```
MACRO ASSEMBLER A51 V8.02c
OBJECT MODULE PLACED IN Project.OBJ
ASSEMBLER INVOKED BY: C:\SiLabs\MCU\IDEfiles\C51\BIN\a51.exe Project.asm XR GEN DB EP NOMOD51 INCDIR(C:\SiLabs\MCU\Inc)

LOC  OBJ            LINE     SOURCE

                       1     $nomod51
                       2     ;*****************************************************************************
                       3     ;  Project
                       4     ;
                       5     ;  YOUR NAME   :  Austin Atteberry
                       6     ;  FILE NAME   :  Project.asm
                       7     ;  DATE        :  11/28/2017
                       8     ;  TARGET MCU  :  C8051F340
                       9     ;  DESCRIPTION :  This program generates a random number, which is used to
                      10     ;              :  control actuators connected to Port 0. The speed of the
                      11     ;              :  actuators is controlled by a keypad connected to Port 3.
                      12     ;              :  The speed is displayed on an LCD display connected to Port
                      13     ;              :  1.
                      14     ;
                      15     ;       NOTES:
                      16     ;
                      17     ;*****************************************************************************
                      18
                     281     $list
                     282
                     283
                     284     ;*****************************************************************************
                     285     ;
                     286     ; EQUATES
                     287     ;
                     288     ;*****************************************************************************
                     289
  0094               290     ENABLE           equ  P1.4                ; Enable signal to LCD
  0092               291     RW               equ  P1.2                ; R/W signal to LCD.
  0093               292     RS               equ  P1.3                ; RS signal to LCD
  00A0               293     LCD              equ  P2                  ; Output port to LCD.
                     294
  00B0               295     keyport          equ P3                   ; Keypad port connected here
  00B0               296     row1             equ P3.0                 ; Row 1 (pin 1)
  00B1               297     row2             equ P3.1                 ; Row 2 (pin 2)
  00B2               298     row3             equ P3.2                 ; Row 3 (pin 3)
  00B3               299     row4             equ P3.3                 ; Row 4 (pin 4)
                     300
  00B4               301     col1             equ P3.4                 ; Column 1 (pin 5)
  00B5               302     col2             equ P3.5                 ; Column 2 (pin 6)
  00B6               303     col3             equ P3.6                 ; Column 3 (pin 7)
  00B7               304     col4             equ P3.7                 ; Column 4 (pin 8)
                     305
                     306
                     307     ;*****************************************************************************
                     308     ;
                     309     ; RESET and INTERRUPT VECTORS
                     310     ;
                     311     ;*****************************************************************************
                     312
                     313              ; Reset Vector
0000                 314                      org 0000H
0000 020003          315                      ljmp Main                ; Locate a jump to the start of
                     316                                               ; code at the reset vector.
                     317
                     318
                     319     ;*****************************************************************************
                     320     ;
                     321     ; MAIN CODE
                     322     ;
                     323     ;*****************************************************************************
                     324
0003                 325     Main:
                     326              ; Disable the WDT.
0003 53D9BF          327                      anl PCA0MD,#NOT(040h)    ; Clear Watchdog Enable bit
                     328
                     329              ; Enable the Port I/O Crossbar
0006 75A6FF          330                      mov P2MDOUT, #0FFH       ; Make P2 output push-pull
0009 75A5FF          331                      mov P1MDOUT, #0FFH       ; Make P1 output push-pull
000C 75F2FF          332                      mov P1MDIN, #0FFH        ; Make port pins input mode digital
000F 75A4FF          333                      mov P0MDOUT, #0FFH       ; Male P0 output push-pull
0012 75A70F          334                      mov P3MDOUT, #0FH        ; Make P3 low nibble output push-pull
0015 75E240          335                      mov XBR1, #40H           ; Enable Crossbar
                     336
0018 758000          337                      mov P0, #0               ; Set Port 0 low
001B 7800            338                      mov R0, #0               ; Initialize R0
001D 791F            339                      mov R1, #1FH             ; Initialize R1
001F 7A00            340                      mov R2, #0               ; Clear mode select (R2)
0021 7B00            341                      mov R3, #0               ; Clears LCD position counter
0023 7C00            342                      mov R4, #0               ; Set initial speed to 1
0025 3112            343                      call Init                ; LCD Initialization proceedure
0027 315E            344                      call Clear               ; Clear LCD Display
0029 316A            345                      call DisplayIntro        ; Call DisplayIntro subroutine
002B 5130            346                      call AutoDisplay         ; Call AutoDisplay subroutine
002D 515E            347                      call DisplaySpeed        ; Call DisplaySpeed subroutine
002F 119F            348                      call Autoroutine         ; Call Autoroutine subroutine
                     349
0031 7800            350     Start:           mov R0, #0               ; clear R0 - the first key is key0
```

```
                    351
                    352             ; scan row1
0033 D2B3           353                     setb row4               ; set row4
0035 C2B0           354                     clr row1                ; clear row1
0037 1177           355                     call colScan            ; call column-scan subroutine
0039 20D526         356                     jb F0, finish           ; if F0 is set, jump to end of program
                    357
                    358             ; scan row2
003C D2B0           359                     setb row1               ; set row1
003E C2B1           360                     clr row2                ; clear row2
0040 1177           361                     call colScan            ; call column-scan subroutine
0042 20D51D         362                     jb F0, finish           ; if F0 is set, jump to end of program
                    363
                    364             ; scan row3
0045 D2B1           365                     setb row2               ; set row2
0047 C2B2           366                     clr row3                ; clear row3
0049 1177           367                     call colScan            ; call column-scan subroutine
004B 20D514         368                     jb F0, finish           ; if F0 is set, jump to end of program
                    369
                    370             ; scan row4
004E D2B2           371                     setb row3               ; set row3
0050 C2B3           372                     clr row4                ; clear row4
0052 1177           373                     call colScan            ; call column-scan subroutine
0054 20D50B         374                     jb F0, finish           ; if F0 is set, jump to end of program
                    375
0057 BA001B         376                     cjne R2, #0H, stagain   ; Jump to stagain if not in Auto mode
005A BD0010         377                     cjne R5, #0H, sdfasfda  ; Jump to sdfasfda if not 0
005D BE0010         378                     cjne R6, #0H, poaefbef  ; Jump to poaefbef if not 0
0060 119F           379                     call Autoroutine        ; Call Autoroutine subroutine
                    380
0062 9002B6         381     finish:         mov DPTR, #Table1       ; Initialize Data Pointer
0065 E8             382                     mov A, R0               ; move keynumber to acc
0066 93             383                     movc A, @A + DPTR       ; Get key character
0067 3189           384                     call control            ; Call control subroutine
0069 C2D5           385                     clr F0                  ; clear flag
006B 80C4           386                     jmp start               ; Continue looking for next key
                    387
006D 1D             388     sdfasfda:       dec R5                  ; Decrement R5
006E 80C1           389                     jmp start               ; Continue looking for next key
                    390
0070 7DFF           391     poaefbef:       mov R5, #0FFH           ; Reset R5
0072 1E             392                     dec R6                  ; Decrement R6
0073 80BC           393                     jmp start               ; Continue looking for next key
                    394
0075 80BA           395     stagain:        jmp start               ; Continue looking for next key
                    396
                    397
                    398     ;*************************************************************************
                    399     ;
                    400     ;  colScan subroutine
                    401     ;
                    402     ;  The subroutine scans columns. It is called during each scan row event.
