

Interfacing Microchip PIC16C54 to Microchip SPI™ Serial EEPROMs

Author: Shannon Poulin
Microchip Technology Inc.

INTRODUCTION

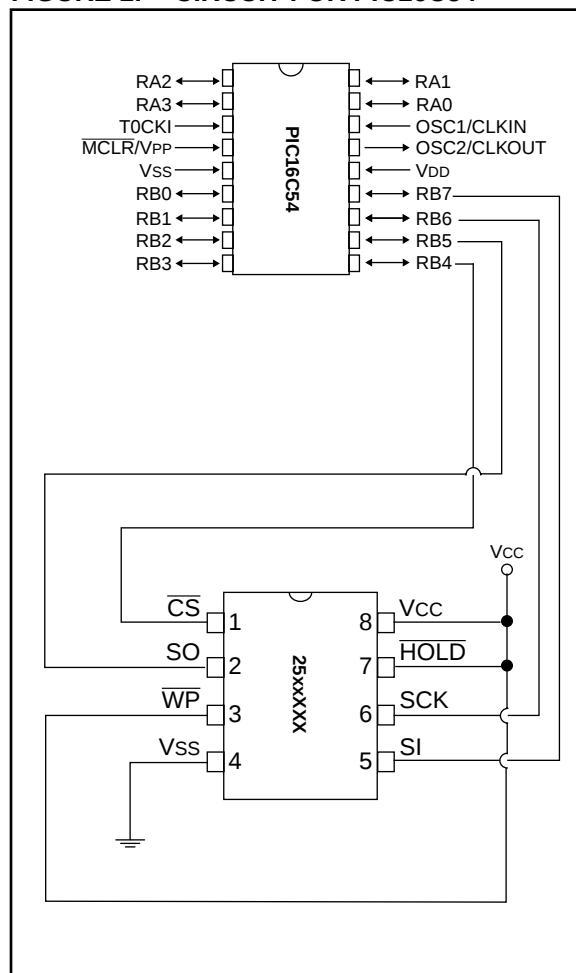
There are many different microcontrollers on the market today that are being used in embedded control applications. Many of these embedded control systems need non-volatile memory. Because of their small footprint, byte level flexibility, low I/O pin requirement, low power consumption, and low cost, serial EEPROMs are a popular choice for non-volatile storage.

Microchip addresses these needs by offering a full line of serial EEPROMs covering industry standard serial communication protocols for 2-wire, 3-wire, and SPI communication. Serial EEPROM devices are available in a variety of densities, operational voltage ranges and packaging options.

This application note provides assistance and source code to ease the design process of interfacing a Microchip PIC16C54 microcontroller and Microchip SPI serial EEPROMs. The connections will be made using port pins from the microcontroller to interface to the memory device. Bit manipulation will be used to provide the proper logic necessary to write and read the part.

Figure 1 describes the hardware schematic for the interface between Microchip's SPI devices and the Microchip PIC16C54 Microcontroller. The schematic shows the connections necessary between the microcontroller and the serial EEPROM, and the software was written assuming these connections. Appendix A contains a listing of the SPI source code.

