Order this document
by AN1759/D

Motorola Semiconductor Application Note

AN1759

Add a Non-Volatile Clock to the MC68HC705J1A

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

Introduction

Many embedded systems require the measurement of time. This can be accomplished internally by some MCUs that have on-chip real-time clocks. Even so, for date, month, and leap year measurement, this task can take substantial amounts of bandwidth and code space.

The DS1307 64x8 serial real-time clock provides calendar and time keeping functions along with system-enhancing non-volatile RAM. With a 2-wire interface, timekeeping can be managed easily.

Some applications of using the DS1307 are:

- Logging of chronological events
- Tracking power down time of a system
- Providing alarm functions

MOTOROLA

The non-volatile RAM (random-access memory) also gives the user additional applications such as:

- Power down information storage for consumer electronics like TVs, VCRs, and hand-held portables
- Identification number storage for remote addressing or security
- Storage of telecommunication information like phone number recall and speed dialing

This application note describes the interface between the MC68HC705J1A (J1A) and the DS1307. Circuitry and example code are given to demonstrate the interface between the two parts.

Features

The DS1307 provides these features:

- Real-time clock counts seconds, minutes, hours, day of the week, date, month, and year.
- Leap year compensation valid up to 2100
- 56 bytes of non-volatile RAM for data storage
- 2-wire serial interface
- Programmable square wave output with frequencies of 1 Hz, 4.096 kHz, 8.192 kHz, and 32.768 kHz
- Automatic power switching to battery when main power fails
- In battery backup mode, less than 500 nA consumed at 25 °C
- 8-pin DIP or SOIC package
- Optional industrial temperature range of –40 °C to +85 °C

Description

The DS1307 is a low-power binary coded decimal (BCD) clock calendar that provides seconds, minutes, hours, day, date, month, and year. In addition, it has 56 bytes of non-volatile RAM. End-of-the-month adjustments are automatic for months with less than 31 days. The device also corrects for leap years. The clock can operate in either 12-hour or 24-hour mode. In 12-hour mode, an a.m./p.m indicator is used.

The DS1307 has built-in power management circuitry to detect power failures on the V_{DD} pin and when detected will switch power over to the battery back-up pin, V_{BAT}. Access to the device is terminated when V_{DD} falls below 1.25 x V_{BAT}. Further accesses to the device are not allowed. On power up, the device switches power from V_{BAT} to V_{DD} when the V_{DD} pin is 0.2 volts above V_{BAT}. Once V_{DD} is higher than 1.25 x V_{BAT}, normal operations can continue.

Address and data are communicated via the 2-wire bus. The DS1307 operates as a slave at all times and is accessed by first transmitting the DS1307's identification code on the bus.

DS1307 Hardware Interface

Pinout and Pin Descriptions

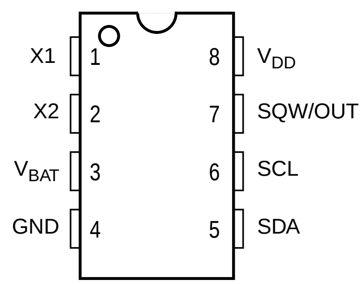

Figure 1. DS1307 Pinout

V_{DD} and GND

These pins serve as the main power source for the device. When +5 volts is applied to this pin, the device is fully accessible and data can be read or written. If the power on the V_{DD} pin falls below 1.25 x V_{DD}, the device switches its power supply to V_{BAT}. At this point, reading and writing to the device is prohibited. The timekeeping function and non-volatile RAM are unaffected.

V_{BAT}

This pin is the power input for any standard 3-volt lithium battery or other 3-volt source. For proper operation, this voltage must be held between 2.5 and 3.5 volts. A lithium battery with at least a 35-mA hours rating will back up the DS1307 for more than 10 years in the absence of power.

X1 and X2

These pins are used to connect a 32.768-kHz crystal to the device. No other capacitors or resistors are needed for this crystal circuit. The internal oscillator circuitry is designed for a crystal with load capacitance of 12.5 pF. For the test circuit described in this application note, an Epson C-001R crystal was used. The Digi-key part number for this device is SE3201-ND.

SQW/OUT

When enabled, this pin outputs one of four selectable frequencies:

- 1 Hz
- 4.096 kHz
- 8.192 kHz
- 32.768 kHz

The 1-Hz signal can be used to feed an external interrupt pin on an MCU. This allows the MCU to use minimal bandwidth when servicing the timekeeping function of a system.

When disabled, the pin acts as a normal output pin. It is controlled via the DS1307 control register.

SCL

The SCL pin is the clock input for the DS1307 2-wire serial interface.

SDA

The SDA pin is an I/O pin used to transmit and receive data off the 2-wire serial interface. SDA is an open-drain pin that requires an external pullup resistor.

