 2053
RCAP4L . . . . . . . . . .  D ADDR   00E4H   A      143# 2052
RCLK . . . . . . . . . . .  B ADDR   00C8H.5 A      227#
REF0CN . . . . . . . . . .  D ADDR   00D1H   A      125#
REN. . . . . . . . . . . .  B ADDR   0098H.4 A      190#
RESET_SOURCES_INIT . . . .  C ADDR   07A7H   A      2043# 2135
RI . . . . . . . . . . . .  B ADDR   0098H.0 A      186#
RS0. . . . . . . . . . . .  B ADDR   00D0H.3 A      235#
RS1. . . . . . . . . . . .  B ADDR   00D0H.4 A      236#
RSTSRC . . . . . . . . . .  D ADDR   00EFH   A      154#
RXOVRN . . . . . . . . . .  B ADDR   00F8H.4 A      265#
SADDR0 . . . . . . . . . .  D ADDR   00A9H   A      88#
SADDR1 . . . . . . . . . .  D ADDR   00F3H   A      158#
SADEN0 . . . . . . . . . .  D ADDR   00B9H   A      102#
SADEN1 . . . . . . . . . .  D ADDR   00AEH   A      93#
SBUF0. . . . . . . . . . .  D ADDR   0099H   A      73#
SBUF1. . . . . . . . . . .  D ADDR   00F2H   A      157# 3543 3558 3934 3958 3982
```

```
SCON0. . . . . . . . . .  D ADDR   0098H   A      72# 186 187 188 189 190 191 192 193
SCON1. . . . . . . . . .  D ADDR   00F1H   A      156# 2058 3545 3547 3559 3561 3936 3938 3960 3962 3984 39
SI . . . . . . . . . . .  B ADDR   00C0H.3 A      216#
SLVSEL . . . . . . . . .  B ADDR   00F8H.2 A      263#
SM0. . . . . . . . . . .  B ADDR   0098H.7 A      193#
SM1. . . . . . . . . . .  B ADDR   0098H.6 A      192#
SM2. . . . . . . . . . .  B ADDR   0098H.5 A      191#
SMB0ADR. . . . . . . . .  D ADDR   00C3H   A      112#
SMB0CN . . . . . . . . .  D ADDR   00C0H   A      109# 213 214 215 216 217 218 219
SMB0CR . . . . . . . . .  D ADDR   00CFH   A      123#
SMB0DAT. . . . . . . . .  D ADDR   00C2H   A      111#
SMB0STA. . . . . . . . .  D ADDR   00C1H   A      110#
SMBFTE . . . . . . . . .  B ADDR   00C0H.1 A      214#
SMBTOE . . . . . . . . .  B ADDR   00C0H.0 A      213#
SP . . . . . . . . . . .  D ADDR   0081H   A      50# 4013
SPI0CFG. . . . . . . . .  D ADDR   009AH   A      74#
SPI0CKR. . . . . . . . .  D ADDR   009DH   A      77#
SPI0CN . . . . . . . . .  D ADDR   00F8H   A      163# 261 262 263 264 265 266 267 268
SPI0DAT. . . . . . . . .  D ADDR   009BH   A      75#
SPIEN. . . . . . . . . .  B ADDR   00F8H.0 A      261#
SPIF . . . . . . . . . .  B ADDR   00F8H.7 A      268#
STA. . . . . . . . . . .  B ADDR   00C0H.5 A      218#
STATE 00 . . . . . . . .  C ADDR   083CH   A      2181 2213#
STATE 01 . . . . . . . .  C ADDR   08A6H   A      2182 2295#
STATE 02 . . . . . . . .  C ADDR   08C3H   A      2183 2322#
STATE 03 . . . . . . . .  C ADDR   08F2H   A      2184 2373#
STATE 04 . . . . . . . .  C ADDR   0926H   A      2185 2419#
STATE 05 . . . . . . . .  C ADDR   095AH   A      2186 2465#
STATE 06 . . . . . . . .  C ADDR   097EH   A      2187 2499#
STATE 07 . . . . . . . .  C ADDR   0A23H   A      2188 2617#
STATE 08 . . . . . . . .  C ADDR   0A7FH   A      2189 2693#
STATE 09 . . . . . . . .  C ADDR   0AC3H   A      2190 2753#
STATE 0A . . . . . . . .  C ADDR   0AEBH   A      2191 2789#
STATE 0B . . . . . . . .  C ADDR   0B0FH   A      2192 2823#
STATE 0C . . . . . . . .  C ADDR   0B48H   A      2193 2869#
STATE 0D . . . . . . . .  C ADDR   0B7AH   A      2194 2913#
STATE 0E . . . . . . . .  C ADDR   0BACH   A      2195 2959#
STATE 0F . . . . . . . .  C ADDR   0BFEH   A      2196 3028#
STATE 10 . . . . . . . .  C ADDR   0C2BH   A      2197 3065#
STATE 11 . . . . . . . .  C ADDR   0C58H   A      2198 3104#
STATE 12 . . . . . . . .  C ADDR   0C7CH   A      2199 3139#
STATE 13 . . . . . . . .  C ADDR   0CAEH   A      2200 3183#
STATE 14 . . . . . . . .  C ADDR   0CE0H   A      2201 3225#
STATE 15 . . . . . . . .  C ADDR   0CFAH   A      2202 3254#
STATE_LOOKUP . . . . . .  C ADDR   0D27H   A      729 735 759 784 856 862 896 902 936 942 976 982 1016 1022
                                                 1134 1140 1174 1180 1213 1219 1559 1566 1586 1593 1600 277
                                                 3292# 4036 4042 4049
```

