Order this document
by AN1758/D

Motorola Semiconductor Application Note

AN1758

Add Addressable Switches to the HC05

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

Introduction

This application note describes the interface between an HC05 microcontroller (MCU) and the DS2405 addressable switch from Dallas Semiconductor Corporation. The address bus structure uses a 1-Wire™ interface that reduces the overhead of control, data, address, and power pins. One pin on the DS2405 combines all of these functions.

The DS2405 provides a means for assigning an identification to a particular node or location with the additional control capability of an open-drain N-channel MOSFET. This MOSFET switch can be remotely addressed and turned on or off via the 1-Wire bus. This structure allows the user to add any number of switches to the bus by simply adding a twisted pair cable containing the 1-Wire bus signal and ground.

The 1-Wire interface also can be used with the DS2407. This device is similar to the DS2405. In addition to the DS2405 features, it has up to two switchable MOSFETS and 1024 bits of user-programmable One-time-programmable EPROM. This allows the user to add information about the system the switch is hooked to. This information can be easily read over the 1-Wire bus to identify the switch and its system properties.

™ 1-Wire is a trademark of Dallas Semiconductor Corporation.

Circuitry and example code are provided in **Code Listing** to demonstrate the interface between the DS2405 and the HC05. Although no example code is given for the DS2407, the serial drivers for the 1-Wire bus make it easy for the user to add application-specific functions to utilize the additional features of the DS2407.

DS2405 and DS2407 Features

DS2405 Features

The DS2405 provides these features:

- Open drain PIO pin controlled by matching 64-bit, laser-engraved registration number associated with each device
- Logic level of open drain output can be determined over the 1-Wire bus for closed-loop control
- PIO pin sink capability is greater than 4 mA at 0.4 V
- Multiple DS2405s can be identified on the bus and switched on or off independent of other devices on the bus
- Unique, factory-lasered and tested 64-bit registration number assures that no two parts 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
- Zero standby power required
- Low-cost TO-92 or 6-pin TSOC package
- 1-Wire bus communicates over a wide voltage range of 2.8 V to 6.0 V from –40 °C to +85 °C

DS2407 Features

In addition to those features on the DS2405, the DS2407 provides these features:

- Dual switches available with the TSOC package

- Better current sink capability:
  - Channel A has 50 mA at 0.4 V
  - Channel B has 8 mA at 0.4 V
- 1024 bits of user one-time programmable (OTP) EPROM
- Seven bytes of user-programmable status memory to control the device

Description

The DS2405 has a 64-bit lasered ROM that is unique to each device. The first 8 bits signify the 1-Wire family member being addressed. The next 48 bits are unique to each DS2405 device allowing over 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 DS2405s to be on the bus. The 1-Wire bus network controller circuitry has a search algorithm embedded to determine the identity of each device on the bus. Once the ROM code of a device is matched, the PIO pin is toggled by an open-drain N-channel MOSFET. The logic level of the PIO pin of each device can also be sensed and reported back to the host of the bus.

DS2405 Hardware Interface

Pinout and Pin Descriptions

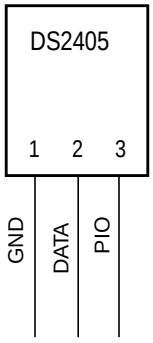

Figure 1. DS2405 TO-92 Pinout

The bidirectional DATA pin is the only interface pin to a microcontroller. Parasitic power is derived from the required pullup resistor on the DATA pin. No other power input is needed for the DS2405. All data transceived between the master and the DS2405 is read and written least significant bit (LSB) first.

Block Diagram

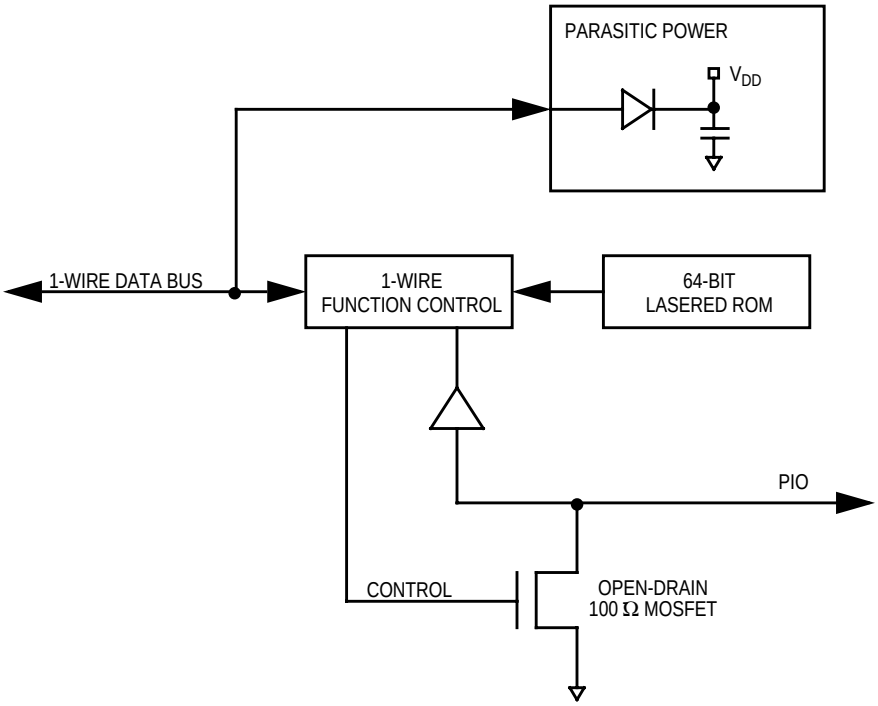

Figure 2. DS2405 Block Diagram

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 DS2405 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 DS2405.