Block Diagram

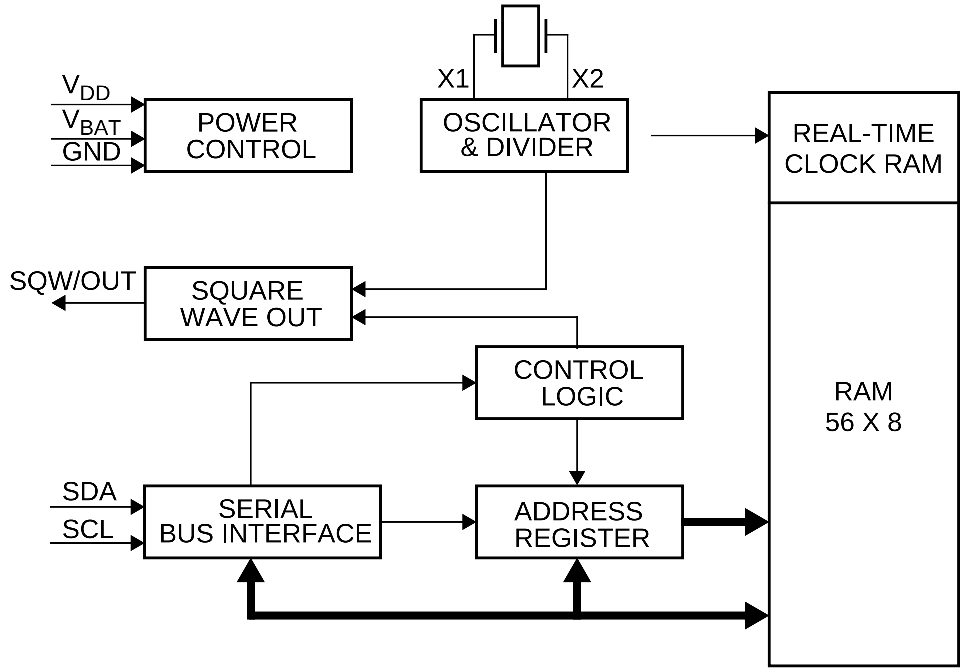

Figure 2. DS1307 Block Diagram

Serial Interface

The DS1307 supports the bidirectional, 2-wire protocol. The protocol has these characteristics:

- Any device sending out data is defined as a transmitter.
- Any device receiving data is defined as a receiver.
- The device controlling the transfer is called the master.
- The device being controlled is called the slave.
- The master initiates all transactions.
- The master always provides the clock for both transmit and receive operations.
- The DS1307 is always considered the slave.
- The clock signal is called SCL.
- The data signal is called SDA.
- All data is sent most significant bit (MSB) first.

Figure 3 shows the 2-wire bus interface between a master and slave.

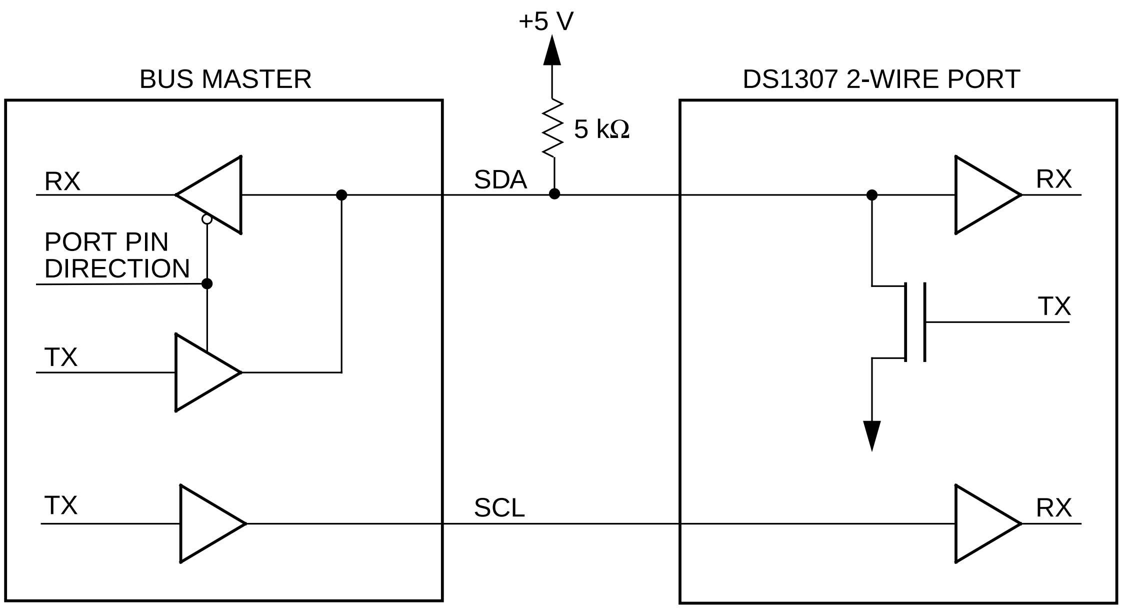

Figure 3. 2-Wire Serial Bus Interface

Bus Idle In idle mode, both the SDA and SCL are held high.

Start Transfer All transfers begin with the start transfer condition. This is done by bringing the SDA pin from HIGH to LOW while the SCL pin is HIGH. The DS1307 is monitoring the bus for this signal and will not start any transactions until this condition is met. See **Figure 4**.

Stop Transfer All transfers must be terminated with the stop transfer condition. This is done by bringing the SDA pin from LOW to HIGH while the SCL pin is HIGH. A stop transfer can be used only after the transmitting device releases the bus. See **Figure 4**.

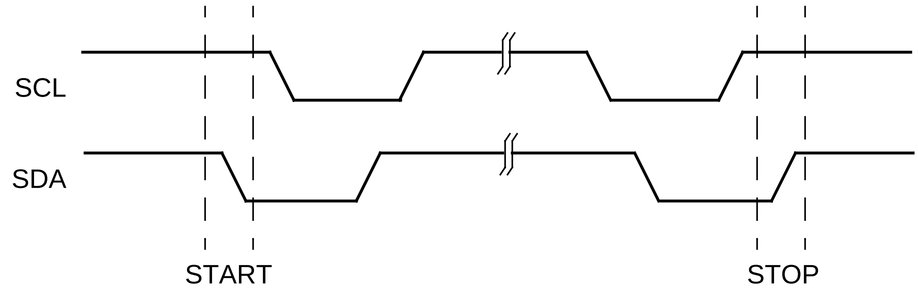

Figure 4. Start and Stop Transfer Timing

Data Transfer

Data is transmitted on the rising edge of SCL. Data can only be changed while SCL is LOW. The receiving device samples the bus after SCL goes HIGH. There is one clock pulse per bit of data transmitted. See **Figure 5**.

