TEMPERATURE COMPENSATED REAL TIME CLOCK REFERENCE DESIGN PROGRAMMER'S GUIDE

Relevant Devices

This application note applies to the following devices:

C8051F300, C8051F302

1. Introduction

Real Time Clocks (RTCs) are used in many applications where it is necessary to keep track of time and date information. The common solution of adding a dedicated RTC device to a board adds to the BOM cost and also increases the board space. A better, cost-effective alternative is to implement the RTC functionality into a microcontroller that performs other useful tasks as well. The Temperature Compensated Real Time Clock (TC-RTC) Reference Design is a complete RTC solution that includes all the firmware and hardware necessary to implement an RTC with full clock and calendar functionality along with temperature compensation. This solution offers significant cost savings compared to solutions involving dedicated RTCs that provide similar functionality. The design uses the Silicon Laboratories C8051F300 Mixed Signal MCU that is available in a 3x3 mm 11-pin QFN package.

The TC-RTC RD firmware implements the following functions:

- Real Time Clock that counts seconds, minutes, hours, date, day of the week, month, and year with leap-year compensation valid up to 2099
- Automatic time compensation for crystal frequency variations due to temperature
- 56-Byte NVRAM with the data stored in internal Flash memory
- SMBus/I^{2}C Interface or UART Interface, depending on firmware loaded
- Square wave output signal at 1/2 Hz rate

This application note describes the design of the TC-RTC hardware and firmware. Refer to the “Temperature Compensated Real Time Clock Reference Design Kit User’s Guide” for step-by-step demonstration instructions.

2. Hardware Overview

The TC-RTC reference design hardware is implemented as an evaluation board that is shown in Figure 1. The Silicon Laboratories C8051F300 MCU uses an external 32 kHz crystal tied to a Timer input as the clock source for the RTC. The RTC evaluation board provides two interfaces—UART and SMBus/I^{2}C.

