

Interfacing the X24C00 to the Motorola 6805 Microcontroller

by Applications Staff, June 1995

The following code demonstrates how the Xicor X24C00 serial E²PROM could be interfaced to the Motorola 6805 microcontroller when connected as shown in Figure 1. The interface uses port A of the 6805, with the PA0 pin connected to the serial data (SDA) and PA1 connected to serial clock (SCL) input of

the E²PROM. Additional code can be found on the Xicor web site at <http://www.xicor.com> that will implement interfaces between the Motorola 6805 microcontroller and other Xicor serial devices, including the X24C01, X24C02/04/08/16 family, X25C02/020/040 SPI family, and X24C44/45 serial NOVRAMs.

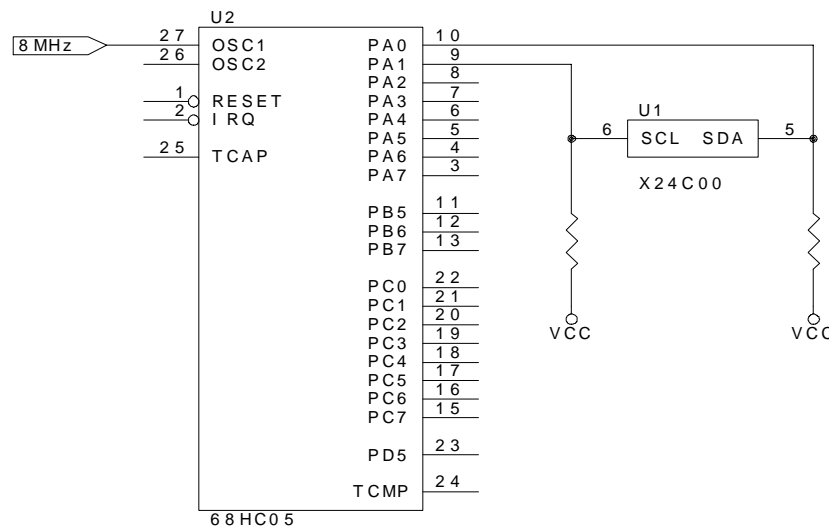


Figure 1. Typical hardware connection for interfacing an X24C00 to a 6805 microcontroller

```

0001      *****
0002      * THIS CODE WAS DESIGNED TO DEMONSTRATE HOW THE XICOR X24C00 PART COULD *
0003      * BE INTERFACED TO THE 68HC05 MICROCONTROLLER. THE INTERFACE USES 2 LINES *
0004      * FROM PORT A (PA0 AND PA1) TO COMMUNICATE. *
0005      * *
0006      * THE CODE SHOWN DEMONSTRATES A 'BYTE READ' AND 'BYTE WRITE'. *
0007      * *
0008      * THE MAINLINE OF THIS PROGRAM READS THE DATA LOCATED AT ADDRESS 04H AND *
0009      * THEN WRITES THAT DATA BACK TO ADDRESS 0BH. *
0010      *****
0011
0012 0001      SCLBIT      EQU 1          BIT INDICATING PORTA SCL POSITION
0013 0000      SDABIT      EQU 0          BIT INDICATING PORTA SDA POSITION
0014 0003      SDAOUT      EQU $03       MAKES SDA AN OUTPUT IF STORED IN DDRA
0015 0002      SDAIN       EQU $02       MAKES SDA AN INPUT IF STORED IN DDRA
0016 0080      DMASK       EQU $80       USED TO MASK BIT TO SEND TO DUT
0017
0018 0000      PORTA       EQU $00       PORT A MEMORY ADDRESS
0019 0004      DDRA        EQU $04       PORT A DIRECTION REGISTER OFFSET
0020
0021 0080      ADDR        EQU $80       LOCATION FOR X24C00 ADDRESS TO ACCESS
0022 0082      DATA       EQU $82       LOCATION FOR X24C00 DATA TRANSFERED
0023 0083      COUNT      EQU $83       COUNTER LOCATION FOR LOOPING
0024 0086      TEMP1      EQU $86       SCRATCH PAD REGISTER
0025
0026      *****
0027      * RESET VECTOR ENTRY POINT *
0028      *****
0029
0030 1ffe      ORG          $1FFE       RESET VECTOR ADDRESS TO PROGRAM ENTRY
0031 1ffe 01 00  FDB        $0100     JUMP TO BEGINNING OF EXECUTABLE CODE
0032
0033      *****
0034      * PROGRAM ENTRY POINT *
0035      *****
0036
0037 0100      ORG          $0100     BEGINNING OF EXECUTABLE CODE
0038
0039 0100 a6 ff  BEGIN:      LDA  #$FF     MAKE PORTA ALL OUTPUTS
0040 0102 b7 04          STA  DDRA
0041 0104 a6 ff          LDA  #$FF     MAKE PORTA ALL ONES
0042 0106 b7 00          STA  PORTA
0043 0108 a6 03          LDA  #$03     MAKE SDA AND SCL OUTPUTS
0044 010a b7 04          STA  DDRA
0045 010c a6 04          LDA  #$04
0046 010e b7 80          STA  ADDR
0047 0110 cd 01 1c      JSR  RDBYT    READ DATA FROM ADDRESS 002DH
0048 0113 a6 0b          LDA  #$0B
0049 0115 b7 80          STA  ADDR
0050 0117 cd 01 32      JSR  WRBYT    WRITE DATA BACK TO ADDRESS 0041H
0051 011a 20 fe          BRA  *        LOOP UNTIL RESET
0052

