



Interfacing 93CX6 Serial EEPROMs to PIC16C5X Microcontrollers

INTRODUCTION

Microchip Technology Inc.'s popular 93C46/56/66 and 93LC46/56/66 Serial EEPROMs feature a three/four wire serial interface bus. The attractive price and simple interface make it the ideal device for additional memory space. This application note is intended for design engineers who wish to incorporate a pre-packaged serial EEPROM interface driver into their application.

THE HARDWARE CONNECTION

A typical 4-wire hardware connection is illustrated in Figure 1A and a typical 3-wire connection is illustrated in Figure 1B. Since all I/O ports on the PIC16C5X are configurable as input and/or output, a 3-wire interface makes optimum utilization of the I/O pins by having a common connection for the DI and DO lines of the Serial EEPROM. The port pin on the PIC16C5X connected to these pins, has a default setting as an output and is configured, when needed, as an input during program execution.

THE SOFTWARE CONNECTION

An example interface driver is listed in Appendix A. A flow diagram is given in Figure 2. The interface driver is written to minimize both overhead to the calling program as well as the program space necessary for its inclusion into the user's code. The driver has been written as a generic driver to service all 93 Series Serial EEPROMs made by Microchip. Processor resources which must be made available to the driver prior to being called are: 1) Two levels of processor stack. 2) Six register locations (four for command/data passing and two for software counters). 3) The File Select Register (FSR), which is used to pass a command/data string pointer to the driver.

Note: The four command/data passing registers have to be defined consecutively in order for the FSR to access them successfully in the program execution.

The user should take the following steps when using the routines provided in Appendix A.

- A) Specify and define a 3-/4-wire interface by defining the common connection to the DI/DO lines and setting the equate 'wire3' TRUE or FALSE (4-wire is automatically assumed if 3-wire is false).
- B) Specify and define if 16-bit or 8-bit data organization is used, by setting equate 'org8' TRUE or FALSE.
- C) The user should assemble the source file by specifying which type of serial EEPROM is being used.

FIGURE 1A - 4 WIRE CONNECTION

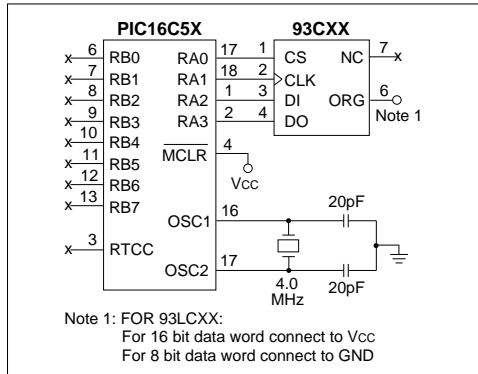
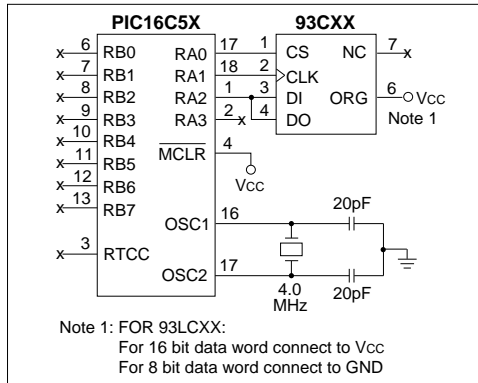


FIGURE 1B - 3 WIRE CONNECTION



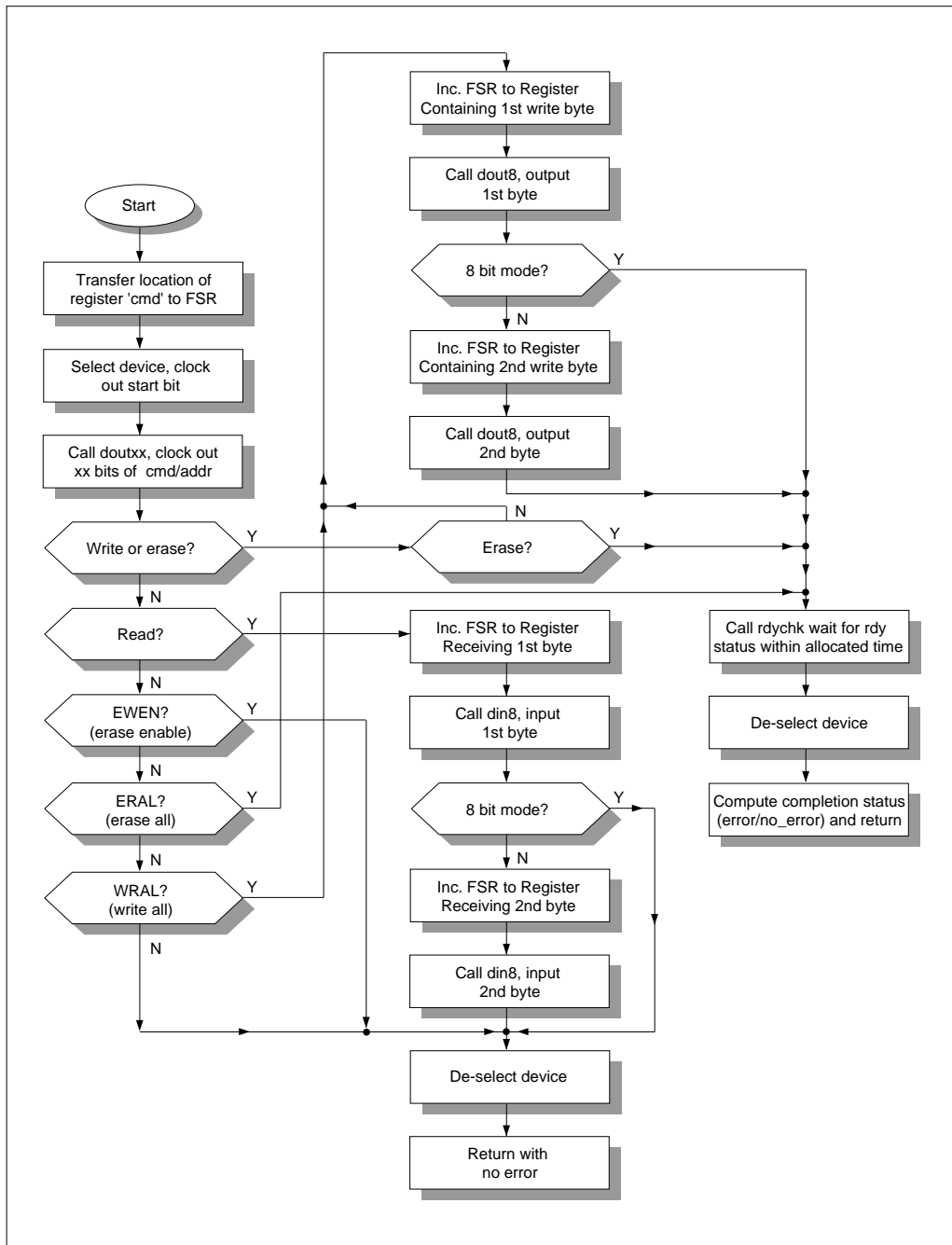
This is done by defining the equate S93C46, S93LC46, etc., as TRUE. Only one device can be TRUE, the rest have to be defined FALSE.

