AN1755

Interfacing the MC68HC705C8A to the DS2430A 256-Bit 1-Wire™ EEPROM

By Mark Glenewinkel
Field Applications Engineering
Consumer Systems Group
Austin, Texas

Introduction

This application note describes the interface between the MC68HC705C8A (C8A) and the DS2430A 1-Wire 256-bit EEPROM from Dallas Semiconductor. The 1-Wire interface reduces the overhead of control, data, address, and power pins found with other EEPROM devices. One pin on the DS2430A combines all of these functions.

More embedded applications today are demanding non-volatile memory storage. For instance, some applications in which EEPROM technology is required are:

- Reprogrammable calibration data for test/measurement equipment
- Power down information storage for consumer electronics like TVs and VCRs
- Identification number storage for remote addressing or security
- Storage of telecommunication information such as phone number recall and speed dialing

TM 1-Wire is a trademark of Dallas Semiconductor.

Circuitry and example code are provided in **Code Listing** to demonstrate the interface between the two parts.

Features

The DS2430A provides these features:

- 256-bit EEPROM (electrically erasable programmable read-only memory) plus a 64-bit one-time programmable (OTP) application register
- EEPROM organized as one page of 32 bytes for random access
- Unique factory-lasered 64-bit registration number ensures absolute identity because no two devices are alike.
- Reduces control, address, data, and power to a single data pin
- Directly connects to a single port pin of a microcontroller
- Communicates up to 16.3 Kbits per second
- Low cost TO-92 or 6-pin TSOC package
- Reads and writes over a wide voltage range of 2.8 volts to 6.0 volts from –40 °C to +85 °C

Description

The DS2430A has 256 bits of EEPROM that are addressed as 32 bytes of memory. The interface to the device is based on a proprietary 1-Wire interface. The DS2430A has four main data components. These are:

- 64-bit lasered ROM
- 256-bit EEPROM memory with 256-bit scratchpad RAM (random-access memory)
- 64-bit EEPROM application register with 64-bit scratchpad RAM
- 8-bit status register

The scratchpad RAM acts as a buffer when writing to the EEPROM. Eight memory functions give the user the ability to write, read, and check the status of the DS2430A.

The DS2430A has a unique 64-bit lasered ROM. The first eight bits signify the 1-Wire Family member being addressed. The next 48 bits are unique to each DS2430A device, allowing more than 281 trillion different devices to be in the field.

The last eight bits are a CRC (cyclic redundancy check) of the first 56 bits. The 64-bit ROM allows multiple DS2430As to be on the bus. Once the ROM code is matched, the memory functions become accessible and the master may issue any of the eight memory commands. All data transceived between the master and the DS2430A is read and written least significant bit (LSB) first.

DS2430A Hardware Interface

Pinout and Pin Descriptions

The bidirectional DATA pin is the only interface pin to a microcontroller (MCU). Parasitic power is derived from the required pullup resistor on the DATA pin. No other power input is needed for the DS2430A.

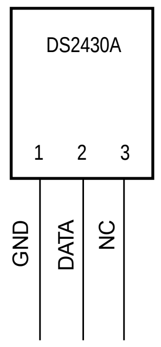


Figure 1. DS2430A TO-92 Pinout

Block Diagram

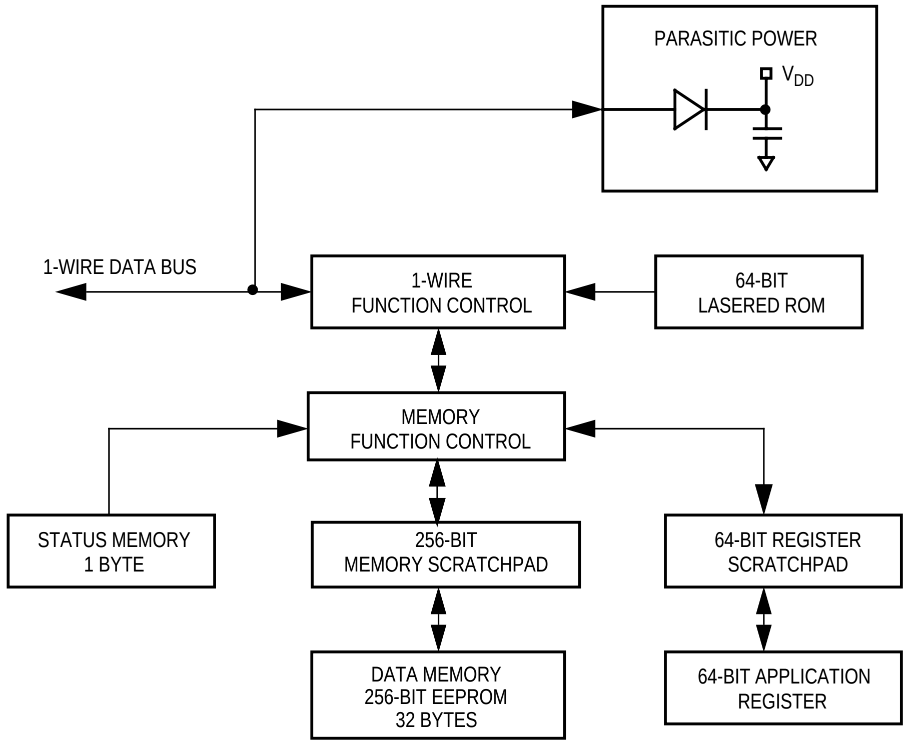


Figure 2. DS2430A Block Diagram

1-Wire Interface

Figure 3 shows the hardware interface of the 1-Wire bus. The bus has a single master and one or more slave devices. In all cases, the DS2430A is a slave. It is important that each device on the bus be able to drive it at the appropriate time. Thus, each device must have open drain or three-state outputs. The maximum bus rate allowed is 16.3 Kbits per second.

The idle state of the bus is high. If for any reason a transaction is suspended, the bus must be left in the idle state if the transaction is to resume at a later time. If the bus is held low for more than 120 µs, one or more of the slave devices could be reset. A pullup resistor is required on the bus to ensure proper idling of the bus and to provide parasitic power to the DS2430A.