                    403     ;  If a key in the current row being scaned has been pressed, the subroutine
                    404     ;  will determine which column. when a key if found to be pressed, the
                    405     ;  subroutine waits until the key has been released before continuing. This
                    406     ;  method debounces the input keys.
                    407     ;
                    408     ;  GLOBAL REGESTERS USED: R0
                    409     ;  GLOBAL BITS USED: F0(PSW.5)
                    410     ;  INPUT: col1(P3.4), col2(P3.5), col3(P3.6), col4(P3.7)
                    411     ;  OUTPUT: R0, F0
                    412     ;
                    413     ;*************************************************************************
                    414
0077 20B405         415     colScan:        jb col1, nextcol        ; check if col1 key is pressed
007A 30B4FD         416                     jnb col1, $             ; If yes, then wait for key release
007D 801D           417                     jmp gotkey              ; Have key, return
007F 08             418     nextcol:        inc R0                  ; Increment keyvalue
0080 20B505         419                     jb col2, nextcol2       ; check if col2 key is pressed
0083 30B5FD         420                     jnb col2, $             ; If yes, then wait for key release
0086 8014           421                     jmp gotkey              ; Have key, return
0088 08             422     nextcol2:       inc R0                  ; Increment keyvalue
0089 20B605         423                     jb col3, nextcol3       ; check if col3 key is pressed
008C 30B6FD         424                     jnb col3, $             ; If yes, then wait for key release
008F 800B           425                     jmp gotkey              ; Have key, return
0091 08             426     nextcol3:       inc R0                  ; Increment keyvalue
0092 20B705         427                     jb col4, nokey          ; check if col4 key is pressed
0095 30B7FD         428                     jnb col4, $             ; If yes, then wait for key release
0098 8002           429                     jmp gotkey              ; Have key, return
009A 08             430     nokey:          inc R0                  ; Increment keyvalue
009B 22             431                     ret                     ; finished scan, no key pressed
009C D2D5           432     gotKey:         setb F0                 ; key found - set F0
009E 22             433                     ret                     ; and return from subroutine
                    434
                    435
                    436     ;*************************************************************************
                    437     ;
                    438     ;  Autoroutine subroutine
                    439     ;
                    440     ;  This subroutine sets pins 1-3 on P0.
                    441     ;
                    442     ;  LOCAL REGISTERS USED: none
                    443     ;  INPUT: none
                    444     ;  OUTPUT: P0
                    445     ;
```

```
                    446     ;*******************************************************************************
                    447
009F 11AA           448     Autoroutine:    call Random             ; Call random subroutine
00A1 9002D0         449                     mov DPTR, #Table4       ; Initialize Data Pointer
00A4 93             450                     movc A, @A + DPTR       ; Get port configuration
00A5 F580           451                     mov P0, A               ; Set port output
00A7 11C3           452                     call Speed              ; Call Speed subroutine
00A9 22             453                     ret                     ; Return
                    454
                    455
                    456     ;*******************************************************************************
                    457     ;
                    458     ;  Random subroutine
                    459     ;
                    460     ;  This subroutine generates a pseudorandom 2-bit number by multiplying R1
                    461     ;  with a prime seed and zeroing out the most significant six bits.
                    462     ;
                    463     ;  LOCAL REGISTERS USED: none
                    464     ;  INPUT: none
                    465     ;  OUTPUT: ACC
                    466     ;
                    467     ;*******************************************************************************
                    468
00AA E9             469     Random:         mov A, R1               ; Assign R1 to A
00AB 7002           470                     jnz random1             ; Jump if not zero
00AD F4             471                     cpl A                   ; Complement A
00AE F9             472                     mov R1, A               ; Assign A to R1
00AF 54B8           473     random1:        anl a, #0B8H            ; And A with 184
00B1 A2D0           474                     mov C, P                ; Move parity bit into C
00B3 E9             475                     mov A, R1               ; Assign R1 to A
00B4 33             476                     rlc A                   ; Rotate A left
00B5 F9             477                     mov R1, A               ; Assign A to R1
00B6 C2E7           478                     clr ACC.7               ; Zero first bit of A
00B8 C2E6           479                     clr ACC.6               ; Zero second bit of A
00BA C2E5           480                     clr ACC.5               ; Zero third bit of A
00BC C2E4           481                     clr ACC.4               ; Zero fourth bit of A
00BE C2E3           482                     clr ACC.3               ; Zero fifth bit of A
00C0 C2E2           483                     clr ACC.2               ; Zero sixth bit of A
00C2 22             484                     ret                     ; Return
                    485
                    486
                    487     ;*******************************************************************************
                    488     ;
                    489     ;  Speed subroutine
                    490     ;
                    491     ;  This subroutine calls the delay function a predetermined number of times
                    492     ;  based on the speed setting.
                    493     ;
                    494     ;  LOCAL REGISTERS USED: none
                    495     ;  INPUT: none
                    496     ;  OUTPUT: none
                    497     ;
                    498     ;*******************************************************************************
                    499
00C3 BC0906         500     Speed:          cjne R4, #9, speed8     ; Jump if (R4 != 9)
00C6 7E1F           501                     mov   R6, #01FH         ; Set register 6
00C8 7D00           502                     mov   R5, #00H          ; Set register 5
00CA 8045           503                     jmp repeat              ; Jump to repeat
00CC BC0806         504     speed8:         cjne R4, #8, speed7     ; Jump if (R4 != 8)
00CF 7E1F           505                     mov   R6, #01FH         ; Set register 6
00D1 7D07           506                     mov   R5, #07H          ; Set register 5
00D3 803C           507                     jmp repeat              ; Jump to repeat
00D5 BC0706         508     speed7:         cjne R4, #7, speed6     ; Jump if (R4 != 7)
00D8 7E1F           509                     mov   R6, #01FH         ; Set register 6
00DA 7D0F           510                     mov   R5, #0FH          ; Set register 5
00DC 8033           511                     jmp repeat              ; Jump to repeat
00DE BC0606         512     speed6:         cjne R4, #6, speed5     ; Jump if (R4 != 6)
00E1 7E2F           513                     mov   R6, #02FH         ; Set register 6
00E3 7D17           514                     mov   R5, #17H          ; Set register 5
00E5 802A           515                     jmp repeat              ; Jump to repeat
00E7 BC0506         516     speed5:         cjne R4, #5, speed4     ; Jump if (R4 != 5)
00EA 7E2F           517                     mov   R6, #02FH         ; Set register 6
00EC 7D1F           518                     mov   R5, #1FH          ; Set register 5
00EE 8021           519                     jmp repeat              ; Jump to repeat
00F0 BC0406         520     speed4:         cjne R4, #4, speed3     ; Jump if (R4 != 4)
00F3 7E2F           521                     mov   R6, #02FH         ; Set register 6
00F5 7D27           522                     mov   R5, #27H          ; Set register 5
00F7 8018           523                     jmp repeat              ; Jump to repeat
00F9 BC0306         524     speed3:         cjne R4, #3, speed2     ; Jump if (R4 != 3)
00FC 7E3F           525                     mov   R6, #03FH         ; Set register 6
00FE 7D2F           526                     mov   R5, #2FH          ; Set register 5
0100 800F           527                     jmp repeat              ; Jump to repeat
0102 BC0206         528     speed2:         cjne R4, #2, speed1     ; Jump if (R4 != 2)
0105 7E3F           529                     mov   R6, #03FH         ; Set register 6
0107 7D37           530                     mov   R5, #37H          ; Set register 5
0109 8006           531                     jmp repeat              ; Jump to repeat
010B 7E3F           532     speed1:         mov   R6, #03FH         ; Set register 6
010D 7D3F           533                     mov   R5, #3FH          ; Set register 5
010F 8000           534                     jmp repeat              ; Jump to repeat
0111 22             535     repeat:         ret                     ; Return
                    536
                    537
                    538     ;*******************************************************************************
                    539     ;
                    540     ;  init subroutine
```

```
                    541     ;
                    542     ;  The subroutine is used initialize the LCD during startup.