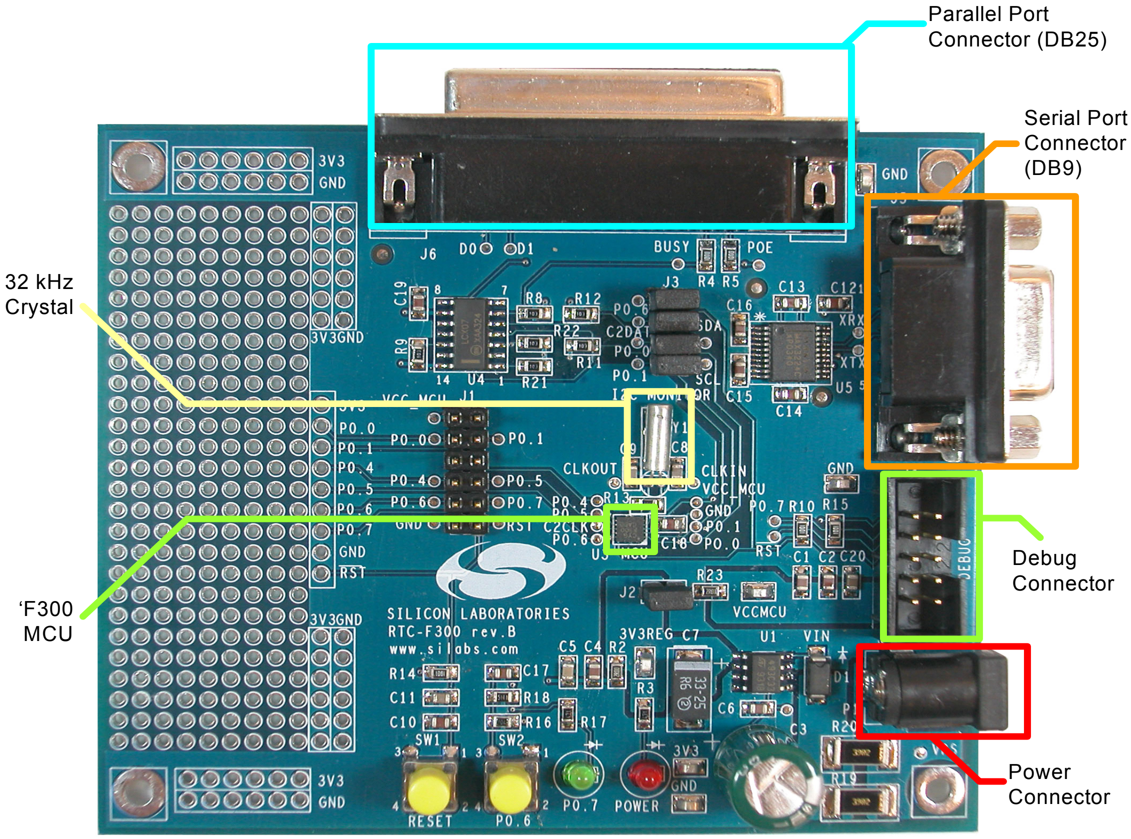

Figure 1. TC-RTC C8051F300 Evaluation Board

The Silicon Laboratories C8051F300 MCU is a small form-factor mixed-signal MCU with a rich feature set. The features of the MCU that are used in this reference design are listed below:

- High-speed 8051 core that delivers up to 25 MIPS throughput with 25 MHz clock
- 8 kB Flash memory (in-system programmable) and 256 bytes internal data RAM
- 25 MHz internal oscillator and external crystal/oscillator inputs
- 8-bit 500 ksps ADC
- On-chip temperature sensor
- Hardware enhanced UART and SMBus serial ports
- Three general-purpose 16-bit counter/timers
- Real time clock mode using timer and external clock source

More detailed information about this MCU can be found in the C8051F300 data sheet that is available here:

http://www.silabs.com/public/documents/tpub_doc/dsheet/Microcontrollers/Small_Form_Factor/en/C8051F30x.pdf

The pin connections of the C8051F300 MCU as used in this design are shown in Figure 2. The board schematic and bill of materials are included in the "Temperature Compensated Real Time Clock Reference Design Kit User's Guide."

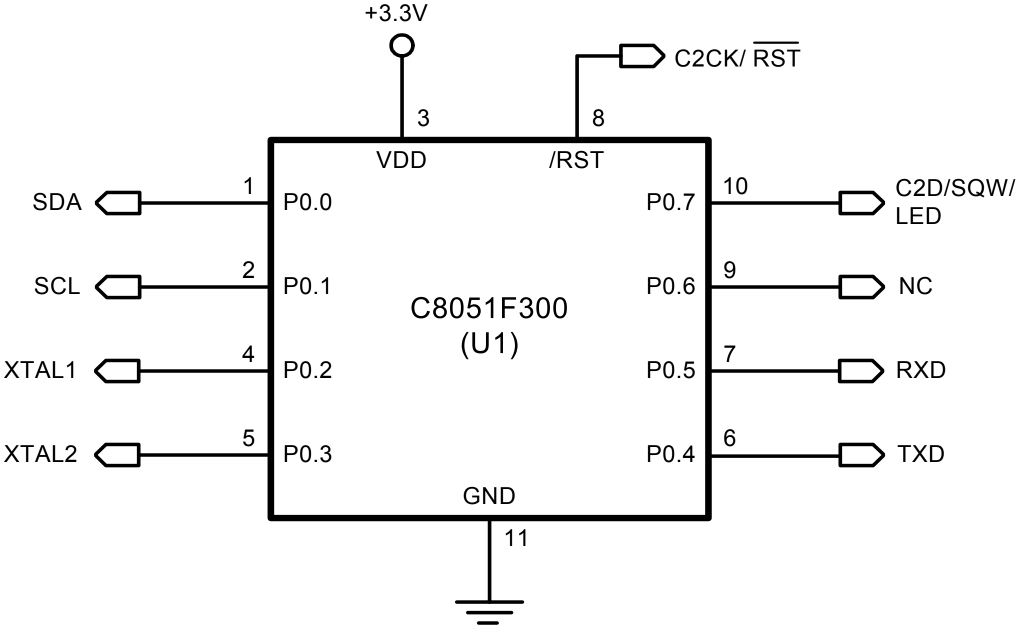

Figure 2. TC-RTC C8051F300 MCU Pin Connections

3. Compensating for Temperature Variations

This section describes the theory of operation behind the temperature compensation feature of the TC-RTC RD.

3.1. Need for Compensation

The primary parameters that cause frequency deviation in a quartz crystal are as follows:

- Ambient temperature
- Age of the crystal
- Power supply voltage

Among these, the dominant factor that affects the crystal frequency is ambient temperature. Figure 3 shows a plot of the variation of crystal frequency versus temperature. From the parabolic curve, it can be seen that the RTC will *lose* time if the temperature is increased or decreased from the room temperature value (25 °C). Note that the RTC will never have a positive error caused by temperature variations (i.e., *gain* time) because the maximum frequency is at room temperature.