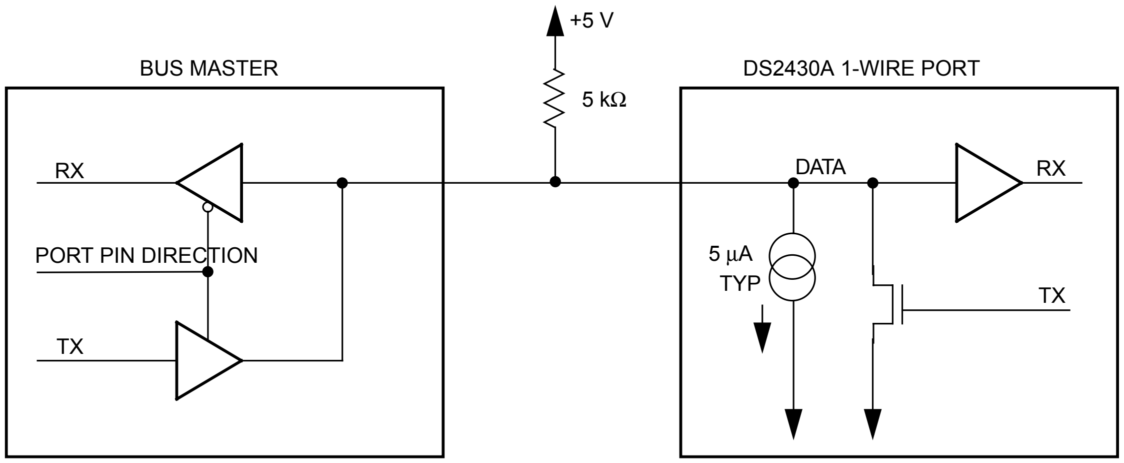


Figure 3. 1-Wire Bus Interface

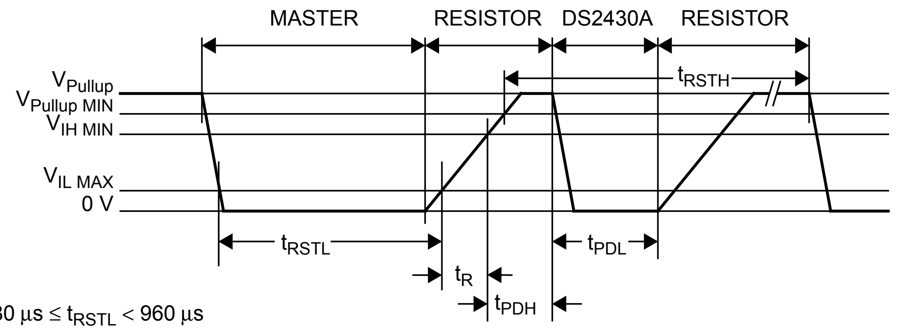
1-Wire Timing

The 1-Wire protocol is divided into two types of transactions. These are:

- Reset and presence pulse
- Write and read one bit of data

When a device is idling in the high state, the master starts communicating to the DS2430A by issuing a reset pulse. The master must drive the bus low for at least 480 µs. After this time, the master turns its port pin into a high impedance input pin and allows the pullup resistor to bring the bus back high. Over the next 480 µs, the master reads the bus looking for a low. If the DS2430A is active and ready to communicate, it will drive the bus low. If the master does not receive a presence pulse, further communication cannot occur.

Figure 4 shows the reset and presence pulse timing.



480 µs ≤ t_{RSTL} < 960 µs
480 µs ≤ t_{RSTH} < INFINITE, INCLUDES RECOVERY TIME
15 µs ≤ t_{PDH} < 60 µs
60 µs ≤ t_{PDL} < 240 µs

Figure 4. Reset and Presence Pulse Timing

After the presence pulse is received, data may now be communicated between the master and the slave. A bit is transceived by specific time slots that are initiated by the master sending a falling edge sync pulse. The sync pulse defines the start of a time slot that is at least 60 µs long. After this time slot is finished, a recovery time of at least 1 µs is required to give the DS2430A time to respond to the next bit being transmitted. The time slot and recovery time together add up to 61 µs, which define the maximum communication speed of 16.3 Kbits per second.

Three different time slots can be generated. They are:

- Write-one time slot
- Write-zero time slot
- Read data time slot

The timing diagrams for these time slots are shown in **Figure 5**, **Figure 6**, and **Figure 7**.

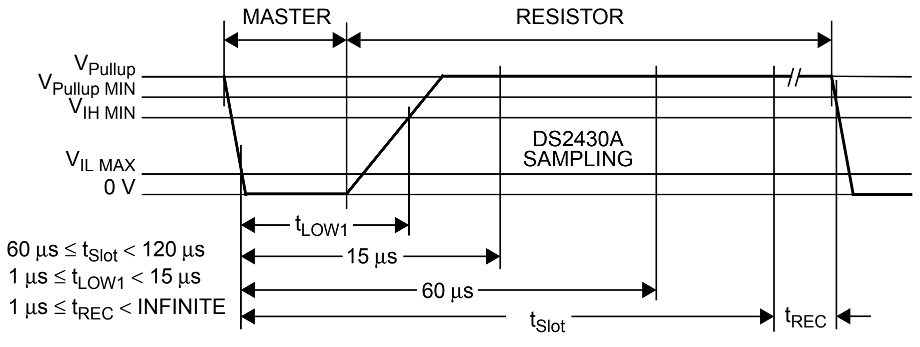


Figure 5. Write-One Time Slot

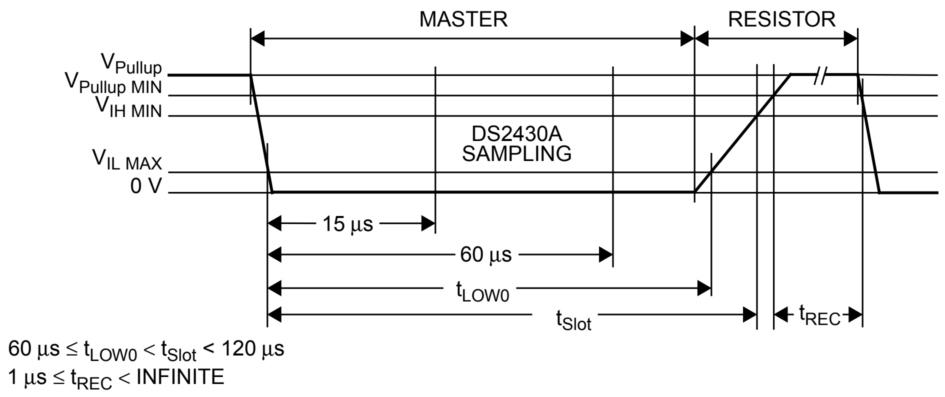


Figure 6. Write-Zero Time Slot

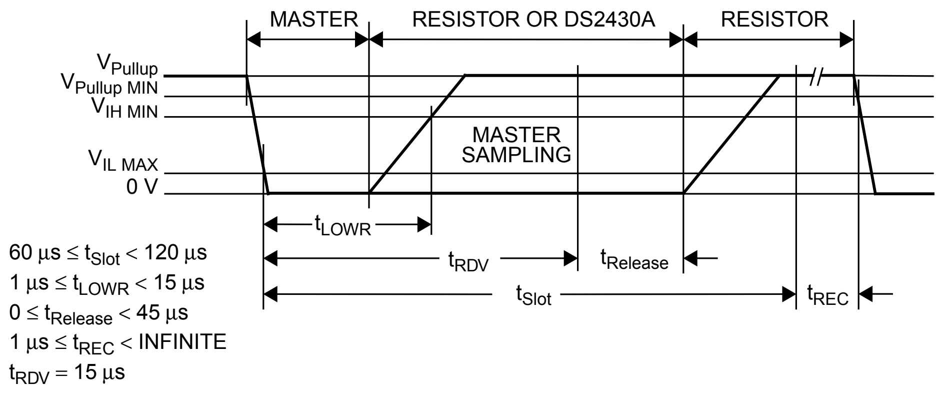


Figure 7. Read Data Time Slot

A step-by-step example of the protocol needed to read the DS2430A's status register is:

1. The master transmits a reset pulse.
2. The master waits for the presence pulse from the DS2430A. Once detected, go to step 3.
3. The master sends out the skip ROM command to the DS2430A. Since the DS2430A is the only device on the bus, a match ROM address command is not needed. The code for skip ROM is $CC or %11001100 and is sent out LSB first.
   a. Write-zero — two times
   b. Write-one — two times
   c. Write-zero — two times
   d. Write-one — two times
4. After the ROM function has been sent, the DS2430A is ready for a memory function command. The code for read status register is $66 or %01100110.
   a. Write-zero
   b. Write-one — two times
   c. Write-zero — two times
   d. Write-one — two times
   e. Write-zero
5. The DS2430A is now ready to send out its status register contents on the bus. The master sends out eight read data time slots to read each bit off the bus and store them in its memory.
6. The transaction is now complete. To issue another command, the DS2430A must be reset again.

DS2430A Software Interface

The transaction sequence to access the DS2430A over the 1-Wire bus is illustrated in **Figure 8**.

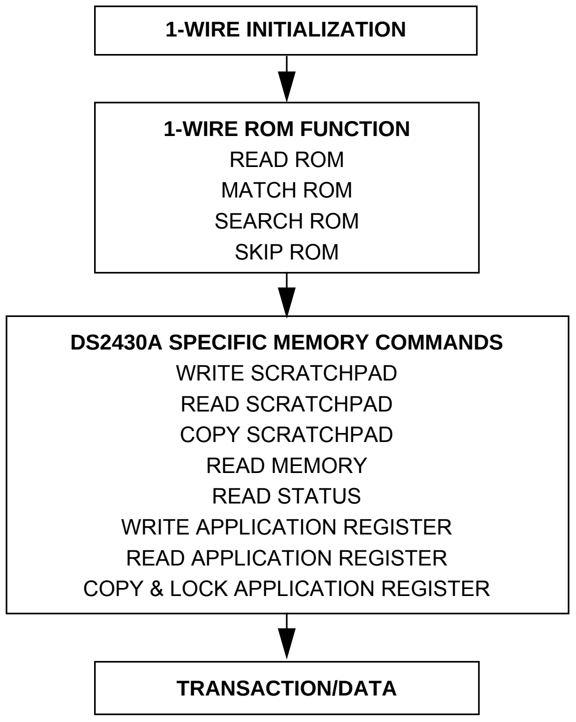


Figure 8. DS2430A Transaction Sequence

Initialization

All transactions start with a device initialization routine. This is accomplished by the master sending a reset pulse and then reading a presence pulse from the DS2430A.

ROM Function Commands

Once the bus master has detected a presence pulse, one of the ROM commands can be issued. All ROM functions are eight bits long.

Read ROM — $33

This command allows the bus master to read the DS2430A's unique 64-bit ROM code. This command can be used only if there is a single DS2430A on the bus. Otherwise, a data collision will occur. All 1-Wire devices have this 64-bit lasered ROM. The first eight bits signify the

1-Wire Family. The next 48 bits are unique to the DS2430A. The last eight bits are a CRC (cyclic redundancy check) of the first 56 bits. Consult the DS2430A data sheet for more detail on the 64-bit ROM.

Match ROM — $55

If more than one DS2430A is on the bus, this command allows the bus master to address a specific DS2430A. Only the ROM that matches will communicate with the master. All other devices will go inactive until a reset pulse is sent.

Search ROM — $F0

When a system has many devices on the bus, the master may not know the number of devices on the bus. This command allows the master to determine the ROM contents of all devices on the bus. Consult the DS2430A data sheet for more detail on this command.

Skip ROM — $CC

This ROM command can save access time if only one device is on the bus. The 64-bit ROM code is not needed to access the memory function commands. If more than one device is on the bus and this command is issued, a data collision will occur. This application note demonstrates this command to access the DS2430A.

Memory Function Commands

After the device has been addressed correctly with the ROM commands, the DS2430A is ready to execute memory functions.

Write Scratchpad — $0F

This command will write data to the data memory's scratchpad RAM. The command is followed by an address byte and then subsequent data. The address is incremented after each data byte is written. Once the address hits location $1F, it wraps around to $00.

Read Scratchpad — $AA

This command will read data from the data memory's scratchpad. The command is followed by an address byte and subsequent data is read. The address is incremented after each data byte is read. Once the address hits location $1F, it wraps around to $00.

Copy Scratchpad — $55

After the data memory scratchpad RAM is verified, it can then be transferred to the data memory EEPROM. After this command and a validation key of $A5 is sent to the device, all 256 bits are written to the EEPROM array. If only a few bytes are to be written to EEPROM, the scratchpad should contain a copy of the latest EEPROM data before the write and copy scratchpad commands are executed. After the copy scratchpad command is issued, the data line must be held high for 100 ms to ensure proper programming of the EEPROM array.

Read Memory — $F0

This command is used to read the data memory EEPROM array. After the command is issued, the whole EEPROM array is transferred to the data memory scratchpad RAM. After the read memory command, a starting address is sent. Data is then read one byte at a time. The address is incremented after each data byte is read. Once the address hits location $1F, it wraps around to $00. If the user wants only to copy the EEPROM array to the scratchpad RAM, issue a reset pulse to the DS2430A after the read memory command is issued.

Write Application Register — $99

This command will write data to the application register's scratchpad RAM. The command is followed by an address byte and then subsequent data. The address is incremented after each data byte is written. Once the address hits location $07, it wraps around to $00. This command can be used as long as the application register is not locked. If the application register already has been locked, the data written to the scratchpad RAM will be lost.

Read Status Register — $66

The status register is used to determine if the application register has been programmed and locked. A validation key of $00 must be sent before the status register can be read. If the register reads $FF, the application register is unlocked. If the register reads $FC, the application register is locked.

Read Application Register — $C3

This command will read data from the application register's scratchpad RAM. If the application register is locked, the data read will come directly from the EEPROM memory. The command is followed by an address byte and subsequent data is read. The address is incremented after reading a data byte. Once the address hits location $07, it wraps around to $00.

Copy and Lock Application Register — $5A

After the application register's scratchpad RAM is verified, then it can be transferred to the application register EEPROM. After this command and a vailidation key of $A5 is sent to the device, all 64 bits are written to the EEPROM array. The application register is now locked. Additional write accesses will be denied. This command can be used only once. After the copy and lock application register command is issued, the data line must be held high for 100 ms to ensure proper programming of the EEPROM array.

