

# M68HC11 and X2402 Family

```

0001 *****
0002 * This code was designed to demonstrate how the X2404 family of parts could *
0003 * be interfaced to the M68HC11 microcontroller. The interface uses 2 lines *
0004 * from Port D (PD0 and PD1) to communicate. Other bus compatible *
0005 * parts can be added to the bus as long as they do not have $A as their *
0006 * device identifier. The routines RDBYT and WRBYT are tailored specifically *
0007 * to the X2404 family. The routines START, STOP, ACK, NACK, OUTBYT, and *
0008 * INBYT can be considered generic bus routines. *
0009 * *
0010 * The code shown demonstrates a 'random read' and 'byte write'. The other *
0011 * modes of operation can be created by expanding upon these routines. *
0012 * Acknowledge polling is used to determine when the write cycle finishes. *
0013 * *
0014 * This code will work with all Xicor bus compatible EEPROMs regardless of *
0015 * their size. As long as the address pins are configured correctly this *
0016 * code will not know the difference between a bus with a single X24C16 and a *
0017 * bus with eight X2402s. *
0018 * *
0019 * The mainline of this program reads the data located at address 002DH and *
0020 * then writes that data back to address 0041H. This program has been tested *
0021 * using the X2402, X2404, X24C04, and X24C16. *
0022 *****
0023
0024 0002 SCLBIT EQU $02 MASK INDICATING PORTD SCL POSITION
0025 0001 SDABIT EQU $01 MASK INDICATING PORTD SDA POSITION
0026 0003 SDAOUT EQU $03 MAKES SDA AN OUTPUT IF STORED IN DDRD
0027 0002 SDAIN EQU $02 MAKES SDA AN INPUT IF STORED IN DDRD
0028 0080 DMASK EQU $80 USED TO MASK BIT TO SEND TO DUT
0029
0030 0008 PORTD EQU $08 PORT D OFFSET IN 'PAGE' $1000
0031 0009 DDRD EQU $09 PORT D DIRECTION REGISTER OFFSET
0032
0033 0080 ADDR EQU $80 LOCATION FOR 2404 ADDRESS TO ACCESS
0034 0082 DATA EQU $82 LOCATION FOR 2404 DATA TRANSFERED
0035 0083 COUNT EQU $83 COUNTER LOCATION FOR LOOPING
0036 0084 PDDATA EQU $84 TEMP REGISTER FOR DATA STORAGE
0037 0085 COUNT2 EQU $85 COUNTER LOCATION FOR ACK POLLING
0038
0039 *****
0040 * RESET VECTOR ENTRY POINT *
0041 *****
0042
0043 FFFE ORG $FFFE RESET VECTOR ADDRESS TO PROGRAM ENTRY
0044 FFFE E0 00 FDB $E000 JUMP TO BEGINNING OF EXECUTABLE CODE
0045
0046 *****
0047 * PROGRAM ENTRY POINT *
0048 *****
0049
0050 E000 ORG $E000 BEGINNING OF EXECUTABLE CODE
0051
0052 E000 8E 00 FF BEGIN: LDS #$00FF INITIALIZE STACK POINTER
0053 E003 CE 10 00 LDX #$1000 INITIALIZE PAGE OFFSET LOCATION
0054 E006 86 FF LDAA #$FF MAKE PORTD ALL ONES
0055 E008 A7 08 STAA PORTD,X
0056 E00A 86 03 LDAA #$03 MAKE SDA AND SCL OUTPUTS
0057 E00C A7 09 STAA DDRD,X
0058 E00E CC 00 2D LDD #$002D
0059 E011 DD 80 STD ADDR
0060 E013 BD E0 23 JSR RDBYT READ DATA FROM ADDRESS 002DH
0061 E016 CC 00 41 LDD #$0041
0062 E019 DD 80 STD ADDR
0063 E01B BD E0 53 JSR WRBYT WRITE DATA BACK TO ADDRESS 0041H
0064 E01E BD E0 D0 JSR ACKPOL PERFORM ACK POLLING
0065 E021 20 FE BRA * LOOP UNTIL RESET

```