

A Guide to FM3127x/FM31L27x Devices

I²C Companion, Typical Application, Pseudo Code



OVERVIEW

The new FM3127x and FM31L27x product families offer integrated Processor Companion, Real Time Clock (RTC), and FRAM memory. The FM3227x and FM32L27x families have the same feature set but do not have the RTC. Each family includes devices that have 4Kb, 16Kb, 64Kb, and 256Kb of nonvolatile FRAM memory. All devices employ an industry standard I²C interface. This interface is used to access the memory, the processor companion, and the RTC. The “L” families offer supply voltage operation from 2.7V to 3.6V, and the standard families offer supply voltage operation from 4.0V to 5.5V. Table 1 summarizes the product families and their major feature sets.

Table 1. Summary of Part Types & Features

Part #	Memory	Companion	RTC	Voltage	2-wire Speed	Package
FM31278	256Kb	✓	✓	4.0 – 5.5V	1MHz	SOIC 14
FM31276	64Kb	✓	✓			
FM31274	16Kb	✓	✓			
FM31272	4Kb	✓	✓			
FM31L278	256Kb	✓	✓	2.7 – 3.6V	1MHz	SOIC 14
FM31L276	64Kb	✓	✓			
FM31L274	16Kb	✓	✓			
FM31L272	4Kb	✓	✓			
FM32278	256Kb	✓		4.0 – 5.5V	1MHz	SOIC 14
FM32276	64Kb	✓				
FM32274	16Kb	✓				
FM32272	4Kb	✓				
FM32L278	256Kb	✓		2.7 – 3.6V	1MHz	SOIC 14
FM32L276	64Kb	✓				
FM32L274	16Kb	✓				
FM32L272	4Kb	✓				

The FM31256/3164/3116/3104 and FM32256/3264/3216/3204 devices are similar to the new ‘27x families. A document that describes the differences between the new and older families is available on the Ramtron website, <http://www.ramtron.com>. Please refer to [AN405 - Comparing FM31xx and FM3127x.pdf](#) and [AN406 - Comparing FM32xx and FM3227x.pdf](#).

TWO LOGICAL DEVICES IN ONE

The processor companion comprises a power-on system reset, low-voltage detect, automatic switchover to backup power, a watchdog timer, an early power fail warning, two event counters, and a lockable 64-bit serial number. All FM3127x/31L27x devices are internally organized as two logical devices. The memory is one logical device, and the companion/RTC is the other logical device. Each has its own address space and is accessed via a 2-wire interface.

Each “device” is accessed using a unique Slave ID, 1010b for the memory and 1101b for the companion/RTC. Device Select bits are used to allow multiple chips on the 2-wire bus to share the same Slave ID, yet are individually addressed. This is used for memory expansion. Since the FM31x/FM32x devices have two address (serial select) pins, up to four devices can share the same I²C bus. This means that three expansion memory devices can be added to an FM31x or FM32x chip.

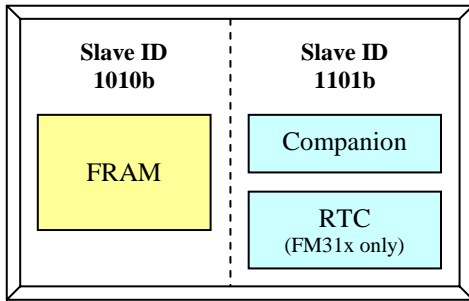


Figure 1. Two Logical Devices Have Unique Slave IDs

A Slave Address is required for every transaction in the 2-wire protocol. The 8-bit address is composed of the Slave ID, Device Select bits, and an R/W bit.

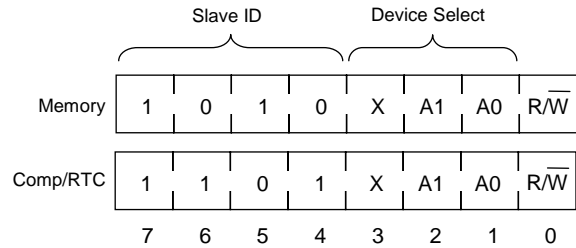


Figure 2. Slave Addresses

PROCESSOR COMPANION

The FM3127x and FM31L27x families integrate all the necessary processor supervisor features a designer may need. The companion features include:

- System power-on-reset (POR) with /RST pin
- Debounced Manual Reset
- Low Voltage Detect with a user programmable trip point
- Early Power-fail Warning (NMI)
- Battery Monitor & Trickle Charger

Each of these features will be covered in more detail.