                    543     ;
                    544     ;  LOCAL REGISTERS USED: None
                    545     ;  INPUT:
                    546     ;  OUTPUT: LCD (P2), ENABLE (P1.4)
                    547     ;
                    548     ;*******************************************************************************
                    549
0112 C293           550     init:           clr RS                  ; Register Select
0114 C292           551                     clr RW                  ; Read/Write ( 1 = Read  ; 0 = Write )
0116 C294           552                     clr ENABLE              ; High to Low Transition Stores the data
0118 517F           553                     call delay              ; Waits for LCD to stabilize
011A 5188           554                     call reset              ; Sends reset bytes to LCD
011C 75A038         555                     mov LCD, #38H           ; Function Set Word
011F 51AC           556                     call Busy               ; Check Busy Flag
0121 D294           557                     setb ENABLE             ; Latched the first byte.
0123 517F           558                     call delay              ; Waits.
0125 C294           559                     clr ENABLE              ; Then resets latch.
0127 51AC           560                     call busy               ; Check Busy Flag
0129 75A008         561                     mov LCD, #08H           ; Display Off word
012C 51AC           562                     call Busy               ; Check Busy Flag
012E D294           563                     setb ENABLE             ; Latched the first byte.
0130 517F           564                     call delay              ; Waits.
0132 C294           565                     clr ENABLE              ; Then resets latch.
0134 51AC           566                     call Busy               ; Check Busy Flag
0136 75A00F         567                     mov LCD, #0FH           ; Display On word.
0139 51AC           568                     call Busy               ; Check Busy Flag
013B D294           569                     setb ENABLE             ; Latched the first byte.
013D 517F           570                     call delay              ; Waits.
013F C294           571                     clr ENABLE              ; Then resets latch
0141 51AC           572                     call Busy               ; Check Busy Flag
0143 75A006         573                     mov LCD, #06H           ; Entry Mode word
0146 51AC           574                     call Busy               ; Check Busy Flag
0148 D294           575                     setb ENABLE             ; Latched the first byte.
014A 517F           576                     call delay              ; Waits.
014C C294           577                     clr ENABLE              ; Then resets latch.
014E 51AC           578                     call Busy               ; Check Busy Flag
0150 75A002         579                     mov LCD, #02H           ; Display Home word
0153 51AC           580                     call Busy               ; Check Busy Flag
0155 D294           581                     setb ENABLE             ; Latched the first byte.
0157 517F           582                     call delay              ; Waits.
0159 C294           583                     clr ENABLE              ; Then resets latch.
015B 51AC           584                     call Busy               ; Check Busy Flag
015D 22             585                     ret                     ; Return
                    586
                    587
                    588     ;*******************************************************************************
                    589     ;
                    590     ;  clear subroutine
                    591     ;
                    592     ;  Clears the LCD.
                    593     ;  Used one 8-bit data move to send the Clear Display Instruction command
                    594     ;  (01H) to the LCD.
                    595     ;
                    596     ;  The subroutine is used during initialization and when the display is full
                    597     ;  to clear the display before it wraps back to DDRAM address 00.
                    598     ;
                    599     ;  INPUT: none
                    600     ;  OUTPUT: Port 2 (LCD) and P1.4 (ENABLE)
                    601     ;
                    602     ;*******************************************************************************
                    603
015E 75A001         604     clear:          mov LCD, #01H           ; Clear Display word
0161 51AC           605                     call Busy               ; Check Busy Flag
0163 D294           606                     setb ENABLE             ; Latched the first byte.
0165 517F           607                     call delay              ; Waits.
0167 C294           608                     clr ENABLE              ; Then resets latch
0169 22             609                     ret                     ; Return
                    610
                    611
                    612     ;*******************************************************************************
                    613     ;
                    614     ;  DisplayIntro subroutine
                    615     ;
                    616     ;  Displays the message "Cat Toy" on the LCD screen for 5 seconds after
                    617     ;  the device powers up.
                    618     ;
                    619     ;  LOCAL REGISTERS USED: R3
                    620     ;  INPUT: none
                    621     ;  OUTPUT: Port 2 (LCD) and P1.4 (ENABLE)
                    622     ;
                    623     ;*******************************************************************************
                    624
016A C293           625     DisplayIntro:   clr RS                  ; Register Select ( 0 = Command )
016C C292           626                     clr RW                  ; Read/Write ( 1 = Read  ; 0 = Write )
016E 7B00           627                     mov R3, #0              ; Clear R3
0170 D294           628                     setb ENABLE             ; Latch the data
0172 517F           629                     call delay              ; Call delay subroutine
0174 C294           630                     clr ENABLE              ; Reset latch
0176 9002D4         631     DisplayIntro1:  mov DPTR, #Table5       ; Initialize Data Pointer
0179 EB             632                     mov A, R3               ; Move table index to acc
017A 93             633                     movc A, @A + DPTR       ; Get character
017B 5173           634                     call display            ; call LCD Display proceedure
017D BB07F6         635                     cjne R3, #7, DisplayIntro1 ; Repeat until R3=7
```

```
0180 7B19          636                     mov R3, #19H              ; Set R3=25
0182 517F          637     DisplayIntro2:  call delay                ; Call delay subroutine
0184 DBFC          638                     djnz R3, DisplayIntro2    ; Run delay subroutine 25 times
0186 315E          639                     call clear                ; Call clear subroutine
0188 22            640                     ret                       ; Return
                   641
                   642
                   643     ;*****************************************************************************
                   644     ;
                   645     ;  control subroutine
                   646     ;
                   647     ;  This subroutine determines what action to take depending on which button
                   648     ;  was pressed.