FIGURE 1: CIRCUIT FOR PIC16C54



AN648

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX A: AN648.ASM

MPASM 01.40.01 Intermediate

AN648.ASM

3-20-1997

15:04:06

PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```
00001          LIST P=16C54
00002 ; Port B pin descriptions
00003 ; PIC16C54                      SPI Device
00004 ; Pin #          Description    Pin #    Description
00005 ; 13             PORTB,7        5        SI, serial data in
00006 ; 12             PORTB,6        6        SCK, serial clock
00007 ; 11             PORTB,5        2        SO, serial data out
00008 ; 10             PORTB,4        1        not CS, not chip select
00009 ;
00010 ;*****RAM register definitions*****
00011 count equ 0x0B          ; count variable location
00012 outbyte equ 0x0C        ; outbyte storage variable
00013 temp1 equ 0x0D          ; temp location 1
00014 temp2 equ 0x0E          ; temp location 2
00015 rxdata equ 0x0F        ; location for received data
00016 ;*****Bit definitions*****
00017 si equ 7                ; SPI data in = portb pin 7
00018 sck equ 6              ; SPI clock = portb pin 6
00019 so equ 5              ; SPI data out = portb pin 5
00020 cs equ 4              ; SPI chip select = portb pin 4
00021 ;*****Include file*****
00022 include "p16c5x.inc"    ; include file for a PIC16C54
00001          LIST
00002 ; P16C5X.INC Standard Header File, Version 3.30 Microchip Technology
00024          LIST
00023 ;*****
01FF          00024 org 01ffh          ; set the reset vector
01FF 0A01      00025 goto start          ; go to the beginning of main
0000          00026 org 000h          ;
0000 0A01      00027 goto start          ;
0001 0C20      00028 ;!!!!!!!!!!!!!!Begin Main Program!!!!!!!!!!!!!!
0002 0006      00029 start movlw 0x20          ; move 20 hex into w
0003 0586      00030 tris PORTB          ; portb all outputs except pin 5
0004 04C6      00031 bsf PORTB,cs          ; set the chip select line
0005 04E6      00032 bcf PORTB,sck          ; clear the clock line
0006 0486      00033 bcf PORTB,si          ; clear the data in line
0007 0C06      00034 ;Send the write enable sequence (WREN)
0008 002C      00035 bcf PORTB,cs          ; clear the chip select (active)
0009 0955      00036 movlw 0x06          ; load WREN sequence
000A 0586      00037 movwf outbyte        ; store in RAM location outbyte
000B 0486      00038 call output          ; call the SPI output routine
000C 0C01      00039 bsf PORTB,cs          ; set the chip select line
000D 002C      00040 ;Send the write status register sequence (WRSR)
000E 0955      00041 bcf PORTB,cs          ; clear the chip select line
000F 0C00      00042 movlw 0x01          ; clear all status register
0010 002C      00043 movwf outbyte        ; store in RAM location outbyte
0011 0955      00044 call output          ; call the SPI output routine
0012 0586      00045 movlw 0x00          ; load up zero to send
0013 0999      00046 movwf outbyte        ; store in RAM location outbyte
00051 ;Send the write enable sequence (WREN) 00047 call output          ; call the SPI output routine
00048 bsf PORTB,cs          ; set the chip select line
00049 ;Wait the required 5mS for the write cycle timer Twc
00050 call delay          ; call the delay subroutine
```

