AN614

Interfacing the 8051 with 2-wire Serial EEPROMs

Author: *Mike Rosenfield*
Memory Products Division

INTERFACING MICROCHIP SERIAL EEPROMS TO THE INTEL 8051 FAMILY OF MICROCONTROLLERS

Many designers today are implementing embedded systems that require low cost non-volatile memory. Microchip has addressed this need with a full line of serial EEPROMs, in a variety of memory configurations, using the industry-standard 2- or 3-wire communication protocols. The theory and application of these protocols are addressed in detail in Microchip's application note AN536.

Microchip recognizes that its broad customer base uses a variety of micro-controllers; many firmware related questions have been asked concerning interfacing the 8051 family and its derivatives.

The purpose of this app note is to provide assembly language examples of 8051 code for the various serial EEPROMs available from Microchip. These routines are intended to provide the basic operating kernels for storing data to or retrieving data from a serial EEPROM.

All of the routines in this app note are available, as source code, for downloading from Microchip's BBS. Information on the BBS is available elsewhere in the Embedded Control Handbook. The file to download is AN61437.zip.

This app note covers all of Microchip's 2-wire serial devices. Note that some devices have features not supported in others, and therefore, some sections of the code presented here may not be applicable to a particular part. We have attempted to label those special sections to minimize confusion.

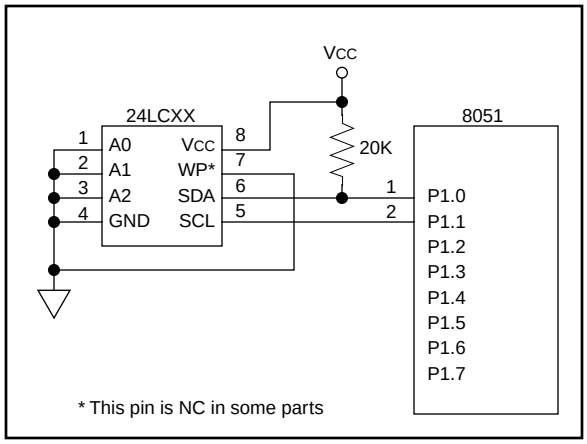
The code includes a simple loop-type shell to enable it to be executed (with an emulator) without the user having to write any other routines. The various address and data pointers must be set to the desired values by hand, before each execution cycle.

TIMING DATA

Clock and data timing is accomplished by software. There are two sets of timing specifications: 100 kHz and 400 kHz. Assuming a 12 MHz 8051 clock, extra NOP's have been added to slow timing down to 100 kHz. See Note 1 in the listing. If a 16 MHz clock is used, additional NOP's are required for 100 kHz operation. See Note 2 in the listing. For 400 kHz operation, the NOP's labelled Note 1 or Note 2 are not needed. If not needed, NOP's may be left out.

Below is the connection diagram used for this app note. Do not forget the pull-up resistor!.

WIRING DIAGRAM



Please check the Microchip BBS for the latest version of the source code. For BBS access information, see Section 6, Microchip Bulletin Board Service information, page 6-3.

