FM33xx SPI Companion Design Guide

Applies to FM3316 and FM33256 Devices

OVERVIEW

The FM33256 and FM3316 products offer integrated Processor Companion, Real Time Clock (RTC), and FRAM memory. The FM33xx family includes devices that have 256Kb and 16Kb of nonvolatile FRAM memory. Both devices employ an industry standard SPI interface. This interface is used to access the memory, the processor companion, and the RTC. The FM33xx operates over a 2.7V to 3.6V power supply range. Table 1 summarizes the product family and their key features.

Table 1. Summary of Part Types & Features

Part #	Memory	Companion	RTC	Voltage	SPI Freq	Package
FM33256	256Kb	✓	✓	2.7 – 3.6V	16MHz	SOIC 14
FM3316	16Kb	✓	✓			

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. Both FM33xx 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 the SPI interface. Each "device" is accessed using either standard memory op-codes for the FRAM memory or special op-codes to control the companion/RTC. Since the FM33xx is a device that uses the SPI protocol, additional SPI chips (FM25xx) may share the clock and data lines but the controller must provide a separate chip select line for each chip.

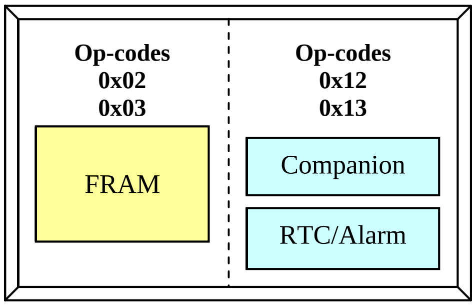

Figure 1. Two Logical Devices Have Unique Op-codes

DESCRIPTION

Figure 2 below is a schematic of a typical application of an FM33xx device. It is shown as an example of external components that could be used, typical values, and their connections to other system devices. In this case, a line-operated (AC powered) system is shown with microcontroller, FM33xx device, and passive components. The micro, in this case, has a dedicated SPI interface. Your micro may not have this port, but an SPI protocol may be written and the GPIO pins may be bit-banged to achieve the same result.

EARLY POWER FAIL

AC line power is rectified and C1 filters the ripple. A 3.3V linear regulator supplies clean DC voltage to the Vdd pins of the FM33xx and microcontroller. The unregulated voltage on C1 is, say, 8V with ripple. The early power fail feature of the FM33xx can be used by connecting a voltage divider (R3, R4) to the PFI input. Once this input goes below 1.5V, the PFO pin which is tied to the controller's NMI pin will drive low, signalling a Non-Maskable Interrupt. The resistor values are chosen based on how early you'd like to warn the microcontroller that the power supply is powering down. The trip voltage V_{TR} that causes the PFO to drive the NMI is determined by this equation:

$$1.5V = V_{TR}\ (R3/(R3+R4))$$

If R3=100K and R4=300K, then V_{TR} is about 6V.