                   649     ;
                   650     ;  LOCAL REGISTERS USED: R2,R3
                   651     ;  INPUT: byte in the Accumulator
                   652     ;  OUTPUT: P0
                   653     ;
                   654     ;*****************************************************************************
                   655
0189 B44510        656     control:        cjne A, #45H, Next1       ; Jump to Next1 if * was not pressed
018C BA0105        657                     cjne R2, #1H, Manual      ; Jump to Manual if currently in Auto
018F 7A00          658                     mov R2, #0H               ; Switch to Auto mode
0191 5130          659                     call AutoDisplay          ; Call AutoDisplay subroutine
0193 22            660                     ret                       ; Return
                   661
0194 7A01          662     Manual:         mov R2, #1H               ; Switch to Manual mode
0196 758000        663                     mov P0, #0H               ; Reset actuators
0199 513F          664                     call ManualDisplay        ; Call ManualDisplay subroutine
019B 22            665                     ret                       ; Return
                   666
019C B44109        667     Next1:          cjne A, #41H, Next2       ; Jump to Next2 if A was not pressed
019F BA0106        668                     cjne R2, #01H, Next2      ; Jump to Next2 if not in Manual mode
01A2 758080        669                     mov P0, #80H              ; Activate Actuator A
01A5 514E          670                     call DisplayActive        ; Call DisplayActive subroutine
01A7 22            671                     ret                       ; Return
                   672
01A8 B44209        673     Next2:          cjne A, #42H, Next3       ; Jump to Next3 if B was not pressed
01AB BA0106        674                     cjne R2, #01H, Next3      ; Jump to Next3 if not in Manual mode
01AE 758040        675                     mov P0, #40H              ; Activate Actuator B
01B1 514E          676                     call DisplayActive        ; Call DisplayActive subroutine
01B3 22            677                     ret                       ; Return
                   678
01B4 B44309        679     Next3:          cjne A, #43H, Next4       ; Jump to Next4 if C was not pressed
01B7 BA0106        680                     cjne R2, #01H, Next4      ; Jump to Next4 if not in Manual mode
01BA 758010        681                     mov P0, #10H              ; Activate Actuator C
01BD 514E          682                     call DisplayActive        ; Call DisplayActive subroutine
01BF 22            683                     ret                       ; Return
                   684
01C0 BA003F        685     Next4:          cjne R2, #0H, NextD       ; Jump to NextD if not in Auto mode
01C3 B43104        686                     cjne A, #31H, Next5       ; Jump to Next5 if 1 was not pressed
01C6 BC003A        687                     cjne R4, #00H, setting1   ; Jump to setting1 if speed 1 is not set
01C9 22            688                     ret                       ; Return
                   689
01CA B43204        690     Next5:          cjne A, #32H, Next6       ; Jump to Next6 if 2 was not pressed
01CD BC0138        691                     cjne R4, #01H, setting2   ; Jump to setting2 if speed 2 is not set
01D0 22            692                     ret                       ; Return
                   693
01D1 B43304        694     Next6:          cjne A, #33H, Next7       ; Jump to Next7 if 3 was not pressed
01D4 BC0236        695                     cjne R4, #02H, setting3   ; Jump to setting3 if speed 3 is not set
01D7 22            696                     ret                       ; Return
                   697
01D8 B43404        698     Next7:          cjne A, #34H, Next8       ; Jump to Next8 if 4 was not pressed
01DB BC0434        699                     cjne R4, #04H, setting4   ; Jump to setting4 if speed 4 is not set
01DE 22            700                     ret                       ; Return
                   701
01DF B43504        702     Next8:          cjne A, #35H, Next9       ; Jump to Next9 if 5 was not pressed
01E2 BC0532        703                     cjne R4, #05H, setting5   ; Jump to setting5 if speed 5 is not set
01E5 22            704                     ret                       ; Return
                   705
01E6 B43604        706     Next9:          cjne A, #36H, NextA       ; Jump to NextA if 6 was not pressed
01E9 BC0630        707                     cjne R4, #06H, setting6   ; Jump to setting6 if speed 6 is not set
01EC 22            708                     ret                       ; Return
                   709
01ED B43704        710     NextA:          cjne A, #37H, NextB       ; Jump to NextB if 7 was not pressed
01F0 BC082E        711                     cjne R4, #08H, setting7   ; Jump to setting7 if speed 7 is not set
01F3 22            712                     ret                       ; Return
                   713
01F4 B43804        714     NextB:          cjne A, #38H, NextC       ; Jump to NextC if 8 was not pressed
01F7 BC092C        715                     cjne R4, #09H, setting8   ; Jump to setting8 if speed 8 is not set
01FA 22            716                     ret                       ; Return
                   717
01FB B43904        718     NextC:          cjne A, #39H, NextD       ; Jump to NextD if 9 was not pressed
01FE BC0A2A        719                     cjne R4, #0AH, setting9   ; Jump to setting9 if speed 9 is not set
0201 22            720                     ret                       ; Return
                   721
0202 22            722     NextD:          ret                       ; Return
                   723
0203 7C00          724     setting1:       mov R4, #00H              ; Set R4
0205 515E          725                     call DisplaySpeed         ; Call DisplaySpeed subroutine
0207 22            726                     ret                       ; Return
                   727
0208 7C01          728     setting2:       mov R4, #01H              ; Set R4
020A 515E          729                     call DisplaySpeed         ; Call DisplaySpeed subroutine
020C 22            730                     ret                       ; Return
```

```
                    731
020D 7C02           732     setting3:       mov R4, #02H            ; Set R4
020F 515E           733                     call DisplaySpeed       ; Call DisplaySpeed subroutine
0211 22             734                     ret                     ; Return
                    735
0212 7C04           736     setting4:       mov R4, #04H            ; Set R4
0214 515E           737                     call DisplaySpeed       ; Call DisplaySpeed subroutine
0216 22             738                     ret                     ; Return
                    739
0217 7C05           740     setting5:       mov R4, #05H            ; Set R4
0219 515E           741                     call DisplaySpeed       ; Call DisplaySpeed subroutine
021B 22             742                     ret                     ; Return
                    743
021C 7C06           744     setting6:       mov R4, #06H            ; Set R4
021E 515E           745                     call DisplaySpeed       ; Call DisplaySpeed subroutine
0220 22             746                     ret                     ; Return
                    747
0221 7C08           748     setting7:       mov R4, #08H            ; Set R4
0223 515E           749                     call DisplaySpeed       ; Call DisplaySpeed subroutine
0225 22             750                     ret                     ; Return
                    751
0226 7C09           752     setting8:       mov R4, #09H            ; Set R4
0228 515E           753                     call DisplaySpeed       ; Call DisplaySpeed subroutine
022A 22             754                     ret                     ; Return
                    755
022B 7C0A           756     setting9:       mov R4, #0AH            ; Set R4
022D 515E           757                     call DisplaySpeed       ; Call DisplaySpeed subroutine
022F 22             758                     ret                     ; Return
                    759
                    760
                    761     ;*************************************************************************
                    762     ;
                    763     ;  AutoDisplay subroutine
                    764     ;
                    765     ;  This subroutine displays "Auto" on the LCD
                    766     ;
                    767     ;  LOCAL REGISTERS USED: R3,R4
                    768     ;  INPUT: byte in the Accumulator
                    769     ;  OUTPUT: LCD
                    770     ;
                    771     ;*************************************************************************
                    772
0230 315E           773     AutoDisplay:    call Clear              ; Clear LCD Display
0232 7B00           774                     mov R3, #0              ; Clear R3
                    775
0234 9002C6         776     AutoDisplay1:   mov DPTR, #Table2       ; Initialize Data Pointer
0237 EB             777                     mov A, R3               ; Move table index to acc
0238 93             778                     movc A, @A + DPTR       ; Get character
0239 5173           779                     call display            ; call LCD Display proceedure
023B BB04F6         780                     cjne R3, #4, AutoDisplay1  ; Repeat until R3=4
023E 22             781                     ret                     ; Return