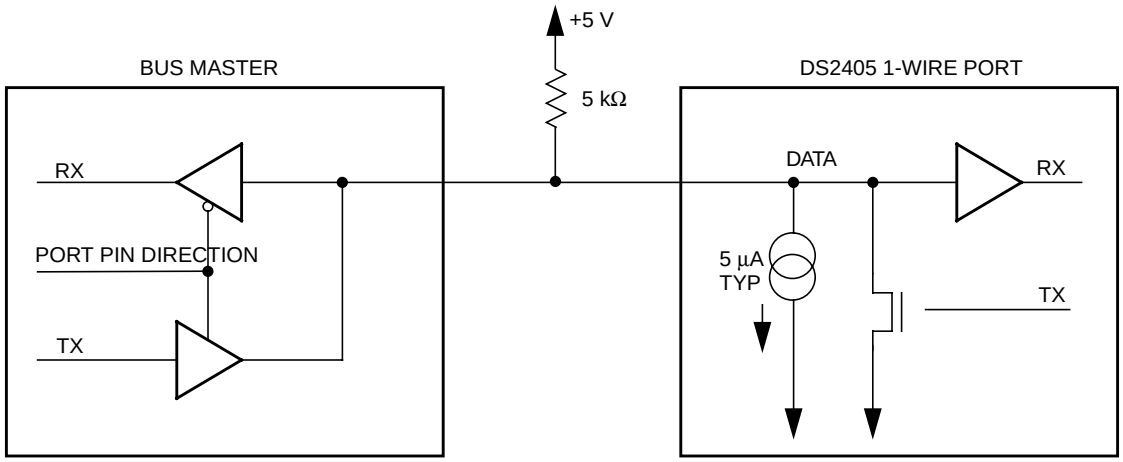

Figure 3. 1-Wire Bus Interface

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 DS2405 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 DS2405 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.

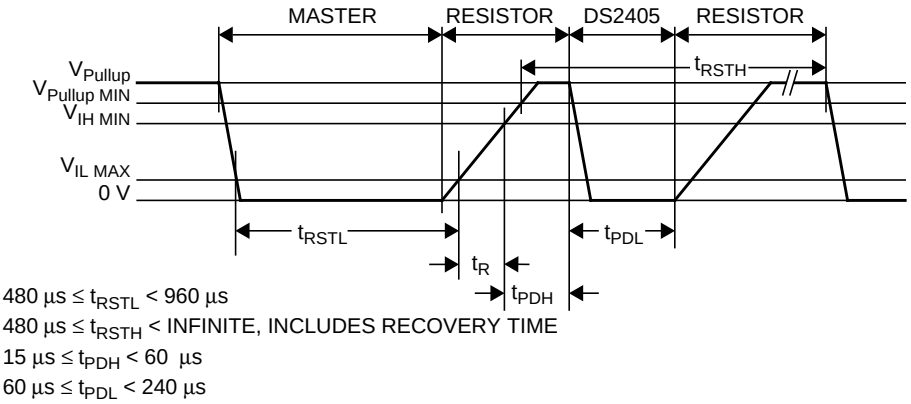

Figure 4. Reset and Presence Pulse Timing

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 DS2405 time to respond to the next bit being transmitted. The time slot and recovery time together add up to 61 µs which defines 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**.