- D) The user would invoke the driver as follows:
 1. Load the register defined as 'cmd' with the 93CXX Command Opcode (only the four upper bits are used in this register; see Figure 3).
 2. If necessary, load the register defined 'addr' with the lower 8/7/6 bit address of the location.
 3. If necessary, load the 9th bit of the address as bit 3 of the register defined as 'cmd' (see Figure 3).

Note: READ, WRITE and ERASE commands need to have an address associated with the command and the 9th bit of the address is only

Interfacing 93 Series Serial EEPROMs

FIGURE 2 - FLOW CHART



Interfacing 93 Series Serial EEPROMs

required when using the 93C56/66 or 93LC56/66 devices in the 8 bit mode (ORG tied to GND).

4. If necessary, load the register defined as 'highb' and 'lowb' with the 16 bits of data, the most significant byte loaded in 'highb'. In 8 bit mode, 'highb' should be loaded with the 8 bit data.

Note: Only the WRITE and WRAL commands require data to be loaded in the 'highb', 'lowb' locations.

5. Call the driver sub-routine 'see'.
6. Upon completion, the driver will return a completion status in the W register (error/no error). Only commands requiring a status check are capable of returning a valid error/no error status, in all other cases a no error is returned.
7. If the READ command is executed, the 16/8 bit data will be loaded in the 'highb' and 'lowb' registers, where 'highb' contains the MSB in the 16 bit mode and 8 bit data in the 8-bit mode.

The Example interface assumes a 4 MHz oscillator clock which gives us a 1 μ S instruction cycle. If a higher clock speed is used, additional NOPs have to be included in the code in order to meet the minimum clock speed requirements of the 93 Series Serial EEPROMs (see data sheet for further details).

Listing in Appendix B is for an interface to 93C46 Serial EEPROM only.

SUMMARY

The 93 Series Serial EEPROMs are a simple and versatile method of increasing read/write memory capability in a PIC16C5X application. The 'generic' code in Appendix A makes it easy to incorporate in any PIC16C5X application. Any of the 93 Series Serial EEPROMs can be applied, while at the same time using a minimal amount of I/O, code and RAM resources.

Code size: 6 bytes of RAM.

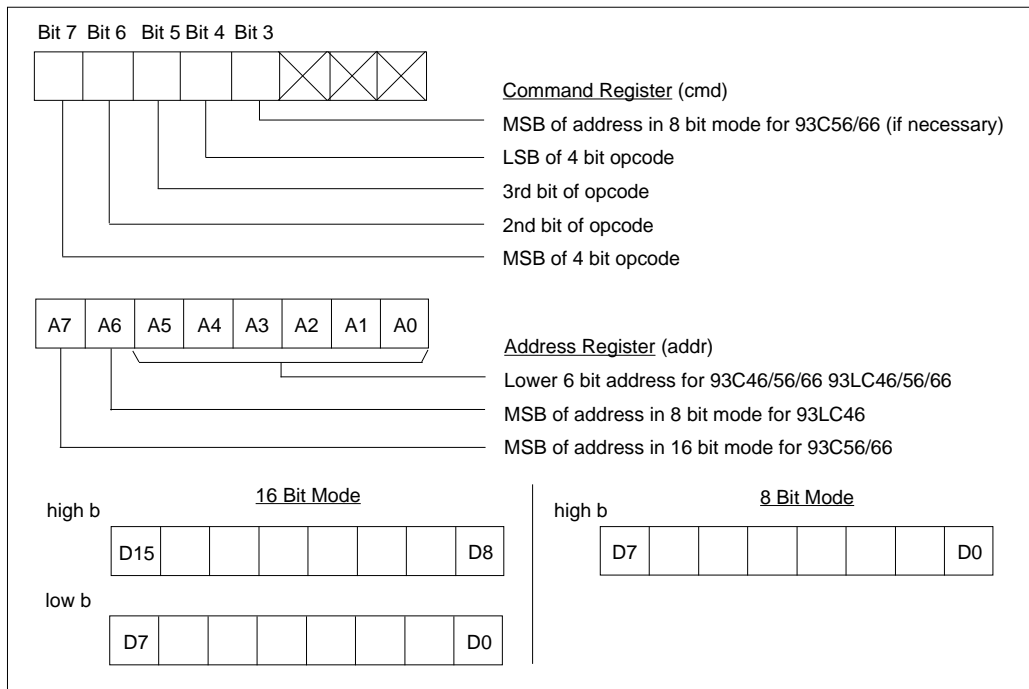
Appendix A listing: 127 bytes program code (max.)

100 bytes program code (min.)

Appendix B listing: 86 bytes program code

*Authors: Stan D'Souza
Logic Products Division
Bob Ward
Field Applications/Sales Organization*

FIGURE 3 - 'CMD', 'ADDR', DATA BYTE DEFINITION



Interfacing 93 Series Serial EEPROMs

Appendix A

Microchip Technology 16c5X Assembler 3.03A Wed Feb 26 11:09:07 1992 Page 1
R/W EEPROM

```
Line  PC  Opcode

0001          LIST  P = 16C54
0002
0003          ;
0004          ;Define Equates:
0005          ;
0005          01FF  PIC54  EQU  1FFH
0006          ;
0007          ;
0008          0000          ORG  0
0009          START
0010          0000          0A7B  goto  main      ;run test program
0011          ;
0012          ;
0013          0001          TRUE  EQU  1
0014          0000          FALSE EQU  0
0015          0000          S93C46 EQU  FALSE
0016          0000          S93LC46 EQU  FALSE
0017          0000          S93C56 EQU  FALSE
0018          0000          S93LC56 EQU  FALSE
0019          0001          S93C66 EQU  TRUE
0020          0000          S93LC66 EQU  FALSE
0021          0001          wire3 equ  TRUE      ;for four-wire setup equate to FALSE
0022          0000          org8  EQU  FALSE
0023          ;
0024          0001          H16   EQU  TRUE
0025          0000          H8    EQU  FALSE
0026          0000          LC468 EQU  FALSE
0027          0000          XC46  EQU  FALSE
0028
0029
0030
0031          ;*****
0032          ;*          Register Assignments
0033          ;*
0034          ;*****
0035          0000          indir equ  0          ;Use this register as source/destination
0036          ;for indirect addressing.
0037          0002          pc     equ  2          ;PIC16/17 Program Counter.
0038          0003          status equ  3          ;PIC16/17 Status Register.
0039          0004          fsr   equ  4          ;File Select Register.
0040          0005          serial equ  5          ;Port used for 93CX6 control.
0041          ;The following four registers must be
0042          ;located consecutively in memory.
0043          001A          cmd    equ  1a         ;This register contains the 4 bit
0044          ;command op code for 93CX6 as follows:
0045          ;bit 7 msb of command op code
0046          ;bit 6 next bit of op code
0047          ;bit 5 next bit of op code
0048          ;bit 4 lsb of op code
0049          ;bit 3 A8 of address in case of
0050          ;56/66 in 8 bit mode.
0051          001B          addr   equ  1b         ;memory address of lower 7/8 bits
0052          001C          highb  equ  1c         ;Used in read/write routines to store the
0053          ;upper byte of a 16 bit data word,
0054          ;or the data in a 8 bit data word
0055          001D          lowb   equ  1d         ;Used in read/write routines to store the
0056          ;lower byte of a 16 bit data word,
0057          ;or not used in 8 bit data word.
0058
0059          001E          cnthi  equ  1e         ;Used as the upper byte of a sixteen bit
0060          ;loop counter in RDYCHK routine.
```

