AN515

Communicating with the I²C™ Bus Using the PIC16C5X

INTRODUCTION

The Microchip Technology Inc.'s 24CXX and 85CXX Serial EEPROMs feature a two wire serial interface bus. The bus protocol is I²C compatible. Interface to a serial port with I²C bus protocol in a microcontroller is trivial. This application note is intended for design engineers who want to develop their software programs to communicate a microcontroller with a 2-wire bus Serial EEPROM through a general purpose I/O port.

Unlike the 3-wire bus Serial EEPROMs, the 24CXX/ 85CXX communicate with any microcontroller only by a serial data I/O line (SDA) and a serial clock (SCL). Chip select is not required. Data transfer may be initiated only when the bus is not busy. During such transfer, the data line (SDA) must remain stable whenever the clock line (SCL) is high. Changes in the data line while the clock line is high are interpreted as a START or STOP condition. A typical transfer format is shown in Figure 1.

After the START condition, a slave address is sent. This address is 7-bits long, the eighth bit is a data direction bit. (R/$\overline{W}$ - a logical '0' indicates a transmission WRITE, a logical '1' represents a request for data READ. A data transfer is always terminated by a STOP condition generated by the master controller. However, if a master still wishes to communicate on the bus, it can generate another START condition and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such transfer.

FIGURE 1 - TRANSFER FORMAT

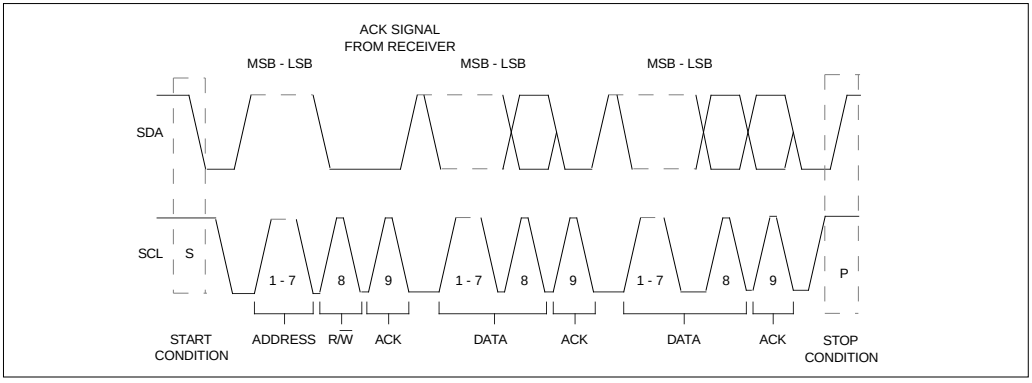
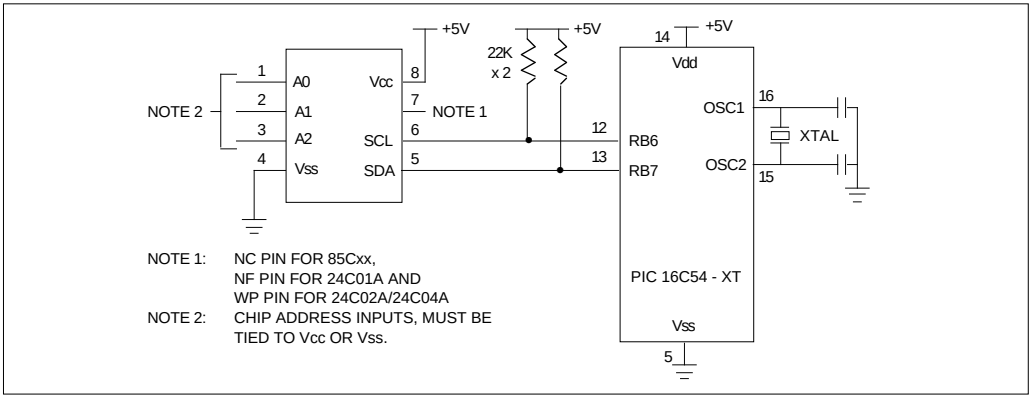