```

```

0053      *****
0054      * READ A BYTE "RANDOM READ SEQUENCE".  THE ADDRESS TO READ IS STORED *
0055      *   IN ADDR.  THE DATA FROM THE DUT IS STORED IN DATA.          *
0056      *****
0057
0058 011c cd 01 aa      RDBYT:  JSR  START          READ A BYTE FROM THE ADDRESS INDICATED
0059 011f b6 80                LDA  ADDR          IN 'ADDR'
0060 0121 48                LSLA
0061 0122 48                LSLA
0062 0123 aa 83                ORA  #$83          BUILD INSTRUCTION
0063 0125 b7 82                STA  DATA
0064 0127 cd 01 85          JSR  OUTI          SEND SLAVE ADDRESS
0065 012a cd 01 50          JSR  INBYT        READ DATA FROM 2404
0066 012d 12 00                BSET #SCLBIT,PORTA
0067 012f 10 00                BSET #SDABIT,PORTA
0068 0131 81                RTS
0069
0070      *****
0071      * WRITE A BYTE "BYTE WRITE SEQUENCE".  THE ADDRESS TO WRITE IS STORED *
0072      *   IN ADDR.  THE DATA TO WRITE IS STORED IN DATA.          *
0073      *****
0074
0075 0132 b6 82          WRBYT:  LDA  DATA          WRITE TO BYTE POINTED TO BY ADDR THE
0076 0134 b7 86                STA  TEMP1        VALUE IN LOCATION 'DATA'
0077 0136 cd 01 aa          JSR  START          SEND START COMMAND
0078 0139 b6 80                LDA  ADDR
0079 013b 48                LSLA
0080 013c 48                LSLA
0081 013d aa 43                ORA  #$43
0082 013f b7 82                STA  DATA
0083 0141 cd 01 85          JSR  OUTI          SEND SLAVE ADDRESS
0084 0144 b6 86                LDA  TEMP1
0085 0146 b7 82                STA  DATA
0086 0148 cd 01 67          JSR  OUTD          SEND WRITE DATA
0087 014b 12 00                BSET #SCLBIT,PORTA
0088 014d 10 00                BSET #SDABIT,PORTA
0089 014f 81                RTS
0090
0091      *****
0092      * READ 8 BITS FROM THE DUT.  THE RESULTS ARE RETURNED IN DATA.  *
0093      *****
0094
0095 0150 a6 02          INBYT:  LDA  #SDAIN        MAKE SDA AN INPUT
0096 0152 b7 04                STA  DDRA
0097 0154 a6 08                LDA  #$08          PREPARE TO SHIFT IN 8 BITS
0098 0156 b7 83                STA  COUNT
0099 0158 cd 01 b3          LOOPI: JSR  CLOCK        CLOCK DATA
0100 015b 46                RORA
0101 015c 39 82                ROL  DATA
0102 015e 3a 83                DEC  COUNT
0103 0160 26 f6                BNE  LOOPI        LOOP UNTIL 8 BITS ARE READ
0104 0162 a6 03                LDA  #SDAOUT       MAKE SDA AN OUTPUT
0105 0164 b7 04                STA  DDRA