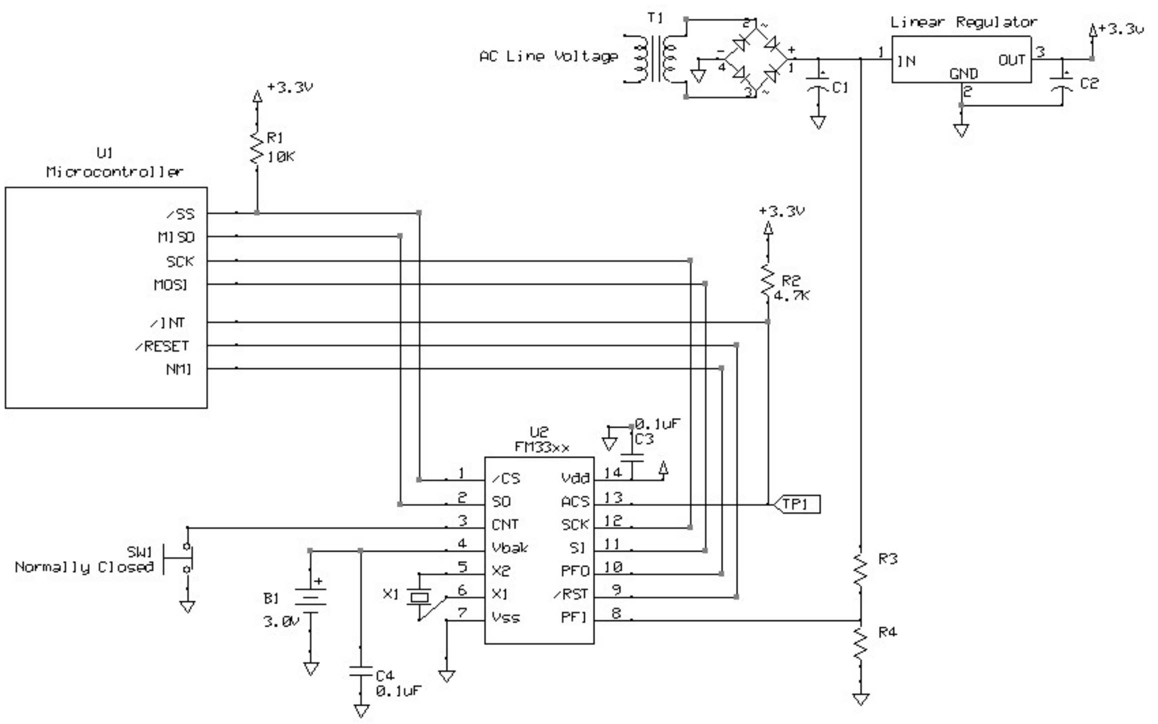

Figure 2. FM33xx Reference Schematic

TAMPER DETECT

A normally closed switch SW1 is shown in this example, and is tied to ground. This can be a switch that is attached to a system case or chassis door. The TMP pin can be set to a polled mode (POLL bit in Register 0Dh) and avoid the DC current from a pullup resistor tied to the battery. There are no external resistors required for this case. The tamper input, in this example, should be programmed to detect a rising edge (CP bit, Register 0Dh).

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 super cap.

From the factory, the RTC oscillator is disabled. To start and configure the RTC, the /OSCEN 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.

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 ACS pin for calibration purposes. The ACS pin in Figure 2 is shown as a test point. It is used to measure the 512Hz frequency in calibration mode. Only a 12.5pF 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 FM33xx uses different op-codes for the companion register space, 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.

RTC CALIBRATION

The schematic shows a test point TP1 attached to ACS pin. It is used to measure the 512Hz frequency in calibration mode. To enter this mode, the CAL bit located in Register 00h must be written to a ‘1’. A frequency counter or other accurate means of measuring frequency should be connected to TP1. Only a crystal may be connected to the X1 and X2 pins – do not probe these pins. When calibrating the RTC oscillator, use the test point TP1. The observed frequency will be slightly higher or lower than 512 Hz. Calibration codes are shown in Table 4 (FM33xx datasheet, pg 10) and are used to adjust the oscillator frequency. For example, if 511.980 Hz is measured, the user must write 10 1001b to Register 01h. After calibration, the RTC will be accurate to ± 5.4 seconds per month at the calibrated temperature. Note that the crystal frequency has a strong temperature dependence the further it deviates from +25C. See application note AN403 – RTC Crystals 6pF vs 12pF for more information.

RTC ALARM

The ACS pin is an open drain output that requires a pullup resistor, R2 in the reference schematic. When the alarm is set using the match bits, ACS pin will drive low when the alarm condition is met.

RTC BACKUP SOURCE

The FM33xx devices may be powered with a supercap or battery as a backup power source. The schematic shows a 3V lithium battery, however a supercap may be used if a battery is not desired. In many cases where a battery is used, a socket is provided to allow replacement by the end user. Some applications are exposed to vibration or mechanical shock which could cause the battery and socket to temporarily lose contact. The Vbak pin is a very high impedance pin which has almost no internal decoupling, therefore a 0.1uF decoupling capacitor is recommended.