```

0014 0486      00052      bcf      PORTB,cs      ; clear the chip select (active)
0015 0C06      00053      movlw     0x06      ; load WREN sequence
0016 002C      00054      movwf     outbyte    ; store in RAM location outbyte
0017 0955      00055      call      output    ; call the SPI output routine
0018 0586      00056      bsf      PORTB,cs      ; set the chip select line
00057 ;Send the read status register sequence (RDSR)
0019 0486      00058      bcf      PORTB,cs      ; clear the chip select line
001A 0C05      00059      movlw     0x05      ; load RDSR sequence
001B 002C      00060      movwf     outbyte    ; store in RAM location outbyte
001C 0955      00061      call      output    ; call the SPI output routine
001D 0955      00062      call      output    ; read the data in status reg.
001E 0586      00063      bsf      PORTB,cs      ; set the chip select line
00064 ;Send the write sequence (WRITE)
001F 0486      00065      bcf      PORTB,cs      ; clear the chip select line
0020 0C02      00066      movlw     0x02      ; load WRITE sequence
0021 002C      00067      movwf     outbyte    ; store in RAM location outbyte
0022 0955      00068      call      output    ; call the SPI output routine
00069 ;****Comment out for use with 25xx010, 25xx020 or 25xx040*****
0023 0C00      00070      movlw     0x00      ; load high address byte
0024 002C      00071      movwf     outbyte    ; store in RAM location outbyte
0025 0955      00072      call      output    ; call the SPI output routine
00073 ;*****
0026 0C55      00074      movlw     0x55      ; load 55 as address into w
0027 002C      00075      movwf     outbyte    ; store in RAM location outbyte
0028 0955      00076      call      output    ; call the SPI output routine
0029 0CAA      00077      movlw     0xAA      ; load data AA into w
002A 002C      00078      movwf     outbyte    ; store in RAM location outbyte
002B 0955      00079      call      output    ; call the SPI output routine
002C 0586      00080      bsf      PORTB,cs      ; set the chip select line
00081 ;Perform data polling (RDSR bit 0)
002D 0486      00082      bcf      PORTB,cs      ; clear the chip select line
002E 0C05      00083      movlw     0x05      ; load RDSR sequence
002F 002C      00084      movwf     outbyte    ; store in RAM location outbyte
0030 0955      00085      call      output    ; call the SPI output routine
00086 ;
0031 0C30      00087      movlw     0x30      ; give the spi device time to
0032 002B      00088      movwf     count      ; set the WIP bit in the
0033 02EB      00089 wait    decfsz    count,1    ; status register before coming
0034 0A33      00090      goto      wait      ; back and doing data polling
00091 ;
0035 0955      00092 polling call    output    ; read the data in status reg.
0036 060F      00093      btfsc    rxdata,0    ; test the WIP bit in status reg.
0037 0A35      00094      goto      polling    ; WIP clear, loop until WIP set
0038 0586      00095      bsf      PORTB,cs      ; set the chip select line
00096 ;Send read sequence (READ), read address 0x55
0039 0486      00097      bcf      PORTB,cs      ; clear the chip select line
003A 0C03      00098      movlw     0x03      ; load READ sequence
003B 002C      00099      movwf     outbyte    ; store in RAM location outbyte
003C 0955      00100      call      output    ; call the SPI output routine
00101 ;****Comment out for use with 25xx010, 25xx020 or 25xx040*****
003D 0C00      00102      movlw     0x00      ; load high address byte
003E 002C      00103      movwf     outbyte    ; store in RAM location outbyte
003F 0955      00104      call      output    ; call the SPI output routine
00105 ;*****
0040 0C55      00106      movlw     0x55      ; move the address into w
0041 002C      00107      movwf     outbyte    ; store in RAM location outbyte
0042 0955      00108      call      output    ; call the SPI output routine
0043 0955      00109      call      output    ; call output to read 1 byte
0044 0586      00110      bsf      PORTB,cs      ; set the chip select line
00111 ;Send the write enable sequence (WREN)
0045 0486      00112      bcf      PORTB,cs      ; clear the chip select (active)
0046 0C06      00113      movlw     0x06      ; load WREN sequence
0047 002C      00114      movwf     outbyte    ; store in RAM location outbyte
0048 0955      00115      call      output    ; call the SPI output routine
0049 0586      00116      bsf      PORTB,cs      ; set the chip select line
00117 ;Send the write disable sequence (WRDI)

```