APPENDIX A: SOURCE CODE

```
MCS-51 MACRO ASSEMBLER    2WIRE                                                    05/31/95

DOS 6.0 (038-N) MCS-51 MACRO ASSEMBLER, V2.2
OBJECT MODULE PLACED IN 2WIRE.OBJ
ASSEMBLER INVOKED BY:  C:\ICE5100\ASM51\ASM51.EXE 2WIRE.TXT


LOC  OBJ            LINE     SOURCE

                       1     $ PAGELENGTH(46) PAGEWIDTH(132) DEBUG NOPAGING XREF
                       2     ;REGISTER ASSIGNMENTS:
                       3     ;
                       4     ;R1     DATA OR DATA POINTER
                       5     ;R2     LOOP COUNTER REGISTER
                       6     ;R3     ADDRESS, HI BYTE
                       7     ;R4     ADDRESS, LOW BYTE
                       8     ;R5
                       9     ;R6     BYTE COUNT FOR PAGE OPERATIONS
                      10
                      11     ;PIN ASSIGNMENTS:
                      12     ;Port 1 bit 0 is data
                      13     ;Port 1 bit 1 is clock
                      14     ;
                      15     ;       These routines assume chip address = 0
                      16     ;
                      17     ;       The oscillator frequency assumed for this app note is 12 MHz.
                      18     ;
                      19     ;       These routines use software timing loops. They may have to be
                      20     ;       adjusted if a different oscillator frequency is used.
                      21     ;
                      22     ;NOTE 1 These NOP'S added for timing delays only on 'C' parts, OR 'LC' parts
                      23     ;       where Vcc is less than 4.5 V. and the oscillator frequency is 12 MHz.
                      24     ;       This allows a bit rate of 100kHz.
                      25     ;NOTE 2 Use these NOP's with a 16 MHz oscillator and 100kHz bit rate.
                      26     ;       For 400kHz bit rate, the NOP's in Note 1 and Note 2 are not required.
                      27     ;
                      28     ; The EEPROM will be busy after a write cycle is initiated (by a stop condition)
                      29     ; for between 1mS to 10 mS per page (or per byte if a byte write). This app note
                      30     ; assumes the user will program appropriate wait times after a write, or check
                      31     ; for Busy status. A subroutine is provided to check the Busy Status.
                      32     ;
```

```
                        ;RAM DEFINITIONS
                        ;
0030                    ORG     30H
0030                    BYTSTR: DS      20H             ;STORAGE FOR READ DATA
                        ;
                        ;CONSTANTS -- REDEFINE AS NECESSARY
                        ;
  00A0                  WTCMD   EQU     10100000B       ;WRITE DATA COMMAND Note 3
  00A1                  RDCMD   EQU     10100001B       ;READ DATA COMMAND  Note 3
  0040                  RDEND   EQU     01000000B       ;READ HIGH-ENDURANCE BLOCK NUMBER COMMAND
  0000                  ADDRH   EQU     0
  0000                  ADDRL   EQU     0
  0055                  DTA     EQU     55H
  0008                  BYTCNT  EQU     8
                        ;
                        ;Note3   Some chip or byte address bits are embedded in the control byte. Refer to
                        ;        the data sheet for exact configuration, which varies from part to part.
                        ;
                        ;*******************************************************************************
                        ; This section contains test loop routines. They form a simple operating shell to
                        ; allow the 2-wire interface code to be tested in a stand-alone mode. Using an
                        ; emulator, change  “NONE” to one of the four listed
                        ; routines to test that funcion. The address and data constants can also be set as
                        ; desired.
                        ; If using a 24XX32 or 24XX65, change the called routines by adding ‘L’ to the end
                        ; of the name. This is required because these parts use TWO address bytes. The ‘L’
                        ; routines send out the extra address byte.
                        ;*******************************************************************************
0000                    ORG     0
0000 020003                     JMP     START
                        
0003 7590FF             START:  MOV     P1,#0FFH        ;INIT PORT 1
0006 12000B                     CALL    NONE            ;TEST LOOP INSERT PROPER ADDRESS HERE
0009 80F8                       JMP     START
                        
000B 22                 NONE:   RET
                        ;*
                        ;* WRITE ONE BYTE TO EEPROM
                        ;* The Address Pointer is the address in the EEPROM. The byte to be sent to the
                        ;* EEPROM is stored in the constant ‘DTA’
                        ;*
000C 7B00               TESTWR: MOV     R3,#ADDRH       ;LOAD ADDRESS POINTER FOR 24XX32 OR 24XX65 ONLY