SYSTEM RESET

The /RST pin on the FM33xx devices may be tied to the MCU's /RESET pin as a system reset. The /RST pin will power up low and continue to stay low for 30 ms min. to 100 ms max. The pin asserts a low state whenever the V_{DD} supply voltage is < the V_{TP} setting or when the watchdog trips.

POWER CYCLE CONSIDERATIONS

To protect the FRAM from corrupting data during power cycles, we recommend that the MCU's /SS control pin be held inactive as V_{DD} powers up and powers down. In many cases, this may be as simple as a pullup resistor R1 on the MCU's output pin that drives the FM33xx /CS pin. As the system microcontroller powers up, its outputs will tri-state before the power supply reaches sufficient voltage to turn various internal circuits on, thereby allowing the pullup resistor to keep the signal at V_{DD}. Likewise, at powerdown there is a V_{DD} voltage reached that causes the outputs to "let go", again allowing the pullup resistor to do its job. See application note AN302 - SPI Writes & Power Cycling for more information.

Summary Table of Op-Codes

Name	Op-Code	Address	Data	Action
WREN	0000_0110b	-	-	Sets WEL
WRITE	0000_0010b	2-byte	Memory Data in	Writes data to F-RAM array if WEL=1. When /CS goes high, WEL is cleared.
READ	0000_0011b	2-byte	Memory Data out	Reads data from F-RAM array
WRDI	0000_0100b	-	-	Clears WEL
RDSR	0000_0101b	-	St Reg data out	Read WPEN, BP(1:0), WEL bits
WRSR	0000_0001b	-	St Reg data in	Write WPEN and BP(1:0) bits
RDPC	0001_0011b	1-byte	Register Data out	Companion/RTC register read
WRPC	0001_0010b	1-byte	Register Data in	Companion/RTC register write

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 Vdd power, but be aware that the battery will see a 1µA 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 lowest (2.6V) of the four settings.
4. Enable RTC oscillator (set /OSCEN = 0)
5. Enter RTC calibration mode (CAL = 1)
6. Determine error and sign by monitoring 512 Hz output (ACS pin). Use a frequency counter.
7. Write error correction value to CALS and CAL(4:0). Use Calibration Adjustments table on pg 10 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. Normal operation

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 RDPC command (0x13), 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
1Dh	/Match	0	0	10 mo	Alarm months				Alarm Month	01-12
1Ch	/Match	0	10 date		Alarm date				Alarm Date	01-31
1Bh	/Match	0	Alarm 10 hours		Alarm hours				Alarm Hours	00-23
1Ah	/Match	Alarm 10 minutes			Alarm minutes				Alarm Minutes	00-59
19h	/Match	Alarm 10 seconds			Alarm seconds				Alarm Seconds	00-59
18h	SNL	AL/SW	F1	F0	VBC	FC	VTP1	VTP0	Companion Control	
17h	Serial Number Byte 7								Serial Number 7	FFh
16h	Serial Number Byte 6								Serial Number 6	FFh
15h	Serial Number Byte 5								Serial Number 5	FFh
14h	Serial Number Byte 4								Serial Number 4	FFh
13h	Serial Number Byte 3								Serial Number 3	FFh
12h	Serial Number Byte 2								Serial Number 2	FFh
11h	Serial Number Byte 1								Serial Number 1	FFh
10h	Serial Number Byte 0								Serial Number 0	FFh
0Fh	Event Counter Byte 1								Event Counter 1	FFh
0Eh	Event Counter Byte 0								Event Counter 0	FFh
0Dh	NVC	-	-	-	RC	WC	POLL	CP	Event Counter Control	
0Ch	WDE	-	-	WDSET4	WDET3	WDET2	WDET1	WDET0	Watchdog Control	
0Bh	-	-	-	WDST4	WDST3	WDST2	WDST1	WDST0	Watchdog Control	
0Ah	-	-	-	-	WR3	WR2	WR1	WR0	Watchdog Restart	
09h	EWDF	LWDF	POR	LB	-	-	-	-	Watchdog Flags	
08h	10 years				years				Years	00-99
07h	0	0	0	10 mo	months				Month	01-12
06h	0	0	10 date		date				Date	01-31
05h	0	0	0	0	0	day			Day	01-07
04h	0	0	10 hours		hours				Hours	00-23
03h	0	10 minutes			minutes				Minutes	00-59
02h	0	10 seconds			seconds				Seconds	00-59
01h	-	-	CALS	CAL4	CAL3	CAL2	CAL1	CAL0	CAL/Control	
00h	/OSCEN	AF	CF	AEN	reserved	CAL	W	R	RTC/Alarm Control	