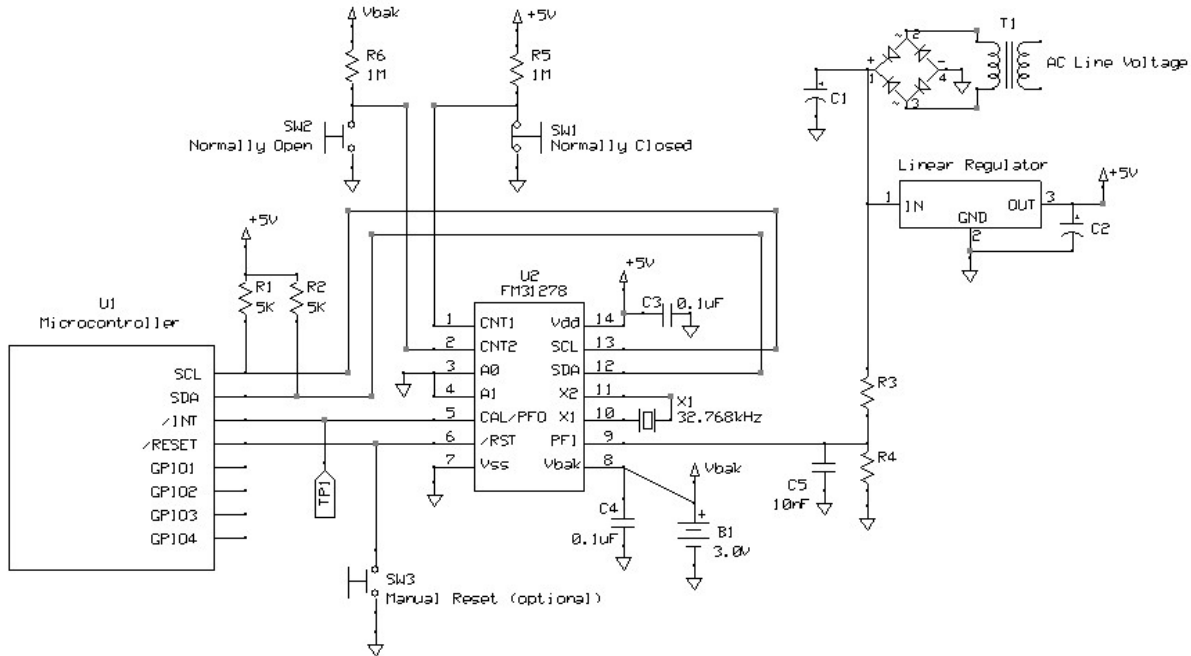


Figure 3. FM3127x Typical Application Circuit

TYPICAL APPLICATION

A typical system that uses the FM3127x might look like Figure 3. It is shown as an example with external components that could be used, typical values, and their connections to other system devices. In this case, an AC line-operated system is shown with a microcontroller (MCU), FM31278 device, and passive components. The MCU, in this case, has a dedicated 2-wire interface. Your micro may not have this port, but a 2-wire protocol may be written and the GPIO pins may be bit-banged to achieve the same. We will cover each pin function in the following sections.

SYSTEM RESET

A reliable system must be powered up and initialized properly. The FM3127x/31L27x products provide a processor reset signal when the system powers up and also whenever a system fault or manual override occurs. The /RST pin is primarily used to drive reset back to the processor but can be used as an input to provide a hardware reset to the system. The /RST pin **DOES NOT** reset or clear any FM3127x/31L27x internal registers.

There are two trigger sources that can drive reset low: a Low Voltage Detect circuit and the Watchdog Timer. At power up, the reset signal drives low as V_{DD} voltage rises toward its nominal operating value. The point at which reset is released is determined by V_{TP} , an internal trip voltage that is always compared to V_{DD} .