```

004A 0486      00118      bcf      PORTB,cs      ; clear the chip select (active)
004B 0C04      00119      movlw     0x04          ; load WRDI sequence
004C 002C      00120      movwf     outbyte       ; store in RAM location outbyte
004D 0955      00121      call      output        ; call the SPI output routine
004E 0586      00122      bsf      PORTB,cs      ; set the chip select line
004F 0486      00123      ;Send the read status register sequence (RDSR)
0050 0C05      00124      bcf      PORTB,cs      ; clear the chip select line
0051 002C      00125      movlw     0x05          ; load RDSR sequence
0052 0955      00126      movwf     outbyte       ; store in RAM location outbyte
0053 0955      00127      call      output        ; call the SPI output routine
0054 0A01      00128      call      output        ; read the data in status reg.
0055 04C6      00129      ;*****Go back to main routine*****
0056 04E6      00130      goto      start
0057 006F      00131
0058 06EC      00132      ;*****SPI output subroutine*****
0059 05E6      00133      output    bcf      PORTB,sck      ; clear the clock line
0060 06CC      00134      bcf      PORTB,si      ; clear the data line
0061 05E6      00135      clrf      rxdata        ; clear the received data location
0062 05C6      00136      ;
0063 036F      00137      btfsc     outbyte,7      ; test outbyte bit 7 , skip if 0
0064 06A6      00138      bsf      PORTB,si      ; set the data line
0065 02AF      00139      bsf      PORTB,sck      ; set the clock line
0066 04C6      00140      rlf      rxdata,1      ; rotate rxdata 1 to the right
0067 04E6      00141      btfsc     PORTB,so      ; test the SPI serial data out
0068 06AC      00142      incf      rxdata,1      ; increment rxdata lsb if set
0069 05E6      00143      bcf      PORTB,sck      ; clear the clock line
0070 05E6      00144      bcf      PORTB,si      ; clear the data line
0071 05E6      00145      ;
0072 05C6      00146      btfsc     outbyte,6      ; test outbyte bit 6 , skip if 0
0073 036F      00147      bsf      PORTB,si      ; set the data line
0074 06A6      00148      bsf      PORTB,sck      ; set the clock line
0075 02AF      00149      rlf      rxdata,1      ; rotate rxdata 1 to the right
0076 04C6      00150      btfsc     PORTB,so      ; test the SPI serial data out
0077 04E6      00151      incf      rxdata,1      ; increment rxdata lsb if set
0078 066C      00152      bcf      PORTB,sck      ; clear the clock line
0079 05E6      00153      bcf      PORTB,si      ; clear the data line
0080 064C      00154      ;
0081 05E6      00155      btfsc     outbyte,5      ; test outbyte bit 5 , skip if 0
0082 036F      00156      bsf      PORTB,si      ; set the data line
0083 06A6      00157      bsf      PORTB,sck      ; set the clock line
0084 02AF      00158      rlf      rxdata,1      ; rotate rxdata 1 to the right
0085 04C6      00159      btfsc     PORTB,so      ; test the SPI serial data out
0086 04E6      00160      incf      rxdata,1      ; increment rxdata lsb if set
0087 066C      00161      bcf      PORTB,sck      ; clear the clock line
0088 05E6      00162      bcf      PORTB,si      ; clear the data line
0089 036F      00163      ;
0090 06A6      00164      btfsc     outbyte,4      ; test outbyte bit 4 , skip if 0
0091 02AF      00165      bsf      PORTB,si      ; set the data line
0092 04C6      00166      bsf      PORTB,sck      ; set the clock line
0093 04E6      00167      rlf      rxdata,1      ; rotate rxdata 1 to the right
0094 066C      00168      btfsc     PORTB,so      ; test the SPI serial data out
0095 05E6      00169      incf      rxdata,1      ; increment rxdata lsb if set
0096 036F      00170      bcf      PORTB,sck      ; clear the clock line
0097 06A6      00171      bcf      PORTB,si      ; clear the data line
0098 02AF      00172      ;
0099 066C      00173      btfsc     outbyte,3      ; test outbyte bit 3 , skip if 0
0100 05E6      00174      bsf      PORTB,si      ; set the data line
0101 05C6      00175      bsf      PORTB,sck      ; set the clock line
0102 036F      00176      rlf      rxdata,1      ; rotate rxdata 1 to the right
0103 06A6      00177      btfsc     PORTB,so      ; test the SPI serial data out
0104 02AF      00178      incf      rxdata,1      ; increment rxdata lsb if set
0105 04C6      00179      bcf      PORTB,sck      ; clear the clock line
0106 04E6      00180      bcf      PORTB,si      ; clear the data line
0107 066C      00181      ;
0108 064C      00182      btfsc     outbyte,2      ; test outbyte bit 2 , skip if 0
0109 05E6      00183      bsf      PORTB,si      ; set the data line

```