Pseudo Code Examples

```
#define WREN      0x06
#define WRITE_MEM 0x02
#define READ_MEM  0x03
#define WRITE_PC  0x12
#define READ_PC   0x13
```

In every case below, the parentheses designate the /CS pin going low "(" and high ")".

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

```
/******************** Enable RTC Oscillator ********************/
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x00          // Sets the starting address to Register 00h
           0x00)         // Write data to 0x00 which clears the /OSCEN bit


/******************** Set RTC time/date ********************/
// Step #1 Set W bit which allows writes to RTC registers
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0xD0 is the command for a write to the Companion/RTC
           0x00          // Sets the starting address to Register 00h
           0x02)         // Write data to 0x02 which sets the W bit

// Step #2 Write time/date to RTC Registers
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x02          // Sets the starting address 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
           0x10          // Month set to March
           0x08)         // Year set to 2008
// RTC does not start to run yet.

// Step #3 Clear W bit to start RTC with the exact time
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x00          // Sets the starting address 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.



/******************** Set VTP Voltage Detect Trip Point ********************/
// From the factory, VTP bits are cleared to 00.  If user wants to set trip point to
// the highest setting, then write VTP bits to 11b in Reg 18h control register.
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x18          // Sets the starting address to Register 0Bh
           0x03)         // Write data to 0x03 which sets both VTP bits
```

```
/******************** Read RTC Registers ********************/
// Step #1 Set R bit which takes snapshot of RTC registers
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x00          // Sets the starting address to Register 00h
           0x01)         // Write data to 0x01 which sets the R bit

// Step #2 Read RTC Registers
READ_PC   (0x13          // Read command tells chip to start reading
           0x02          // Sets the starting address to Register 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

// Step #3 Clear R bit
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x00          // Sets the starting address to Register 00h
           0x00)         // Data=0x00 clears the R bit to allow RTC read next time



/******************** Configure Event Counter ********************/
// Configure polarity bits for rising edge detection on the Counter
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x0D          // Sets the starting address to Register 0Dh
           0x01)         // Data=0x01 which sets CP to 1



/******************** Read Event Counters ********************/
// Step #1 Set RC bit which takes snapshot of both counter registers
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x0D          // Sets the starting address to Register 0Dh
           0x09)         // Data=0x09 which sets the RC bit and keeps CP=1

// Step #2 Read Event Counters
READ_PC  (0x13           // Read command tells chip to start reading
          0x0D           // Sets the starting address to Register 0Dh
          0x1A           // CounterByte0 reads out LSB 0x1A (decimal 26)
          0x00)          // CounterByte1 reads out MSB 0x00

// Step #3 Clear RC bit
WREN      (0x06)         // Sets WEL bit.  WREN must precede WRITE or WRITE_PC opcode.
WRITE_PC  (0x12          // 0x12 is the command for a write to the Companion/RTC
           0x0D          // Sets the starting address to Register 0Dh
           0x01)         // Data=0x01 clears the RC bit and keeps CP=1
```