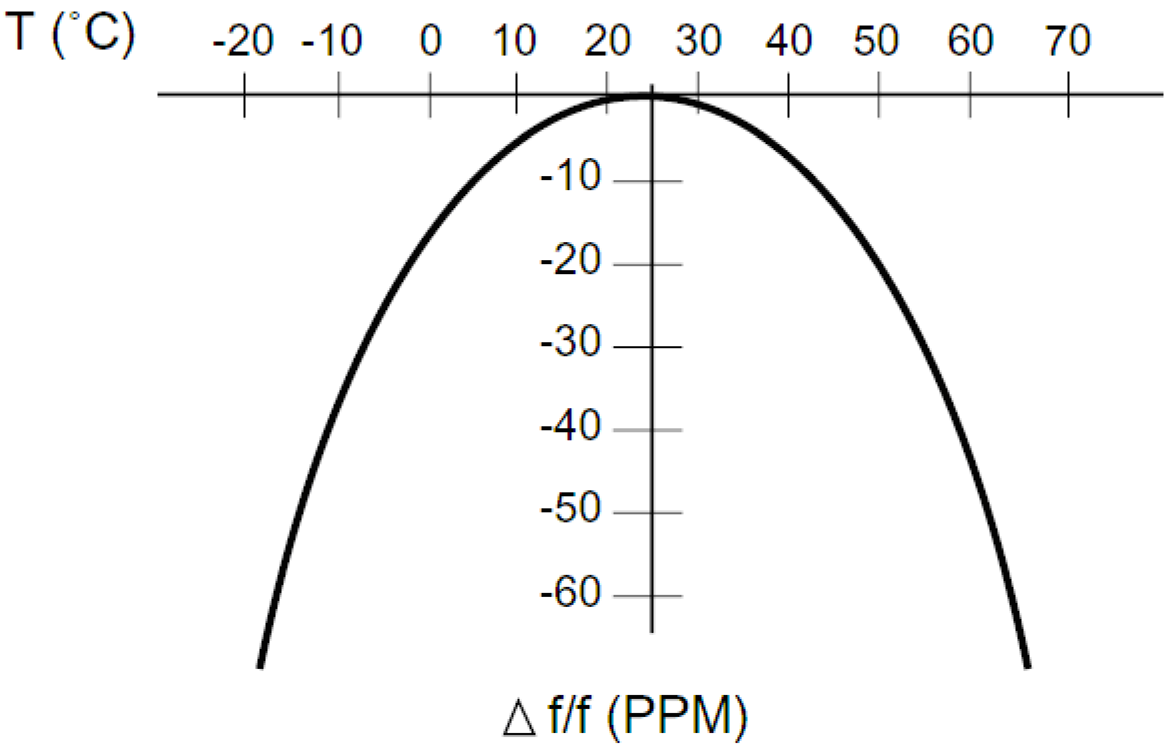

Figure 3. Parabolic Temperature Curve

The maximum frequency variation is approximately –0.04 ppm/ºC^2. So, the frequency deviation can be expressed as the following:

$\Delta f/f = 0.04 \text{ ppm} \times (\Delta T)^2$

where,

ΔT = Ambient Temperature – 25 ºC

3.2. Calculating Time Compensation

The TC-RTC Reference Design firmware repeats the following steps once per minute to calculate and accumulate lost time.

1. The ADC is used to measure the die temperature from the on-chip temperature sensor. “3.3. Calculating Ambient Temperature” describes the calculation involved.
2. The value measured by the ADC is then used to calculate the deviation in ppm, and the result is stored in memory. This indicates the number of microseconds that need to be compensated.

At the end of a 24-hour period, the total accumulated error is added to the RTC time to complete the compensation process. The temperature is assumed to not vary widely within a one-minute period. Refer to the functions in the module “F30x_TCRTC_Temperature.c” for more details.

3.3. Calculating Ambient Temperature

The on-chip temperature sensor produces a voltage output that is proportional to the absolute temperature of the ‘F300 die as shown in Figure 4. The typical relationship between this voltage and the die temperature is shown below:

$$V_{TEMP} = \left(3.35\frac{mV}{C}\right) \times TEMP_C + 897mV$$

where,

V_{TEMP} is the temperature sensor voltage output (mV)

$TEMP_C$ is the die temperature (ºC)

"4.2. ADC Configuration" on page 7 describes how the ADC is configured to read the temperature sensor value. More information about the temperature sensor and the ADC can be found in Chapter 5 of the C8051F30x device data sheet.