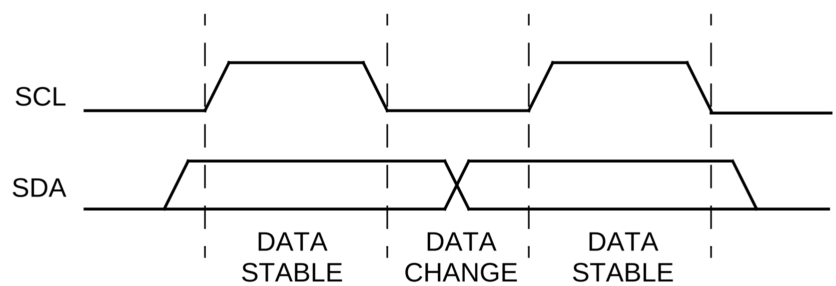

Figure 5. Data Transfer Timing

Acknowledge Transfer

The acknowledge transfer is a type of handshaking convention used to signify that a successful transfer of data has taken place. After the transmitting device sends out the eighth bit of a byte of data, it releases the bus. The master sends out a ninth clock signal and the receiver acknowledges the transfer by pulling SDA LOW. Once the transmitter reads the LOW condition of SDA, it proceeds by taking over the bus and sending out the next byte of data.

If the DS1307 is transmitting data and the master wants to end further transmissions, the master sends a NO ACK signal (HIGH) back to the DS1307. This tells the DS1307 that no more transfers are needed and the stop transfer condition will be initiated soon. See **Figure 6** for these different timing patterns.

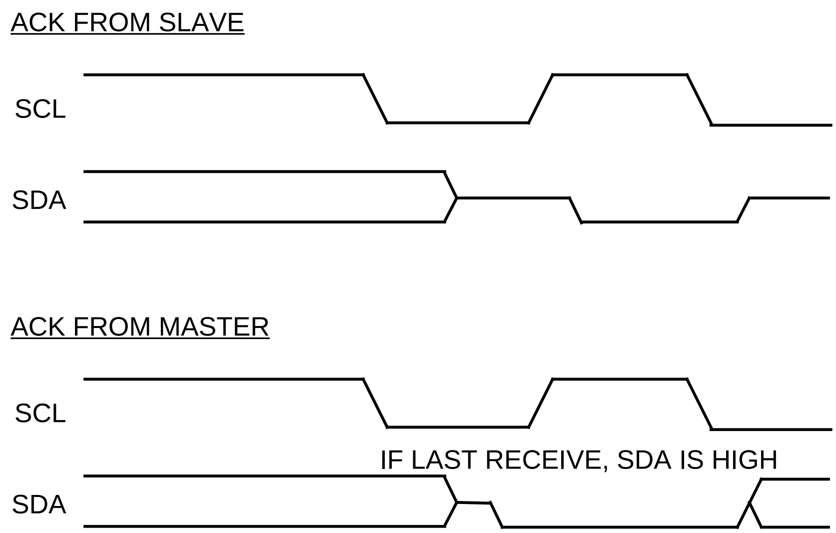

Figure 6. Acknowledge Timing

2-Wire Protocol Example

An example of the protocol needed to write $10 to address $07 of the DS1307 is:

1. The master transmits a start transfer.
2. The master transmits the DS1307 7-bit identification code, %1101000.
3. Since this is a data write transfer, the master then transmits a 0.
4. Since a byte has just been transmitted, the receiver (DS1307) will now send out a LOW to acknowledge the transfer.
5. The master reads the SDA pin for a LOW.
6. The master sends out the address of $07 to the DS1307 and receives back an acknowledge.
7. The master sends out the data, $10, to the DS1307 and receives back an acknowledge. The DS1307 writes $10 to address $07.
8. Finally, a stop transfer is sent to the DS1307 to complete the transaction.

DS1307 Software Interface

Memory Map

The DS1307's memory map is shown in **Figure 7**. The real-time clock registers are located in address locations $00 to $07. The 56 bytes of non-volatile RAM are located in address locations $08 to $3F. During multibyte addresses, the address pointer wraps around to $00 after it reaches $3F.

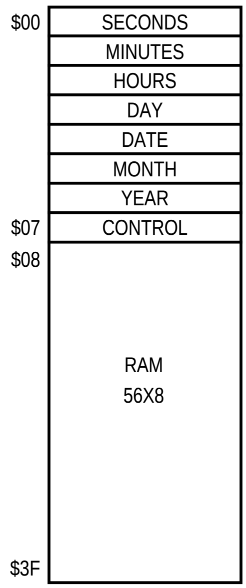

Figure 7. DS1307 Memory Map

Register Map

The real-time clock registers are shown in detail in **Figure 8**. The time and calendar are set by writing to the appropriate registers. The information is in binary coded decimal (BCD) format.

To enable the processor, write a 0 to the CLOCK HALT bit in register $00. The DS1307 is shipped with this bit set to 1.

Either 12-hour or 24-hour clock format can be used. If bit 6 of register $02 is a 0, the device is in 24-hour mode. Likewise, when bit 6 is a 1, the device is in 12-hour mode. Bit 5 of address $02 is used for the second

10 hours when in 24-hour mode. When using 12-hour mode, bit 5 is a 1 for p.m. and a 0 for a.m.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$00	CLOCK HALT	10 SECONDS			SECONDS			
$01	X	10 MINUTES			MINUTES			
$02	X	12 / 24	A/P / 10 HR	10 HR	HOURS			
$03	X	X	X	X	X	DAY		
$04	X	X	10 DATE		DATE			
$05	X	X	10 MONTH		MONTH			
$06	10 YEAR				YEAR			
$07	OUT	X	X	SQWE	X	X	RS1	RS0

Figure 8. DS1307 Register Map

Control Register

The control register is used to control the SQW/OUT pin.