000E 7C00                       MOV     R4,#ADDRL       ;LOAD ADDRESS POINTER FOR ALL DEVICES
0010 7955                       MOV     R1,#DTA         ;LOAD DATA BYTE
0012 120037                     CALL    BYTEW           ;CALL  BYTE WRITE ROUTINE
0015 22                         RET
                        
```

```
                        80      ;*
                        81      ;* WRITE A BLOCK OF DATA TO EEPROM
                        82      ;* The address pointer is the address in EEPROM where data will start. The byte
                        83      ;* pointer is the starting address of  RAM containing the block of data to be sent.
                        84      ;* The byte count indicates how many bytes to send to the EEPROM.
                        85      ;* The number of bytes that can be sent before a STOP command is issued is
                        86      ;* dependant on EEPROM type. Refer to the data book for specific values.
                        87      ;*
0016 7B00               88      BLKWR:  MOV     R3,#ADDRH       ;LOAD ADDRESS POINTER FOR 24XX32 OR 24XX65 ONLY
0018 7C00               89              MOV     R4,#ADDRL       ;LOAD ADDRESS POINTER FOR ALL DEVICES
001A 7930               90              MOV     R1,# BYTSTR     ;LOAD BYTE POINTER
001C 7E08               91              MOV     R6,#BYTCNT      ;LOAD BYTE COUNT
001E 120048             92              CALL    PAGEW           ;CALL PAGE WRITE ROUTINE
0021 22                 93              RET
                        94
                        95      ;*
                        96      ;* READ ONE BYTE FROM EEPROM
                        97      ;* The address pointer is the address of the desired byte in EEPROM.
                        98      ;* The byte will be returned in R1.
                        99      ;*
0022 7B00              100      TESTRD: MOV     R3,#ADDRH       ;LOAD ADDRESS POINTER FOR 24XX32 OR 24XX65 ONLY
0024 7C00              101              MOV     R4,#ADDRL       ;LOAD ADDRESS POINTER FOR ALL DEVICES
0026 120082            102              CALL    BYTERD          ;CALL BYTE READ ROUTINE.
0029 F9                103              MOV     R1,A            ;SAVE THE BYTE
002A 22                104              RET
                       105
                       106      ;*
                       107      ;* READ A BLOCK FROM EEPROM
                       108      ;* The address pointer is the starting address of the desired data block in EEPROM.
                       109      ;* The data pointer is the starting address in RAM where data will be stored.
                       110      ;* The byte count indicates how many bytes should be read.
                       111      ;* The entire EEPROM may be read with one READ command this way.
                       112      ;*
002B 7B00              113      BLOKRD: MOV     R3,#ADDRH       ;LOAD ADDRESS POINTER FOR 24XX32 OR 24XX65 ONLY
002D 7C00              114              MOV     R4,#ADDRL       ;LOAD ADDRESS POINTER FOR ALL DEVICES
002F 7930              115              MOV     R1,#BYTSTR      ;LOAD DATA POINTER
0031 7E08              116              MOV     R6,#BYTCNT      ;LOAD BYTE COUNT
0033 12005C            117              CALL    BLKRD           ;CALL BLOCK READ ROUTINE
0036 22                118              RET
                       119
                       120      ;*END OF TEST LOOP
                       121      ;*************************************************************************************
                       122
                       123
                       124      ;*********************************************************************
                       125      ; This routine writes a byte of data to EEPROM
                       126      ; The EEPROM address is assumed to be in R4. See NOTE 3.
```

```
                        ; The DATA to be written is assumed to be in R1
                        ;*********************************************************************
0037 74A0               BYTEW:  MOV     A,#WTCMD        ;LOAD WRITE COMMAND
0039 120120                     CALL    OUTS            ;SEND IT
003C EC                         MOV     A,R4            ;GET BYTE ADDRESS
003D 120158                     CALL    OUT             ;SEND IT
0040 E9                         MOV     A,R1            ;GET DATA
0041 120158                     CALL    OUT             ;SEND IT
0044 120197                     CALL    STOP            ;SEND STOP CONDITION
0047 22                         RET

                        ;*************************************************************************
                        ; THIS ROUTINE WRITES A PAGE OF DATA TO EEPROM
                        ; The EEPROM start address is assumed to be in R4. See NOTE 3.
                        ; The DATA pointer is in R1
                        ; The BYTE count is in R6
                        ; The number of bytes that can be transfered depends on the EEPROM used.