FIGURE 2 - A SIMPLE HARDWARE CONNECTION



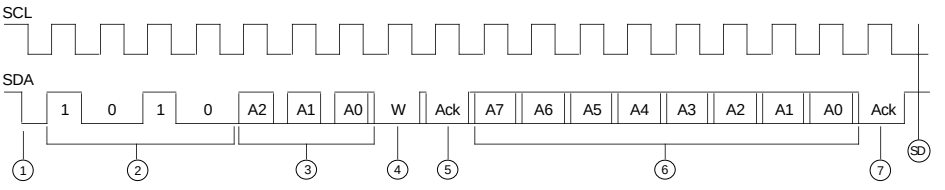
An example program has been provided in Appendix A containing all PIC16C54 routines needed to exercise a 24CXX or 85CXX device. A simple hardware connection is illustrated in Figure 2. A maximum of eight 24C01A/24C02A/85C72/85C82's, or four 24C04A/85C92's can be addressed by a microcontroller on the same two wire bus without additional interfaces. Each device is identified by its Chip Address and will only respond to the correct slave address. A detailed bus flow is shown in Figure 3.

Figure 3 as shown below describes how the bit stream is set up for READ and WRITE mode in the microcomputer programming software prior to sending it on the two wire serial bus.

The stop condition, after the write sequence, starts the internal self-timed write cycle which may last up to 6 milliseconds (.7 ms per byte). Acknowledge signal should be monitored during this period.

Author: *Bruce Negley*
Memory Products Division

FIGURE 3A - SETTING THE INTERNAL WORD ADDRESS OF THE 24CXX/85CXX



1. START CONDITION
2. 4 BIT DEVICE CODE 1010 FOR 24CXX/85CXX
3. DEVICE ADDRESS A0=PA, HIGH ORDER ADDRESS BIT A8 FOR 24C04A/85C92
4. 0=WRITE TO ADDRESS REGISTER
5. EEPROM DRIVES BUS LOW WITH ACKNOWLEDGEMENT (TIMEOUT IF NO RESPONSE)
6. ADDRESS FOR DATA, A7-A0 (MSB-LSB)
7. BUS LOW FOR ACKNOWLEDGEMENT

FIGURE 3B - BYTE WRITE SEQUENCE

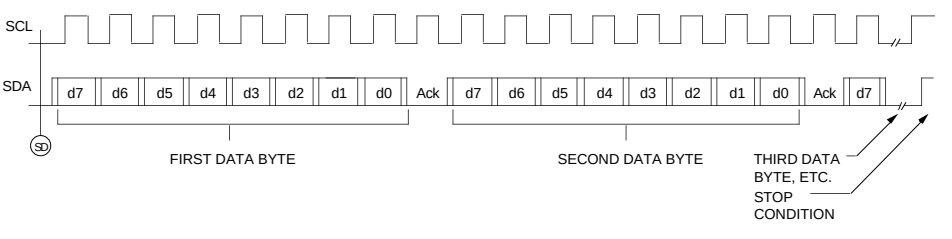
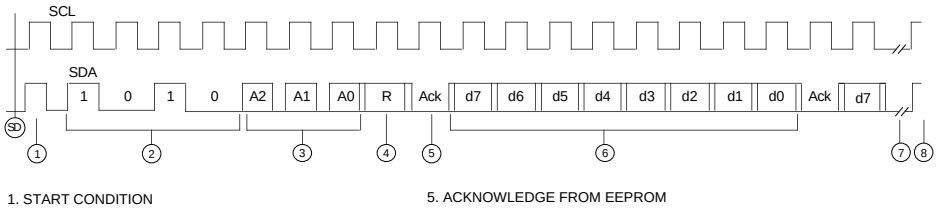


FIGURE 3C - READ MODE SEQUENCE



1. START CONDITION
2. 4 BIT DEVICE CODE 1010 FOR 24CXX/85CXX
3. DEVICE ADDRESS A2, A1, A0. A0 = PA, HIGH ORDER ADDRESS BIT A8 FOR 24C04A/85C92
4. 1 = READ MODE

READ UP TO 128 BYTES (24C01A, 85C72)
READ UP TO 256 BYTES (24C02A/04A, 85C82/92)