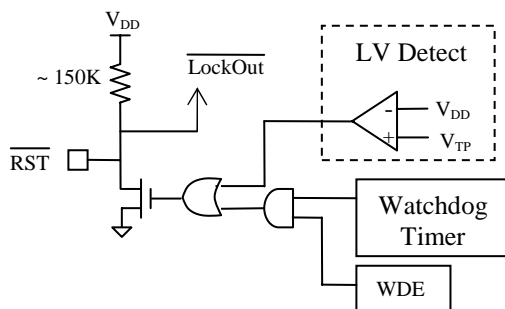


Figure 4. Reset Trigger Sources

Note that an internal pullup resistor (~150K) is provided on the /RST pin to eliminate the need for an external resistor. Once tripped, the reset circuit times out after ~ 150 ms (100ms min., 200ms max.). The user may set V_{TP} to one of two values: 3.9V or 4.4V on the FM3127x devices and 2.6V or 2.9V on the FM31L27x devices.

The other reset trigger source is the Watchdog Timer. The watchdog is a free-running, user-programmable

timer that can be set to time out as soon as 100 ms or as long as 3 seconds. The timer settings are stored in nonvolatile registers, so there is no need to re-initialize these or the V_{TP} trip voltage again. In order for the watchdog to trigger a reset, both the timer must expire and the watchdog enable bit (WDE) must be set. The timer may be restarted at any time within the timeout period by writing 1010b to the Watchdog Restart register. To conserve power, the watchdog counter may be disabled by writing 1111b to the Watchdog Timeout register.

Once a reset condition occurs, the device will set one of two flags to indicate the source of the reset. These flags are located in register 09h and are called WTR and POR. These bits are battery-backed. After power is restored, the system host can read these bits to determine the cause of the reset. The host must clear these bits after reading them.

TIP: During system prototyping and debug, the user can clear the WDE bit and still see the watchdog function in action by reading the WTR flag. A Watchdog Timer value is set to, say, 01101b or 1.3 seconds. The cleared WDE bit avoids driving the /RST pin low and locking out the I²C interface for a watchdog timer fault, yet you may read the WTR flag in register 09h to see if your software has a problem.

A manual hardware reset may be provided in the system by simply connecting a momentary contact switch to the /RST pin. Switch noise is filtered and debounced by the FM3127x/31L27x device. Note that in all reset cases, the chip internally drives a /LockOut signal that serves to block any 2-wire interface traffic and abort any pending write accesses to the FRAM memory array. A reset that occurs prior to all 8 data bits and an ACK will cause the write to abort. However, write accesses that have been acknowledged by the FRAM device will be completed, i.e. if a reset occurs after the 9th bit of the data phase.

EVENT COUNTERS

The FM3127x/31L27x devices integrate two 16-bit event counters for tamper-detect or other event logging purposes. Each counter has an input pin that is edge-triggered and polarity that is user-defined. The event, or edge, must be a CMOS logic level. The event counter control register and counter values are battery-backed.

In Figure 3, a normally open and a normally closed switch are shown, each switch tied to ground (or a system case or chassis). The normally open switch is

preferred if you don't want extra current pulled from the backup power source. If the system is normally powered up (V_{DD}) most of the time, an extra 3 μ A may not be a problem. On the other hand, an extra few microamps could reduce your backup time when using a capacitor as a backup source.

FEATURE: At the first powerup with no backup source, the event counters are cleared to all zeroes.

CAUTION: The CNT pins do not have a pullup or pulldown internally. Do not leave these pins floating.

EARLY POWER-FAIL WARNING

Most electronic systems use voltage-regulated power to provide a stable voltage to guarantee circuit operation. An unregulated power supply always precedes the regulator. As the unregulated voltage varies, the regulated voltage remains stable – to a point. As the unregulated supply voltage drops, due to a failure or normal system shutdown, the processor and memory subsystem is better served by knowing well before the regulator dies that power is lost. An early warning mechanism can be provided to the processor host of this impending doom. A non-maskable interrupt (NMI) is a means by which the host can prepare for this soon-to-be loss of power.