                    782
                    783
                    784     ;*************************************************************************
                    785     ;
                    786     ;  ManualDisplay subroutine
                    787     ;
                    788     ;  This subroutine displays "Manual" on the LCD
                    789     ;
                    790     ;  LOCAL REGISTERS USED: R3
                    791     ;  INPUT: byte in the Accumulator
                    792     ;  OUTPUT: LCD
                    793     ;
                    794     ;*************************************************************************
                    795
023F 315E           796     ManualDisplay:  call Clear              ; Clear LCD Display
0241 7B00           797                     mov R3, #0              ; Clear R3
                    798
0243 9002CA         799     ManDisplay1:    mov DPTR, #Table3       ; Initialize Data Pointer
0246 EB             800                     mov A, R3               ; Move table index to acc
0247 93             801                     movc A, @A + DPTR       ; Get character
0248 5173           802                     call display            ; call LCD Display proceedure
024A BB06F6         803                     cjne R3, #6, ManDisplay1   ; Repeat until R3=6
024D 22             804                     ret                     ; Return
                    805
                    806
                    807     ;*************************************************************************
                    808     ;
                    809     ;  DisplayActive subroutine
                    810     ;
                    811     ;  This subroutine displays the letter of the active actuator on the second
                    812     ;  line of the LCD when the program is in manual mode
                    813     ;
                    814     ;  LOCAL REGISTERS USED: none
                    815     ;  INPUT: byte in the Accumulator
                    816     ;  OUTPUT: LCD
                    817     ;
                    818     ;*************************************************************************
                    819
024E C293           820     DisplayActive:  clr RS                  ; Register Select ( 0 = Command )
0250 C292           821                     clr RW                  ; Read/Write ( 1 = Read  ; 0 = Write )
0252 75A0C0         822                     mov LCD, #0C0H          ; Set cursor position
0255 D294           823                     setb ENABLE             ; Latches the data.
0257 517F           824                     call delay              ; Waits.
0259 C294           825                     clr ENABLE              ; Then resets latch.
```

```
025B 5173              826                      call display              ; call LCD Display proceedure
025D 22                827                      ret                       ; Return
                       828
                       829
                       830     ;*****************************************************************************
                       831     ;
                       832     ;  DisplaySpeed subroutine
                       833     ;
                       834     ;  This subroutine displays the speed of the active actuator on the second
                       835     ;  line of the LCD.
                       836     ;
                       837     ;  LOCAL REGISTERS USED: R4
                       838     ;  INPUT: none
                       839     ;  OUTPUT: LCD
                       840     ;
                       841     ;*****************************************************************************
                       842
025E C293              843     DisplaySpeed:    clr RS                   ; Register Select ( 0 = Command )
0260 C292              844                      clr RW                   ; Read/Write ( 1 = Read  ; 0 = Write )
0262 75A0C0            845                      mov LCD, #0C0H           ; Set cursor position
0265 D294              846                      setb ENABLE              ; Latches the data.
0267 517F              847                      call delay               ; Waits.
0269 C294              848                      clr ENABLE               ; Then resets latch.
026B 9002B6            849                      mov DPTR, #Table1        ; Initialize Data Pointer
026E EC                850                      mov A, R4                ; Move table index to acc
026F 93                851                      movc A, @A + DPTR        ; Get character
0270 5173              852                      call display             ; call LCD Display proceedure
0272 22                853                      ret                      ; Return
                       854
                       855
                       856     ;*****************************************************************************
                       857     ;
                       858     ;  display subroutine
                       859     ;
                       860     ;  Moves the control or ASCII byte in the accumulator into the LCD 8-bits at
                       861     ;  a time.
                       862     ;
                       863     ;  LOCAL REGISTERS USED: R3
                       864     ;  INPUT: byte in the Accumulator
                       865     ;  OUTPUT: One byte to the LCD.
                       866     ;
                       867     ;*****************************************************************************
                       868
0273                   869     display:                                  ; The data to be sent is in A.
0273 D293              870                      setb RS                  ; Register Select ( 1 = Data )
0275 F5A0              871                      mov LCD, A               ; Sends data to LCD
0277 D294              872                      setb ENABLE              ; Latches the data.
0279 517F              873                      call delay               ; Waits.
027B C294              874                      clr ENABLE               ; Then resets the latch.
                       875
027D 0B                876                      inc R3                   ; R3 is used to keep track of LCD DDRAM.
                       877                                               ; After an ASCII char is sent, R3 is
                       878                                               ;  incremented.
                       879
027E 22                880                      ret                      ; Return
                       881
                       882
                       883     ;*****************************************************************************
                       884     ;
                       885     ;  delay subroutine
                       886     ;
                       887     ;  This subroutine is a simple delay loop that is used to provide timing for
                       888     ;  the LCD interface.
                       889     ;
                       890     ;  LOCAL REGISTERS USED: R5, R6
                       891     ;  INPUT: none
                       892     ;  OUTPUT: none
                       893     ;  ACTION: Provides time delay for the LCD interface.
                       894     ;
                       895     ;*****************************************************************************
                       896
027F 7E00              897     delay:           mov   R6, #00h           ; Set register 6 to 0
0281 7D00              898     Loop0:           mov   R5, #00h           ; Set register 5 to 0
0283 DDFE              899                      djnz  R5, $              ; Decrement register 5
0285 DEFA              900                      djnz  R6, Loop0          ; Decrement register 6
0287 22                901                      ret                      ; Return
                       902
                       903
                       904     ;*****************************************************************************
                       905     ;
                       906     ;  reset
                       907     ;
                       908     ;  Initialization by instruction
                       909     ;  This subroutine sends a Function Set byte (30H) to the LCD three times so
                       910     ;  that the LCD will reset correctly and communicate with the 8051.
                       911     ;
                       912     ;  INPUT: none
                       913     ;  OUTPUT: LCD (P2), ENABLE (P1.4)
                       914     ;
                       915     ;*****************************************************************************
                       916
0288 517F              917     reset:           call delay
028A 75A030            918                      mov LCD, #30H            ; Writes Function Set.
028D D294              919                      setb ENABLE              ; Latches Instruction.
028F 517F              920                      call delay               ; Waits.
```

```
0291 C294            921                    clr ENABLE              ; Then resets latch.
0293 51AC            922                    call Busy               ; Check Busy Flag delay
0295 75A030          923                    mov LCD, #30H           ; Writes Function Set.
0298 D294            924                    setb ENABLE             ; Latches Instruction.
029A 517F            925                    call delay              ; Waits.