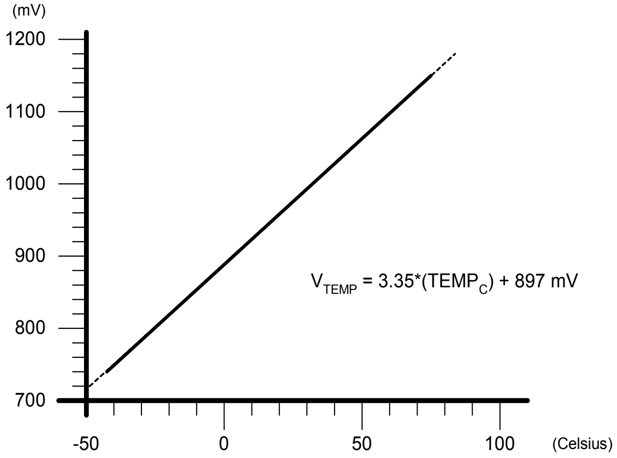

Figure 4. Typical Temperature Sensor Transfer Function

4. Firmware Implementation

The following sections describe the register memory map and the primary components that constitute the TC-RTC firmware.

4.1. RTC Registers and Non-volatile User RAM

Figure 5 shows the addresses of the RTC registers and the NVRAM. The RTC registers store the current time and date information. These registers are maintained in the ‘F300 device internal volatile RAM. " Appendix A—TC-RTC: Supported Commands" on page 9 describes the UART commands used to read and write these registers. The SMBus/I^{2}C access sequence is described in Section 4.4.

The TC-RTC also provides a 56-byte User NVRAM that is stored in the ‘F300 device internal Flash memory. This can be used as a general-purpose non-volatile storage area. The current values of the Flash NVRAM are always mirrored in a 56-byte array in RAM. This is because the ‘F300 Flash memory can be erased only in 512-byte pages. So, writing one or more bytes to the NVRAM is a three-step process—the bytes are first written to the RAM array, the Flash page is erased, and the 56-byte array is copied from RAM to Flash memory.

One useful application of this NVRAM is to store the time and date information just before a power failure. The stored information can later be used for debugging or some other purpose. To do this, the power supply line should be monitored and as soon as a dip is detected, the time and date information should be read from the RTC registers and written to the NVRAM. This should be managed by an external device and is not automatically done by the TC-RTC firmware.

Address	Register
00H	Seconds
01H	Minutes
02H	Hours
03H	Day of Week
04H	Date
05H	Month
06H	Year
07H	Control (Reserved)
08H 3FH	NVRAM 56 x 8

Figure 5. RTC and NVRAM Registers

4.2. ADC Configuration

The ADC is configured to use the temperature sensor as the positive input and ground as the negative input. V_{DD} is used as the voltage reference, and the SAR conversion clock is set to 5 MHz. The Programmable Gain Amplifier (PGA) is set to a gain of 2. Oversampling is performed to improve the accuracy of the measurements by collecting and averaging 65536 (64k) samples of the temperature sensor output.

The first time the TC-RTC board is powered-on after loading the firmware, a 1-point offset calibration is performed as follows:

1. Wait for 15 seconds (soak time). This allows the die to heat up to normal operating temperature.
2. Measure temperature sensor output using the above ADC settings.
   Note: The calibration measurement assumes a die temperature of 28 ºC.
3. Store the measured value in non-volatile Flash memory.

4.3. Timers Configuration

The ‘F300 MCU has three general-purpose timers. All these three timers are used by the TC-RTC firmware. The timers are configured and used for the following purposes:

- Timer0—This is used to provide the SMBus SCL Low Timeout.
- Timer1—This is either used as a UART baud rate generator, or for SMBus Free Timeout (SCL High) detection.
- Timer2—This uses the external oscillator as its clock source, and is used for RTC time measurement.

4.4. UART Interface

The UART interface provided by the TC-RTC is a simple 2-wire interface that uses only the TXD and RXD lines. It supports three commands, which are described in detail with examples in " Appendix A—TC-RTC: Supported Commands" on page 9. Refer to the “Temperature Compensated Real Time Clock Reference Design Kit User’s Guide” for step-by-step demonstration instructions to use this interface.