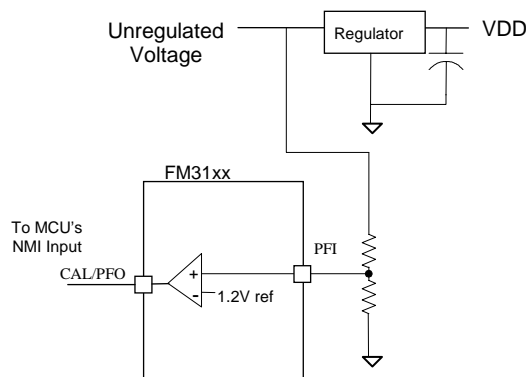


Figure 5. Early Power-fail Warning

A simple voltage divider, tied to the unregulated supply, can be applied to an on-chip comparator that senses this condition. The comparator output is tied to the processor NMI input. The trip point at which the comparator trips is a simple matter of resistor selection, knowing the other comparator input is internally set to 1.2V.

Referring back to Figure 3, AC line power is rectified and C1 filters the ripple. A 3.3V linear regulator supplies clean DC voltage to the V_{DD} pins of the FM31278 and microcontroller. The unregulated voltage on C1 is, say, 8V with ripple. The early

power fail feature of the FM31x/FM32x can be used by connecting a voltage divider (R3, R4) to the PFI input. Once this input goes below 1.2V, the PFI pin (tied to the controller's /INT) will drive low, signalling an interrupt. The resistor values are chosen based on how early you'd like to warn the micro that the power supply has been turned off. The trip voltage that causes the PFI to drive is determined by this equation:

$$1.2 = V_{TR} (R3/(R3+R4))$$

If R3=100K and R4=330K, then V_{TR} is about 5.1V.

Additional information on this topic is available in application note [AN400](#) on the Ramtron website.

REAL TIME CLOCK

A real time clock (RTC) is used to keep track of the time, day, and date, and to record this information whenever a system event occurs. The RTC consists of an oscillator and counters that derive time/calendar information, and a number of registers that hold said information and RTC control settings as well. The RTC runs continuously even if the main power fails. A backup power source keeps the RTC operating. The backup supply may be a 3V battery or a capacitor.

From the factory, the RTC oscillator is disabled. To start and configure the RTC, the /OSCFEN bit must first be set to 0. Then, the clock and calendar registers must be written to reflect the current time, day, and date. Be sure to write a value to each of the timekeeping registers 02h through 08h.

CAUTION: If a battery is applied to V_{BAK} prior to V_{DD} , the chip will draw about 1 μ A I_{BAK} . This occurs even if the oscillator is disabled. In order to maximize battery life, V_{DD} must be powered before a battery is applied to V_{BAK} .

The 32KHz oscillator is divided down through a series of counters. The first counter divides it by 32,768 to derive the 1-Hz signal for the Seconds counter. A 512Hz tap is used to drive out to the CAL/PFO pin for calibration purposes. The CAL/PFO pin in Figure 3 is shown as a test point. It is used to measure the 512Hz frequency in calibration mode. Only a crystal may be connected to the X1 and X2 pins – when calibrating the RTC oscillator, please use the test point TP1. The next counter uses seconds to drive a signal once per minute to the Minutes counter. Subsequent counters continue to divide down until there is one pulse per hour, month, and

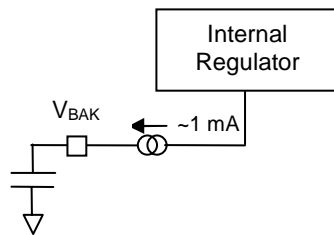
year driving their respective counters. The counters hold time and date information that is memory-mapped into RTC address space. These are locations 02h through 08h. Since the FM31x/FM32x uses two different Device IDs, they do not appear as FRAM locations. Users may read and write the time and date by reading and writing these locations without disturbing the FRAM locations.

The RTC data may be read without interrupting the RTC operation. When the RTC is read by writing the R bit to a “1”, a snapshot is taken of the current time-day-date and stored in registers where it can be read by the system host. The snapshot ensures that the time is not changing between successive reads from the host. Likewise during RTC writes, the registers hold the data written by the host and wait to transfer it to the RTC counters after all the time-day-date information has been written and W bit cleared. If W=0, writes to the RTC registers are ignored.