Memory Map

The DS2430A's memory is comprised of three sections called the status register, data memory, and application register, all shown in **Figure 9**. The status register indicates if the application register is locked. As long as the application register is unprogrammed, the status register reads $FF. When the application register has been programmed and locked, the status register reads $FC.

The data memory consists of a 256-bit EEPROM array and a 256-bit scratchpad RAM array. The arrays are addressed by 32 8-bit sections from $00 to $1F. The scratchpad RAM acts as a buffer when writing to the device.

The application register consists of a 64-bit EEPROM array and a 64-bit scratchpad RAM array. The 64 bits are addressed by eight 8-bit sections from $00 to $07. Like the data memory, the scratchpad RAM acts as a buffer when writing to the device.

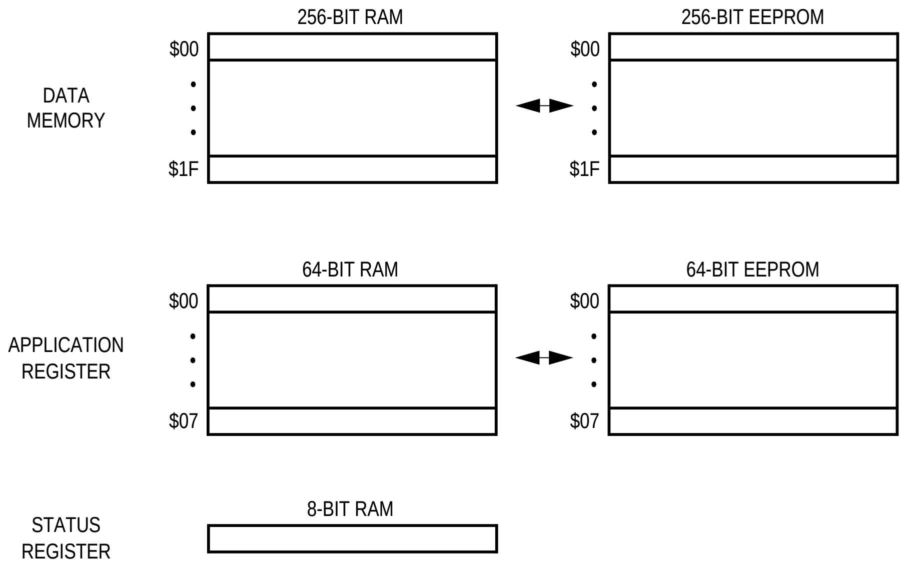


Figure 9. DS2430A Memory Map

MC68HC705C8A Hardware Interface

The MC68HC705C8A (C8A) is one of the most popular members of the HC05 Family. It has a total of 7744 bytes of EPROM and 176 bytes of RAM. The part includes a total of 24 I/O pins and seven input-only pins. Peripherals include the serial peripheral interface (SPI) bus, serial communications interface (SCI), and a 16-bit capture/compare timer.

The schematic used for testing the C8A-to-DS2430A interface on the MMEVS development system is shown in **Figure 10**. Bit 0 of port A is used to transmit and receive data on the DATA pin of the DS2430A.

For further information on the C8A, consult the *MC68HC705C8A Technical Data,* Freescale document order number MC68HC705C8A/D.

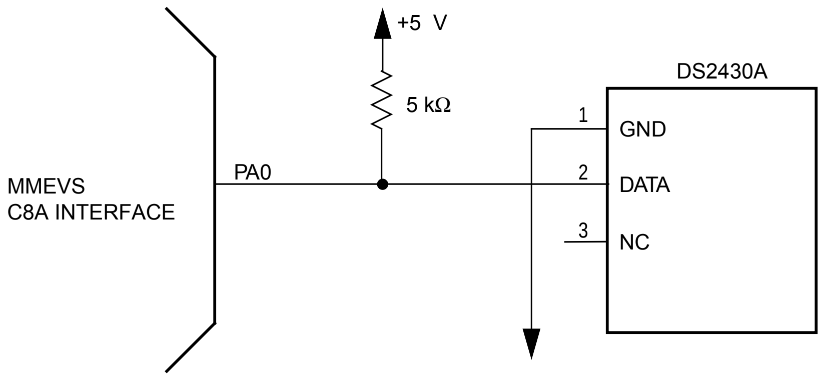


Figure 10. C8A-to-DS2430A Interface Test Circuit

MC68HC705C8A Software Interface

I/O driving or manipulation is the process of toggling I/O pins with software instructions to create a certain hardware peripheral. The HC05 CPU provides special instructions to specifically manipulate single I/O pins.

The serial transmission driver has been put into three subroutines:

- RESET_PULSE — The RESET_PULSE routine sends out a reset pulse to the DS2430A and waits for a presence pulse. If no presence pulse is found, the routine goes into an error loop.
- TXD — The TXD routine takes a byte of data and creates eight time slots of either write-one or write-zero depending on the bit being transmitted.
- RXD — The RXD routine transmits eight read data time slots. Each bit is read and shifted into a byte of RAM.

The flowcharts for the DS2430A serial I/O drivers are shown in **Figure 11**, **Figure 12**, and **Figure 13**.

Figure 14 shows the flowcharts for the main test routine.

Follow this step-by-step sequence of tests. Each memory command is preceded by a reset pulse and skip ROM command.

1. Read memory. This will transfer the contents of the EEPROM array into the scratchpad RAM.
2. Write scratch memory. Starting at address $06, write $55.
3. Read scratch memory. Starting at address $06, read the contents of the scratchpad RAM.
4. Verify that the value read from scratchpad RAM location $06 is $55. If not, go to error loop.
5. Copy scratchpad. Follow this command with a validation key of $A5. This will copy the contents of the scratchpad RAM array to the EEPROM array. Hold the DATA pin high for 100 ms to allow proper programming of the EEPROM.
6. Read memory. This will transfer the contents of the EEPROM array into the scratchpad RAM.
7. Read scratch memory. Starting at address $06, read the contents of the scratchpad RAM.
8. Verify that the value read from scratchpad RAM location $06 is $55. If not, go to error loop.

This routine demonstrates the interface software needed to communicate with the DS2430A. Although the C8A was used, any HC05 Family device could utilize this interface code. Minor adjustments of port pins and memory maps might be necessary.

The assembly code for the test routine is provided in **Code Listing**.