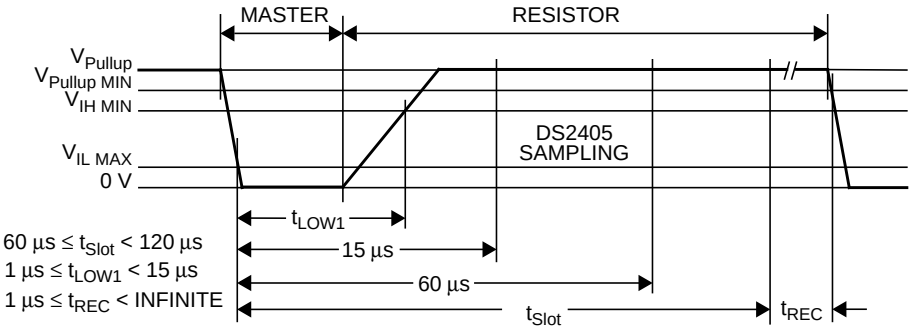

Figure 5. Write-One Time Slot

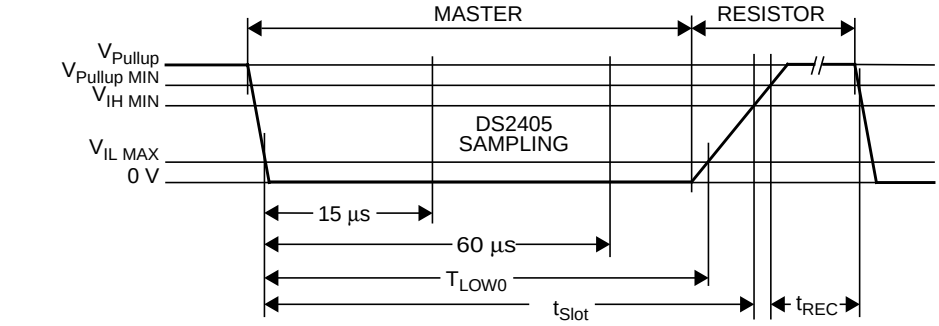

Figure 6. Write-Zero Time Slot

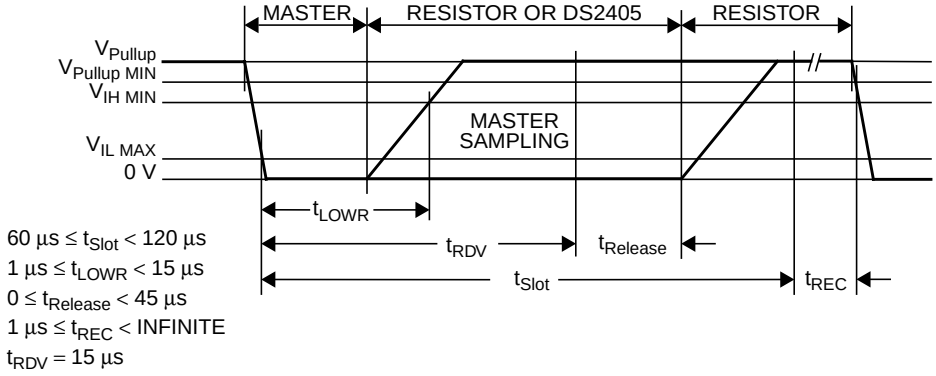

Figure 7. Read Data Time Slot

A step-by-step example of the protocol needed to match the DS2405's 64-bit ROM code is:

1. The master transmits a reset pulse.
2. The master waits for the presence pulse from the DS2405. Once detected, go to step 3.
3. The master sends out the match ROM command to the DS2405. The code for match ROM is $55 or %01010101 and is sent out LSB first.
    a. Write-one
    b. Write-zero
    c. Write-one
    d. Write-zero
    e. Write-one
    f. Write-zero
    g. Write-one
    h. Write-zero
4. After the match ROM function has been sent, the DS2405 will toggle its PIO pin.
5. The transaction is now complete. To issue another command, the DS2405 must be reset again.

DS2405 Software Interface

The transaction sequence to access the DS2405 over the 1-Wire bus is illustrated in **Figure 8**.

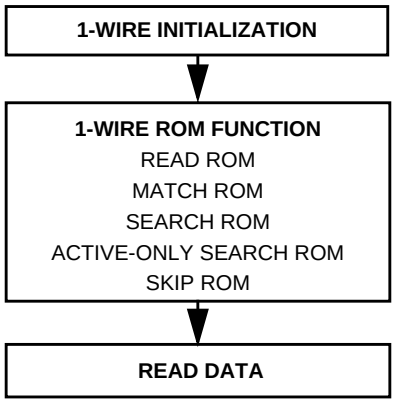

Figure 8. DS2405 Transaction Sequence

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 DS2405.

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 DS2405's unique 64-bit ROM code. This command can only be used if there is a single DS2405 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 DS2405. The last eight bits are a CRC (cyclic redundancy check) of the first 56 bits. Consult the DS2405 data sheet for more detail on the 64-bit ROM.

Match ROM — $55

If more than one DS2405 is on the bus, this command allows the bus master to address a specific DS2405. The DS2405 that exactly matches the 64-bit ROM sequence will toggle its PIO pin. If the open-drain

N-channel device was off, it will now be on. Likewise, if it was on, it will be now be off. All other devices will go inactive until a reset pulse is sent.

After the last bit of the ROM has been read and the PIO pin has been toggled, additional read data time slots will cause the DS2405 to transmit the logic state of the PIO pin. If the pulldown device is turned on and the PIO pin is a logic 0, the DS2405 will transmit a 0 in a read data time slot. If the pulldown device is turned off and the PIO pin is a logic 1, the DS2405 will transmit a 1 in a read data time slot. An external pullup resistor is required on the PIO pin to bring it high when the MOSFET has been turned off. Additional read data time slots will indicate the state of the PIO pin until a reset pulse is sent on the bus.