BACKUP POWER

The RTC, event counters, and various control settings are maintained even if the primary power source is removed. A backup power source on V_{BAK} allows the user to retain RTC operation and settings. The V_{BAK} source may be a battery or a large value capacitor (i.e. SuperCap). The RTC operates at very low current, 1 μ A maximum. Even at a fraction of a microamp, the capacitor will discharge to a point where the V_{BAK} voltage drops out of spec, and the RTC stops functioning and backup data is lost. For this situation, the charge on the capacitor can be maintained while the primary power source is applied. The FM31x/32x/4005 employs a trickle charger that provides this feature and is user-enabled.

The figure below shows the trickle charger circuit in simplified form. An internally regulated power supply drives a user-controlled current source attached to the VBAK pin. The VBC bit in register 0Bh enables the trickle charger. There is a Fast Charge mode which is enabled by the FC bit (register 0Bh, bit 5). In this mode the trickle charger current is set to approximately 1 mA, allowing a large backup capacitor to charge more quickly.



When a SuperCap is used as a backup power source, the maximum voltage on the VBAK pin is self-limiting. It is defined by the internally regulated voltage. If a battery is used on VBAK, the maximum VBAK voltage is 3.75V. Internal damage may result if the VBAK voltage is driven above 3.75V with an external charging circuit.

You may notice a 0.1 μ F decoupling capacitor on the V_{BAK} pin. This is recommended especially if the battery is socketed.

CALCULATING CAPACITOR BACKUP TIME

A capacitor (e.g. SuperCap) is an alternative to a battery as a backup power source to keep the RTC running when the main supply (V_{DD}) is powered off. The V_{BAK} voltage is specified to operate down to 2.0V. The initial voltage on the backup capacitor is determined either by an external charging circuit that the customer builds or by the internal trickle charger. The trickle charger voltage self-limits to ~3.3V if V_{DD} is above 3.75V (FM3127x 5V devices). For 3.3V systems (FM31L27x devices), the capacitor will charge up to V_{DD} . Note that the internal trickle charger, typically 80 μ A, will require about 21 minutes of charge time per volt, assuming a 0.1F SuperCap is used.

When main power is off ($V_{DD} < 2.5V$), then the internal power is switched over to V_{BAK} and the trickle charger is turned off. The backup capacitor will have a load of $< 1\mu$ A. The discharge time is related to the load current, capacitance on V_{BAK} , and starting voltage and ending voltage.

From $i = C \, dV/dt$,

we can calculate time, $dt = C \, dV/i$

For systems that use a 0.1F SuperCap (assuming the capacitor is fully charged),

$$dt = 0.1 (3.3-2.0)/1\mu A$$

or 130,000 seconds, which is about 36 hours. A 1F SuperCap will allow a backup time of 15 days.

For designs that implement an external charging circuit, you must take into account this circuit's off current, which may not be negligible.

Setup Example

The following example is a setup procedure that describes the proper sequence for power-up, initialization, and register settings.

1. Apply V_{DD} power.
2. Apply battery to V_{BAK} . If SuperCap is used, the trickle charger may be set ($VBC = 1$) and optionally set the FC bit. If a battery is used, be sure that $VBC=0$. The battery may be installed before applying V_{DD} power, but be aware that the battery will see a $1\mu A$ load even if the oscillator is off.
3. Set the V_{DD} voltage trip point, VTP bit, if you prefer the other higher setting. Default is the lower of the two settings.
4. Enable RTC oscillator (clear /OSCEN bit)
5. Enter RTC calibration mode ($CAL = 1$)
6. Determine error and sign by monitoring 512 Hz output. Use a frequency counter.
7. Write error correction value to CALS and CAL(4:0). Use Calibration Adjustments table on pg 9 in the datasheet.
8. Exit Calibration mode ($CAL = 0$)
9. Set W bit to enable RTC time-day-date
10. Write time-day-date values to registers 02h – 08h.
11. Clear W bit to start RTC with new values
12. Set Watchdog Timeout value WDT(4:0)
13. Issue Restart Watchdog command WR(3:0) = 1010b to ensure the timeout value is loaded
14. Enable Watchdog timer ($WDE = 1$)
15. Normal operation
16. MCU must issue Restart Watchdog WR(3:0) = 1010b at least once within the timeout period

Note: Steps 4 through 11 do not apply to the FM3227x/FM32L27x families.