OUT — Controls the output level of the SQW/OUT pin when SQWE = 0.
1 = SQW/OUT pin HIGH
0 = SQW/OUT pin LOW

SQWE — Enables the oscillator square wave on the SQW/OUT pin
1 = Square wave enabled
0 = Square wave disabled

RS — Square wave output frequency

- RS1 = 0 and RS0 = 0 —> 1 Hz
- RS1 = 0 and RS1 = 1 —> 4.096 kHz
- RS1 = 1 and RS1 = 0 —> 8.192 kHz
- RS1 = 1 and RS1 = 1 —> 32.768 kHz

Data Write Sequence

The first byte transmitted in a write to the DS1307 is its 7-bit identification code followed by the $R/\overline{W}$ bit. For writes, this bit will be 0. The next byte transmitted is the DS1307 address pointer. After this, bytes of data to be written to the DS1307 RAM are transmitted. After each byte of data is written, the address pointer is incremented. See **Figure 9**.

	DS1307 ADDRESS	$R/\overline{W}$		ADDRESS POINTER		DATA (N)		DATA (N+?)		
START	1101000	0	ACK	XXXX,XXXX	ACK	XXXX,XXXX	ACK	XXXX,XXXX	ACK	STOP

Figure 9. Data Write Sequence

Data Read Sequence

The first byte transmitted in a read from the DS1307 is its 7-bit identification code followed by the $R/\overline{W}$ bit. For reads, this bit will be 1. Then the DS1307 will begin transmitting data back to the master. As long as the DS1307 receives clocks and acknowledgments, it keeps transmitting data. The starting address is the previous address pointer from the last write transaction. If needed, a write sequence with only an address can be used to initialize the address pointer for reads.

NOTE: *Remember that for the last byte read, the master sends back a No ACK to the DS1307.*

	DS1307 ADDRESS	$R/\overline{W}$		DATA (N)		DATA (N+?)		
START	1101000	1	ACK	XXXX,XXXX	ACK	XXXX,XXXX	$\overline{ACK}$	STOP

Figure 10. Data Read Sequence

MC68HC705J1A Hardware Interface

With only 20 pins, the J1A is one of the smallest members of the HC05 Family. It has a total of 1240 bytes of erasable programmable read-only memory (EPROM) and includes 14 I/O (input/output) pins. The schematic used for testing the J1A to DS1307 interface on the MMEVS development system is shown in **Figure 11**. The pins used to drive the DS1307 on the J1A are listed here also.

- Port A, bit 0 — This I/O pin (SCL) is configured as an output to drive the serial clock pin, SCL, of the DS1307.
- Port A, bit 1 — This I/O pin (SDA) is used to transmit and receive data on the SDA pin of the DS1307.

For further information on the HC705J1A, consult the *MC68HC705J1A Technical Databook* (MC68HC705J1A/D).

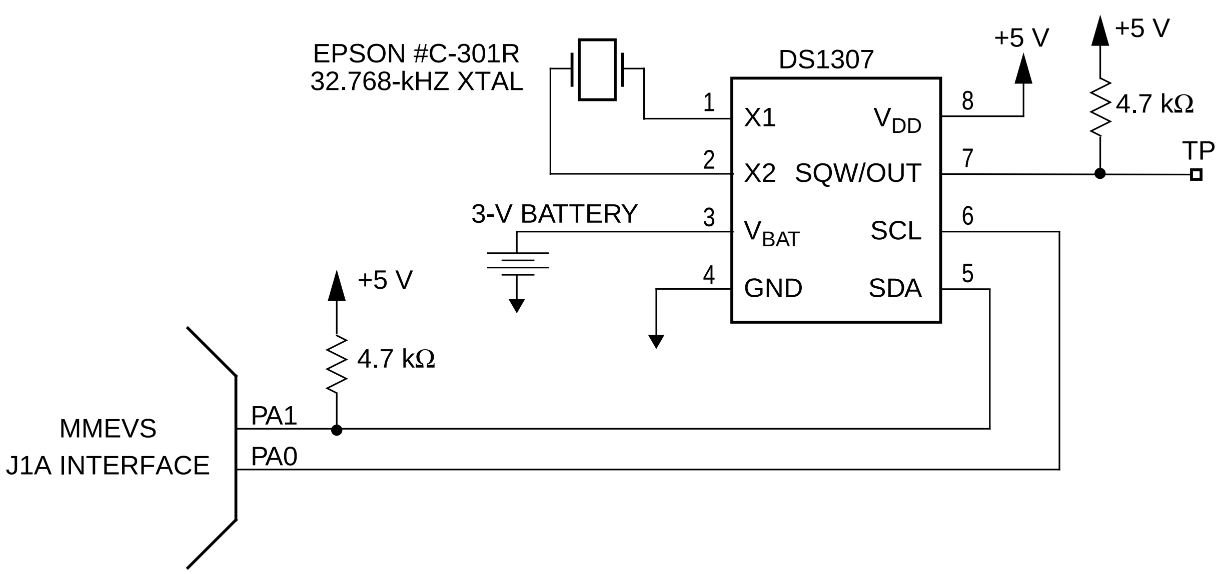

Figure 11. J1A-to DS1307 Interface Test Circuit

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 specifically to manipulate single I/O pins.