Development Tools

The interface was created and tested using these development tools:

- M68MMPFB0508 — Freescale MMEVS platform board
- X68EM05C9A — Freescale C series emulation module
- Win IDE Version 1.02 — Editor, assembler, and debugger by P&E Microcomputer Systems

Flowcharts for the Serial Drivers

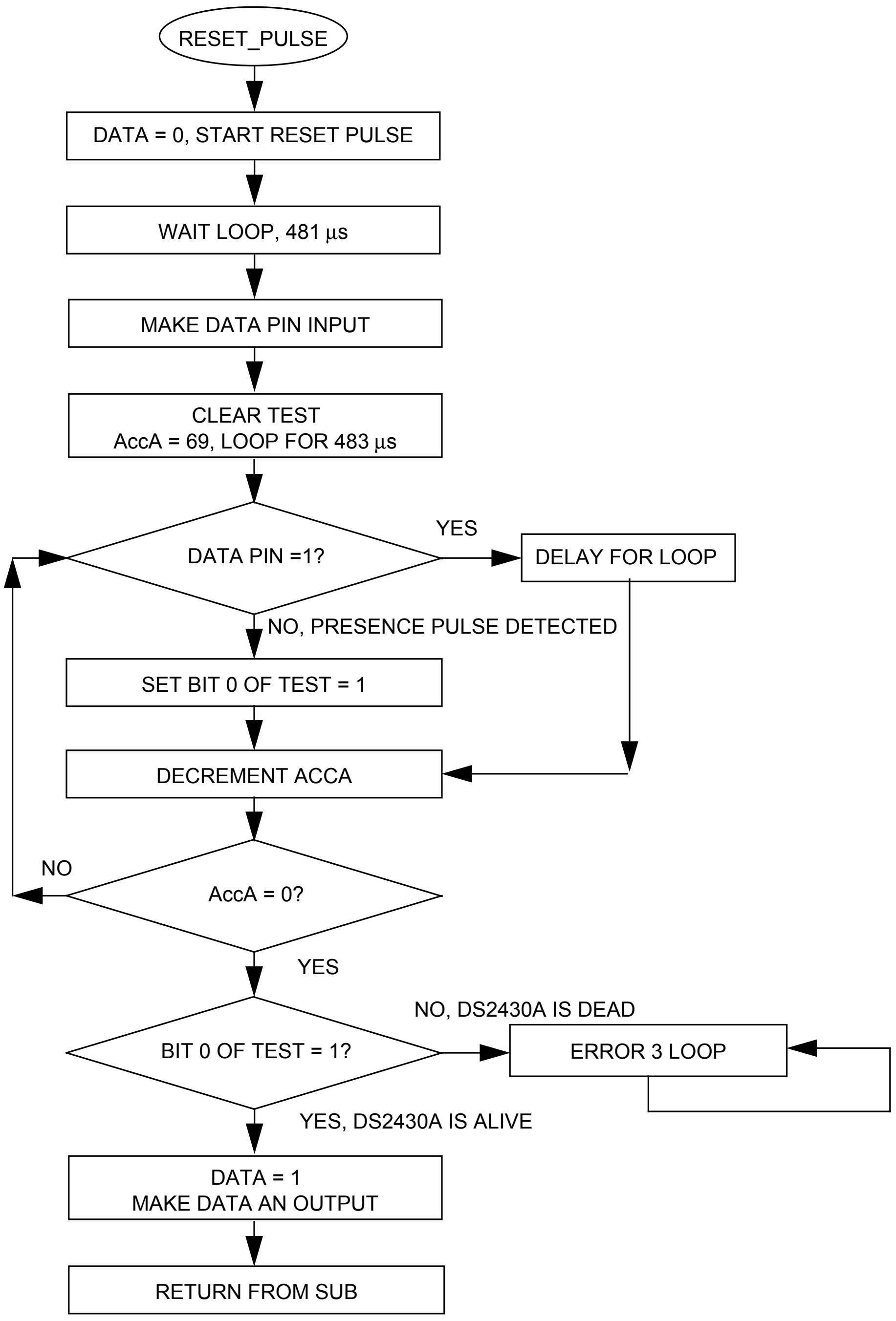


Figure 11. RESET_PULSE Subroutine Flowchart