To read the RTC, you may follow this simple 3-step procedure.

1. Set R bit which takes snapshot of RTC registers (assumes R previously logic 0)
2. Issue Selective Read command (Slave ID 1101b), starting at address 02h, and read seven RTC bytes (02h – 08h)
3. Clear R bit to prepare for next RTC read

RTC REGISTER MAP

Address	D7	D6	D5	D4	D3	D2	D1	D0	Function	Range
18h	Serial Number Byte 7								Serial Number 7	FFh
17h	Serial Number Byte 6								Serial Number 6	FFh
16h	Serial Number Byte 5								Serial Number 5	FFh
15h	Serial Number Byte 4								Serial Number 4	FFh
14h	Serial Number Byte 3								Serial Number 3	FFh
13h	Serial Number Byte 2								Serial Number 2	FFh
12h	Serial Number Byte 1								Serial Number 1	FFh
11h	Serial Number Byte 0								Serial Number 0	FFh
10h	Counter 2 MSB								Event Counter 2 MSB	FFh
0Fh	Counter 2 LSB								Event Counter 2 LSB	FFh
0Eh	Counter 1 MSB								Event Counter 1 MSB	FFh
0Dh	Counter 1 LSB								Event Counter 1 LSB	FFh
0Ch					RC	CC	C2P	C1P	Event Count Control	
0Bh	SNL	-	FC	WP1	WP0	VBC	-	VTP	Companion Control	
0Ah	WDE	-	-	WDT4	WDT3	WDT2	WDT1	WDT0	Watchdog Control	
09h	WTR	POR	LB	-	WR3	WR2	WR1	WR0	Watchdog Restart/Flags	
08h	10 years				years				Years	00-99
07h	0	0	0	10 mo	months				Month	1-12
06h	0	0	10 date		date				Date	1-31
05h	0	0	0	0	0	day			Day	1-7
04h	0	0	10 hours		hours				Hours	0-23
03h	0	10 minutes			minutes				Minutes	0-59
02h	0	10 seconds			seconds				Seconds	0-59
01h	/OSCEN	reserved	CALS	CAL4	CAL3	CAL2	CAL1	CAL0	CAL/Control	
00h	reserved	CF	reserved	reserved	reserved	CAL	W	R	RTC Control	

Pseudo Code Examples

```
#define WRITE_MEM 0xA0
#define READ_MEM  0xA1
#define WRITE_RTC 0xD0
#define READ_RTC  0xD1
```

NOTE: Text in **BLUE** indicates data being sent by the controller. Text in **RED** indicates data being received by the controller.

```

/***** Enable RTC Oscillator *****/
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
           (0x01) // Sets the address pointer to Register 01h
           (0x00) // Write data to 0x00 which clears the /OSCEN bit
STOP // SDA rising edge while SCL is high

/***** Set RTC time/date *****/
// Step #1 Set W bit which allows writes to RTC registers
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
           (0x00) // Sets the address pointer to Register 00h
           (0x02) // Write data to 0x02 which sets the W bit
STOP // SDA rising edge while SCL is high

// Step #2 Write time/date to RTC Registers
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
           (0x02) // Sets the address pointer to Register 02h
           (0x00) // Seconds set to 00
           (0x10) // Minutes set to 10
           (0x14) // Hours set to 14 (2 PM)
           (0x03) // Day set to the third day of the week
           (0x04) // Date set to the fourth day in March
           (0x03) // Month set to March
           (0x08) // Year set to 2008
STOP // SDA rising edge while SCL is high
// RTC does not start to run yet.

// Step #3 Clear W bit to start RTC with the exact time
START // SDA falling edge while SCL is high
WRITE (0xD0) // 0xD0 is the command for a write to the Companion
           (0x00) // Sets the address pointer to Register 00h
           (0x00) // Data=0x00 clears the W bit to start RTC with time defined
                  // in Step #2. The 8th clock of this byte defines the actual
                  // start of the RTC.
STOP // SDA rising edge while SCL is high

/***** Set VTP Voltage Detect Trip Point *****/
// From the factory, VTP bit is cleared to 0. If user wants to set trip point to
// the higher setting, then write VTP bit to 1 in Reg 0Bh control register.
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
           (0x0B) // Sets the address pointer to Register 0Bh
           (0x01) // Write data to 0x01 which sets the VTP bit
STOP // SDA rising edge while SCL is high