Search ROM — $F0

When a system has many devices on the bus, the master may not know the number of devices on the bus or their 64-bit ROM codes. This command allows the master to determine the ROM contents of all devices on the bus. The search ROM command uses a tedious process of elimination to determine the identity of the devices on the bus. A 3-step process is used on each bit of the 64-bit code.

The steps are:

1. The master reads a bit position of the 64-bit code.
2. The master reads the complement of the bit position.
3. With this information, the master writes a bit on the bus to match those devices that have the same bit value. This in turn deselects the other devices that do not have a matched bit.

This process is repeated until eventually all devices are eliminated except the device that matches exactly to the 64-bit code that was just transmitted. Once one device is found, the master may issue read data time slots to read the PIO pin. The process is started over to find the next device on the bus. The search ROM command deserves further treatment but is beyond the scope of this application note. Consult the DS2405 data sheet for a more detailed explanation on how to use this command.

Active-Only Search ROM — $EC

The active-only search ROM command is very similar to the search ROM command. The difference is that this command only searches for devices that have their MOSFET turned on and the PIO pin pulled low. Consult the DS2405 data sheet for a more detailed explanation on how to use this command.

Skip ROM — $CC

The protocol of the 1-Wire bus includes a command to bypass the 64-bit ROM code. This is useful if only one device is on the bus. Many 1-Wire bus devices have other functions that are different to each device. The skip ROM command shortens the access time needed to execute internal functions of a particular device. Since the DS2405 contains only the 64-bit ROM with no additional data fields or functions, the skip ROM command is not applicable to the DS2405.

Memory Map

The DS2405 has no memory other than the 64-bit ROM code. The ROM code is shown in **Figure 9**.

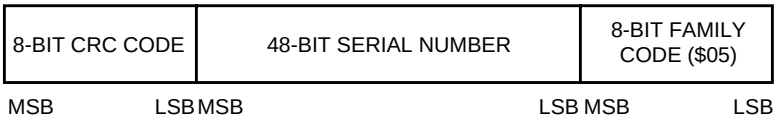

Figure 9. DS2405 Memory Map

MC68HC705J1A Hardware Interface

This application note uses the MC68HC705J1A (J1A) member of the HC05 Family to test the interface between the DS2405 and the HC05. With only 20 pins, the J1A is one of the smaller members of the HC05 Family.

The MC68HC705J1A has:

- 1240 bytes of erasable programmable read-only memory (EPROM)
- 64 bytes of RAM
- Timer
- 14 I/O pins

The schematic used for testing the J1A-to-DS2405 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 DS2405.

For further information on the HC705J1A, consult the *MC68HC705J1A Technical Data*.

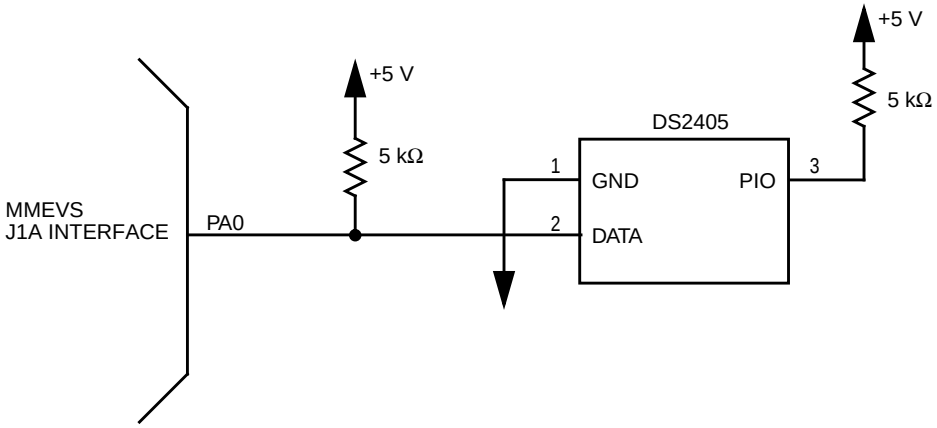

Figure 10. J1A to DS2405 Interface Test Circuit

MC68HC705J1A 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 — Sends out a reset pulse to the DS2405 and waits for a presence pulse. If no presence pulse is found, the routine goes into an error loop.
- TXD — Takes a byte of data and creates eight time slots of either write-one or write-zero, depending on the bit being transmitted.
- RXD — Transmits eight read data time slots. Each bit is read and shifted into a byte of RAM.

The flowcharts for the DS2405 serial I/O drivers are shown in **Figure 11**, **Figure 12**, and **Figure 13**.

The flowchart for the main test routine is in **Figure 14**. The step-by-step sequence of tests is:

1. Read ROM. This command asks the DS2405 to put its 64-bit ROM code on the bus.
2. Set up a loop to retrieve a total of eight bytes.
3. Store each byte into the RAM buffer DS2405_ROM. The location DS2405_ROM+7 will have the contents of the LSB of the ROM code. The location DS2405_ROM will have the contents of the MSB of the ROM code.
4. Match ROM. This command allows the user to select a DS2405 on the bus and toggle its PIO pin.
5. Set up a loop to transmit a total of eight bytes.
6. Transmit each byte from the RAM buffer DS2405_ROM starting LSB first.
7. Execute a RXD command. This will read the value of the PIO pin.