Interfacing 93 Series Serial EEPROMs

```
0061      001F      cnt      equ      1f          ;Used as the lower byte of a sixteen bit
0062                                          ;loop counter in RDYCHK routine, and
0063                                          ;elsewhere as an eight bit counter.
0064      001E      temp_cmd equ      1e          ;doubles as a temp register for cmd
0065      001F      temp_addr equ     1f          ;doubles as a temp register for addr
0066
0067      ;*****
0068      ;*                               Bit Assignments
0069      ;* The following assignments are for 3-wire. For 4-wire please assign
0070      ;* din and dout to two separate pins.
0071      0000      carry   equ      0           ;Carry Flag of Status Register.
0072      0002      zflag   equ      2           ;Zero Flag of Status Register.
0073
0074      0002      cs      equ      2           ;Port pin tied to CS on 93CX6.
0075      0001      din     equ      1           ;Port pin tied to DI on 93CX6. 3-wire setup
0076      0001      dout    equ      1           ;Port pin tied to DO on 93CX6. 3-wire setup
0077      0003      cclock  equ      3           ;Port pin tied to CLK on 93CX6.
0078
0079      ;*****
0080      ;*                               General Assignments
0081      ;*
0082      ;*****
0083      0000      no_err  equ      0           ;After issuing a WRITE, ERASE, ERAL, or WRAL
0084      0001      error   equ      1           ;command, the approximate number of machine
0085      0020      tries  equ      20          ;cycles X 256 to wait for the RDY status.
0086                                          ;This value must be adjusted for operating
0087                                          ;frequencies other than 4 MHz.
0088
0089
0090
0091      0080      read    equ      b'10000000' ;read command op code
0092      0040      write   equ      b'01000000' ;write command op code
0093      00C0      erase   equ      b'11000000' ;erase command op code
0094      0030      ewen    equ      b'00110000' ;erase enable command opcode
0095      0000      ewds    equ      b'00000000' ;erase disable command opcode
0096      0020      eral    equ      b'00100000' ;erase all command op code
0097      0010      wral    equ      b'00010000' ;write all command op code
0098
0099      ;*****
0100      ;*                               Macro Definitions
0101      ;*
0102      ;*****
0103
0103      sel      MACRO                               ;Selects the 93CX6 device
0104      bsf      serial,cs                          ;Chip Select (CS) = '1' to select
0105      ENDM                                         ;the device
0106
0107      dsel     MACRO                               ;De-select the 93CX6 device.
0108      bcf      serial,cs                          ;Chip Select (CS) = '0' to de-select
0109      ;the device.
0110      ENDM
0111
0112      strtbt   MACRO                               ;Issue the Start Bit to the 93CX6.
0113      bsf      serial,din                         ;Start Bit = '1'.
0114      clkkit   ;Clock it out.
0115      ENDM
0116
0117      clkkit   MACRO                               ;Clocks a serial data bit into or out
0118      ;of the 93CX6 device.
0119      bsf      serial,clock                       ;Clock (CLK) = '1'.
0120
0121      nop
0122      ;Adjust the number of nop instructions
0123      ;between the assertion and de-assertion of
0124      ;CLK in proportion to the PIC16/17 operating
0125      ;frequency. Refer to the 93CX6 data for the
0126      ;minimum CLK period.
```

Interfacing 93 Series Serial EEPROMs

```

0127         bcf     serial,clock      ;Clock (CLK) = '0'.
0128         ENDM
0129         ;
0130         ;*****
0131         ;*           DOUTx
0132         ;*
0133         ;*****
0134         ;doutxx, outputs up to 11 bits of op code/data, depending on whether
0135         ;a 46/56/66 serial eeprom is being used. The number of bits over 8 are
0136         ;saved in the cmd register and the rest in the addr register. Before
0137         ;calling this routine the cmd and the addr registers should be loaded
0138         ;as follows:
0139         ;cmd reg.bits 7|6|5|4|3|2|1|0
0140         ;-----|---|---|---|---|---|
0141         ;           X|X|X|X|X|X|X|Y  -> not used
0142         ;           X|X|X|X|X|X|Y|X  -> mot used
0143         ;           X|X|X|X|X|Y|X|X  -> not used
0144         ;           X|X|X|Y|X|X|X|X  -> 9th bit of address if necessary
0145         ;           X|X|Y|X|X|X|X|X  -> 1sb of command op code
0146         ;           X|Y|X|X|X|X|X|X  -> 3rd bit of command opcode
0147         ;           Y|X|X|X|X|X|X|X  -> 2nd bit of command opcode
0148         ;
0149         ;addr reg. 6/7/8 bits of address if necessary.
0150
0151         dout10
0152         0001 0360         rlf     indir      ;rotate thru carry
0153         0002 0425         bcf     serial,din   ;set output to 0
0154         0003 0603         btfsc   status,carry ;set?
0155         0004 0525         bsf     serial,din   ;else set output to 1
0156         clkkit          ;clk data
0157         m
0158         m
0159         0005 0565         bsf     serial,clock ;of the 93CX6 device.
0160         m
0161         0006 0000         nop
0162         m
0163         m
0164         m
0165         m
0166         m
0167         0007 0465         bcf     serial,clock ;Clock (CLK) = '0'.
0168         dout9
0169         0008 0360         rlf     indir      ;rotate thru carry
0170         0009 0425         bcf     serial,din   ;set output to 0
0171         000A 0603         btfsc   status,carry ;set?
0172         000B 0525         bsf     serial,din   ;else set output to 1
0173         clkkit          ;clk data
0174         m
0175         m
0176         000C 0565         bsf     serial,clock ;of the 93CX6 device.
0177         m
0178         000D 0000         nop
0179         m
0180         m
0181         m
0182         m
0183         m
0184         000E 0465         bcf     serial,clock ;Clock (CLK) = '0'.
0185         000F 02A4         incf   fsr
0186         dout8
0187         0010 0C08         movlw  8
0188         0011 003F         movwf  cnt
0189         ;
0190         0012 0425         d_o_8 bcf     serial,din   ;Assume that the bit to be transferred is a
0191         '0'. Hence, de-assert DI.
0192         0013 0360         rlf     indir      ;Rotate the actual bit to be transferred into
0193         ;the carry bit.
0194         0014 0603         btfsc  status,carry ;Test the carry, if our assumption was
0195         ;correct, skip the next instruction.