Five subroutines were created to provide an easy application programming interface (API). These routines are:

- START_SER — Sends out a start condition on the bus
- STOP_SER — Sends out a stop condition on the bus
- TXD — The master takes the contents of AccA and transmits it MSB first to the DS1307. The master also checks for acknowledgement from the DS1307.
- RXD — After the master addresses the DS1307 with its identification code and the read bit, the DS1307 transmits a byte of data back to the master. This routine reads that byte and puts it into AccA. The master also generates an acknowledgment back to the DS1307.
- RXD_LAST — This routine is just like RXD but it is used for the last byte read from the DS1307. It does not generate an acknowledgment back to the DS1307.

The flowcharts for the DS1620 serial I/O drivers are shown in **Flowcharts for the Test Interface**. These routines were written especially for the DS1307 and may not be able to properly drive other MCU peripherals with 2-wire serial buses.

A typical application would use the SQW/OUT pin on the DS1307. When configuring this pin for a 1-Hz signal, feed the signal to the $\overline{\text{IRQ}}$ pin of an MCU. An interrupt routine can be created to read the contents of the DS1307 every time a 1-Hz signal hits the $\overline{\text{IRQ}}$ pin. This should take minimal CPU bandwidth and provide the user an easy way to retrieve time and date information.

The main test routine was written to verify the bus interface between the DS1307 and the J1A. It writes a known date and time into the DS1307 and then reads it back out. The data read is put into a RAM buffer on the HC05. When the emulator is stopped, read the contents of the HC05 RAM buffer to verify the transmission process.

The test routine sequence is shown in **Figure 15**. The assembly code for the test routine is provided in the section titled **Code Listing**.

The sequence of tests is:

1. Configure the device to turn on a 1-Hz signal on the SQW/OUT pin.
   a. Transmit a start condition.
   b. Transmit the DS1307 code to write to the device of %11010000.
   c. Transmit the control register address and then $10.
   d. Transmit a stop condition.
2. Write start time.
   a. Transmit a start condition.
   b. Transmit the DS1307 code to write to the device of %11010000.
   c. Transmit the starting address of $00, the seconds register.
   d. Transmit Saturday, June 20, 1998, 4:30:00 p.m. (By writing a 0 to bit 7 of the seconds register, the crystal circuit has been turned on.)
   e. Transmit a stop condition.
3. Read time and date, store away to HC05 RAM buffer.
   a. Transmit a start condition.
   b. Transmit the DS1307 code to write to the device of %11010000.
   c. Transmit the starting address of $00.
   d. Transmit a stop condition.
   e. Transmit a start condition.

f. Transmit the DS1307 code to read from the device of %11010001.

g. Read the date and time and store away to HC05 RAM.

h. Transmit a stop condition.

Since the real-time clock is running, you can restart the code at step 3 and verify that it is keeping time.

This routine demonstrates the interface software needed to communicate with the DS1307. Although the J1A was used, any HC05 device could utilize this interface code. Minor adjustments of port pins and memory maps might be necessary.

Development Tools

The interface was created and tested using these development tools:

- M68MMPFB0508 — Motorola MMEVS platform board
- X68EM05J1A — Motorola J1A emulation module
- Win IDE Version 1.02 — Editor, assembler, and debugger by P&E Microcomputer Systems