```

```

0106 0166 81          RTS
0107
0108      *****
0109      * WRITE 8 BITS TO THE DUT.  THE DATA TO SEND IS IN DATA.  IF THE LAST *
0110      * BIT TO SEND IS A ONE THE SDA LINE IS MADE AN INPUT BEFORE THE EIGHTH*
0111      * CLOCK PULSE TO AVOID BUS CONTENTION WHEN THE DUT ACKNOWLEDGES.  THE *
0112      * ROUTINE FINISHES WITH SDA IN AN INPUT STATE.                          *
0113      *****
0114
0115 0167 a6 03      OUTD:   LDA  #SDAOUT
0116 0169 b7 04          STA  DDRA
0117 016b a6 08          LDA  #$08          PREPARE TO SHIFT OUT 8 BITS
0118 016d b7 83          STA  COUNT
0119 016f b6 82      LOOPO:  LDA  DATA
0120 0171 a4 80          AND  #DMASK          IS THE DATA TO BE SHIFTED 1 OR 0
0121 0173 27 04          BEQ  IS0          JUMP IF DATA SHOULD BE 0
0122 0175 10 00          BSET #SDABIT,PORTA
0123 0177 20 02          BRA  IS1
0124 0179 11 00      IS0:   BCLR #SDABIT,PORTA  MAKE SDA A 0
0125 017b cd 01 b3      IS1:   JSR  CLOCK          SEND CLOCK SIGNAL
0126 017e 38 82          LSL  DATA
0127 0180 3a 83          DEC  COUNT
0128 0182 26 eb          BNE  LOOPO          LOOP UNTIL ALL 8 BITS HAVE BEEN SENT
0129 0184 81          RTS
0130
0131 0185 a6 03      OUTI:   LDA  #SDAOUT
0132 0187 b7 04          STA  DDRA
0133 0189 a6 07          LDA  #$07          PREPARE TO SHIFT OUT 8 BITS
0134 018b b7 83          STA  COUNT
0135 018d b6 82      LOOPOI: LDA  DATA
0136 018f a4 80          AND  #DMASK          IS THE DATA TO BE SHIFTED A 1 OR A 0
0137 0191 27 04          BEQ  IS0I          JUMP IF DATA SHOULD BE 0
0138 0193 10 00          BSET #SDABIT,PORTA
0139 0195 20 02          BRA  IS1I
0140 0197 11 00      IS0I:  BCLR #SDABIT,PORTA  MAKE SDA A 0
0141 0199 cd 01 b3      IS1I:  JSR  CLOCK          SEND CLOCK SIGNAL
0142 019c 38 82          LSL  DATA
0143 019e 3a 83          DEC  COUNT
0144 01a0 26 eb          BNE  LOOPOI          LOOP UNTIL ALL 8 BITS HAVE BEEN SENT
0145 01a2 a6 02          LDA  #SDAIN
0146 01a4 b7 04          STA  DDRA
0147 01a6 cd 01 b3      JSR  CLOCK
0148 01a9 81          RTS
0149
0150      *****
0151      * ISSUE A START COMMAND *
0152      *****
0153
0154 01aa 10 00      START:  BSET #SDABIT,PORTA  MAKE SURE THAT SDA IS HIGH
0155 01ac 12 00          BSET #SCLBIT,PORTA  MAKE SURE THAT SCL IS HIGH
0156 01ae 11 00          BCLR #SDABIT,PORTA  FORCE SDA LOW
0157 01b0 13 00          BCLR #SCLBIT,PORTA  FORCE SCL LOW
0158 01b2 81          RTS

```

```
0159
0160 *****
0161 * ISSUE A CLOCK PULSE. WHILE THE CLOCK IS HIGH THE VALUE ON THE *
0162 * SDA LINE IS PLACED IN THE LSB OF REG A. WHEN A READ IS TAKING*
0163 * PLACE THE LSB OF REG A WILL INDICATE THE VALUE FROM THE DUT. *
0164 *****
0165
0166 01b3 12 00      CLOCK:      BSET #SCLBIT,PORTA    PROVIDE A CLOCK ON SCL, START HIGH
0167 01b5 b6 00      LDA  PORTA          READ SDA WHILE SCL IS HIGH
0168 01b7 13 00      BCLR #SCLBIT,PORTA
0169 01b9 a4 01      AND  #$01           SDA VALUE IS IN LOWER BIT OF A REG
0170 01bb 81        RTS
```