```

Interfacing 93 Series Serial EEPROMs

```

0196 0015 0525      bsf    serial,din    ;No, actual bit was a '1'. Assert DI.
0197              clklt          ;Clock the 93CX6.
0198              m
0199              m
0200 0016 0565      bsf    serial,clock ;of the 93CX6 device.
0201              m                ;Clock (CLK) = '1'.
0202 0017 0000      nop
0203              m                ;Adjust the number of nop instructions
                                ;between the assertion and de-assertion of
                                ;
0204              m                ;CLK in proportion to the PIC16/17 operating
0205              m                ;frequency. Refer to the 93CX6 data for the
0206              m                ;minimum CLK period.
0207              m
0208 0018 0465      bcf    serial,clock ;Clock (CLK) = '0'.
0209 0019 02FF      decfsz cnt        ;Repeat until cnt = 0.
0210 001A 0A12      goto   d_o_8        ;Cnt still > 0.
0211 001B 0360      rlf    indir        ;Restore register to its original condition.
0212 001C 0425      bcf    serial,din    ;make sure din is low
0213 001D 0800      retlw   no_err       ;Exit with good status.
0214
0215                ;*****
0216                ;*                               DIN8
0217                ;*
0218                ;*****
0219                ;Din8 will input 8 bits of data from the
0220                ;93CX6. Before calling this routine, the FSR
0221                ;must point to the register being used to
0222                ;hold the incoming data.
0223                din8
0224 001E 0C02      movlw  b'0000010'    ;set up the RAL as a input before proceeding
0225 001F 0005      tris   serial        ;set up porta
0226 0020 0C08      movlw  8                ;Initialize loop counter.
0227 0021 003F      movwf  cnt                ;
0228
0229                d_i_8
0230 0022 0360      rlf    indir        ;Make room for the incoming bit in the
0231                ;destination register.
0232 0023 0400      bcf    indir,0        ;Assume that the incoming bit is a '0' and
0233                ;clear the LSB of the destination register.
0234                clklt          ;Clock a bit in the 93CX6.
0235                m
0236                m                ;of the 93CX6 device.
0237 0024 0565      bsf    serial,clock ;Clock (CLK) = '1'.
0238                m
0239 0025 0000      nop
0240                m                ;Adjust the number of nop instructions
0241                m                ;between the assertion and de-assertion of
0242                m                ;CLK in proportion to the PIC16/17 operating
0243                m                ;frequency. Refer to the 93CX6 data for the
0244                m                ;minimum CLK period.
0245 0026 0465      bcf    serial,clock ;Clock (CLK) = '0'.
0246 0027 0625      btfsz  serial,dout    ;Test the incoming bit, if our assumption
0247                ;was correct, skip the next instruction.
0248 0028 0500      bsf    indir,0        ;No, actual bit is a '1'. Set the LSB of the
0249                ;destination register.
0250 0029 02FF      decfsz cnt        ;Repeat until cnt = 0.
0251 002A 0A22      goto   d_i_8        ;Cnt still > 0.
0252                ;setup RAL back to output
0253 002B 0C00      movlw  0                ;set RAL as output
0254 002C 0005      tris   serial        ; /
0255 002D 0800      retlw   no_err       ;Exit with good status.
0256
0257                ;*****
0258                ;*                               RDYCHK
0259                ;*
0260                ;*****
0261                ;Rdychk will read the 93CX6 READY/BUSY status
0262                ;and wait for RDY status within the allotted
                                ;number of processor cycles. If RDY status

```

Interfacing 93 Series Serial EEPROMs

```
0263                                     ;is not present after this set period, the
0264                                     ;routine will return with an error status.
0265
0266
0267     rdychk
0268     002E 0C02             movlw    b'00000010'    ;set up RAL as a input before proceeding
0269     002F 0005             tris    serial        ;set up porta
0270     0030 0C20             movlw    tries         ;
0271     0031 003E             movwf   cnthi         ;Initialize time-out counter
0272     0032 007F             clrfsz  cnt           ;
0273                                     ;De-select the 93CX6.
0274
0275     0033 0445             m        bcf    serial,cs    ;Chip Select (CS) = '0' to de-select
0276                                     ;the device.
0277
0278     ;        nop          ;NOTE: Check the 93CX6 data sheet for
0279                                     ;minimum CS low time. Depending upon
0280                                     ;processor frequency, a nop(s) may be
0281                                     ;between the assertion and de-assertion of
0282                                     ;Chip Select.
0283
0284                                     sel          ;Re-select the 93CX6.
0285
0286     0034 0545             m        bsf    serial,cs    ;Chip Select (CS) = '1' to select
0287     0035 0625             notrdy  btfsc  serial,dout   ;If DO is a '0', 93CX6 has yet to completed
0288                                     ;the last operation (still busy).
0289     0036 0A3E             goto    rdynoerr        ;skip to no error
0290     0037 02FF             decfsz  cnt           ;No, not yet ready. Decrement the LSB of
0291                                     ;16 bit timer and check for expiration.
0292     0038 0A35             goto    notrdy         ;Still some time left. Try again.
0293     0039 02FE             decfsz  cnthi         ;Least significant byte expired - decrement
0294                                     ;and check for expiration of the MSB.
0295     003A 0A35             goto    notrdy         ;Still some time left. Try again.
0296                                     ;setup RAL back to output
0297     003B 0C00             movlw    0            ;set RAL as output
0298     003C 0005             tris    serial        ;
0299     003D 0801             retlw   error         ;RDY status was not present in the allotted
0300                                     ;time, return with error status.
0301
0302     rdynoerr
0303     003E 0C00             movlw    0            ;setup RAL back to output
0304     003F 0005             tris    serial        ;set porta as output
0305     0040 0800             retlw   no_err       ;
0306
0307     ;*****
0308     ;*                               SEE
0309     ;*
0310     ;*****
0311
0312     ;See will control the entire operation of a
0313     ;93CX6 device. Prior to calling the routine,
0314     ;load a valid command/memory address into
0315     ;location cmd, and for WRITE or WRAL
0316     ;commands, load registers highb and lowb with
0317     ;16 bits of write data. Upon exit, the W
0318     ;register will contain the completion status.
0319     ;Only 93CX6 instructions which require a
0320     ;status check can return with an error as the
0321     ;completion status. The values that denote
0322     ;the completion status are defined as
0323     ;variables 'error' and 'no_err' in the
0324     ;general assignments section.
0325
0326     see
0327     0041 021A             movf    cmd,w         ;save cmd
0328     0042 003E             movwf  temp_cmd      ;
0329     0043 021B             movf    addr,w       ;save addr
0330     0044 003F             movwf  temp_addr     ;
0331     0045 0C1A             movlw  cmd            ;Load W with the location of the cmd
0332                                     ;register.
```