4.5. SMBus Interface

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL (see Figure 6). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL. See the SMBus chapter in the C8051F30x device data sheet for more information on arbitration and other SMBus details.

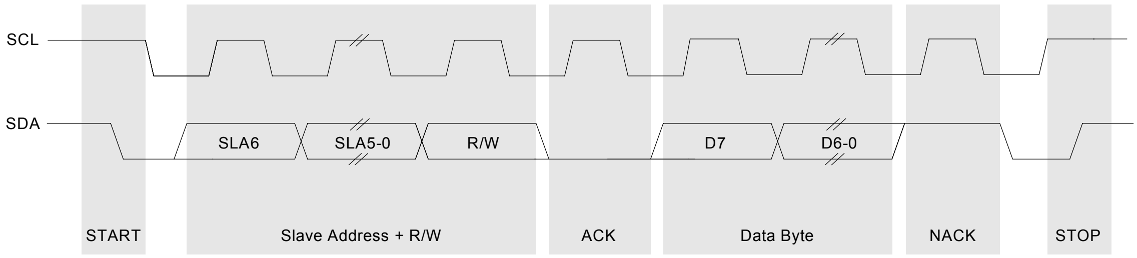

Figure 6. Typical SMBus Transaction

The TC-RTC firmware uses the SMBus interface as a Slave, with a Slave Address of 0xD0. This can be changed by modifying the SLA_ADD macro definition in “F30x_TCRTC_Interface.h”. The Slave TC-RTC can be in one of two modes, which are described in the following sections.

4.5.1. Slave Receiver Mode

If the Master transmits a START condition, followed by the correct slave address and the direction bit set to “0” (write), the slave enters this mode and sends an ACK. The first received byte sets the register address pointer in the TC-RTC firmware. Subsequent received bytes are considered data and are written to the register pointed to by the register address pointer, while incrementing the pointer after every write. The Slave sends an ACK after each byte is received. The RTC exits this mode when the Master sends a STOP condition. Figure 7 shows a Typical Slave Receiver Sequence.

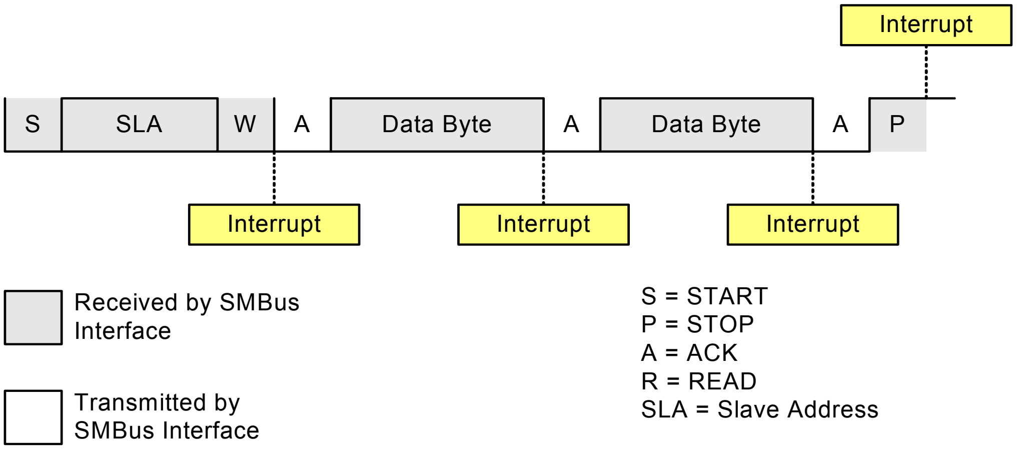

Figure 7. Typical Slave Receiver Sequence

4.5.2. Slave Transmitter Mode

If the Master transmits a START condition, followed by the correct slave address and the direction bit set to “1” (read), the slave receiver enters this mode and sends an ACK. Then, the RTC starts sending the byte pointed to by the register address pointer set by the previous write operation, while incrementing the pointer after sending each byte. It waits for an ACK from the Master after sending every byte. The RTC exits this mode when the Master sends a NACK. Figure 8 shows a Typical Slave Transmitter Sequence.