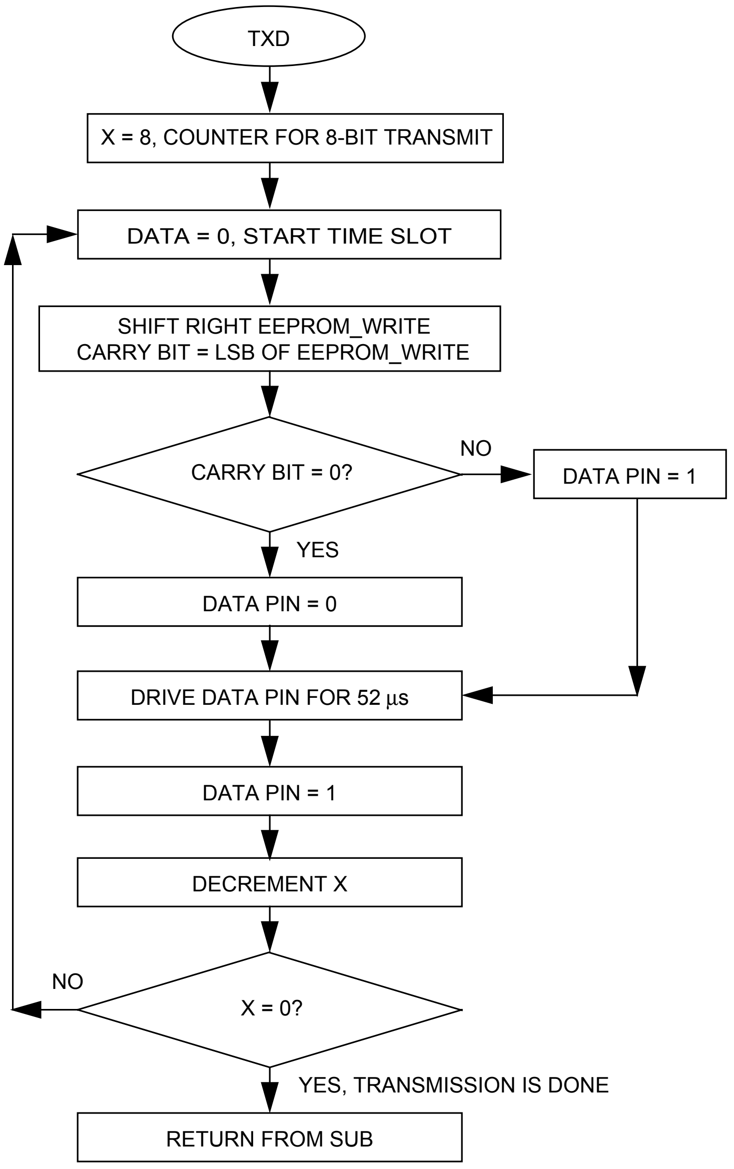


Figure 12. TXD Subroutine Flowchart

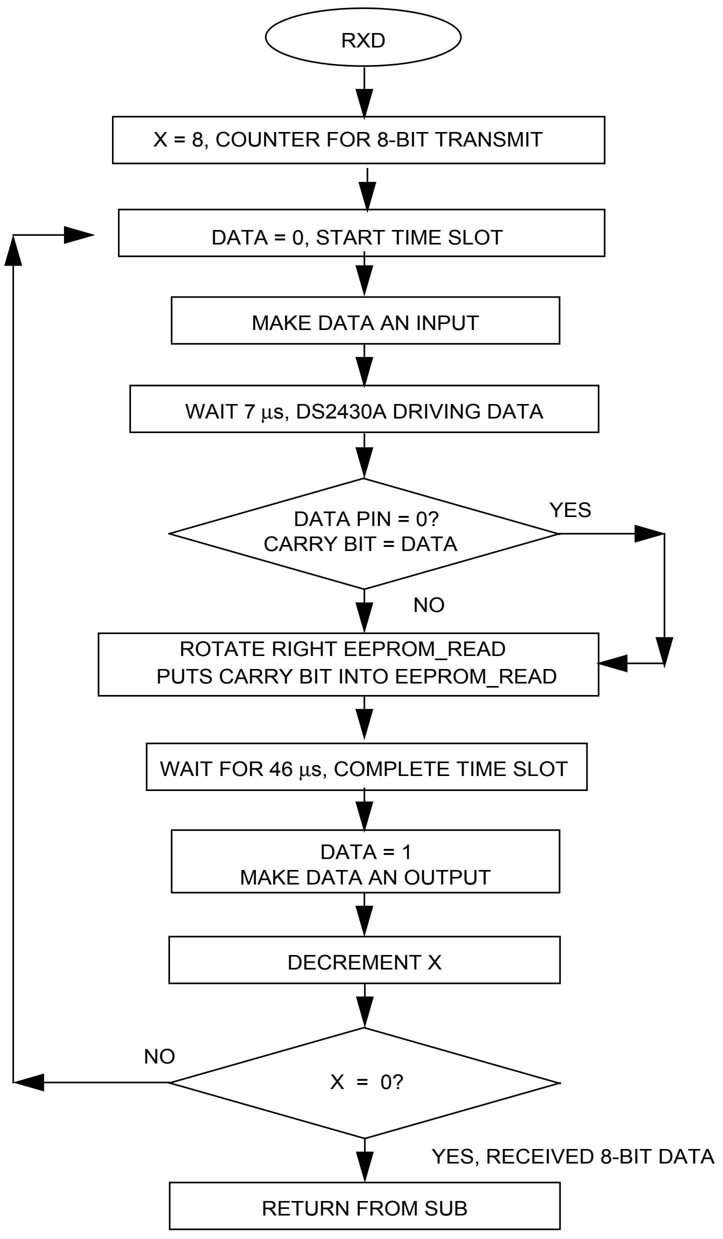


Figure 13. RXD Subroutine Flowchart

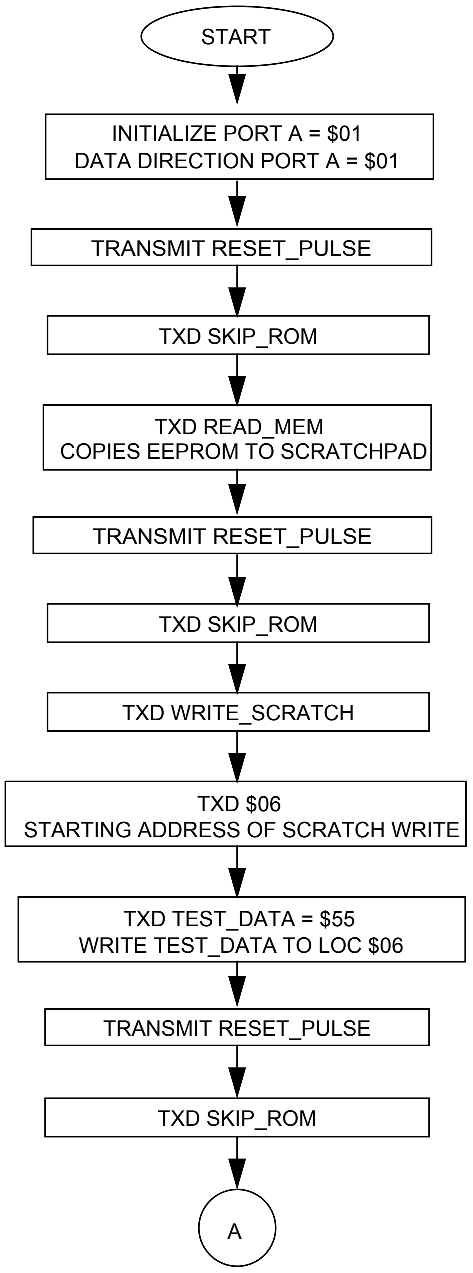


Figure 14. Flowchart for Main Test Routine (Sheet 1 of 3)

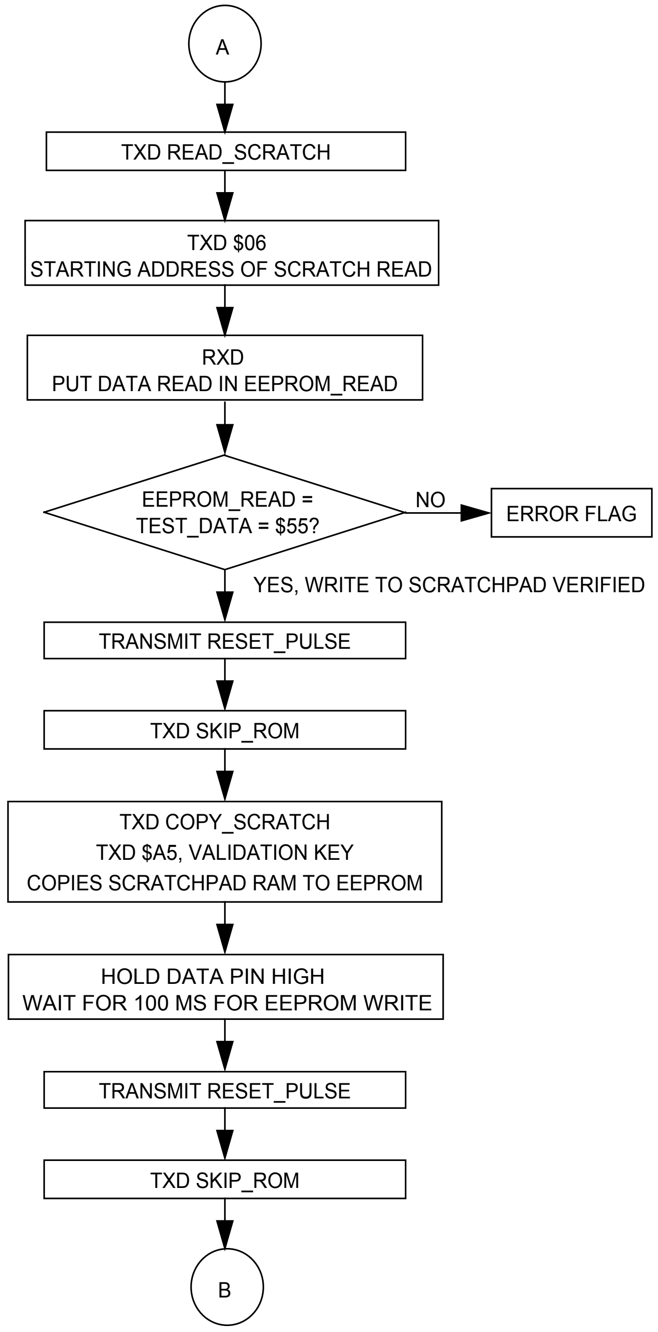


Figure 14. Flowchart for Main Test Routine (Sheet 2 of 3)