Interfacing 93 Series Serial EEPROMs

```

0332 0046 0024      movwf   fsr           ;Transfer that information into the File
0333                                     ;Select Register. The fsr now points to
0334                                     ;location cmd.
0335                                     ;Select the 93CX6.
0336                                     m
0337 0047 0545      bsf     serial,cs    ;Chip Select (CS) = '1' to select
0338      strtbt        ;Send a start bit.
0339                                     m
0340 0048 0525      bsf     serial,din   ;Start Bit = '1'.
0341      clkkit        ;Clock it out.
0342                                     m
0343                                     m
0344 0049 0565      bsf     serial,clock ;Clock (CLK) = '1'.
0345                                     m
0346 004A 0000      nop                    ;Adjust the number of nop instructions
0347                                     m
0348                                     m
0349                                     m
0350                                     m
0351                                     m
0352 004B 0465      bcf     serial,clock ;Clock (CLK) = '0'.
0353                                     ;
0354 004C 06FA      btfsc  cmd,7           ;bit 7 = 0?
0355 004D 0A73      goto    sca10        ;xfer 10 bit cmd/adr
0356                                     ;
0357                                     ;
0358 004E 07DA      btfss  cmd,6           ;bit 6 = 0 ?
0359 004F 0A75      goto    sc10         ;xfer 10 bit cmd/adr
0360                                     ;
0361 0050 0A73      goto    sca10        ;xfer 10 bit cmd/adr
0362                                     ;
0363                                     ;
0364      see1          goto    set_cmd_addr ;else set cmd/addr
0365 0051 021E      movf   temp_cmd,w     ;retore cmd
0366 0052 003A      movwf  cmd            ; /
0367 0053 021F      movf   temp_addr,w   ;restore addr
0368 0054 003B      movwf  addr          ; /
0369 0055 06DA      btfsc  cmd,6         ;Check for a WRITE or ERASE command.
0370 0056 0A5F      goto   see2          ;Yes, parse the command further.
0371 0057 06FA      btfsc  cmd,7         ;Check for a READ command.
0372 0058 0A69      goto   read_        ;Yes, process READ command.
0373 0059 06BA      btfsc  cmd,5         ;Check for a EWEN or ERAL command.
0374 005A 0A66      goto   see3          ;Yes, parse the command further.
0375 005B 069A      btfsc  cmd,4         ;Check for a WRAL command.
0376 005C 0A6E      goto   write_       ;Yes, process WRITE/WRAL command.
0377
0378      exit_        dsel        ;No further processing required; 93CX6
0379      m
0380 005D 0445      bcf     serial,cs    ;Chip Select (CS) = '0' to de-select
0381                                     ;the device.
0382                                     ;command completed.
0383                                     ;
0384 005E 0800      retlw  no_err        ;Return with good completion status.
0385
0386 005F 07FA      see2    btfss  cmd,7           ;Check for a ERASE command.
0387 0060 0A6E      goto    write_       ;No, process WRITE command.
0388 0061 092E      exit2_ call    rdychk        ;ERASE command requires a status check.
0389                                     ;De-select the 93CX6.
0390      m
0391 0062 0445      m      bcf     serial,cs    ;Chip Select (CS) = '0' to de-select
0392                                     ;the device.
0393 0063 01E2      addwf  pc            ;Compute completion status from results of
0394                                     ;status check.
0395 0064 0800      retlw  no_err        ;Return with good completion status.
0396 0065 0801      retlw  error        ;Return with bad completion status.
0397
0398 0066 069A      see3    btfsc  cmd,4         ;Check for a EWEN command
0399 0067 0A5D      goto    exit_        ;Yes, no further processing required, exit
0400                                     ;now.
0401 0068 0A61      goto    exit2_       ;No, ERAL command which requires a status

```

Interfacing 93 Series Serial EEPROMs

```
0402                                     ;check.
0403
0404 0069 02A4   read_   incf   fsr           ;Increment the File Select Register to point
0405                                     ;to the register receiving the upper byte of
0406                                     ;the incoming 93CX6 data word.
0407 006A 091E           call   din8         ;Input the upper byte.
0408 006B 02A4           incf   fsr           ;Increment the File Select Register to point
0409                                     ;to the register receiving the lower byte.
0410 006C 091E           call   din8         ;Input 8 more bits.
0411 006D 0A5D           goto   exit_        ;No further processing required, exit now.
0412
0413 006E 02A4   write_  incf   fsr           ;Increment the File Select Register to point
0414                                     ;to the upper byte of the 16 bit 93CX6 data
0415                                     ;word to be transmitted.
0416 006F 0910           call   dout8        ;Output that byte.
0417 0070 02A4           incf   fsr           ;Increment the File Select Register to point
0418                                     ;to the lower byte.
0419 0071 0910           call   dout8        ;Output the lower byte of the 16 bit 93CX6
0420                                     ;data word.
0421 0072 0A61           goto   exit2_       ;Exit with a status check.
0422
0423 ;
0424 ;
0425 ;
0426   sca10
0427 0073 0901           call   dout10       ;output cmd reg
0428 0074 0A51           goto   see1         ;return
0429 ;
0430 ;
0431 ;
0432 ;
0433 ;
0434   sc10
0435 0075 037A           rlf    cmd          ;rotate cmd
0436 0076 035A           rlf    cmd,w        ;left twice
0437 0077 003B           movwf  addr         ;save in addr
0438 0078 007A           clrf  cmd          ;clear command
0439 0079 0901           call  dout10        ;xfer 10 bits
0440 007A 0A51           goto  see1         ;return
0441 ;
0442 ;*****
0443 ;*                               Test Program
0444 ;*
0445 ;*****
0446   main                                     ;We've include a sample program to exercise
0447                                     ;the PIC16/17 to 93C66 interface using a simple
0448                                     ;erase, write and verify routine.
0449                                     ;8 bit organization has been used
0450                                     ;with a 3 wire interface.
0451 007B 0065           clrf  serial        ;Clear the port tied to the 93C66 device.
0452 007C 0CF0           movlw b'11110000'  ;Intialize the data direction register for
0453 007D 0005           tris  serial        ;that port.
0454
0455 007E 0C30           movlw ewen          ;Load W with the Erase/Write Enable command.
0456 007F 003A           movwf cmd           ;Transfer W into cmd register.
0457 0080 0941           call  see           ;Enable the 93C66 device.
0458
0459 0081 0C20           movlw eral          ;Load W with the Erase All command.
0460 0082 003A           movwf cmd           ;Transfer W into cmd register.
0461 0083 0941           call  see           ;Erase the 93C66.
0462 0084 0F01           xorlw error         ;Check completion status.
0463 0085 0643           btfs  status, zflag ;Test for error condition.
0464 0086 0AA6           goto  errloop       ;Yes, bad completion status, error-out.
0465
0466                                     ;Write loop:
0467 00AA           tstptrn            equ  0xAA           ;Define the test pattern to be written.
0468
0469 0087 0C40           movlw write         ;Load W with the Write command.
0470 0088 003A           movwf cmd           ;Transfer W into cmd register.
```