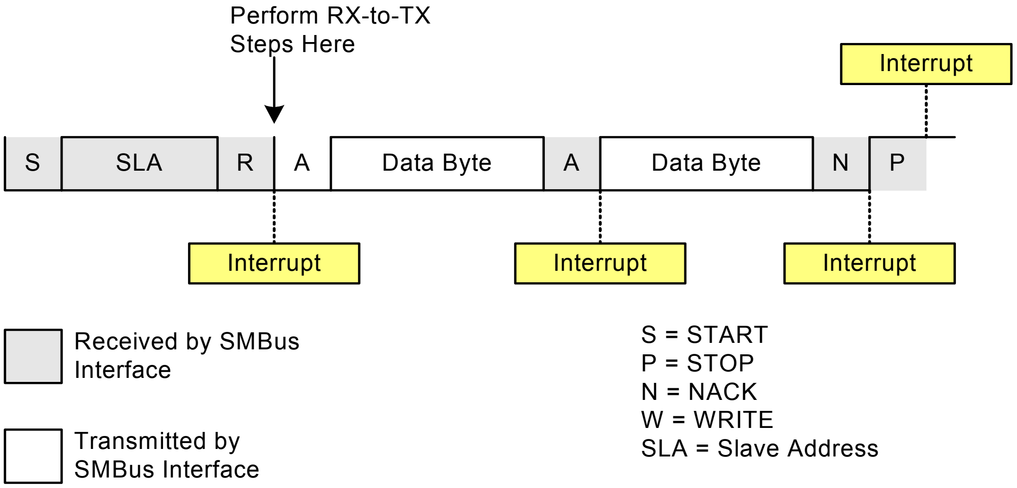

Figure 8. Typical Slave Transmitter Sequence

The register address pointer wraps around to address 0x00 when incremented from address 0x3F. Refer to the SMBus ISR in the module “F30x_TCRTC_Main.c” for more details about the implementation of the SMBus state machine.

To easily evaluate the SMBus interface provided with the TC-RTC board, a test software utility is provided in AN293.zip. This Windows software uses the PC parallel port to emulate a SMBus Master and communicates with the TC-RTC via its SMBus interface. This is provided only to demonstrate the SMBus interface and is not meant as a real-world application of this design.

Appendix A—TC-RTC: Supported Commands

TC-RTC Firmware with UART Interface

The TC-RTC firmware with UART interface supports a set of commands to read/write date and time. The supported commands along with explanations are listed below:

- s – Toggles auto-display mode that displays time once every second.
- r xx yy – Read yy bytes starting at address xx.
- w xx yy data – Write yy bytes starting at address xx.

Notes:

1. The commands are case-sensitive. Only lower-case letters are accepted.
2. The addresses should be in decimal format (e.g., address 0x1F is entered as 31).
3. See “ Appendix B—TC-RTC: Implementation Details” for information on how the RTC register data is formatted.

Write Command Examples

- Date/Time: 11:05:39 AM, Saturday, December, 08, 1979→39 05 51 06 08 12 79
  w000739055106081279
- Time: 21: 48: 22→22 48 21
  w0003224821
- Day: Tuesday→02
  w030102
- NVRAM data write: 01 02 03 starting at address 0x1F (31)→31 01 02 03
  w3103010203

Read Command Examples

- Date/Time: r0007
- NVRAM (all 56 bytes): r0856
- NVRAM (4 bytes starting at address 26): r2604

Appendix B—TC-RTC: Implementation Details

UART Communication Parameters

- Baud rate: 9600 bps
- Data format: 8 data bits, 1 stop bit, no parity
- Flow control: None

RTC and NVRAM Registers—Data Format

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Range
00H	CH	Seconds							00-59
01H	0	Minutes							00-59
02H	0	12 / 24	Hours A/P						01-12 01-23
03H	0	0	0	0	0	Day of Week			1-7
04H	0	0	Date						01-31
05H	0	0	0	Month					01-12
06H	Year								00-99
07H	Control (Reserved)								
08H 3FH	NVRAM 56 x 8 (Stored in non-volatile Flash Memory)								

Figure 9. RTC and NVRAM Registers—Data Format

Notes:

1. The contents of the time and calendar registers are in binary coded decimal (BCD) format.
2. The Clock Halt (CH) bit, when set to zero, will disable the RTC.
3. The 12/24 bit selects 12-hour mode if set to one.
4. In 12-hour mode, the AM/PM bit selects “PM” if set to one. In 24-hour mode, this bit is the second 10-hour bit (20-23 hours).

NOTES:

CONTACT INFORMATION

Silicon Laboratories Inc.

4635 Boston Lane
Austin, Texas 78735
Email: MCUinfo@silabs.com
Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.