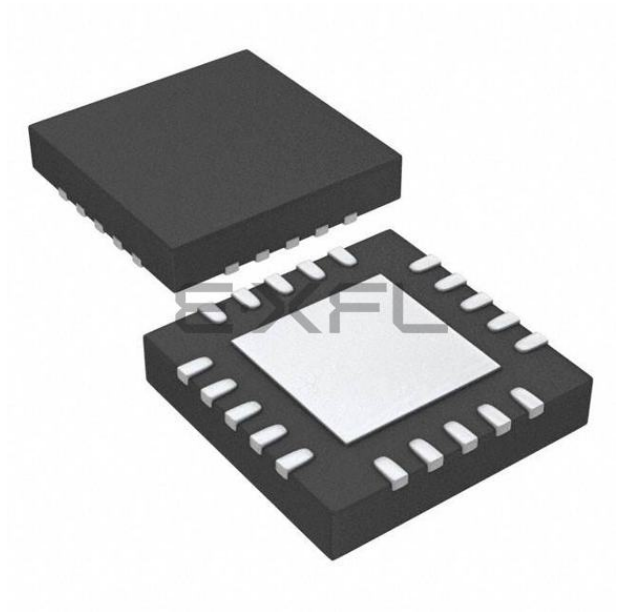


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 16x10b; D/A 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f392-a-gmr

Table 7.15. IDAC Electrical Characteristics

V_{DD} = 3.0 V, –40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), full-scale output current set to 2 mA, unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Static Performance					
Resolution			10		bits
Integral Nonlinearity		—	<±1	±3	LSB
Differential Nonlinearity	0 to +105 °C, Guaranteed Monotonic	—	<±0.5	±1	LSB
	–40 to 0°C	–1.0	<±0.5	1.3	LSB
Output Compliance Range		0	—	V_{DD} – 1.0	V
Offset Error		—	0	—	µA
Full Scale Error	2 mA Full Scale Output Current	–122	0	40	µA
	1 mA Full Scale Output Current	–61	0	20	µA
	0.5 mA Full Scale Output Current	–31	0	10	µA
Full Scale Error Tempco		—	80	—	ppm/°C
V_{DD} Power Supply Rejection Ratio		—	1.05	—	µA/V
Dynamic Performance					
Output Settling Time to 1/2 LSB	IDA0H:L = 0x3FF to 0x000	—	7	—	µs
Startup Time		—	6.5	—	µs
Gain Variation	1 mA Full Scale Output Current	—	0.1	—	%
	0.5 mA Full Scale Output Current	—	0.1	—	%
Power Consumption					
Power Supply Current (V_{DD} supplied to IDAC)	2 mA Full Scale Output Current	—	2065	—	µA
	1 mA Full Scale Output Current	—	1065	—	µA
	0.5 mA Full Scale Output Current	—	565	—	µA

SFR Definition 9.2. ADC0H: ADC0 Data Word MSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBE; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0H[7:0]	**ADC0 Data Word High-Order Bits.** For AD0LJST = 0: Bits 7–2 will read 000000b. Bits 1–0 are the upper 2 bits of the 10-bit ADC0 Data Word. For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word.

SFR Definition 9.3. ADC0L: ADC0 Data Word LSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBD; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0L[7:0]	**ADC0 Data Word Low-Order Bits.** For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit Data Word. For AD0LJST = 1: Bits 7–6 are the lower 2 bits of the 10-bit Data Word. Bits 5–0 will read 000000b.

10. Temperature Sensor (C8051F390/2/4/6/8 and C8051F370/4 Only)

A fully C8051F33x-compatible temperature sensor is included on the C8051F390/2/4/6/8 and C8051F370/4 and accessed via the ADC multiplexer in single-ended configuration. For the self-contained precision temperature sensor, refer to Section 8.

To use the ADC to measure the temperature sensor, the ADC mux channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 10.1. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 12.1. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 7.11 for the slope and offset parameters of the temperature sensor.