                        ;*************************************************************************
0048 74A0               PAGEW:  MOV     A,#WTCMD        ;LOAD WRITE COMMAND
004A 120120                     CALL    OUTS            ;SEND IT
004D EC                         MOV     A,R4            ;GET LOW BYTE ADDRESS
004E 120158                     CALL    OUT             ;SEND IT
0051 E7                 BTLP:   MOV     A,@R1           ;GET DATA
0052 120158                     CALL    OUT             ;SEND IT
0055 09                         INC     R1              ;INCREMENT DATA POINTER
0056 DEF9                       DJNZ    R6,BTLP         ;LOOP TILL DONE
0058 120197                     CALL    STOP            ;SEND STOP CONDITION
005B 22                         RET


                        ;*************************************************************************
                        ; THIS ROUTINE READS A BLOCK OF DATA FROM EEPROM AT A SPECIFIED ADDRESS
                        ; EEPROM address in R4. See NOTE 3.
                        ; Stores data at RAM location pointed to by R1
                        ; Byte count specified in R6. May be 1 to 256 bytes
                        ;*************************************************************************
005C 74A0               BLKRD:  MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
005E 120120                     CALL    OUTS            ;SEND IT
0061 EC                         MOV     A,R4            ;GET LOW BYTE ADDRESS
0062 120158                     CALL    OUT             ;SEND IT
0065 74A1                       MOV     A,#RDCMD        ;LOAD READ COMMMAND
0067 120120                     CALL    OUTS            ;SEND IT
006A 12017E             BRDLP:  CALL    IN              ;READ DATA
006D F7                         MOV     @R1,A           ;STORE DATA
006E 09                         INC     R1              ;INCREMENT DATA POINTER
006F DE04                       DJNZ    R6,AKLP         ;DECREMENT LOOP COUNTER
0071 120197                     CALL    STOP            ;IF DONE, ISSUE STOP CONDITION
```

```
0074 22             174             RET                     ;DONE, EXIT ROUTINE
                    175
0075 C290           176     AKLP:   CLR     P1.0            ;NOT DONE, ISSUE ACK
0077 D291           177             SETB    P1.1
0079 00             178             NOP                     ;NOTE 1
007A 00             179             NOP
007B 00             180             NOP
007C 00             181             NOP                     ;NOTE 2
007D 00             182             NOP
007E C291           183             CLR     P1.1
0080 80E8           184             JMP     BRDLP           ;CONTINUE WITH READS
                    185
                    186     ;*********************************************************************
                    187     ; THIS ROUTINE READS A BYTE OF DATA FROM THE EEPROM
                    188     ; The EEPROM address is in R4. See NOTE 3.
                    189     ; Returns the data byte in R1
                    190     ;*********************************************************************
0082 74A0           191     BYTERD: MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
0084 120120         192             CALL    OUTS            ;SEND IT
0087 EC             193             MOV     A,R4            ;GET LOW BYTE ADDRESS
0088 120158         194             CALL    OUT             ;SEND IT
008B 12008F         195             CALL    CREAD           ;GET DATA BYTE
008E 22             196             RET
                    197
                    198     ;*********************************************************************
                    199     ; THIS ROUTINE READS A BYTE OF DATA FROM EEPROM
                    200     ; From EEPROM current address pointer.
                    201     ; Returns the data byte in R1
                    202     ;*********************************************************************
008F 74A1           203     CREAD:  MOV     A,#RDCMD        ;LOAD READ COMMMAND
0091 120120         204             CALL    OUTS            ;SEND IT
0094 12017E         205             CALL    IN              ;READ DATA
0097 F9             206             MOV     R1,A            ;STORE DATA
0098 120197         207             CALL    STOP            ;SEND STOP CONDITION
009B 22             208             RET
                    209
                    210     ;*************************************************************************
                    211     ; The next four routines are used with the 24XX32 or 24XX65 only. These parts
                    212     ; require two address bytes, and these routines send the second byte out.
                    213     ; Other than this, these routines are the same as the previous four.
                    214     ;*************************************************************************
                    215     ; THIS ROUTINE READS A BLOCK OF DATA FROM EEPROM AT A SPECIFIED ADDRESS