If the PIO pin is high (MOSFET turned off), the value put in DS2405_PIO will be $FF. If the PIO pin is low (MOSFET turned on), the value put in DS2405_PIO will be $00.

8. To toggle the PIO pin, restart the code at step 4.

This routine demonstrates the interface software needed to communicate with the DS2405. Although the J1A was used, any HC05 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 the following development tools.

- M68MMPFB0508 — Motorola MMEVS platform board
- M68EM05J1A — Motorola J1A emulation module
- Win IDE Version 1.02 — Editor, assembler, and debugger by P&E Microcomputer Systems

Flowcharts for the Serial Drivers

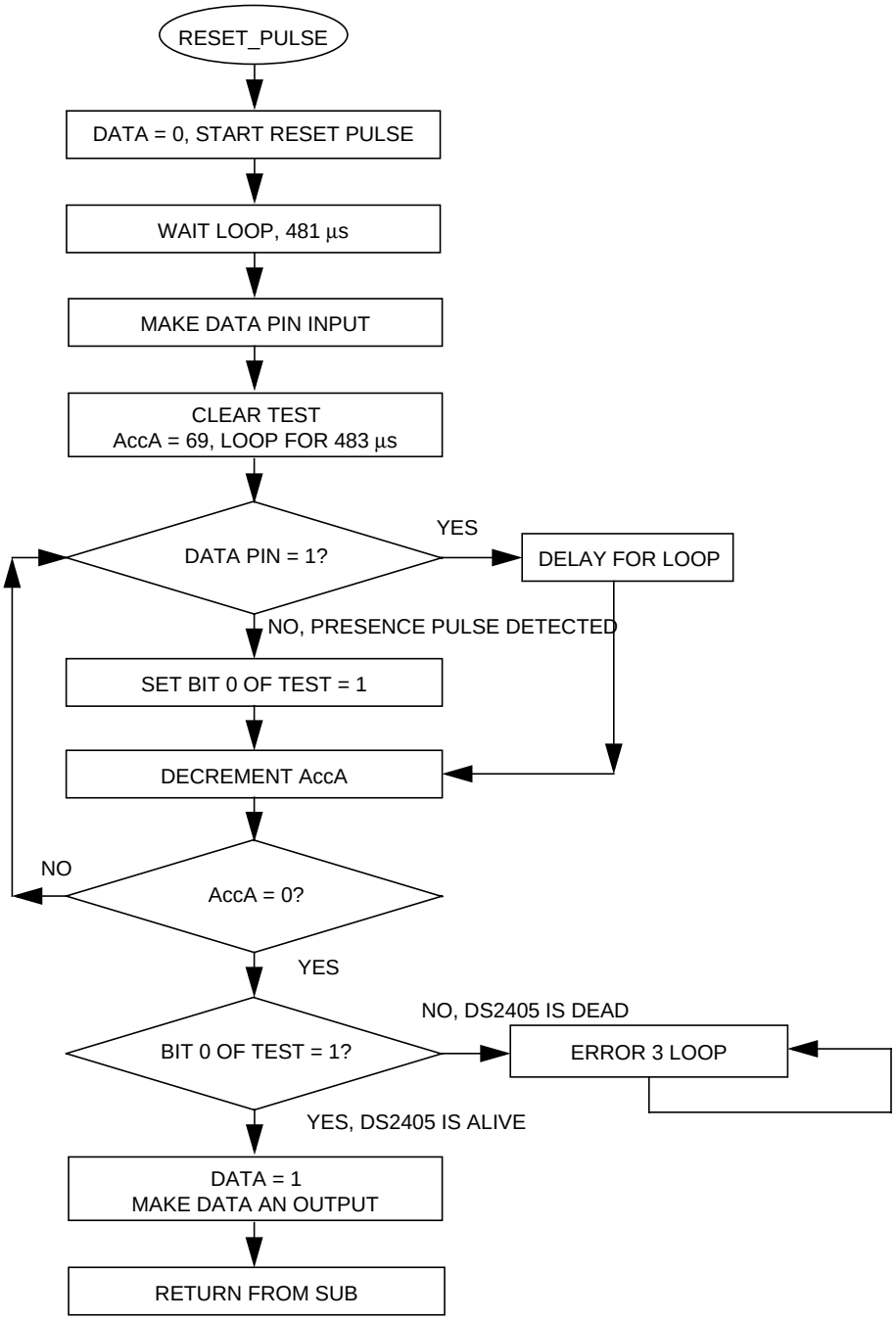

Figure 11. RESET_PULSE Subroutine Flowchart