Flowcharts for the Test Interface

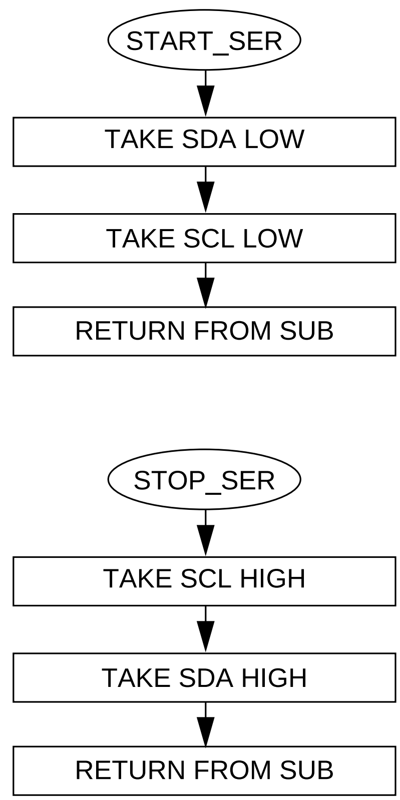

Figure 12. START_SER and STOP_SER Subroutines

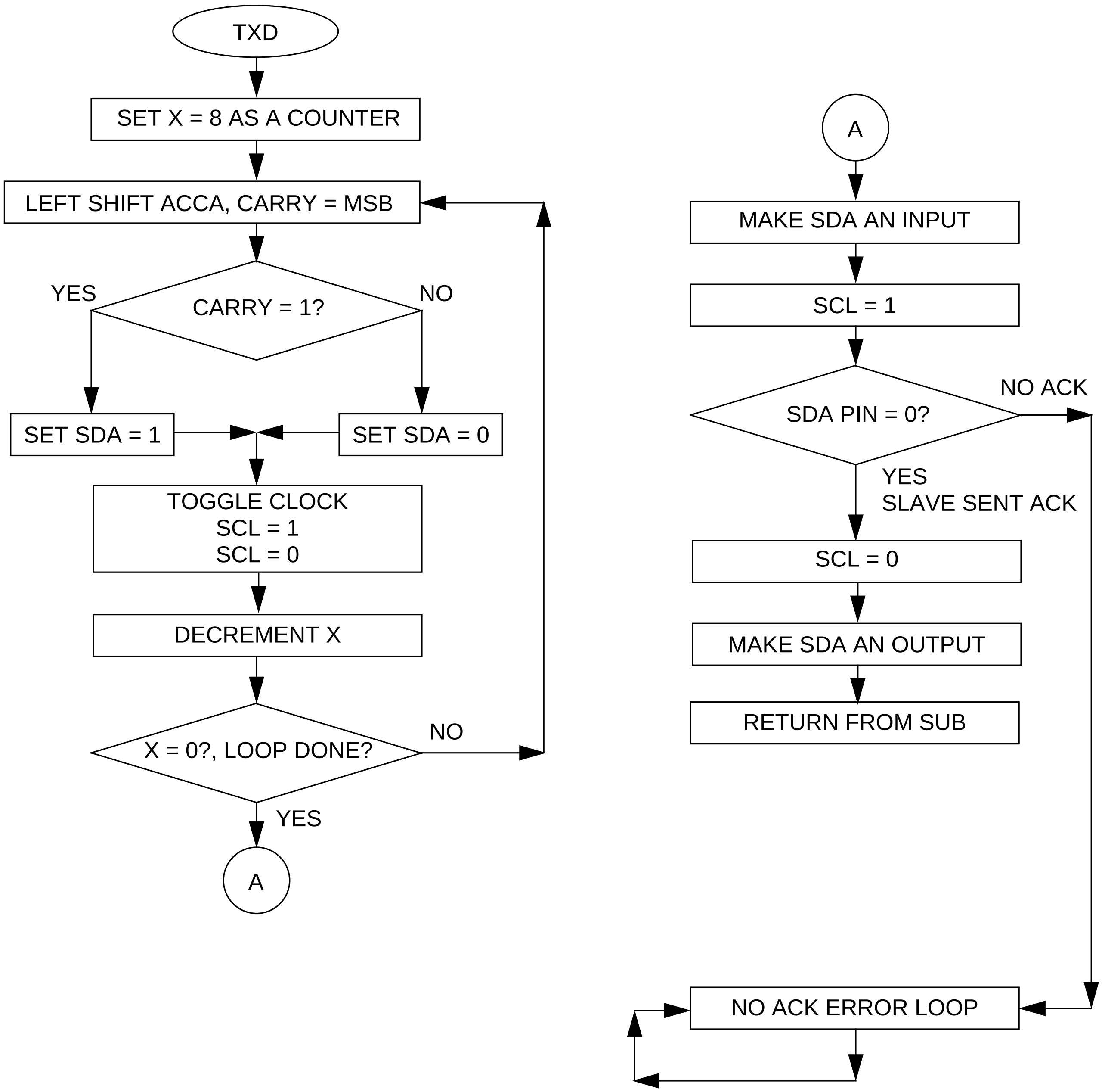

Figure 13. TXD Subroutine

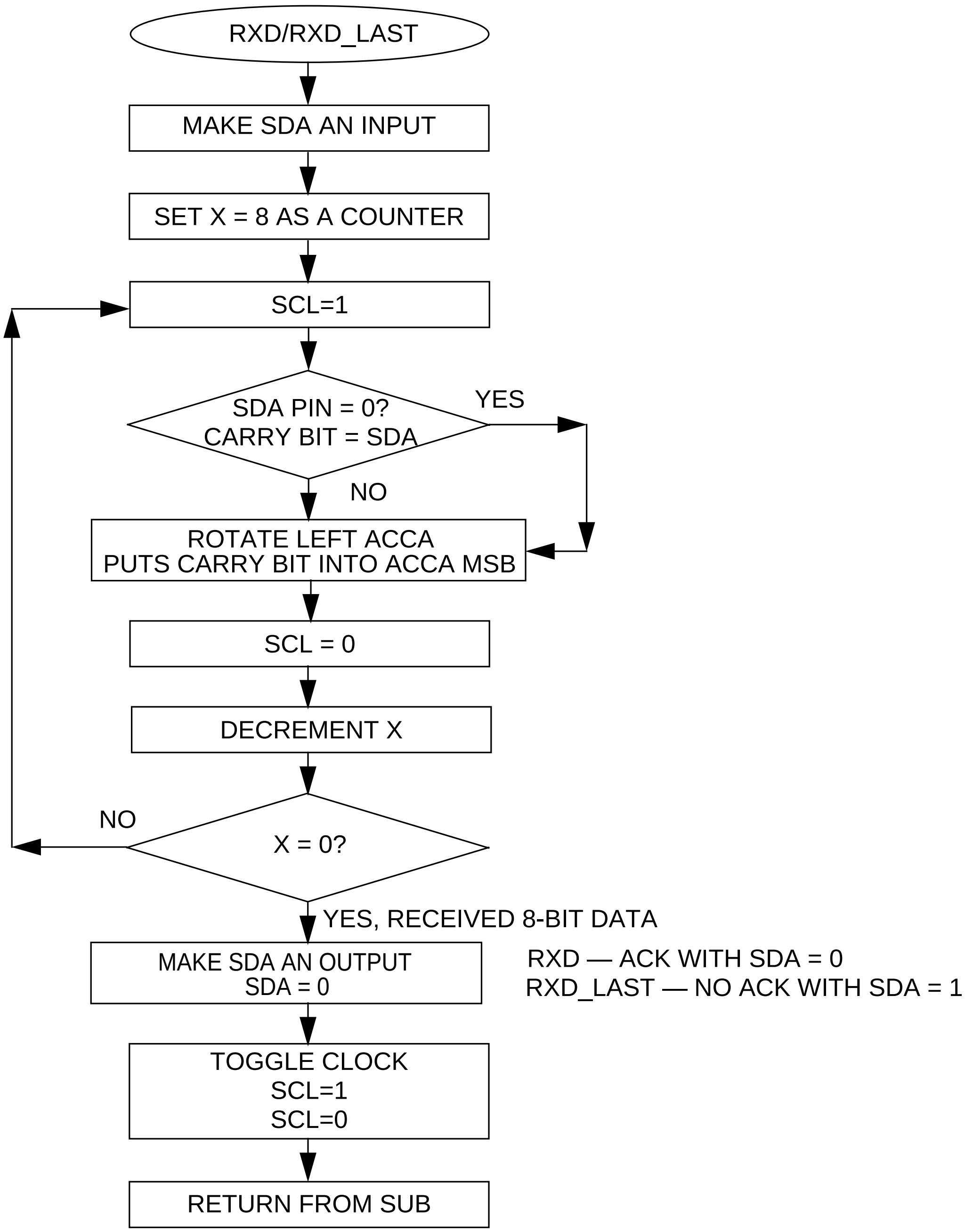

Figure 14. RXD/RXD_LAST Subroutines

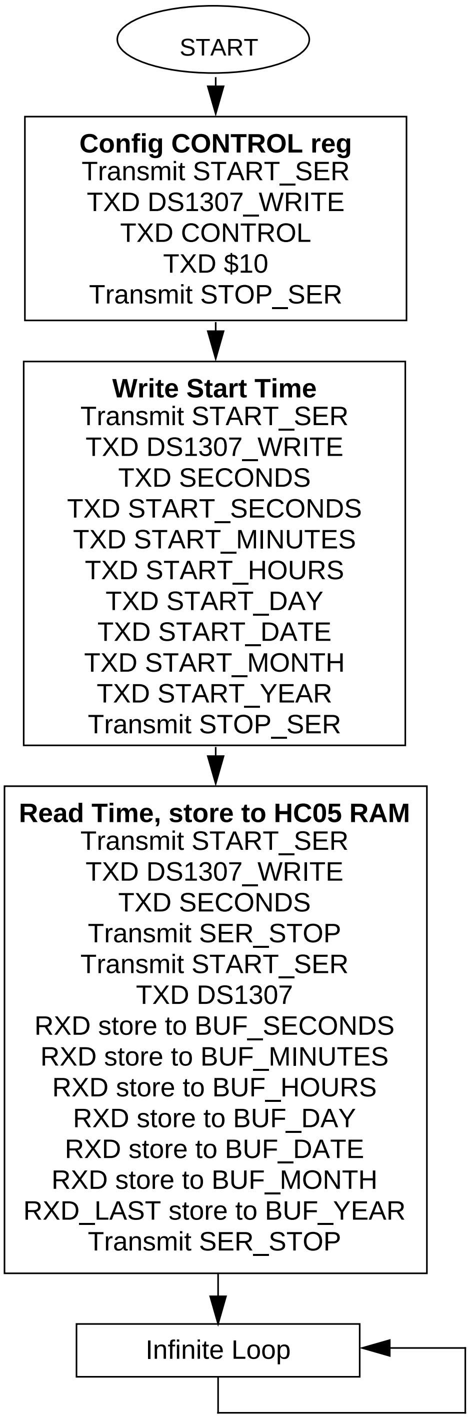

Figure 15. Flowchart for Main Test Routine

Code Listing

```
*********************************************************************************
*
* File name: DS1307.ASM
* Example Code for the MC68HC705J1A Interface to the
*     Dallas DS1307 Serial Real Time Clock
* Ver: 1.0
* Date: June 1, 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 a Non-volatile Clock to the MC68HC705J1A" Literature # AN1759/D
*
*********************************************************************************

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

*** Application Specific Definitions
SER_PORT        EQU    $00                      ;PORTA is SER_PORT
SCL             EQU    0T                       ;PORTA, bit 0, clock signal
SDA             EQU    1T                       ;PORTA, bit 1, data signal

DS1307_WRITE    EQU    $D0                      ;addresses the DS1307 for write
DS1307_READ     EQU    $D1                      ;addresses the DS1307 for read
SECONDS         EQU    $00                      ;DS1307 address for seconds
MINUTES         EQU    $01                      ;DS1307 address for minutes
HOURS           EQU    $02                      ;DS1307 address for hours
DAY             EQU    $03                      ;DS1307 address for the day
DATE            EQU    $04                      ;DS1307 address for the date
MONTH           EQU    $05                      ;DS1307 address for the month
YEAR            EQU    $06                      ;DS1307 address for the year
CONTROL         EQU    $07                      ;DS1307 address for control

*** Memory Definitions
EPROM           EQU    $300                     ;start of EPROM mem
RAM             EQU    $C0                      ;start of RAM mem
RESET           EQU    $7FE                     ;vector for reset

*** Time Start Definitions for test
*** Start on Saturday, June 20th, 1998, 4:30:00 PM
START_SECONDS   EQU    $00                      ;0 seconds
START_MINUTES   EQU    $30                      ;30 minutes
START_HOURS     EQU    $64                      ;4 hours, PM, 12 Hour mode
START_DAY       EQU    $06                      ;Saturday
START_DATE      EQU    $20                      ;20th
START_MONTH     EQU    $06                      ;June
START_YEAR      EQU    $98                      ;1998
```

```
*** RAM VARIABLES ****************************************************************
* Buffer for test reading data from the DS1307
               ORG      RAM
BUF_SECONDS    DB       1                    ;buffer on HC05 for seconds
BUF_MINUTES    DB       1                    ;buffer on HC05 for hours
BUF_DAY        DB       1                    ;buffer on HC05 for the day
BUF_DATE       DB       1                    ;buffer on HC05 for the date
BUF_MONTH      DB       1                    ;buffer on HC05 for the month
BUF_YEAR       DB       1                    ;buffer on HC05 for the year


*** MAIN ROUTINE *****************************************************************
               ORG      EPROM                ;start at begining of EPROM
*** Intialize Ports
START          lda      #$03                 ;init SER_PORT