```
STATE TABLE. . . . . . .  C ADDR   0810H   A      2180# 3293
STO. . . . . . . . . . .  B ADDR   00C0H.4 A      217#
T2CON. . . . . . . . . .  D ADDR   00C8H   A      117# 222 223 224 225 226 227 228 229
T4CON. . . . . . . . . .  D ADDR   00C9H   A      118# 2051
TB8. . . . . . . . . . .  B ADDR   0098H.3 A      189#
TCLK . . . . . . . . . .  B ADDR   00C8H.4 A      226#
TCON . . . . . . . . . .  D ADDR   0088H   A      57# 176 177 178 179 180 181 182 183
TF0. . . . . . . . . . .  B ADDR   0088H.5 A      181# 447 448
TF1. . . . . . . . . . .  B ADDR   0088H.7 A      183#
TF2. . . . . . . . . . .  B ADDR   00C8H.7 A      229#
TH0. . . . . . . . . . .  D ADDR   008CH   A      61# 442
TH1. . . . . . . . . . .  D ADDR   008DH   A      62#
TH2. . . . . . . . . . .  D ADDR   00CDH   A      122#
TH4. . . . . . . . . . .  D ADDR   00F5H   A      160#
TI . . . . . . . . . . .  B ADDR   0098H.1 A      187#
TIMER INIT . . . . . . .  C ADDR   07AEH   A      2048# 2136
TL0. . . . . . . . . . .  D ADDR   008AH   A      59# 443
TL1. . . . . . . . . . .  D ADDR   008BH   A      60#
TL2. . . . . . . . . . .  D ADDR   00CCH   A      121#
```

```
TL4. . . . . . . . . . . .  D ADDR   00F4H   A      159#
TMOD . . . . . . . . . . .  D ADDR   0089H   A      58# 2050
TMR3CN . . . . . . . . . .  D ADDR   0091H   A      66#
TMR3H. . . . . . . . . . .  D ADDR   0095H   A      70#
TMR3L. . . . . . . . . . .  D ADDR   0094H   A      69#
TMR3RLH. . . . . . . . . .  D ADDR   0093H   A      68#
TMR3RLL. . . . . . . . . .  D ADDR   0092H   A      67#
TR0. . . . . . . . . . . .  B ADDR   0088H.4 A      180# 444 450
TR1. . . . . . . . . . . .  B ADDR   0088H.6 A      182# 2357 3876 3918
TR2. . . . . . . . . . . .  B ADDR   00C8H.2 A      224#
TXBSY. . . . . . . . . . .  B ADDR   00F8H.3 A      264#
UART_INIT. . . . . . . . .  C ADDR   07BEH   A      2056# 2137
WCOL . . . . . . . . . . .  B ADDR   00F8H.6 A      267#
WDTCN. . . . . . . . . . .  D ADDR   00FFH   A      170# 2044 2045
XBR0 . . . . . . . . . . .  D ADDR   00E1H   A      140#
XBR1 . . . . . . . . . . .  D ADDR   00E2H   A      141# 2108
XBR2 . . . . . . . . . . .  D ADDR   00E3H   A      142# 2109


REGISTER BANK(S) USED: 0


ASSEMBLY COMPLETE.  0 WARNING(S), 0 ERROR(S)
```