029C C294            926                    clr ENABLE              ; Then resets the latch.
029E 51AC            927                    call Busy               ; Check Busy Flagdelay
02A0 75A030          928                    mov LCD, #30H           ; Writes Function Set.
02A3 D294            929                    setb ENABLE             ; Latches Instruction
02A5 517F            930                    call delay              ; Waits
02A7 C294            931                    clr Enable              ; Then resets the latch
02A9 51AC            932                    call Busy               ; Check Busy Flag
02AB 22              933                    ret                     ; Return
                     934
                     935
                     936    ;*************************************************************************
                     937    ;
                     938    ;  Busy
                     939    ;
                     940    ;  This Subroutine checks the Busy Flag (DB7) to ensure the LCD is not busy
                     941    ;
                     942    ;  INPUT  P2.7
                     943    ;
                     944    ;*************************************************************************
                     945
02AC C293            946    Busy:           clr RS                  ; Clear RS
02AE D292            947                    setb RW                 ; Set RW
02B0 20A7FD          948                    jb P2.7, $              ; Wait while Pin 2.7 is active
02B3 C292            949                    clr RW                  ; Clear RW
02B5 22              950                    ret                     ; Return
                     951
                     952
                     953    ;*************************************************************************
                     954    ;
                     955    ;  Tables
                     956    ;
                     957    ;*************************************************************************
                     958
02B6 31323341        959    Table1:         db  31H,32H,33H,41H,34H,35H,36H,42H,37H,38H,39H,43H,45H,30H,46H,44H
02BA 34353642
02BE 37383943
02C2 45304644
02C6 4175746F        960    Table2:         db  41H,75H,74H,6FH
02CA 4D616E75        961    Table3:         db  4DH,61H,6EH,75H,61H,6CH
02CE 616C
02D0 00804010        962    Table4:         db  00H,80H,40H,10H
02D4 43617420        963    Table5:         db  43H,61H,74H,20H,54H,6FH,79H
02D8 546F79
                     964
                     965                    end
```

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

N A M E	T Y P E	V A L U E	ATTRIBUTES / REFERENCES
AC	B ADDR	00D0H.6 A	240#
ACC	D ADDR	00E0H A	141# 478 479 480 481 482 483
ACK	B ADDR	00C0H.1 A	225#
ACKRQ	B ADDR	00C0H.3 A	223#
AD0BUSY	B ADDR	00E8H.4 A	262#
AD0CM0	B ADDR	00E8H.0 A	266#
AD0CM1	B ADDR	00E8H.1 A	265#
AD0CM2	B ADDR	00E8H.2 A	264#
AD0EN	B ADDR	00E8H.7 A	259#
AD0INT	B ADDR	00E8H.5 A	261#
AD0TM	B ADDR	00E8H.6 A	260#
AD0WINT	B ADDR	00E8H.3 A	263#
ADC0CF	D ADDR	00BCH A	108#
ADC0CN	D ADDR	00E8H A	149# 259 260 261 262 263 264 265 266
ADC0GTH	D ADDR	00C4H A	115#
ADC0GTL	D ADDR	00C3H A	114#
ADC0H	D ADDR	00BEH A	110#
ADC0L	D ADDR	00BDH A	109#
ADC0LTH	D ADDR	00C6H A	117#
ADC0LTL	D ADDR	00C5H A	116#
AMX0N	D ADDR	00BAH A	106#
AMX0P	D ADDR	00BBH A	107#
ARBLOST	B ADDR	00C0H.2 A	224#
AUTODISPLAY	C ADDR	0230H A	346 659 773#
AUTODISPLAY1	C ADDR	0234H A	776# 780
AUTOROUTINE	C ADDR	009FH A	348 379 448#
B	D ADDR	00F0H A	157#
BUSY	C ADDR	02ACH A	556 560 562 566 568 572 574 578 580 584 605 922 927 932 946#
CCF0	B ADDR	00D8H.0 A	256#
CCF1	B ADDR	00D8H.1 A	255#
CCF2	B ADDR	00D8H.2 A	254#
CCF3	B ADDR	00D8H.3 A	253#
CCF4	B ADDR	00D8H.4 A	252#
CF	B ADDR	00D8H.7 A	249#
CKCON	D ADDR	008EH A	64#
CLEAR	C ADDR	015EH A	344 604# 639 773 796
CLKMUL	D ADDR	00B9H A	105#
CLKSEL	D ADDR	00A9H A	91#
COL1	B ADDR	00B0H.4 A	301# 415 416
COL2	B ADDR	00B0H.5 A	302# 419 420
COL3	B ADDR	00B0H.6 A	303# 423 424
COL4	B ADDR	00B0H.7 A	304# 427 428
COLSCAN	C ADDR	0077H A	355 361 367 373 415#
CONTROL	C ADDR	0189H A	384 656#
CPT0CN	D ADDR	009BH A	77#
CPT0MD	D ADDR	009DH A	79#
CPT0MX	D ADDR	009FH A	81#
CPT1CN	D ADDR	009AH A	76#
CPT1MD	D ADDR	009CH A	78#
CPT1MX	D ADDR	009EH A	80#
CR	B ADDR	00D8H.6 A	250#
CY	B ADDR	00D0H.7 A	239#
DELAY	C ADDR	027FH A	553 558 564 570 576 582 607 629 637 824 847 873 897# 917 920 925 930
DISPLAY	C ADDR	0273H A	634 779 802 826 852 869#
DISPLAYACTIVE	C ADDR	024EH A	670 676 682 820#
DISPLAYINTRO	C ADDR	016AH A	345 625#
DISPLAYINTRO1	C ADDR	0176H A	631# 635
DISPLAYINTRO2	C ADDR	0182H A	637# 638
DISPLAYSPEED	C ADDR	025EH A	347 725 729 733 737 741 745 749 753 757 843#
DPH	D ADDR	0083H A	53#
DPL	D ADDR	0082H A	52#
EA	B ADDR	00A8H.7 A	199#
EIE1	D ADDR	00E6H A	147#
EIE2	D ADDR	00E7H A	148#
EIP1	D ADDR	00F6H A	163#
EIP2	D ADDR	00F7H A	164#
EMI0CF	D ADDR	0085H A	55#
EMI0CN	D ADDR	00AAH A	92#
EMI0TC	D ADDR	0084H A	54#
ENABLE	B ADDR	0090H.4 A	290# 552 557 559 563 565 569 571 575 577 581 583 606 608 628 630 823 825 846 848 872 874 919 921 924 926 929 931