               sta      SER_PORT
               lda      #$03                 ;make SER_PORT pins outputs
               sta      DDRA

*** DS1307 configuration
*** Turn on osc, turn on SQW/OUT pin with 1 Hz signal
               jsr      START_SER            ;start serial transmission

               lda      #DS1307_WRITE        ;address the DS1307 device, write
               jsr      TXD
               lda      #CONTROL             ;send address of control reg
               jsr      TXD
               lda      #$10                 ;send config data
               jsr      TXD

               jsr      STOP_SER             ;stop serial transmission

*** Write Starting Time to DS1307
               jsr      START_SER            ;start serial transmission

               lda      #DS1307_WRITE        ;address the DS1307 device, write
               jsr      TXD
               lda      #SECONDS             ;start address of DS1307
               jsr      TXD
               lda      #START_SECONDS       ;write seconds
               jsr      TXD
               lda      #START_MINUTES       ;write minutes
               jsr      TXD
               lda      #START_HOURS         ;write hours
               jsr      TXD
               lda      #START_DAY           ;write day
               jsr      TXD
               lda      #START_DATE          ;write date
               jsr      TXD
               lda      #START_MONTH         ;write month
               jsr      TXD
               lda      #START_YEAR          ;write year
               jsr      TXD

               jsr      STOP_SER             ;stop serial transmission
```

```
*** Read Time, store away in HC05 time buffer for verification
* Write starting address
              jsr     START_SER             ;start serial transmission

              lda     #DS1307_WRITE         ;address the DS1307 device, write
              jsr     TXD
              lda     #SECONDS              ;start address of DS1307 read
              jsr     TXD

              jsr     STOP_SER              ;stop serial transmission

* Read Time data put in HC05 buffer
              jsr     START_SER             ;start serial transmission

              lda     #DS1307_READ          ;address the DS1307 device, read
              jsr     TXD
              jsr     RXD
              sta     BUF_SECOND            ;read seconds, store to buffer
              jsr     RXD
              sta     BUF_MINUTES           ;read minutes, store to buffer
              jsr     RXD
              sta     BUF_HOURS             ;read hours, store to buffer
              jsr     RXD
              sta     BUF_DAY               ;read the day, store to buffer
              jsr     RXD
              sta     BUF_DATE              ;read the date, store to buffer
              jsr     RXD
              sta     BUF_MONTH             ;read the month, store to buffer
              jsr     RXD_LAST
              sta     BUF_YEAR              ;read the year, store to buffer

              jsr     STOP_SER              ;stop serial transmission

DUMMY         bra     DUMMY                 ;test sequence is over

*** SUBROUTINES ******************************************************************
*** Sends out Start command on bus
START_SER     bclr    SDA,SER_PORT          ;SDA=0
              bclr    SCL,SER_PORT          ;SCL=0
              rts