```

```

/***** Read RTC Registers *****/
// Step #1 Set R bit which takes snapshot of RTC registers
START
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion
            (0x00) // Sets the address pointer to Register 00h
            (0x01) // Write data to 0x01 which sets the R bit
STOP

// Step #2 Read RTC Registers
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // Selective Read starts with a write command
            (0x02) // Sets the address pointer to Register 02h
START // This is a re-start
READ_RTC (0xD1) // Read command tells chip to start reading from addr 02h
          (0x59) // The chip reads out 59 seconds
          (0x15) // The chip reads out 15 minutes
          (0x11) // The chip reads out 11 hours
          (0x03) // The chip reads out 03 for third day of the week
          (0x04) // The chip reads out 04 fourth day in March
          (0x03) // The chip reads out 03 for March
          (0x08) // The chip reads out 08 for 2008
STOP // SDA rising edge while SCL is high

// Step #3 Clear R bit
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion
            (0x00) // Sets the address pointer to Register 00h
            (0x00) // Data=0x00 clears the R bit to allow RTC read next time
STOP // SDA rising edge while SCL is high

/***** Calibrate RTC *****/
// Step #1 Set CAL bit which turns on the 512Hz sq wave on the CAL/PFO pin. Setting
// the CAL bit also allows writes to NV bits in Reg 01h CAL/Control register.
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
            (0x00) // Sets the address pointer to Register 00h
            (0x04) // Write data to 0x04 which sets the CAL bit
STOP // SDA rising edge while SCL is high

// Step #2 Use a freq counter to accurately measure the 512Hz, use lookup table in
// datasheet to find to that matches your freq measurement, write cal code to Reg 01h
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion/RTC
            (0x01) // Sets the address pointer to Register 01h
            (0x2C) // Data=0x2C for 511.9722Hz and 511.9744Hz
STOP // SDA rising edge while SCL is high
// RTC does not start to run yet.

// Step #3 Clear CAL bit to exit cal mode and turn off 512Hz
START // SDA falling edge while SCL is high
WRITE_RTC (0xD0) // 0xD0 is the command for a write to the Companion
            (0x00) // Sets the address pointer to Register 00h
            (0x00) // Data=0x00 clears the CAL bit
STOP // SDA rising edge while SCL is high

```

```

/***** Configure Event Counters *****/
// Configure polarity bits for rising edge detection on Counter1 and
// falling edge detection on Counter2
START
WRITE_RTC (0xD0)          // 0xD0 is the command for a write to the Companion
                (0x0C)          // Sets the address pointer to Register 0Ch
                (0x01)          // Data=0x01 which sets C2P to 0, C1P to 1
STOP

/***** Read Event Counters *****/
// Step #1 Set RC bit which takes snapshot of both counter registers
START
WRITE_RTC (0xD0)          // 0xD0 is the command for a write to the Companion
                (0x0C)          // Sets the address pointer to Register 0Ch
                (0x09)          // Data=0x09 which sets the RC bit and keeps C1P=1
STOP

// Step #2 Read Event Counters
START                // SDA falling edge while SCL is high
WRITE_RTC (0xD0)          // Selective Read starts with a write command
                (0x0D)          // Sets the address pointer to Register 0Dh
START                // This is a re-start
READ_RTC  (0xD1)          // Read command tells chip to start reading from addr 0Dh
                (0x1A)          // Counter1 reads out LSB 0x1A (decimal 26)
                (0x00)          // Counter1 reads out MSB 0x00
                (0x31)          // Counter2 reads out LSB 0x31 (decimal 49)
                (0x02)          // Counter2 reads out MSB 0x02
STOP                // SDA rising edge while SCL is high

// Step #3 Clear RC bit
START                // SDA falling edge while SCL is high
WRITE_RTC (0xD0)          // 0xD0 is the command for a write to the Companion
                (0x0C)          // Sets the address pointer to Register 00h
                (0x01)          // Data=0x01 clears the RC bit and keeps C1P=1
STOP                // SDA rising edge while SCL is high

```