ES0	B ADDR	00A8H.4 A	202#
ESPI0	B ADDR	00A8H.6 A	200#
ET0	B ADDR	00A8H.1 A	205#
ET1	B ADDR	00A8H.3 A	203#
ET2	B ADDR	00A8H.5 A	201#
EX0	B ADDR	00A8H.0 A	206#
EX1	B ADDR	00A8H.2 A	204#
F0	B ADDR	00D0H.5 A	241# 356 362 368 374 385 432
F1	B ADDR	00D0H.1 A	245#
FINISH	C ADDR	0062H A	356 362 368 374 381#
FLKEY	D ADDR	00B7H A	103#
FLSCL	D ADDR	00B6H A	102#
GOTKEY	C ADDR	009CH A	417 421 425 429 432#
IE	D ADDR	00A8H A	90# 199 200 201 202 203 204 205 206
IE0	B ADDR	0088H.1 A	185#
IE1	B ADDR	0088H.3 A	183#
INIT	C ADDR	0112H A	343 550#
IP	D ADDR	00B8H A	104# 210 211 212 213 214 215 216

```
IT0. . . . . B ADDR   0088H.0 A      186#
IT01CF . . . . . D ADDR   00E4H   A      145#
IT1. . . . . B ADDR   0088H.2 A      184#
KEYPORT. . . . . D ADDR   00B0H   A      295#
LCD. . . . . D ADDR   00A0H   A      293# 555 561 567 573 579 604 822 845 871 918 923 928
LOOP0. . . . . C ADDR   0281H   A      898# 900
MAIN . . . . . C ADDR   0003H   A      315 325#
MANDISPLAY1. . . . C ADDR   0243H   A      799# 803
MANUAL . . . . . C ADDR   0194H   A      657 662#
MANUALDISPLAY. . . C ADDR   023FH   A      664 796#
MASTER . . . . . B ADDR   00C0H.7 A      219#
MCE0 . . . . . B ADDR   0098H.5 A      191#
MODF . . . . . B ADDR   00F8H.5 A      271#
NEXT1. . . . . C ADDR   019CH   A      656 667#
NEXT2. . . . . C ADDR   01A8H   A      667 668 673#
NEXT3. . . . . C ADDR   01B4H   A      673 674 679#
NEXT4. . . . . C ADDR   01C0H   A      679 680 685#
NEXT5. . . . . C ADDR   01CAH   A      686 690#
NEXT6. . . . . C ADDR   01D1H   A      690 694#
NEXT7. . . . . C ADDR   01D8H   A      694 698#
NEXT8. . . . . C ADDR   01DFH   A      698 702#
NEXT9. . . . . C ADDR   01E6H   A      702 706#
NEXTA. . . . . C ADDR   01EDH   A      706 710#
NEXTB. . . . . C ADDR   01F4H   A      710 714#
NEXTC. . . . . C ADDR   01FBH   A      714 718#
NEXTCOL. . . . . C ADDR   007FH   A      415 418#
NEXTCOL2 . . . . . C ADDR   0088H   A      419 422#
NEXTCOL3 . . . . . C ADDR   0091H   A      423 426#
NEXTD. . . . . C ADDR   0202H   A      685 718 722#
NOKEY. . . . . C ADDR   009AH   A      427 430#
NSSMD0 . . . . . B ADDR   00F8H.2 A      274#
NSSMD1 . . . . . B ADDR   00F8H.3 A      273#
OSCICL . . . . . D ADDR   00B3H   A      99#
OSCICN . . . . . D ADDR   00B2H   A      98#
OSCLCN . . . . . D ADDR   0086H   A      56#
OSCXCN . . . . . D ADDR   00B1H   A      97#
OV . . . . . B ADDR   00D0H.2 A      244#
P. . . . . B ADDR   00D0H.0 A      246# 474
P0 . . . . . D ADDR   0080H   A      50# 337 451 663 669 675 681
P0MDIN . . . . . D ADDR   00F1H   A      158#
P0MDOUT. . . . . D ADDR   00A4H   A      86# 333
P0SKIP . . . . . D ADDR   00D4H   A      129#
P1 . . . . . D ADDR   0090H   A      66# 290 291 292
P1MDIN . . . . . D ADDR   00F2H   A      159# 332
P1MDOUT. . . . . D ADDR   00A5H   A      87# 331
P1SKIP . . . . . D ADDR   00D5H   A      130#
P2 . . . . . D ADDR   00A0H   A      82# 293 948
P2MDIN . . . . . D ADDR   00F3H   A      160#
P2MDOUT. . . . . D ADDR   00A6H   A      88# 330
P2SKIP . . . . . D ADDR   00D6H   A      131#
P3 . . . . . D ADDR   00B0H   A      96# 295 296 297 298 299 301 302 303 304
P3MDIN . . . . . D ADDR   00F4H   A      161#
P3MDOUT. . . . . D ADDR   00A7H   A      89# 334
P3SKIP . . . . . D ADDR   00DFH   A      140#
P4 . . . . . D ADDR   00C7H   A      118#
P4MDIN . . . . . D ADDR   00F5H   A      162#
P4MDOUT. . . . . D ADDR   00AEH   A      94#
PCA0CN . . . . . D ADDR   00D8H   A      133# 249 250 252 253 254 255 256
PCA0CPH0 . . . . . D ADDR   00FCH   A      169#
PCA0CPH1 . . . . . D ADDR   00EAH   A      151#
PCA0CPH2 . . . . . D ADDR   00ECH   A      153#
PCA0CPH3 . . . . . D ADDR   00EEH   A      155#
PCA0CPH4 . . . . . D ADDR   00FEH   A      171#
PCA0CPL0 . . . . . D ADDR   00FBH   A      168#
PCA0CPL1 . . . . . D ADDR   00E9H   A      150#
PCA0CPL2 . . . . . D ADDR   00EBH   A      152#
PCA0CPL3 . . . . . D ADDR   00EDH   A      154#
PCA0CPL4 . . . . . D ADDR   00FDH   A      170#
PCA0CPM0 . . . . . D ADDR   00DAH   A      135#
PCA0CPM1 . . . . . D ADDR   00DBH   A      136#
PCA0CPM2 . . . . . D ADDR   00DCH   A      137#
PCA0CPM3 . . . . . D ADDR   00DDH   A      138#
PCA0CPM4 . . . . . D ADDR   00DEH   A      139#
PCA0H. . . . . D ADDR   00FAH   A      167#
PCA0L. . . . . D ADDR   00F9H   A      166#
PCA0MD . . . . . D ADDR   00D9H   A      134# 327
PCON . . . . . D ADDR   0087H   A      57#
PFE0CN . . . . . D ADDR   00AFH   A      95#
POAEFBEF . . . . . C ADDR   0070H   A      378 391#
PS0. . . . . B ADDR   00B8H.4 A      212#
PSCTL. . . . . D ADDR   008FH   A      65#
PSPI0. . . . . B ADDR   00B8H.6 A      210#
PSW. . . . . D ADDR   00D0H   A      125# 239 240 241 242 243 244 245 246
PT0. . . . . B ADDR   00B8H.1 A      215#
PT1. . . . . B ADDR   00B8H.3 A      213#
PT2. . . . . B ADDR   00B8H.5 A      211#
PX0. . . . . B ADDR   00B8H.0 A      216#
PX1. . . . . B ADDR   00B8H.2 A      214#
RANDOM . . . . . C ADDR   00AAH   A      448 469#
RANDOM1. . . . . C ADDR   00AFH   A      470 473#
RB80 . . . . . B ADDR   0098H.2 A      194#
REF0CN . . . . . D ADDR   00D1H   A      126#