*** Sends out Stop command on bus
STOP_SER      bset    SCL,SER_PORT          ;SCL=1
              bset    SDA,SER_PORT          ;SDA=1
              rts

*** Routine takes contents of AccA and transmits it serially to
*** the DS1307, MSB first
*** Looks for ACK, goes to ERROR routine if no ACK
TXD           ldx     #8T                   ;set counter

WRITE         asla                          ;Carry bit = MSB
              bcc     J1
              bset    SDA,SER_PORT          ;SDA=1
              bra     CLOCK_IT              ;branch to clock_it
J1            bclr    SDA,SER_PORT          ;SDA=0
              brn     J1                    ;evens it out
```

```
CLOCK_IT       bset    SCL,SER_PORT          ;SCL=1
               bclr    SCL,SER_PORT          ;SCL=0
               decx                          ;decrement counter
               bne     WRITE
* Check for ACK
               bclr    SDA,DDRA              ;SDA is input
               bset    SCL,SER_PORT          ;SCL=1
               brclr   SDA,SER_PORT,J2       ;if SDA=0, slave ACK

ACK_ERROR      bra     ACK_ERROR             ;no slave ACK, error loop

J2             bclr    SCL,SER_PORT          ;SCL=0
               bset    SDA,DDRA              ;SDA is output
               rts                           ;return from sub


*** Routine clocks the DS1307 to read data from SDA, MSB first
*** 8 bit contents are put in AccA
*** Generates ACK back to slave
RXD            bclr    SDA,DDRA              ;make the SDA pin on J1A input
               ldx     #8T                   ;set counter

READ           bset    SCL,SER_PORT          ;SCL=1
               brclr   SDA,SER_PORT,J3       ;carry bit = SDA
J3             rola                          ;put carry bit into AccA MSB
               bclr    SCL,SER_PORT          ;SCL=0

               decx                          ;decrement counter
               bne     READ
* ACK back to slave
               bset    SDA,DDRA              ;make the SDA pin on J1A output
               bclr    SDA,SER_PORT          ;SDA=0
               bset    SCL,SER_PORT          ;SCL=1
               bclr    SCL,SER_PORT          ;SCL=0
               rts                           ;return from sub

*** Routine clocks the DS1307 to read data from SDA, MSB first
*** 8 bit contents are put in AccA
*** Generates NO ACK back to slave, signals last read to DS1307
RXD_LAST       bclr    SDA,DDRA              ;make the SDA pin on J1A input
               ldx     #8T                   ;set counter

READ_LAST      bset    SCL,SER_PORT          ;SCL=1
               brclr   SDA,SER_PORT,J4       ;carry bit = SDA
J4             rola                          ;put carry bit into AccA MSB
               bclr    SCL,SER_PORT          ;SCL=0

               decx                          ;decrement counter
               bne     READ_LAST
* NO ACK back to slave
               bset    SDA,DDRA              ;make the SDA pin on J1A output
               bset    SDA,SER_PORT          ;SDA=1
               bset    SCL,SER_PORT          ;SCL=1
               bclr    SCL,SER_PORT          ;SCL=0
               rts                           ;return from sub


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

References

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

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

DS1307 Datasheet, Dallas Semiconductor, 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/

MOTOROLA