5. ACKNOWLEDGE FROM EEPROM
6. FIRST DATA BYTE
7. SECOND DATA BYTE, ETC.
8. STOP CONDITION

Appendix A:

```
MPALC CROSS ASSEMBLER  2.00  d:\seeprom\appnotes\i2cbus.asm              Apr 11
15:36:02 1990  PAGE 1

TWO WIRE/I2C BUS INTERFACE WITH PIC16C5x

0001              TITLE "TWO WIRE/I2C BUS INTERFACE WITH PIC16C5x"
0002         ;
0003              LIST          P=16C54
0004         ;
0005         ;*****************************************************
0006         ;**  Two wire/I2C Bus READ/WRITE Sample Routines
             ;    of Microchip's 24CXX/85CXX serial CMOS
0007         ;**  EEPROM interfacing to a PIC16C54 8-bit CMOS
0008         ;**  single chip microcomputer
0009         ;**
0010         ;**  Part use = PIC16C54-XT/JW
0011         ;**  Note: 1)  All timings are based on a
             ;    reference crystal frequency of 2 MHz which
             ;    is equivalent to an instruction cycle
0012         ;**  time of 2 usec.
0013         ;**  2)  Address and literal values are read
             ;    in octal unless otherwise specified.
             ;    3)  The following sample program is
             ;    intended to interface a two wire/I2C
             ;    serial EEPROM with a PIC16C54 on a
             ;    stand-alone application only.
             ;    In the case where the two wire bus is
             ;    multiplexing with other circuitry, it is
             ;    recommended to check the 24CXX/85CXX in
             ;    standby mode to avoid bus contention.
0014         ;**
0015         ;*****************************************************
0016         ;
0017         ;-----------------------------------------------------
0018         ;    Files Assignment
0019         ;-----------------------------------------------------
0020         ;
0021    0002    PC       EQU     2           ; Program counter
0022    0004    FSR      EQU     4           ; File Select Register
0023    0005    RA       EQU     5           ; Port A use to select
                                             ; device address
0024    0006    RB       EQU     6           ; RB7 = SDA, RB6 = SCL
0025       ;
0026    0010    STATUS   EQU     10          ; Status register
0027    0011    FLAG     EQU     11          ; Common flag bits
                                             ; register
0028    0012    EEPROM   EQU     12          ; Bit buffer
0029    0013    ERCODE   EQU     13          ; Error code (to indicate
                                             ; bus status)
0030    0020    ADDR     EQU     20          ; Address register
0031    0021    DATAI    EQU     21          ; Stored data input
                                             ; register
```

```
0032      0022    DATAO    EQU      22          ; Stored data output
                                                ; register
0033      0023    SLAVE    EQU      23          ; Device address
                                                          ; (1010xxx0)
0034      0024    TXBUF    EQU      24          ; TX buffer
0035      0025    RXBUF    EQU      25          ; RX buffer
0036      0026    COUNT    EQU      26          ; Bit counter
0037         ;
0038      0030    TIMER0   EQU      30          ; Delay timer0
0039      0031    TIMER1   EQU      31          ; Delay timer1
0040         ;
0041         ;
0042         ;-----------------------------------------------------
0043         ;                    Bit Assignments
0044         ;-----------------------------------------------------
0045         ;
0046         ;    FLAG Bits
0047         ;
0048      0000    ERROR    EQU      0           ; Error flag
0049         ;
0050         ;    EEPROM Bits
0051         ;
0052      0007    DI       EQU      7           ; EEPROM input
0053      0006    DO       EQU      6           ; EEPROM output
0054         ;
0055         ;    I2C Device Bits
0056         ;
0057      0007    SDA              EQU         7   ; RB7, data in/out
0058      0006    SCL              EQU         6   ; RB6, serial clock
0059         ;
0060         ;    END FILES/BITS EQUATE
0061         ;
0062         ;
0063         ;-----------------------------------------------------
0064         ;    Two wire/I2C - CPU communication error status table
             ;    and subroutine
0065         ;-----------------------------------------------------
0066         ;    input  :  W-reg              = error code
0067         ;    output :  ERCODE = error code
0068         ;               FLAG(ERROR)       = 1
0069         ;
0070         ;      code           error status mode
0071         ;    -------- ----------------------------------
0072         ;        1    :       SCL locked low by device (bus is still
                                                busy)
0073         ;        2    :       SDA locked low by device (bus is still
                                                busy)
0074         ;        3    :       No acknowledge from device (no
                                                handshake)
0075         ;        4    :       SDA bus not released for master to
                                    generate STOP bit
```

```
0076         ;-----------------------------------------------------
0077         ;
0078         ;    Subroutine to identify the status of the serial clock
;                         (SCL) and serial data
0079         ;    (SDA) condition according to the error status table. 0080
;                         Codes generated are useful for bus/device diagnosis.
0081         ;
0082        ERR
0083      0000    3411     BTFSS    FLAG,ERROR       ; Remain as first error
                                                     ; encountered
0084      0001    0053     MOVWF    ERCODE           ; Save error code
0085      0002    2411     BSF      FLAG,ERROR       ; Set error flag
0086      0003    4000     RETLW    0
0087              ;
0088              ;-----------------------------------------------------
0089              ;        START bus communication routine
0090              ;-----------------------------------------------------
0091              ;        input   : none