REG0CN . . . . . D ADDR   00C9H   A      120#
REN0 . . . . . B ADDR   0098H.4 A      192#
REPEAT . . . . . C ADDR   0111H   A      503 507 511 515 519 523 527 531 534 535#
```

```
RESET. . . . . C ADDR   0288H   A      554 917#
RI0. . . . . . B ADDR   0098H.0 A      196#
ROW1 . . . . . B ADDR   00B0H.0 A      296# 354 359
ROW2 . . . . . B ADDR   00B0H.1 A      297# 360 365
ROW3 . . . . . B ADDR   00B0H.2 A      298# 366 371
ROW4 . . . . . B ADDR   00B0H.3 A      299# 353 372
RS . . . . . . B ADDR   0090H.3 A      292# 550 625 820 843 870 946
RS0. . . . . . B ADDR   00D0H.3 A      243#
RS1. . . . . . B ADDR   00D0H.4 A      242#
RSTSRC . . . . D ADDR   00EFH   A      156#
RW . . . . . . B ADDR   0090H.2 A      291# 551 626 821 844 947 949
RXOVRN . . . . B ADDR   00F8H.4 A      272#
S0MODE . . . . B ADDR   0098H.7 A      189#
SBCON1 . . . . D ADDR   00ACH   A      93#
SBRLH1 . . . . D ADDR   00B5H   A      101#
SBRLL1 . . . . D ADDR   00B4H   A      100#
SBUF0. . . . . D ADDR   0099H   A      75#
SBUF1. . . . . D ADDR   00D3H   A      128#
SCON0. . . . . D ADDR   0098H   A      74# 189 191 192 193 194 195 196
SCON1. . . . . D ADDR   00D2H   A      127#
SDFASFDA . . . C ADDR   006DH   A      377 388#
SETTING1 . . . C ADDR   0203H   A      687 724#
SETTING2 . . . C ADDR   0208H   A      691 728#
SETTING3 . . . C ADDR   020DH   A      695 732#
SETTING4 . . . C ADDR   0212H   A      699 736#
SETTING5 . . . C ADDR   0217H   A      703 740#
SETTING6 . . . C ADDR   021CH   A      707 744#
SETTING7 . . . C ADDR   0221H   A      711 748#
SETTING8 . . . C ADDR   0226H   A      715 752#
SETTING9 . . . C ADDR   022BH   A      719 756#
SI . . . . . . B ADDR   00C0H.0 A      226#
SMB0CF . . . . D ADDR   00C1H   A      112#
SMB0CN . . . . D ADDR   00C0H   A      111# 219 220 221 222 223 224 225 226
SMB0DAT. . . . D ADDR   00C2H   A      113#
SMOD1. . . . . D ADDR   00E5H   A      146#
SP . . . . . . D ADDR   0081H   A      51#
SPEED. . . . . C ADDR   00C3H   A      452 500#
SPEED1 . . . . C ADDR   010BH   A      528 532#
SPEED2 . . . . C ADDR   0102H   A      524 528#
SPEED3 . . . . C ADDR   00F9H   A      520 524#
SPEED4 . . . . C ADDR   00F0H   A      516 520#
SPEED5 . . . . C ADDR   00E7H   A      512 516#
SPEED6 . . . . C ADDR   00DEH   A      508 512#
SPEED7 . . . . C ADDR   00D5H   A      504 508#
SPEED8 . . . . C ADDR   00CCH   A      500 504#
SPI0CFG. . . . D ADDR   00A1H   A      83#
SPI0CKR. . . . D ADDR   00A2H   A      84#
SPI0CN . . . . D ADDR   00F8H   A      165# 269 270 271 272 273 274 275 276
SPI0DAT. . . . D ADDR   00A3H   A      85#
SPIEN. . . . . B ADDR   00F8H.0 A      276#
SPIF . . . . . B ADDR   00F8H.7 A      269#
STA. . . . . . B ADDR   00C0H.5 A      221#
STAGAIN. . . . C ADDR   0075H   A      376 395#
START. . . . . C ADDR   0031H   A      350# 386 389 393 395
STO. . . . . . B ADDR   00C0H.4 A      222#
T2CE . . . . . B ADDR   00C8H.4 A      232#
T2CSS. . . . . B ADDR   00C8H.1 A      235#
T2SPLIT. . . . B ADDR   00C8H.3 A      233#
T2XCLK . . . . B ADDR   00C8H.0 A      236#
TABLE1 . . . . C ADDR   02B6H   A      381 849 959#
TABLE2 . . . . C ADDR   02C6H   A      776 960#
TABLE3 . . . . C ADDR   02CAH   A      799 961#
TABLE4 . . . . C ADDR   02D0H   A      449 962#
TABLE5 . . . . C ADDR   02D4H   A      631 963#
TB80 . . . . . B ADDR   0098H.3 A      193#
TCON . . . . . D ADDR   0088H   A      58# 179 180 181 182 183 184 185 186
TF0. . . . . . B ADDR   0088H.5 A      181#
TF1. . . . . . B ADDR   0088H.7 A      179#
TF2H . . . . . B ADDR   00C8H.7 A      229#
TF2L . . . . . B ADDR   00C8H.6 A      230#
TF2LEN . . . . B ADDR   00C8H.5 A      231#
TH0. . . . . . D ADDR   008CH   A      62#
TH1. . . . . . D ADDR   008DH   A      63#
TI0. . . . . . B ADDR   0098H.1 A      195#
TL0. . . . . . D ADDR   008AH   A      60#
TL1. . . . . . D ADDR   008BH   A      61#
TMOD . . . . . D ADDR   0089H   A      59#
TMR2CN . . . . D ADDR   00C8H   A      119# 229 230 231 232 233 234 235 236
TMR2H. . . . . D ADDR   00CDH   A      124#
TMR2L. . . . . D ADDR   00CCH   A      123#
TMR2RLH. . . . D ADDR   00CBH   A      122#
TMR2RLL. . . . D ADDR   00CAH   A      121#
TMR3CN . . . . D ADDR   0091H   A      67#
TMR3H. . . . . D ADDR   0095H   A      71#
TMR3L. . . . . D ADDR   0094H   A      70#
TMR3RLH. . . . D ADDR   0093H   A      69#
TMR3RLL. . . . D ADDR   0092H   A      68#
TR0. . . . . . B ADDR   0088H.4 A      182#
TR1. . . . . . B ADDR   0088H.6 A      180#
TR2. . . . . . B ADDR   00C8H.2 A      234#
TXBMT. . . . . B ADDR   00F8H.1 A      275#
TXMODE . . . . B ADDR   00C0H.6 A      220#
USB0ADR. . . . D ADDR   0096H   A      72#
USB0DAT. . . . D ADDR   0097H   A      73#
USB0XCN. . . . D ADDR   00D7H   A      132#
```

```
VDM0CN . . . . . .  D ADDR   00FFH   A      172#
WCOL . . . . . . .  B ADDR   00F8H.6 A      270#
XBR0 . . . . . . .  D ADDR   00E1H   A      142#
XBR1 . . . . . . .  D ADDR   00E2H   A      143# 335
XBR2 . . . . . . .  D ADDR   00E3H   A      144#


REGISTER BANK(S) USED: 0


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