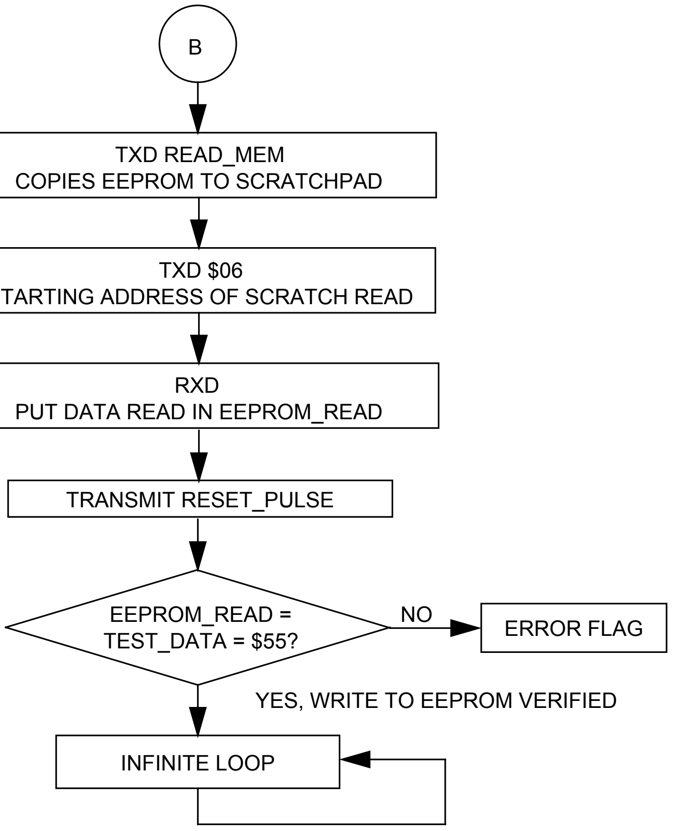

B
TXD READ_MEM
COPIES EEPROM TO SCRATCHPAD
TXD $06
STARTING ADDRESS OF SCRATCH READ
RXD
PUT DATA READ IN EEPROM_READ
TRANSMIT RESET_PULSE
EEPROM_READ =
TEST_DATA = $55?
NO
ERROR FLAG
YES, WRITE TO EEPROM VERIFIED
INFINITE LOOP

Figure 14. Flowchart for Main Test Routine (Sheet 3 of 3)