Interfacing 93 Series Serial EEPROMs

```

0471 0089 0CAA          movlw  tstptrn      ;Intialize the 93C66 data registers with
0472                                     ;write data.
0473 008A 003C          movwfw highb       ;load in high byte only
0474                                     ;since 8 bit low byte is ignored
0475 008B 007B          clrf   addr        ;start at addr 0
0476 008C 0941 test1    call   see         ;Write data word into 93C66 device.
0477 008D 0F01          xorlw  error       ;Check completion status.
0478 008E 0643          btfsc status,zflag ;Test for error condition.
0479 008F 0AA6          goto  errloop     ;Yes, bad completion status, error-out.
0480 0090 03FB          incfsz addr      ;No, increment the 8 bit memory address
0481                                     ;field.
0482 0091 0A8C          goto  test1      ;write another location
0483 0092 077A          btfss cmd,3     ;see if all done
0484 0093 0A95          goto  wrt_nxt_pg ;no then write next page
0485 0094 0A97          goto  read_tst  ;read written data
0486 wrt_nxt_pg
0487 0095 057A          bsf   cmd,3     ;set page bit
0488 0096 0A8C          goto  test1     ;No, write another location.

0489
0490                                     ;Read loop:
0491 read_tst
0492 0097 0C80          movlw  read      ;Load W with the Read command.
0493 0098 003A          movwfw cmd      ;Transfer W into cmd register.
0494 0099 0941 test2    call   see         ;Read addressed data word from 93C66 device.
0495 009A 0CAA          movlw  tstptrn  ;Load W with the pattern written.
0496 009B 009C          subwfw highb,w ;Verify the data read against what was
0497                                     ;written.
0498 009C 0743          btfss status,zflag ;Same?
0499 009D 0AA6          goto  errloop   ;No, error-out.
0500 009E 03FB          incfsz addr    ;Yes, both byte correct, increment the 8 bit
0501                                     ;memory address field.
0502 009F 0A99          goto  test2     ;do next byte
0503 00A0 077A          btfss cmd,3    ;check page bit
0504 00A1 0AA3          goto  rd_nxt_pg ;no then chk next page
0505 00A2 0AA5          goto  alloc     ;all done!!!
0506 rd_nxt_pg
0507 00A3 057A          bsf   cmd,3     ;set page bit
0508 00A4 0A99          goto  test2     ;No, read another location.
0509
0510 00A5 0AA5  alloc   goto  alloc     ;Home safe!
0511 ;
0512 ;
0513 errloop
0514 00A6 0AA6          goto  errloop
0515 ;
0516 ;
0517 ;
0518 ;KEY DEFINITIONS
0519 ;
0520 0000          ORG    PIC54
0521 SYS_RESET
0522 01FF 0A00      GOTO  START
0523 ;
0524 0000          END

```

Interfacing 93 Series Serial EEPROMs

Appendix B

Microchip Technology PIC16C5X Assembler 3.03A Wed Feb 26 11:11:15 1992 Page 1

```
Line  PC  Opcode
0001  ;*****
0002  ;*                               MPALC Directives Section
      ;*
0003  ;*****
0004
0005  LIST      P=16C54,N=40,C=132
0006
0007
0008  ;*****
0009  ;*                               Register Assignments      *
0010  ;*****
0011
0012  0000  indir  equ    0x00      ;Use this register as source/destination for
0013      ;indirect addressing.
0014  0002  pc    equ    0x02      ;PIC16/17 Program Counter.
0015  0003  status equ    0x03      ;PIC16/17 Status Register.
0016  0004  fsr   equ    0x04      ;File Select Register.
0017  0005  serial equ    0x05      ;Port used for 93C46 control. Since Port A
0018      ;is 4 bits wide, we'll use 3 or 4 pins of Port A.
0019
0020      ;The following three registers must be
0021      ;located consecutively in memory.
0022  0010  cmd   equ    0x10      ;This register contains the 2 bit 93C46
0023      ;command is the upper 2 bit positions and
0024      ;memory address in the lower 6.
0025  0011  highb equ    0x11      ;Used in read/write routines to store the
0026      ;upper byte of a 16 bit 93C46 data word.
0027  0012  lowb  equ    0x12      ;Used in read/write routines to store the
0028      ;lower byte of a 16 bit 93C46 data word.
0029
0030  0013  cnthi equ    0x13      ;Used as the upper byte of a sixteen bit loop
0031      ;counter in RDYCHK routine.
0032  0014  cnt   equ    0x14      ;Used as the lower byte of a sixteen bit loop
0033      ;counter in RDYCHK routine, and elsewhere as
0034      ;an eight bit counter.
0035
0036  ;*****
0037  ;* Bit Assignments: The following assignment is for 3-wire setup.
0038  ;* For four wire setup, assign dout equ 2.
0039
0040  0000  carry  equ    0          ;Carry Flag of Status Register.
0041  0002  zflag  equ    2          ;Zero Flag of Status Register.
0042
0043  0000  cs     equ    0          ;Port pin tied to CS on 93C46.
0044  0001  din    equ    1          ;Port pin tied to DI on 93C46. 3-wire setup.
0045  0001  dout   equ    1          ;Port pin tied to DO on 93C46. 3-wire setup.
0046  0003  clock  equ    3          ;Port pin tied to CLK on 93C46.
0047
0048  ;*****
0049  ;*                               General Assignments      *
0050  ;*****
0051
0052  0000  no_err equ    0          ;
0053  0001  error  equ    1          ;
0054  0004  tries  equ    0x04      ;After issuing a WRITE, ERASE, ERAL, or WRAL
0055      ;command, the approximate number of machine
0056      ;cycles X 256 to wait for the RDY status.
0057      ;This value must be adjusted for operating
0058      ;frequencies other than 4 MHz.
0059
0060  0080  read   equ    0x80      ;93C46 Read command.
0061  0040  write  equ    0x40      ;93C46 Write command.
0062  00C0  erase  equ    0xC0      ;93C46 Erase command.
0063  0030  ewen  equ    0x30      ;93C46 Erase/Write Enable command.
```

Interfacing 93 Series Serial EEPROMs

```

0064    0000    ewds    equ    0x00                ;93C46 Erase/Write Disable command.
0065    0020    eral    equ    0x20                ;92CXX Erase All command.
0066    0010    wral    equ    0x10                ;92CXX Write All command.
0067
0068
0069
0070
0071
0072
0073    sel
0074
0075
0076    dsel
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103    0000    reset_  org    0x1FF
0104    01FF    0A57    goto   main
0105
0106
0107
0108
0109    0000
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119    0000    0C08    dout8  movlw  0x08                ;Initialize loop counter.
0120    0001    0034    movwf  cnt
0121
0122    0002    0425    d_o_8  bcf    serial,din        ;Assume that the bit to be transferred is a
0123
0124    0003    0360    rlf    indir            ;Rotate the actual bit to be transferred into
0125
0126    0004    0603    btfsc  status,carry;    Test the carry, if our assumption was
0127
0128    0005    0525    bsf    serial,din        ;No, actual bit was a '1'. Assert DI.
0129
0130
0131
0132    0006    0565    m      bsf    serial,clock    ;Clock (CLK) = '1'.
0133
0134    0007    0000    m      nop