                    216     ; This routine is for the 24LC32 or 24LC64
                    217     ; EEPROM address in R3:R4
                    218     ; Stores data at RAM location pointed to by R1
                    219     ; Byte count specified in R6. May be 1 to 256 bytes
                    220     ;*************************************************************************
```

```
009C 74A0          221     BLKRDL: MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
009E 120120        222             CALL    OUTS            ;SEND IT
00A1 EB            223             MOV     A,R3            ;GET HI BYTE ADDRESS
00A2 120158        224             CALL    OUT             ;SEND IT
00A5 EC            225             MOV     A,R4            ;GET LOW BYTE ADDRESS
00A6 120158        226             CALL    OUT             ;SEND IT
00A9 74A1          227             MOV     A,#RDCMD        ;LOAD READ COMMMAND
00AB 120120        228             CALL    OUTS            ;SEND IT
00AE 80BA          229             JMP     BRDLP           ;CONTINUE WITH DATA READ
                   230
                   231     ;*********************************************************************
                   232     ; This routine writes a byte of data to EEPROM
                   233     ; This routine is for the 24LC32 or 24LC64
                   234     ; The EEPROM address is assumed to be in R3:R4
                   235     ; The DATA to be written is assumed to be in R1
                   236     ;*********************************************************************
00B0 74A0          237     BYTEWL: MOV     A,#WTCMD        ;LOAD WRITE COMMAND
00B2 120120        238             CALL    OUTS            ;SEND IT
00B5 EB            239             MOV     A,R3            ;GET HI BYTE ADDRESS
00B6 120158        240             CALL    OUT             ;SEND IT
00B9 EC            241             MOV     A,R4            ;GET LOW BYTE ADDRESS
00BA 120158        242             CALL    OUT             ;SEND IT
00BD E9            243             MOV     A,R1            ;GET DATA
00BE 120158        244             CALL    OUT             ;SEND IT
00C1 120197        245             CALL    STOP            ;SEND STOP CONDITION
00C4 22            246             RET
                   247
                   248     ;*************************************************************************
                   249     ; THIS ROUTINE WRITES A PAGE OF DATA TO EEPROM
                   250     ; This routine is for the 24LC32 or 24LC64
                   251     ; The EEPROM start address is assumed to be in R3:R4
                   252     ; The DATA pointer is in R1
                   253     ; The BYTE count is in R6
                   254     ; The number of bytes that can be transfered depends on the EEPROM in use.
                   255     ;*************************************************************************
00C5 74A0          256     PAGEWL: MOV     A,#WTCMD        ;LOAD WRITE COMMAND
00C7 120120        257             CALL    OUTS            ;SEND IT
00CA EB            258             MOV     A,R3            ;GET HI BYTE ADDRESS
00CB 120158        259             CALL    OUT             ;SEND IT
00CE EC            260             MOV     A,R4            ;GET LOW BYTE ADDRESS
00CF 120158        261             CALL    OUT             ;SEND IT
00D2 E7            262     BTLPL:  MOV     A,@R1           ;GET DATA
00D3 120158        263             CALL    OUT             ;SEND IT
00D6 09            264             INC     R1              ;INCREMENT DATA POINTER
00D7 DEF9          265             DJNZ    R6,BTLPL        ;LOOP TILL DONE
00D9 120197        266             CALL    STOP            ;SEND STOP CONDITION
00DC 22            267             RET
```

```
                        268
                        269     ;*********************************************************************
                        270     ; THIS ROUTINE READS A BYTE OF DATA FROM THE EEPROM
                        271     ; This routine is for the 24LC32 or 24LC64
                        272     ; The EEPROM address is in R3:R4
                        273     ; Returns the data byte in R1
                        274     ;*********************************************************************
00DD 74A0               275     BYTERDL:        MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
00DF 120120             276             CALL    OUTS            ;SEND IT
00E2 EB                 277             MOV     A,R3            ;GET HI BYTE ADDRESS
00E3 120158             278             CALL    OUT             ;SEND IT
00E6 EC                 279             MOV     A,R4            ;GET LOW BYTE ADDRESS
00E7 120158             280             CALL    OUT             ;SEND IT
00EA 118F               281             CALL    CREAD           ;GET DATA BYTE
00EC 22                 282             RET
                        283
                        284     ;
                        285     ;SUBROUTINES
                        286     ;