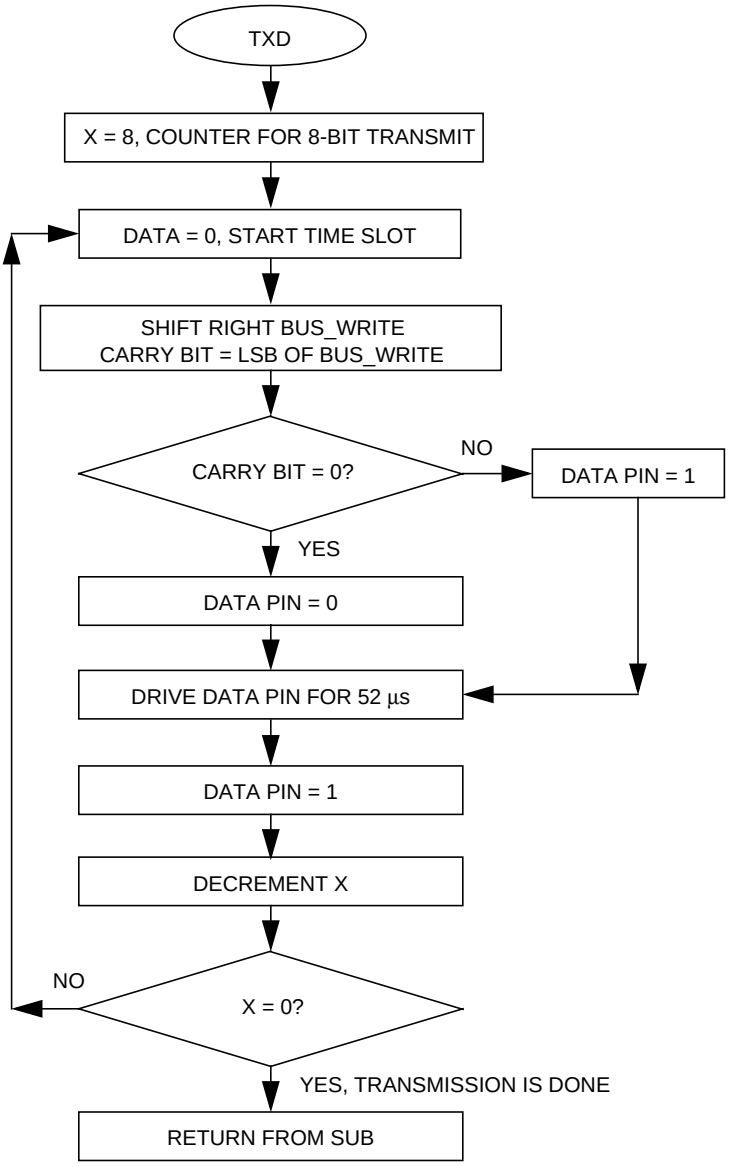

Figure 12. TXD Subroutine Flowchart

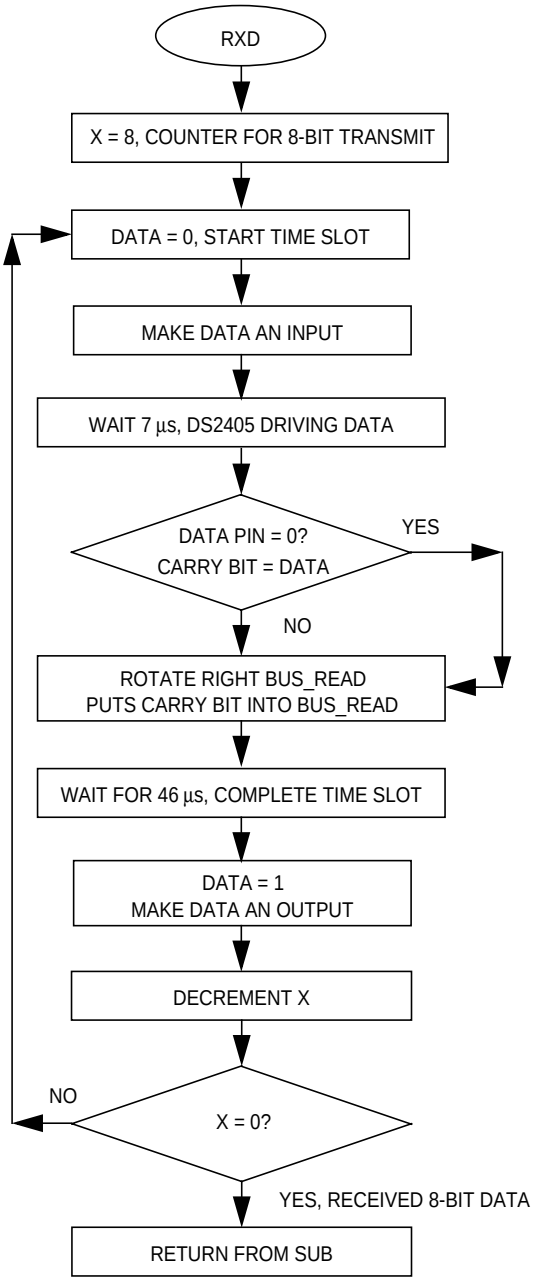

Figure 13. RXD Subroutine Flowchart

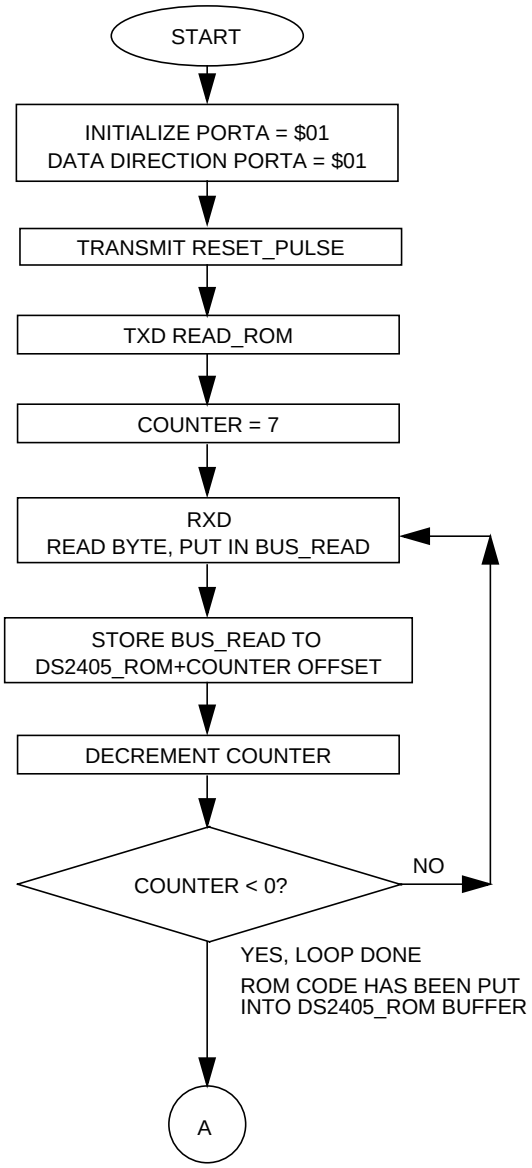

Figure 14. Flowchart for Main Test Routine (Sheet 1 of 2)

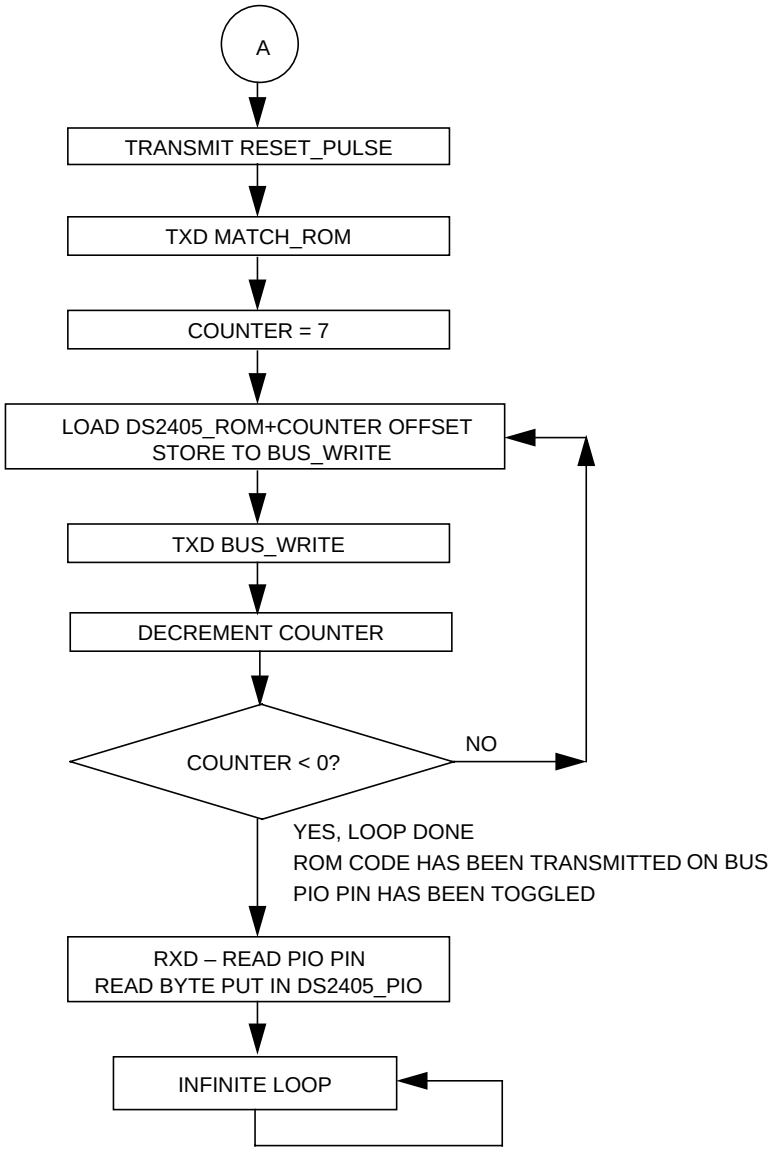

Figure 14. Flowchart for Main Test Routine (Sheet 2 of 2)