```



Interfacing 93 Series Serial EEPROMs

```

0135          m          ;between the assertion and de-assertion of
0136          m          ;CLK in proportion to the PIC16/17 operating
0137          m          ;frequency. Refer to the 93C46 data for the
0138          m          ;minimum CLK period.
0139          m
0140 0008 0465          bcf      serial,clock      ;Clock (CLK) = '0'.
0141 0009 02F4          decfsz  cnt              ;Repeat until cnt = 0.
0142 000A 0A02          goto    d_o_8              ;Cnt still > 0.
0143 000B 0360          rlf      indir            ;Restore register to its original condition.
0144 000C 0425          bcf      serial,din        ;make din low before return
0145 000D 0800          retlw   no_err          ;Exit with good status.
0146
0147          ;*****
0148          ;*                      DIN8                      *
0149          ;*****
0150          ;Din8 will input 8 bits of data from the
0151          ;93C46. Before calling this routine, the FSR
0152          ;must point to the register being used to
0153          ;hold the incoming data.
0154          din8
0155          ;*****
0156          ;set up RAL as a input before proceeding
0157 000E 0C02          movlw   b'00000010'      ;set up PORTA
0158 000F 0005          tris   serial          ; /
0159          ;*****
0160 0010 0C08          movlw   0x08            ;Initialize loop counter.
0161 0011 0034          movwf  cnt              ;
0162
0163          d_i_8          clklt          ;Clock a bit out of the 93C46.
0164          m
0165          m          ;93C46 device.
0166 0012 0565          m          bsf      serial,clock      ;Clock (CLK) = '1'.
0167          m
0168 0013 0000          m          nop
0169          m          ;Adjust the number of nop instructions
0170          m          ;between the assertion and de-assertion of
0171          m          ;CLK in proportion to the PIC16/17 operating
0172          m          ;frequency. Refer to the 93C46 data for the
0173          m          ;minimum CLK period.
0174 0014 0465          bcf      serial,clock      ;Clock (CLK) = '0'.
0175 0015 0360          rlf      indir            ;Make room for the incoming bit in the
0176          ;destination register.
0177 0016 0400          bcf      indir,0         ;Assume that the incoming bit is a '0' and
0178          ;clear the LSB of the destination register.
0179 0017 0625          btfsz  serial,dout      ;Test the incoming bit, if our assumption
0180          ;was correct, skip the next instruction.
0181 0018 0500          bsf      indir,0         ;No, actual bit is a '1'. Set the LSB of the
0182          ;destination register.
0183 0019 02F4          decfsz  cnt              ;Repeat until cnt = 0.
0184 001A 0A12          goto    d_i_8              ;Cnt still > 0
0185          ;*****
0186          ;Restore RAL back as output
0187 001B 0C00          movlw   0              ;set RAL as output
0188 001C 0005          tris   serial          ; /
0189          ;*****
0190 001D 0800          retlw   no_err          ;Exit with good status.
0191
0192          ;*****
0193          ;*                      RDYCHK                      *
0194          ;*****
0195          ;Rdychk will read the 93C46 READY/BUSY status
0196          ;and wait for RDY status within the allotted
0197          ;number of processor cycles. If RDY status
0198          ;is not present after this set period, the
0199          ;routine will return with an error status.
0200
0201          rdychk
0202          ;*****
0203          ;set up RAL as a input before proceeding
0204 001E 0C02          movlw   b'00000010'      ;set up PORTA

```

Interfacing 93 Series Serial EEPROMs

```

0205 001F 0005      tris    serial      ; /
0206                ;*****
0207 0020 0C04      movlw   tries      ;Initialize time-out counter.
0208 0021 0033      movwf   cnthi      ;
0209 0022 0074      clr    cnt        ;
0210                dsel    serial      ;De-select the 93C46.
0211                m
0212 0023 0405 m      bcf     serial,cs  ;Chip Select (CS) = '0' to de-select the
0213                ;device.
0214
0215                ;      nop          ;NOTE: Check the 93C46 data sheet for
0216                ;      minimum CS low time. Depending upon
0217                ;      processor frequency, a nop(s) may be
0218                ;      between the assertion and de-assertion of
0219                ;      Chip Select.
0220
0221                sel     serial      ;Re-select the 93C46.
0222                m
0223 0024 0505      bsf     serial,cs  ;Chip Select (CS) = '1' to select the device.
0224 0025 0625 notrdy btfsz   serial,dout  ;If DO is a '0', 93C46 has yet to completed
0225                ;the last operation (still busy).
0226 0026 0A2E      goto    rdynoerr   ;skip to no error
0227 0027 02F4      decfsz  cnt        ;No, not yet ready. Decrement the LSB of our
0228                ;16 bit timer and check for expiration.
0229 0028 0A25      goto    notrdy     ;Still some time left. Try again.
0230 0029 02F3      decfsz  cnthi      ;Least significant byte expired - decrement
0231                ;and check for expiration of the MSB.
0232 002A 0A25      goto    notrdy     ;Still some time left. Try again.
0233                ;*****
0234                ;Restore RAL back as output
0235 002B 0C00      movlw   0          ;set RAL as output
0236 002C 0005      tris    serial      ; /
0237                ;*****
0238 002D 0801      retlw   error      ;RDY status was not present in the allotted
0239                ;time, return with error status.
0240                rdynoerr
0241                ;*****
0242                ;Restore RAL back as output
0243 002E 0C00      movlw   0          ;set RAL as output
0244 002F 0005      tris    serial      ; /
0245                ;*****
0246 0030 0800      retlw   no_err     ;
0247                ;
0248                ;*****
0249                ;*
0250                ;* SEE *
0251                ;*****
0252
0253                ;See will control the entire operation of a
0254                ;93C46 device. Prior to calling the routine,
0255                ;load a valid command/memory address into
0256                ;location cmd, and for WRITE or WRAL
0257                ;commands, load registers highb and lowb with
0258                ;16 bits of write data. Upon exit, the W
0259                ;register will contain the completion status.
0260                ;Only 93C46 instructions which require a
0261                ;status check can return with an error as the
0262                ;completion status. The values that denote
0263                ;the completion status are defined as
0264                ;variables 'error' and 'no_err' in the
0265                ;general assignments section.
0266
0267 0031 0C10 see      movlw   cmd        ;Load W with the location of the cmd
0268                ;register.
0269 0032 0024      movwf   fsr        ;Transfer that information into the File
0270                ;Select Register. The fsr now points to
0271                ;location cmd.
0272                sel     serial      ;Select the 93C46.
0273                m
0274 0033 0505      bsf     serial,cs  ;Chip Select (CS) = '1' to select the device.
0275                strtbt          ;Send a start bit.