0092              ;        output  : initialize bus communication
0093              ;-----------------------------------------------------
0094              ;
0095              ;  Generate START bit (SCL is high while SDA goes from
                  ;  high to low transition) and check status of the
0096              ;  serial clock.
0097              BSTART
0098      0004    6077     MOVLW    B'00111111'  ; Put SCL, SDA line in
                                                           ; output state
0099      0005    0006     TRIS     RB
0100      0006    2706     BSF      RB,SCL       ; Set clock high
0101      0007    6001     MOVLW    1            ; Ready error status
                                                  ; code 1
0102      0010    3706     BTFSS    RB,SCL       ; Locked?
0103      0011    4400     CALL     ERR          ; SCL locked low by device
0104      0012    2346     BCF      RB,SDA       ; SDA goes low during SCL
                                    ; high
0105      0013    0000     NOP                   ; Timing adjustment
0106      0014    0000     NOP
0107      0015    0000     NOP
0108      0016    2306     BCF      RB,SCL       ; Start clock train
0109      0017    4000     RETLW    0
0110              ;
0111              ;END SUB
0113              ;
0114              ;-----------------------------------------------------
0115              ;        STOP bus communication routine
0116              ;-----------------------------------------------------
0117              ;  Input :  None
0118              ;  Output        :  Bus communication, STOP condition
0119              ;-----------------------------------------------------
0120              ;
0121              ; Generate STOP bit (SDA goes from low to high during
                  ; SCL high state)
```

```
0122                ; and check bus conditions.
0123                ;
0124                BSTOP
0125        0020    2346     BCF     RB,SDA       ; Return SDA to low
0126        0021    0000     NOP
0127        0022    0000     NOP
0128        0023    2706     BSF     RB,SCL       ; Set SCL high
0129        0024    6001     MOVLW   1            ; Ready error code 1
0130        0025    3706     BTFSS   RB,SCL       ; High?
0131        0026    4400     CALL    ERR          ; No, SCL locked low by
                                                  ; device
0132        0027    2746     BSF     RB,SDA       ; SDA goes from low to
                                                  ; high during SCL high
0133        0030    6004     MOVLW   4            ; Ready error code 4
0134        0031    3746     BTFSS   RB,SDA       ; High?
0135        0032    4400     CALL    ERR          ; No, SDA bus not release
                                     ; for STOP
0136        0033    4000     RETLW   0
0137                ;
0138                ;END SUB
0139                ;
0040                ;-----------------------------------------------------
0141                ;  Serial data send from PIC16CXX to serial EEPROM,
                    ;  bit-by-bit subroutine
0142                ;-----------------------------------------------------
0143                ;  Input :  None
0144                ;  Output           :  To (DI) of serial EEPROM device
00145               ;-----------------------------------------------------
0146                ;
0147                BITIN
0148        0034    6277     MOVLW   B'10111111'  ; Force SDA line as input
0149        0035    0006     TRIS    RB
0150        0036    2746     BSF     RB,SDA       ; Set SDA for input
0151        0037    2352     BCF     EEPROM,DI
0152        0040    2706     BSF     RB,SCL       ; Clock high
0153        0041    6001     MOVLW   1
0154        0042    3306     BTFSC   RB,SCL       ; Skip if SCL  is high
0155        0043    5047     GOTO    BIT1
0156        0044    3411     BTFSS   FLAG,ERROR   ; Remain as first error
                                                  ; encountered
0157        0045    0053     MOVWF   ERCODE       ; Save error code
0158        0046    2411     BSF     FLAG,ERROR   ; Set error flag
0159                    BIT1
0160        0047    3346     BTFSC   RB,SDA       ; Read SDA pin
0161        0050    2752     BSF     EEPROM,DI    ; DI = 1
0162        0051    0000     NOP                  ; Delay
0163        0052    2306     BCF     RB,SCL       ; Return SCL to low
0164        0053    4000     RETLW   0
0165                ;
0166                ;END SUB
0168                ;
```

```
0169             ;-----------------------------------------------------
0170             ;  Serial data receive from serial EEPROM to PIC16CXX,
                 ;  bit-by-bit subroutine
0171             ;-----------------------------------------------------
0172             ;  Input :  EEPROM file
0173             ;  Output           :  From (DO) of serial EEPROM device
          ;                      to PIC
0174             ;-----------------------------------------------------
0175             ;
0176             BITOUT
0177      0054   6077     MOVLW   B'00111111'  ; Set SDA, SCL as outputs
0178      0055   0006     TRIS    RB
0179      0056   3712     BTFSS   EEPROM,DO
0180      0057   5070     GOTO    BIT0
0181      0060   2746     BSF     RB,SDA       ; Output bit 0
0182      0061   6002     MOVLW   2
0183      0062   3346     BTFSC   RB,SDA       ; Check for error code 2
0184      0063   5074     GOTO    CLK1
0185      0064   3411     BTFSS   FLAG,ERROR   ; Remain as first error
                                                ; encountered
0186      0065   0053     MOVWF   ERCODE       ; Save error code
0187      0066   2411     BSF     FLAG,ERROR   ; Set error flag
0188      0067   5074     GOTO    CLK1         ; SDA locked low by device
0189                 ;
0190                 BIT0
0191      0070   2346     BCF     RB,SDA       ; Output bit 0
0192      0071   0000     NOP                  ; Delay
0193      0072   0000     NOP
0194      0073   0000     NOP
0195                 CLK1
0196      0074   2706     BSF     RB,SCL
0197      0075   6001     MOVLW   1            ; Error code 1