Code Listing

```
*************************************************************************************
*
* File name: DS2430A.ASM
* Example Code for the MC68HC705C8A Interface to the
*     Dallas DS2430A 1-Wire 256-Bit EEPROM
* Ver: 1.0
* Date: June 7, 1998
* Author: Mark Glenewinkel
*         Freescale Field Applications
*         Consumer Systems Group
* Assembler: P&E IDE ver 1.02
*
* For code explanation and flow charts,
* please consult Freescale Application Note
*    "Interfacing the MC68HC705C8A to the DS2430A 1-Wire 256-bit EEPROM"
*    Literature # AN1755/D
*
* NOTE: All timing functions are based on a 2-MHz internal bus clock
*
*************************************************************************************

*** SYSTEM DEFINITIONS AND EQUATES **********************************************
*** Internal Register Definitions
PORTA           EQU     $00             ;PortA
DDRA            EQU     $04             ;data direction for PortA

*** Application Specific Definitions
DATA            EQU     0T              ;PORTA, bit 0, data signal
DATA_DIR        EQU     0T              ;PortA Data Dir for DATA signal

*** ROM Function Commands
READ_ROM        EQU     $33             ;read ROM
MATCH_ROM       EQU     $55             ;match ROM
SKIP_ROM        EQU     $CC             ;skip ROM
SEARCH_ROM      EQU     $F0             ;search ROM

*** Memory Function Commands
WRITE_SCRATCH   EQU     $0F             ;write to scratchpad memory
READ_SCRATCH    EQU     $AA             ;read from scratchpad memory
COPY_SCRATCH    EQU     $55             ;copy scratchpad to EEPROM array
READ_MEM        EQU     $F0             ;read from EEPROM memory
WRITE_APP       EQU     $99             ;write to the application reg
READ_STATUS     EQU     $66             ;read from the status reg
READ_APP        EQU     $C3             ;read from application reg
COPY_LOCK       EQU     $5A             ;copy & lock application reg

*** Memory Definitions
EPROM           EQU     $160            ;start of EPROM mem
RAM             EQU     $50             ;start of RAM mem
RESET           EQU     $1FFE           ;vector for reset
```

```
*** RAM VARIABLES *****************************************
              ORG      RAM
EEPROM_WRITE  DB       $00              ;storage for EEPROM write
EEPROM_READ   DB       $00              ;storage for EEPROM read
TEST          DB       $00              ;test result for presence
TEST_DATA     DB       $55              ;test data for EEPROM testing


*** MAIN ROUTINE ******************************************
              ORG      EPROM            ;start at begining of EPROM
*** Intialize Ports
START         lda      #$01             ;init PORTA
              sta      PORTA
              lda      #$01             ;config outputs on PORTA
              sta      DDRA

*** Issue "SKIP ROM" command
              jsr      RESET_PULSE      ;send reset to DS2430A
              lda      #SKIP_ROM
              sta      EEPROM_WRITE
              jsr      TXD              ;send Skip ROM cmd

*** "Read Memory" Command, copies EEPROM to scratchpad memory
              lda      #READ_MEM
              sta      EEPROM_WRITE
              jsr      TXD              ;send Read Memory cmd

*** Issue "SKIP ROM" command
              jsr      RESET_PULSE      ;send reset to DS2430A
              lda      #SKIP_ROM
              sta      EEPROM_WRITE
              jsr      TXD              ;send Skip ROM cmd

*** Issue "WRITE SCRATCHPAD" command, start at location $06
*** Write 1 byte of info - TEST_DATA
              lda      #WRITE_SCRATCH
              sta      EEPROM_WRITE
              jsr      TXD              ;send Write Sratch memory cmd
              lda      #$06
              sta      EEPROM_WRITE
              jsr      TXD              ;send address of $06
              lda      TEST_DATA
              sta      EEPROM_WRITE
              jsr      TXD              ;send TEST_DATA

*** Issue "SKIP ROM" command
              jsr      RESET_PULSE      ;send reset to DS2430A
              lda      #SKIP_ROM
              sta      EEPROM_WRITE
              jsr      TXD              ;send Skip ROM cmd
```

```
*** "Read Scratchpad" Command, start address at $06, read 1 byte
*** Is it equal to TEST_DATA?
               lda      #READ_SCRATCH
               sta      EEPROM_WRITE
               jsr      TXD                ;send Read Scratch memory cmd
               lda      #$06
               sta      EEPROM_WRITE
               jsr      TXD                ;send address of $06
               jsr      RXD                ;receive data from addr $06

               lda      EEPROM_READ
               cmp      TEST_DATA          ;does data read = TEST_DATA?
               beq      PASS
ERROR1         bra      ERROR1             ;bad read, stay here
PASS

*** Issue "SKIP ROM" command
               jsr      RESET_PULSE        ;send reset to DS2430A
               lda      #SKIP_ROM
               sta      EEPROM_WRITE
               jsr      TXD                ;send Skip ROM cmd

*** Issue "Copy Scratchpad" command with validation key $A5
               lda      #COPY_SCRATCH
               sta      EEPROM_WRITE
               jsr      TXD                ;send Copy Scratch memory cmd
               lda      #$A5
               sta      EEPROM_WRITE
               jsr      TXD                ;send validation key of $A5

*** Hold DATA=1 and wait for 100ms, EEPROM being programmed
               bset     DATA,PORTA         ;hold DATA = 1
               jsr      NV_WAIT            ;wait routine for 100ms

*** Issue "SKIP ROM" command
               jsr      RESET_PULSE        ;send reset to DS2430A
               lda      #SKIP_ROM
               sta      EEPROM_WRITE
               jsr      TXD                ;send Skip ROM cmd

*** Issue "READ MEMORY" command, copies EEPROM to scratchpad memory
*** Start address at $06
               lda      #READ_MEM
               sta      EEPROM_WRITE
               jsr      TXD                ;send Read Memory cmd
               lda      #$06
               sta      EEPROM_WRITE
               jsr      TXD                ;send addr of $06

*** Read 1 byte, verify against TEST_DATA
               jsr      RXD                ;receive data from addr $06
               jsr      RESET_PULSE        ;send reset to DS2430A

               lda      EEPROM_READ
               cmp      TEST_DATA          ;does data read = TEST_DATA?
               beq      DUMMY
ERROR2         bra      ERROR2             ;bad read, stay here

DUMMY          bra      DUMMY              ;test sequence is over
```

```
*** SUBROUTINES ********************************************
*** Routine creates a reset pulse and then checks for the
*** presence pulse from the DS2430A
*** If no presence pulse, goto error loop

* Create a greater than 480usec reset pulse
RESET_PULSE    bclr     DATA,PORTA
               lda      #160T            ;2 wait for 481usec
J1             deca                      ;3
               bne      J1               ;3
               bclr     DATA,DDRA        ;DATA is now an input

* Wait for greater than 480usec, look for presence pulse,
* TEST will be equal to $01 if presence is detected
               clr      TEST
               lda      #69T             ;wait for 483usec
J2             brset    DATA,PORTA,J3    ;5 DATA=1?
               bset     0,TEST           ;5 DATA=0, presence detected
               bra      J4               ;3    set TEST bit 0 = 1

J3             brn      J3               ;3 branch has same time
               brn      J3               ;3
               nop                       ;2

J4             deca                      ;3 decrement Acca
               bne      J2               ;3 done?

* Check TEST, if TEST=$01, then ok
* if TEST=$00, then goto error routine
               brset    0,TEST,J5        ;TEST bit 0 = 1?
ERROR3         bra      ERROR3           ;presence pulse not detected, error
J5             bset     DATA,PORTA       ;TEST passed, DATA=1
               bset     DATA,DDRA        ;DATA is output now
               rts

*** Routine takes contents of EEPROM_WRITE and transmits it serially to
*** it serially to the DS2430A, LSB first
TXD            ldx      #8T              ;set counter

* Drive DATA=0 for 9usec
WRITE          bclr     DATA,PORTA       ;5 DATA=0, start time slot

               asr      EEPROM_WRIT      ;5 Carry bit = LSB
               bcc      J6               ;3
               bset     DATA,PORTA       ;5 DQ=1
               bra      J7               ;3 branch to clock_it
J6             bclr     DATA,PORTA       ;5DQ=0
               brn      J6               ;3evens it out

* At this point, 10.5usec has expired and either a 1 or 0
* is being transmitted on the DATA pin
* We must now wait for at least 49.5usec, routine below is 52usec
J7             lda      #17T             ;2
J8             deca                      ;3
               bne      J8               ;3
```

```
* Make sure DATA=1, then wait for more than 1usec for recovery time
               bset     DATA,PORTA       ;5 DATA=1
               decx                      ;3
               bne      WRITE            ;3 all 8 bits transmitted?
               rts                       ;return from sub


*** Routine clocks the DS2430A to read data from DATA, LSB first
*** 8 bit contents are put in EEPROM_READ
RXD            ldx      #8T              ;set counter

* Drive DATA=0 for 1usec,
READ           bclr     DATA,PORTA       ;5 DATA=0, start time slot
               nop                       ;2
               bclr     DATA,DDRA        ;5 make DATA an input

* Wait for 7 usec, then sample DATA
               lda      #2T              ;2
J9             deca                      ;3
               bne      J9               ;3

* Now read data and wait for 50µsec
               brclr    DATA,PORTA,J10   ;5 carry bit = DATA
J10            ror      EEPROM_READ      ;5 carry bit into EEPROM_READ

* At this point, 15.5µsec has expired since time slot started
* We must now wait for at least 44.5µsec, routine below is 46µsec
               lda      #15T             ;2
J11            deca                      ;3
               bne      J11              ;3

* Make sure DATA=1, then wait for more than 1usec for recovery time
               bset     DATA,PORTA       ;5 DATA=1
               bset     DATA,DDRA        ;5 make DATA an output
               decx                      ;3
               bne      READ             ;3 all 8 bits received?
               rts                       ;return from sub


*** Routine creates a ~100ms routine with a 2MHz MCU internal bus for
*** NV memory to be set correctly
NV_WAIT        ldx      #132T
J12            lda      #255T
J13            deca                      ;3
               bne      J13              ;3
               decx
               bne      J12
               rts


*** VECTOR TABLE ******************************************
               ORG      RESET
               DW       START
```

Application Note

References

MC68HC705C8A Technical Data, MC68HC705C8A/D, Freescale

M68HC05 Applications Guide, M68HC05AG/AD, Freescale

DS2430A Datasheet, Dallas Semiconductor, 1997.

freescale™ semiconductor

AN1755/D