```

0082 05C6      00184      bsf      PORTB,sck      ; set the clock line
0083 036F      00185      rlf      rxdata,1      ; rotate rxdata 1 to the right
0084 06A6      00186      btfsc    PORTB,so      ; test the SPI serial data out
0085 02AF      00187      incf      rxdata,1      ; increment rxdata lsb if set
0086 04C6      00188      bcf      PORTB,sck      ; clear the clock line
0087 04E6      00189      bcf      PORTB,si      ; clear the data line
                00190 ;
0088 062C      00191      btfsc    outbyte,1      ; test outbyte bit 1 , skip if 0
0089 05E6      00192      bsf      PORTB,si      ; set the data line
008A 05C6      00193      bsf      PORTB,sck      ; set the clock line
008B 036F      00194      rlf      rxdata,1      ; rotate rxdata 1 to the right
008C 06A6      00195      btfsc    PORTB,so      ; test the SPI serial data out
008D 02AF      00196      incf      rxdata,1      ; increment rxdata lsb if set
008E 04C6      00197      bcf      PORTB,sck      ; clear the clock line
008F 04E6      00198      bcf      PORTB,si      ; clear the data line
                00199
0090 060C      00200      btfsc    outbyte,0      ; test outbyte bit 0 , skip if 0
0091 05E6      00201      bsf      PORTB,si      ; set the data line
0092 05C6      00202      bsf      PORTB,sck      ; set the clock line
0093 036F      00203      rlf      rxdata,1      ; rotate rxdata 1 to the right
0094 06A6      00204      btfsc    PORTB,so      ; test the SPI serial data out
0095 02AF      00205      incf      rxdata,1      ; increment rxdata lsb if set
0096 04C6      00206      bcf      PORTB,sck      ; clear the clock line
0097 04E6      00207      bcf      PORTB,si      ; clear the data line
                00208 ;
0098 0800      00209      retlw    0      ; return from subroutine
                00210
                00211 ; 250 x 400nS x 50 = 5mS (plus overhead)
0099 0C32      00212 delay  movlw    0x32      ; move 50 decimal into w
009A 002D      00213      movwf    temp1      ; move 50 decimal into temp1
009B 0CFA      00214 dec1   movlw    0xFA      ; move 250 decimal into w
009C 002E      00215      movwf    temp2      ; move 250 decimal into temp2
009D 02EE      00216 dec2   decfsz   temp2,1      ; decrement temp2, skip if zero
009E 0A9D      00217      goto     dec2      ; goto decrement 2 if not zero
009F 02ED      00218      decfsz   temp1,1      ; decrement temp1, skip if zero
00A0 0A9B      00219      goto     dec1      ; goto decrement 1 if not zero
00A1 0800      00220      retlw    0      ; return both locations = 0
                00221
                00222
                00223      END

```

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XX-----X
01C0 : -----X

```

All other memory blocks unused.

Program Memory Words Used: 163
Program Memory Words Free: 349

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 0 suppressed

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.microchip.com>

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: 630-285-0071 Fax: 630-285-0075

Dallas

Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton

Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
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

Toronto

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong

Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India

Microchip Technology India
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Korea

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

Shanghai

Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore

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

Taiwan, R.O.C

Microchip Technology Taiwan
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: 44-1628-851077 Fax: 44-1628-850259

France

Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
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 München, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Arizona Microchip Technology SRL
Centro Direzionale Colleone
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81-4-5471- 6166 Fax: 81-4-5471-6122

5/8/97



MICROCHIP

All rights reserved. © 1997, Microchip Technology Incorporated, USA. 5/97

Information contained in this publication regarding device applications and the like is intended for 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. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.