0198      0076   3306     BTFSC   RB,SCL       ; SCL locked low?
0199      0077   5103     GOTO    BIT2         ; No.
0200      0100   3411     BTFSS   FLAG,ERROR   ; Yes.
0201      0101   0053     MOVWF   ERCODE       ; Save error code
0202      0102   2411     BSF     FLAG,ERROR   ; Set error flag
0203                 BIT2
0204      0103   0000     NOP
0205      0104   0000     NOP
0206      0105   2306     BCF     RB,SCL       ; Return SCL to low
0207      0106   4000     RETLW   0
0208             ;
0209             ;END SUB
0211             ;
0212             ;
0213             ;-----------------------------------------------------
0214             ;        RECEIVE DATA subroutine
0215             ;-----------------------------------------------------
0216             ;        Input    :  None
```

```
0217             ;       Output  :  RXBUF = Receive 8-bit data
0218             ;-------------------------------------------------
0219             ;
0220             RX
0221     0107    6010    MOVLW   .8          ; 8 bits of data
0222     0110    0066    MOVWF   COUNT
0223     0111    0165    CLRF    RXBUF
0224                 ;
0225                 RXLP
0226     0112    1565    RLF     RXBUF       ; Shift data to buffer
0227     0113    SKPC
0228     0113    3403  + BTFSS   3,0
0228     0114    2025    BCF     RXBUF,0     ; carry –> f(0)
0229     0115            SKPNC
0230     0115    3003  + BTFSC   3,0
0230     0116    2425    BSF     RXBUF,0
0231     0117    4434    CALL    BITIN
0232     0120    3352    BTFSC   EEPROM,DI
0233     0121    2425    BSF     RXBUF,0     ; Input bit =1
0234     0122    1366    DECFSZ  COUNT       ; 8 bits?
0235     0123    5112    GOTO    RXLP
0236     0124    2712    BSF     EEPROM,DO   ; Set acknowledge bit = 1
0237     0125    4454    CALL    BITOUT      ; to STOP further input
0238     0126    4000    RETLW   0
0239                 ;
0240                 ;END SUB
0241                 ;
0242                 ;-------------------------------------------------
0243                 ;  TRANSMIT DATA subroutine
0244                 ;-------------------------------------------------
0245                 ;  Input    :  TXBUF
0246                 ;  Output   :  Data X'mitted to EEPROM device
0247                 ;-------------------------------------------------
0248                 ;
0249                 TX
0250     0127    6010    MOVLW   .8
0251     0130    0066    MOVWF   COUNT
0252                ;
0253                     TXLP
0254     0131    2312    BCF     EEPROM,DO   ; Shift data bit out.
0255     0132    3364    BTFSC   TXBUF,7     ; If shifted bit=0, data
                                             ; bit = 0
0256     0133    2712    BSF     EEPROM,DO   ; Otherwise data bit = 1
0257     0134    4454    CALL    BITOUT      ; Serial data out
0258     0135    1564    RLF     TXBUF       ; Rotate TXBUF left
0259     0136            SKPC                ; f(6) –> f(7)
0260     0136    3403  + BTFSS   3,0
0260     0137    2024    BCF     TXBUF,0     ; f(7) –> carry
0261     0140            SKPNC               ; carry –> f(0)
0262     0140    3003  + BTFSC   3,0
```

```
0262    0141    2424    BSF     TXBUF,0
0263    0142    1366    DECFSZ  COUNT           ; 8 bits done?
0264    0143    5131    GOTO    TXLP            ; No.
0265    0144    4434    CALL    BITIN           ; Read acknowledge bit
0266    0145    6003    MOVLW   3
0267    0146    3352    BTFSC   EEPROM,DI       ; Check for
                                                ; acknowledgement
0268    0147    4400    CALL    ERR             ; No acknowledge from
                                                ; device
0269    0150    4000    RETLW   0
0270                ;
0271                ;END SUB
0273                ;
0274                ;-------------------------------------------------
0275                ;   BYTE-WRITE, write one byte to EEPROM device
0276                ;-------------------------------------------------
0277                ;   Input   :   DATAO = data to be written
0278                ;               ADDR = destination address
0279                ;               SLAVE = device address (1010xxx0)
0280                ;   Output  :   Data written to EEPROM device
0281                ;-------------------------------------------------
0282                ;
0283    0200            ORG     200             ; The location for BYTE-
                                                ; WRITE routine can be
0284                ;                           ; assigned anywhere
                                                ; between (377- 777)
                                                ; octal.
0285                    WRBYTE
0286    0200    1023    MOVF    SLAVE,W         ; Get SLAVE address
0287    0201    0064    MOVWF   TXBUF           ; to TX buffer
0288    0202    4404    CALL    BSTART          ; Generate START bit
0289    0203    4527    CALL    TX              ; Output SLAVE address
0290    0204    1020    MOVF    ADDR,W          ; Get WORD address
0291    0205    0064    MOVWF   TXBUF           ; into buffer
0292    0206    4527    CALL    TX              ; Output WORD address
0293    0207    1022    MOVF    DATAO,W         ; Move DATA
0294    0210    0064    MOVWF   TXBUF           ; into buffer
0295    0211    4527    CALL    TX              ; Output DATA and detect
                                                ; acknowledgement
0296    0212    4420    CALL    BSTOP           ; Generate STOP bit
0297                ;
0298                ;
0299                ;
0300                ;-------------------------------------------------
0301                ;   BYTE-READ, read one byte from serial EEPROM
                    ;   device
0302                ;-------------------------------------------------
0303                ;   Input   :       ADDR  = source address
0304                ;                   SLAVE = device address (1010xxx0)
0305                ;   Output  :       DATAI = data read from serial
                                        EEPROM
```