```



Interfacing 93 Series Serial EEPROMs

```

0276          m
0277 0034 0525 m      bsf    serial,din      ;Start Bit = '1'.
0278          m      clkout          ;Clock it out.
0279          m
0280          m      ;93C46 device.
0281 0035 0565 m      bsf    serial,clock    ;Clock (CLK) = '1'.
0282          m
0283 0036 0000 m      nop                    ;Adjust the number of nop instructions
0284          m      ;between the assertion and de-assertion of
0285          m      ;CLK in proportion to the PIC16/17 operating
0286          m      ;frequency. Refer to the 93C46 data for the
0287          m      ;minimum CLK period.
0288          m
0289 0037 0465          bcf    serial,clock    ;Clock (CLK) = '0'.
0290 0038 0900          call   dout8          ;Transmit the 2 bit command and six bit
0291          m      ;address.
0292 0039 06D0          btfs   cmd,6          ;Check for a WRITE or ERASE command.
0293 003A 0A43          goto   see2          ;Yes, parse the command further.
0294 003B 06F0          btfs   cmd,7          ;Check for a READ command.
0295 003C 0A4D          goto   read_         ;Yes, process READ command.
0296 003D 06B0          btfs   cmd,5          ;Check for a EWEN or ERAL command.
0297 003E 0A4A          goto   see3          ;Yes, parse the command further.
0298 003F 0690          btfs   cmd,4          ;Check for a WRAL command.
0299 0040 0A52          goto   write_        ;Yes, process WRITE/WRAL command.
0300
0301          m      exit_dsel          ;No further processing required; 93C46
0302          m
0303 0041 0405 m      bcf    serial,cs        ;Chip Select (CS) = '0' to de-select the
0304          m      ;device.
0305          m      ;command completed.
0306 0042 0800          retlw  no_err        ;Return with good completion status.
0307
0308 0043 07F0 see2     btfs   cmd,7          ;Check for a ERASE command.
0309 0044 0A52          goto   write_        ;No, process WRITE command.
0310 0045 091E exit2_   call   rdychk        ;ERASE command requires a status check.
0311          m      dsel              ;De-select the 93C46.
0312          m
0313 0046 0405 m      bcf    serial,cs        ;Chip Select (CS) = '0' to de-select the
0314          m      ;device.
0315 0047 01E2          addwf  pc              ;Compute completion status from results of
0316          m      ;status check.
0317 0048 0800          retlw  no_err        ;Return with good completion status.
0318 0049 0801          retlw  error         ;Return with bad completion status.
0319
0320 004A 0690 see3     btfs   cmd,4          ;Check for a EWEN command.
0321 004B 0A41          goto   exit_         ;Yes, no further processing required, exit
0322          m      ;now.
0323 004C 0A45          goto   exit2_        ;No, ERAL command which requires a status
0324          m      ;check.
0325
0326 004D 02A4 read_    incf   fsr            ;Increment the File Select Register to point
0327          m      ;to the register receiving the upper byte of
0328          m      ;the incoming 93C46 data word.
0329 004E 090E          call   din8          ;Input the upper byte.
0330 004F 02A4          incf   fsr            ;Increment the File Select Register to point
0331          m      ;to the register receiving the lower byte.
0332 0050 090E          call   din8          ;Input 8 more bits.
0333 0051 0A41          goto   exit_         ;No further processing required, exit now.
0334
0335 0052 02A4 write_   incf   fsr            ;Increment the File Select Register to point
0336          m      ;to the upper byte of the 16 bit 93C46 data
0337          m      ;word to be transmitted.
0338 0053 0900          call   dout8        ;Output that byte.
0339 0054 02A4          incf   fsr            ;Increment the File Select Register to point
0340          m      ;to the lower byte.
0341 0055 0900          call   dout8        ;Output the lower byte of the 16 bit 93C46
0342          m      ;data word.
0343 0056 0A45          goto   exit2_        ;Exit with a status check.
0344
0345          m      ;*****

```


Interfacing 93 Series Serial EEPROMs

```

0346                ;*                Test Program                *
0347                ;*****
0348                main                ;We've include a sample program to exercise
0349                ;the PIC16/17 to 93C46 interface using a simple
0350                ;erase, write and varify routine.
0351
0352    0057    0065                clr    serial                ;Clear the port tied to the 93C46 device.
0353    0058    00CF                movlw b'11110100'            ;Intialize the data direction register for
0354    0059    0005                tris   serial                ;that port.
0355
0356    005A    0C30                movlw ewen            ;Load W with the Erase/Write Enable command.
0357    005B    0030                movwf cmd            ;Transfer W into cmd register.
0358    005C    0931                call   see            ;Enable the 93C46 device.
0359
0360    005D    0C20                movlw eral            ;Load W with the Erase All command.
0361    005E    0030                movwf cmd            ;Transfer W into cmd register.
0362    005F    0931                call   see            ;Erase the 93C46.
0363    0060    0F01                xorlw  error          ;Check completion status.
0364    0061    0643                btfsc status, zflag  ;Test for error condition.
0365    0062    0A82                goto  errloop        ;Yes, bad completion status, error-out.
0366                ;Write loop:
0368                001F    loopcnt equ    0x1F                ;Define an unused location for our test
0369                ;program loop counter.
0370                00AA    tstptrn equ    0xAA                ;Define the test pattern to be written.
0371
0372    0063    0C40                movlw  .64            ;Initialize that counter.
0373    0064    003F                movwf loopcnt        ;
0374    0065    0C40                movlw  write          ;Load W with the Write command.
0375    0066    0030                movwf cmd            ;Transfer W into cmd register.
0376    0067    0CAA                movlw  tstptrn        ;Intialize the 93C46 data registers with
0377                ;write data.
0378    0068    0031                movwf highb          ;
0379    0069    0032                movwf lowb           ;
0380    006A    0931    test1    call   see            ;Write data word into 93C46 device.
0381    006B    0F01                xorlw  error          ;Check completion status.
0382    006C    0643                btfsc status,zflag   ;Test for error condition.
0383    006D    0A82                goto  errloop        ;Yes, bad completion status, error-out.
0384    006E    02B0                incf  cmd              ;No, increment the 6 bit memory address
0385                ;field.
0386    006F    02FF                decfsz loopcnt        ;Have we written all 64 locations?
0387    0070    0A6A                goto  test1           ;No, write another location.
0388
0389                ;Read loop:
0390
0391    0071    0C40                movlw  .64            ;Initialize loop counter.
0392    0072    003F                movwf loopcnt        ;
0393    0073    0C80                movlw  read           ;Load W with the Read command.
0394    0074    0030                movwf cmd            ;Transfer W into cmd register.
0395    0075    0931    test2    call   see            ;Read addressed data word from 93C46 device.
0396    0076    0CAA                movlw  tstptrn        ;Load W with the pattern written.
0397    0077    0091                subwf highb,0        ;Verify the data read against what was
0398                ;written.
0399    0078    0743                btfss status,zflag   ;Same?
0400    0079    0A82                goto  errloop        ;No, error-out.
0401    007A    0CAA                movlw  tstptrn        ;Repeat with the lower byte read.
0402    007B    0092                subwf lowb,0         ;
0403    007C    0743                btfss status,zflag   ;Same?
0404    007D    0A82                goto  errloop        ;No, error-out.
0405    007E    02B0                incf  cmd              ;Yes, both byte correct, increment the 6 bit
0406                ;memory address field.
0407    007F    02FF                decfsz loopcnt        ;Have we read all 64 locations?
0408    0080    0A75                goto  test2           ;No, read another location.
0409
0410    0081    0A81    alloc    goto  alloc            ;Home safe!
0411
0412    0082    0A82    errloop goto  errloop          ;Bad news!
0413
0414                0000                END                ;Thats all folks!

```

Interfacing 93 Series Serial EEPROMs

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/microhip>

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

AMERICAS (continued)

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

All rights reserved. © 1995, Microchip Technology Incorporated, USA.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.