                        287     ;*********************************************************************
                        288     ; This routine tests for WRITE DONE condition
                        289     ; by testing for an ACK.
                        290     ; This routine can be run as soon as a STOP condition
                        291     ; has been generated after the last data byte has been
                        292     ; sent to the EEPROM. No ACK will be returned until
                        293     ; the EEPROM is done with the write operation.
                        294     ;*********************************************************************
00ED 74A0               295     ACKTST: MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
00EF 7A08               296             MOV     R2,#8           ;LOOP COUNT  -- EQUAL TO BIT COUNT
00F1 C290               297             CLR     P1.0            ;START CONDITION -- DATA = 0
00F3 00                 298             NOP                     ;NOTE 1
00F4 00                 299             NOP
00F5 00                 300             NOP
00F6 00                 301             NOP                     ;NOTE 2
00F7 00                 302             NOP
00F8 C291               303             CLR     P1.1            ;CLOCK = 0
00FA 33                 304     AKTLP:  RLC     A               ;SHIFT BIT
00FB 5005               305             JNC     AKTLS
00FD D290               306             SETB    P1.0            ;DATA = 1
00FF 020104             307             JMP     AKTL1           ;CONTINUE
0102 C290               308     AKTLS:  CLR     P1.0            ;DATA = 0
0104 D291               309     AKTL1:  SETB    P1.1            ;CLOCK HI
0106 00                 310             NOP                     ;NOTE 1
0107 00                 311             NOP
0108 00                 312             NOP
0109 00                 313             NOP                     ;NOTE 2
010A 00                 314             NOP
```

```
010B C291           315             CLR     P1.1            ;CLOCK LOW
010D DAEB           316             DJNZ    R2,AKTLP        ;DECREMENT COUNTER
010F D290           317             SETB    P1.0            ;TURN PIN INTO INPUT
0111 00             318             NOP                     ;NOTE 1
0112 00             319             NOP                     ;NOTE 2
0113 D291           320             SETB    P1.1            ;CLOCK ACK
0115 00             321             NOP                     ;NOTE 1
0116 00             322             NOP
0117 00             323             NOP
0118 00             324             NOP                     ;NOTE 2
0119 00             325             NOP
011A 309002         326             JNB     P1.0,EXIT       ;EXIT IF NO ACK (WRITE NOT DONE)
011D C291           327             CLR     P1.1
                    328     ;                               ;SET DONE FLAG
011F 22             329     EXIT:   RET
                    330
                    331
                    332
                    333     ;*************************************************************************
                    334     ; THIS ROUTINE SENDS OUT CONTENTS OF THE ACCUMULATOR
                    335     ; to the EEPROM and includes START condition. Refer to the data sheets
                    336     ;for discussion of START and STOP conditions.
                    337     ;*************************************************************************
                    338
0120 7A08           339     OUTS:   MOV     R2,#8           ;LOOP COUNT  -- EQUAL TO BIT COUNT
0122 D290           340             SETB    P1.0            ;INSURE DATA IS HI
0124 D291           341             SETB    P1.1            ;INSURE CLOCK IS HI
0126 00             342             NOP                     ;NOTE 1
0127 00             343             NOP
0128 00             344             NOP
0129 00             345             NOP                     ;NOTE 2
012A 00             346             NOP
012B C290           347             CLR     P1.0            ;START CONDITION -- DATA = 0
012D 00             348             NOP                     ;NOTE 1
012E 00             349             NOP
012F 00             350             NOP
0130 00             351             NOP                     ;NOTE 2
0131 00             352             NOP
0132 C291           353             CLR     P1.1            ;CLOCK = 0
0134 33             354     OTSLP:  RLC     A               ;SHIFT BIT
0135 5005           355             JNC     BITLS
0137 D290           356             SETB    P1.0            ;DATA = 1
0139 02013E         357             JMP     OTSL1           ;CONTINUE
013C C290           358     BITLS:  CLR     P1.0            ;DATA = 0
013E D291           359     OTSL1:  SETB    P1.1            ;CLOCK HI
0140 00             360             NOP                     ;NOTE 1
0141 00             361             NOP
```

```
0142 00            362             NOP
0143 00            363             NOP                     ;NOTE 2