7

```
0306                ;-------------------------------------------------
0307                ;
0308    0300            ORG     300         ; The location for BYTE-
                                            ; READ routine can be
                                            ; assigned anywhere
0309                ;                       ; between (377-777) octal.
0310                RDBYTE
0311    0300    1023    MOVF    SLAVE,W     ; Move SLAVE address
0312    0301    0064    MOVWF   TXBUF       ; into buffer (R/W = 0)
0313    0302    4404    CALL    BSTART      ; Generate START bit
0314    0303    4527    CALL    TX          ; Output SLAVE address.
                                            ; Check ACK.
0315    0304    1020    MOVF    ADDR,W      ; Get WORD address
0316    0305    0064    MOVWF   TXBUF
0317    0306    4527    CALL    TX          ; Output WORD address.
                                            ; Check ACK.
0318    0307    4404    CALL    BSTART      ; START READ (if only one
                                            ; device
0319    0310    1023    MOVF    SLAVE,W     ; is connected to the I2C
                                            ; bus)
0320    0311    0064    MOVWF   TXBUF
0321    0312    2424    BSF     TXBUF,0     ; Specify READ mode
                                            ; (R/W = 1)
0322    0313    4527    CALL    TX          ; Output SLAVE address
0323    0314    4507    CALL    RX          ; READ in data and
                                            ; acknowledge
0324    0315    4420    CALL    BSTOP       ; Generate STOP bit
0325    0316    1065    MOVF    RXBUF       ; Save data from buffer
0326    0317    0061    MOVWF   DATAI       ; to DATAI file.
0327                ;
0328                ;
0329                ;
0330                END

%ASM-I, No Errors, No Warnings
```

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