Code Listing

```
*****************************************************************************
*
* File name: DS2405.ASM
* Example Code for the MC68HC705J1A Interface to the
*     Dallas DS2405 Addressable Switch
* Ver: 1.0
* Date: June 17, 1998
* Author: Mark Glenewinkel
*         Motorola Field Applications
*         Consumer Systems Group
* Assembler: P&E IDE ver 1.02
*
* For code explanation and flow charts,
* please consult Motorola Application Note
*    "Add Addressable Switches to a Microcontroller System"
*    Literature # AN1758/D
*
* NOTE: All timing functions are based on a 2MHz 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
SEARCH_ROM     EQU         $F0               ;search ROM
ACT_SEARCH_ROM EQU         $EC               ;active-only search ROM
SKIP_ROM       EQU         $CC               ;skip ROM

*** Memory Definitions
EPROM          EQU         $300              ;start of EPROM mem
RAM            EQU         $C0               ;start of RAM mem
RESET          EQU         $07FE             ;vector for reset

*** RAM VARIABLES ***********************************************************
               ORG         RAM
BUS_WRITE      DB          $00               ;storage for byte write on bus
BUS_READ       DB          $00               ;storage for byte read on bus
TEST           DB          $00               ;test result for presence
```

```
DS2405_PIO     DB          $FF                ;PIO pin level
                                              ;power-on reset makes PIO high
DS2405_ROM     RMB         8                  ;storage for DS2405 64-bit ROM code
                                              ;DS2405_ROM   = MSB
                                              ;DS2405_ROM+7 = LSB
COUNTER        DB          $00                ;temporary counter

*** 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 "READ ROM" command
               jsr         RESET_PULSE        ;send reset on bus
               lda         #READ_ROM
               sta         BUS_WRITE
               jsr         TXD                ;send Read ROM cmd

               lda         #7T
               sta         COUNTER            ;set counter to rxd 8 bytes

ROM_Cycle1     jsr         RXD                ;receive data from bus - 1 byte
               lda         BUS_READ
               ldx         COUNTER
               sta         DS2405_ROM,x       ;store byte read into DS2405_ROM
               dec         COUNTER
               bpl         ROM_Cycle1         ;loop done?

*** Issue "MATCH ROM" command
               jsr         RESET_PULSE        ;send reset on bus
               lda         #MATCH_ROM
               sta         BUS_WRITE
               jsr         TXD                ;send Match ROM cmd

               lda         #7T
               sta         COUNTER            ;set counter to txd 8 bytes

ROM_Cycle2     ldx         COUNTER
               lda         DS2405_ROM,x       ;read byte from DS2405_ROM
               sta         BUS_WRITE
               sr          TXD                ;transmit data on bus -  1 byte
               dec         COUNTER
               bpl         ROM_Cycle2         ;loop done?
*** Read DS2405, read PIO pin
               jsr         RXD                ;receive data from bus - 1 byte
               lda         BUS_READ
               sta         DS2405_PIO         ;store result in DS2405_PIO
DUMMY          bra         DUMMY              ;test sequence is over
```

```
*** SUBROUTINES ***************************************************************
*** Routine creates a reset pulse and then checks for the
*** presence pulse from the DS2405
*** If no presence pulse, goto error loop
* Create a greater then 480 µsec reset pulse
RESET_PULSE    bclr        DATA,PORTA
               lda         #160T              ;2 wait for 481 µsec
J              deca                           ;3
               bne         J1                 ;3
               bclr        DATA,DDRA          ;DATA is now an input

* Wait for greater than 480 µsec, look for presence pulse,
* TEST will be equal to $01 if presence is detected
               clr         TEST
               lda         #69T               ;wait for 483 µsec
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 BUS_WRITE and transmits it serially to
*** it serially on the bus, LSB first
TXD            ldx         #8T                ;set counter

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

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

* Make sure DATA=1, then wait for more than 1 µsec 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 bus to read data from DATA, LSB first
*** 8 bit contents are put in BUS_READ
RXD             ldx         #8T                 ;set counter

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

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

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

* At this point, 15.5 µsec has expired since time slot started
* We must now wait for at least 44.5 µsec, routine below is 46 µsec
                lda         #15T                ;2
J11             deca                            ;3
                bne         J11                 ;3
* Make sure DATA=1, then wait for more than 1 µsec 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


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

References

MC68HC705J1A Technical Data, Motorola document order number C68HC705J1A/D, 1996.

M68HC05 Applications Guide, Motorola document order number M68HC05AG/AD, 1996.

DS2405 Datasheet, Dallas Semiconductor Corporation, 1997.

DS2407 Datasheet, Dallas Semiconductor Corporation, 1997.

HC05/08 Website

http://design-net.com/csic/welcome.htm

Development Tools Website

http://design-net.com/csic/devsys/sg173/sg173.htm

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Nippon Motorola Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan. 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/