0144 00            364             NOP
0145 C291          365             CLR     P1.1            ;CLOCK LOW
0147 DAEB          366             DJNZ    R2,OTSLP        ;DECREMENT COUNTER
0149 D290          367             SETB    P1.0            ;TURN PIN INTO INPUT
014B 00            368             NOP                     ;NOTE 1
014C 00            369             NOP                     ;NOTE 2
014D 00            370             NOP
014E D291          371             SETB    P1.1            ;CLOCK ACK
0150 00            372             NOP                     ;NOTE 1
0151 00            373             NOP
0152 00            374             NOP
0153 00            375             NOP                     ;NOTE 2
0154 00            376             NOP
0155 C291          377             CLR     P1.1
0157 22            378             RET
                   379
                   380     ;*************************************************************************
                   381     ; THIS ROUTINE SENDS OUT CONTENTS OF ACCUMULATOR TO EEPROM
                   382     ; without sending a START condition.
                   383     ;*************************************************************************
                   384
0158 7A08          385     OUT:    MOV     R2,#8           ;LOOP COUNT  -- EQUAL TO BIT COUNT
015A 33            386     OTLP:   RLC     A               ;SHIFT BIT
015B 5005          387             JNC     BITL
015D D290          388             SETB    P1.0            ;DATA = 1
015F 020164        389             JMP     OTL1            ;CONTINUE
0162 C290          390     BITL:   CLR     P1.0            ;DATA = 0
0164 D291          391     OTL1:   SETB    P1.1            ;CLOCK HI
0166 00            392             NOP                     ;NOTE 1
0167 00            393             NOP
0168 00            394             NOP
0169 00            395             NOP                     ;NOTE 2
016A 00            396             NOP
016B C291          397             CLR     P1.1            ;CLOCK LOW
016D DAEB          398             DJNZ    R2,OTLP         ;DECREMENT COUNTER
016F D290          399             SETB    P1.0            ;TURN PIN INTO INPUT
0171 00            400             NOP                     ;NOTE 1
0172 00            401             NOP                     ;NOTE 2
0173 00            402             NOP
0174 D291          403             SETB    P1.1            ;CLOCK ACK
0176 00            404             NOP                     ;NOTE 1
0177 00            405             NOP
0178 00            406             NOP
0179 00            407             NOP                     ;NOTE 2
017A 00            408             NOP
```

```
017B C291           409             CLR     P1.1
017D 22             410             RET
                    411
                    412     ;*************************************************************************
                    413     ; THIS ROUTINE READS IN A BYTE FROM THE EEPROM
                    414     ; and stores it in the accumulator
                    415     ;*************************************************************************
                    416
017E 7A08           417     IN:     MOV     R2,#8           ;LOOP COUNT
0180 D290           418             SETB    P1.0            ;SET DATA BIT HIGH FOR INPUT
0182 C291           419     INLP:   CLR     P1.1            ;CLOCK LOW
0184 00             420             NOP                     ;NOTE 1
0185 00             421             NOP
0186 00             422             NOP
0187 00             423             NOP
0188 00             424             NOP                     ;NOTE 2
0189 00             425             NOP
018A D291           426             SETB    P1.1            ;CLOCK HIGH
018C C3             427             CLR     C               ;CLEAR CARRY
018D 309001         428             JNB     P1.0,INL1       ;JUMP IF DATA =0
0190 B3             429             CPL     C               ;SET CARRY IF DATA =1
0191 33             430     INL1:   RLC     A               ;ROTATE DATA INTO ACCUMULATOR
0192 DAEE           431             DJNZ    R2,INLP         ;DECREMENT COUNTER
0194 C291           432             CLR     P1.1            ;CLOCK LOW
0196 22             433             RET
                    434
                    435
0197 C290           436     STOP:   CLR     P1.0            ;STOP CONDITION SET DATA LOW
0199 00             437             NOP                     ;NOTE 1
019A 00             438             NOP
019B 00             439             NOP
019C 00             440             NOP                     ;NOTE 2
019D 00             441             NOP
019E D291           442             SETB    P1.1            ;SET CLOCK HI
01A0 00             443             NOP                     ;NOTE 1
01A1 00             444             NOP
01A2 00             445             NOP
01A3 00             446             NOP                     ;NOTE 2
01A4 00             447             NOP
01A5 D290           448             SETB    P1.0            ;SET DATA HI
01A7 22             449             RET
                    450
                    451     ;*************************************************************************
                    452     ;These routines contain special commands only for the 24LC65 SMART SERIAL EEPROM
                    453     ;*
                    454     ; SET SECURE BLOCK
                    455     ; ASSUMES START BLOCK 0 & BLOCK LENGTH OF 1
```

```
                        ; The numbers are implicit in the commands. Refer to the data sheet for details.
                        ;*
01A8 7B80               SETSEC: MOV     R3,#80H         ;LOAD COMMAND AND STARTING BLOCK NUMBER
01AA 7C00                       MOV     R4,#0
01AC 7981                       MOV     R1,#81H         ;SET COMMAND FOR NUMBER OF BLOCKS TO SECURE
01AE 1137                       CALL    BYTEW           ;EXECUTE
01B0 22                         RET
                        ;*
                        ; READ SECURE BLOCK NUMBER(S)
                        ; RETURNS BLOCK NUMBER IN R1 AND BLOCK COUNT IN R2
                        ; (UPPER NIBBLES WILL BE 1'S)
                        ;*
01B1 74A0               RDSEC:  MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
01B3 3120                       CALL    OUTS            ;SEND IT
01B5 7480                       MOV     A, #80H         ;LOAD COMMAND
01B7 3158                       CALL    OUT             ;SEND IT
01B9 7400                       MOV     A,#0            ;LOAD COMMAND
01BB 3158                       CALL    OUT             ;SEND IT
01BD 74C0                       MOV     A,#0C0H         ;LOAD COMMMAND
01BF 3158                       CALL    OUT             ;SEND IT
01C1 317E                       CALL    IN              ;READ STARTING BLOCK NUMBER
01C3 F9                         MOV     R1,A            ;STORE IT
01C4 00                         NOP                     ;NOTE 1
01C5 00                         NOP
01C6 00                         NOP
01C7 00                         NOP                     ;NOTE 2
01C8 00                         NOP
01C9 C290                       CLR     P1.0            ;ISSUE ACK
01CB D291                       SETB    P1.1
01CD 00                         NOP                     ;NOTE 1
01CE 00                         NOP
01CF 00                         NOP
01D0 00                         NOP                     ;NOTE 2
01D1 00                         NOP
01D2 C291                       CLR     P1.1
01D4 317E                       CALL    IN              ;READ NUMBER OF BLOCKS
01D6 FA                         MOV     R2,A            ;STORE IT
01D7 3197                       CALL    STOP            ;SEND STOP CONDITION
01D9 22                         RET
                        ;*
                        ; SET HIGH-ENDURANCE BLOCK NUMBER
                        ; ASSUMES BLOCK 0
                        ;*
```

```
01DA 7B80            503     SETHI:  MOV     R3,#80H         ;LOAD COMMAND AND BLOCK NUMBER
01DC 7C00            504             MOV     R4,#0
01DE 7900            505             MOV     R1,#0           ;SET DATA =0
01E0 1137            506             CALL    BYTEW           ;EXECUTE
01E2 22              507             RET
                     508
                     509     ;*
                     510     ; READ HIGH-ENDURANCE BLOCK NUMBER
                     511     ; RETURNS BLOCK NUMBER IN R1 (UPPER NIBBLE WILL BE 1'S)
                     512     ;*
01E3 7B80            513     READHI: MOV     R3,#80H         ;LOAD COMMAND
01E5 7C00            514             MOV     R4,#0
01E7 1201EB          515             CALL    HIEND           ;EXECUTE
01EA 22              516             RET
                     517
01EB 74A0            518     HIEND:  MOV     A,#WTCMD        ;LOAD WRITE COMMMAND TO SEND ADDRESS
01ED 3120            519             CALL    OUTS            ;SEND IT
01EF EB              520             MOV     A,R3            ;GET HI BYTE ADDRESS
01F0 3158            521             CALL    OUT             ;SEND IT
01F2 EC              522             MOV     A,R4            ;GET LOW BYTE ADDRESS
01F3 3158            523             CALL    OUT             ;SEND IT
01F5 7440            524             MOV     A,#RDEND        ;LOAD READ COMMMAND
01F7 3158            525             CALL    OUT             ;SEND IT
01F9 317E            526             CALL    IN              ;READ DATA
01FB F9              527             MOV     R1,A            ;STORE DATA
01FC 3197            528             CALL    STOP            ;SEND STOP CONDITION
01FE 22              529             RET
                     530
                     531     ;END of 24LC65 Routines
                     532     ;*********************************************************************
                     533     END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

NOTES:

NOTES:

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microchip

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95

MICROCHIP