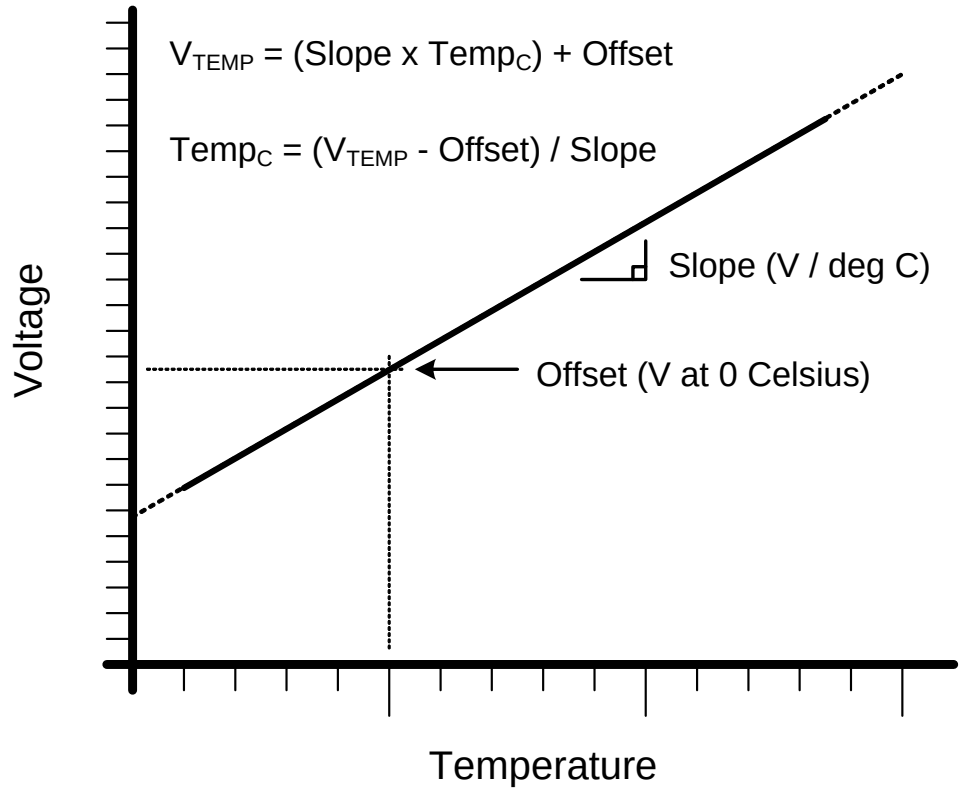


Figure 10.1. Temperature Sensor Transfer Function

SFR Definition 15.3. SP: Stack Pointer

Bit	7	6	5	4	3	2	1	0
Name	SP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	1	1	1

SFR Address = 0x81; SFR Page = All Pages

Bit	Name	Function
7:0	SP[7:0]	**Stack Pointer.** The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 15.4. ACC: Accumulator

Bit	7	6	5	4	3	2	1	0
Name	ACC[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	ACC[7:0]	**Accumulator.** This register is the accumulator for arithmetic operations.

SFR Definition 15.5. B: B Register

Bit	7	6	5	4	3	2	1	0
Name	B[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

SFR Definition 20.10. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Address = 0xE4; SFR Page = All Pages

Table 21.1. Flash Security Summary

Erase page containing Lock Byte—Unlock all pages (if any page is locked)	C2 Device Erase Only	Flash Error Reset	Flash Error Reset
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Read, Write or Erase Reserved Area	Not Permitted	Flash Error Reset	Flash Error Reset

C2 Device Erase—Erases all Flash pages including the page containing the Lock Byte.

Flash Error Reset —Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is '1' after reset).

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
- Locking any Flash page also locks the page containing the Lock Byte.
- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

The 16-bit C8051F39x/37x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

Table 23.1 lists several input values and the associated outputs using the 16-bit C8051F39x/37x CRC algorithm:

Table 23.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

SFR Definition 28.1. SMB0CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB0	INH0	BUSY0	EXTHOLD0	SMB0TOE	SMB0FTE	SMB0CS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0

Bit	Name	Function
7	ENSMB0	**SMBus0 Enable.** This bit enables the SMBus0 interface when set to 1. When enabled, the interface constantly monitors the SDA0 and SCL0 pins.
6	INH0	**SMBus0 Slave Inhibit.** When this bit is set to logic 1, the SMBus0 does not generate an interrupt when slave events occur. This effectively removes the SMBus0 slave from the bus. Master Mode interrupts are not affected.
5	BUSY0	**SMBus0 Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD0	**SMBus0 Setup and Hold Time Extension Enable.** This bit controls the SDA0 setup and hold times according to Table 28.2. 0: SDA0 Extended Setup and Hold Times disabled. 1: SDA0 Extended Setup and Hold Times enabled.
3	SMB0TOE	**SMBus0 SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus0 forces Timer 3 to reload while SCL0 is high and allows Timer 3 to count when SCL0 goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL0 is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus0 communication.
2	SMB0FTE	**SMBus0 Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL0 and SDA0 remain high for more than 10 SMBus clock source periods.
1:0	SMB0CS[1:0]	**SMBus0 Clock Source Selection.** These two bits select the SMBus0 clock source, which is used to generate the SMBus0 bit rate. The selected device should be configured according to Equation 28.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

SFR Definition 28.2. SMB1CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB1	INH1	BUSY1	EXTHOLD1	SMB1TOE	SMB1FTE	SMB1CS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = F

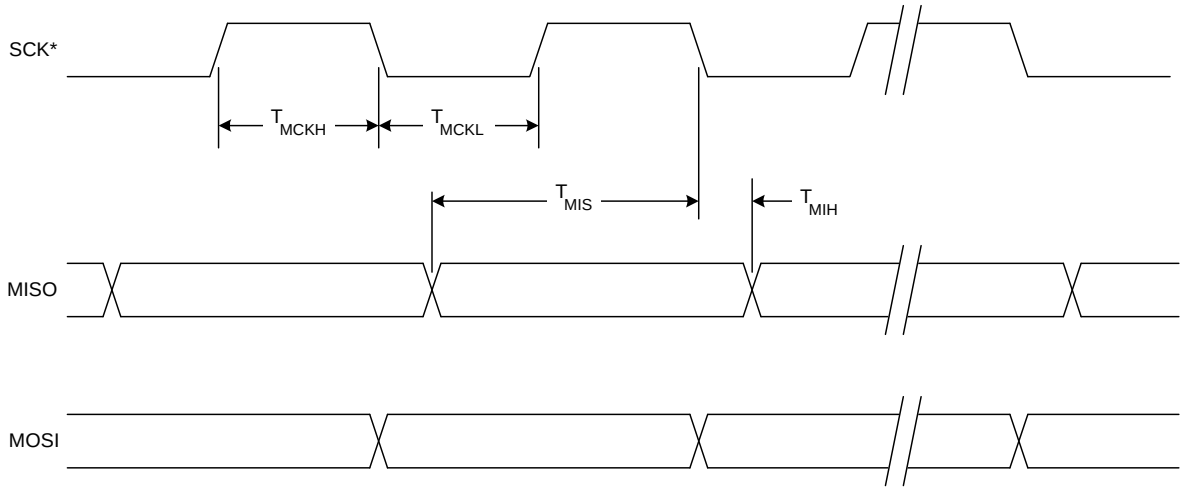
Bit	Name	Function
7	ENSMB1	**SMBus1 Enable.** This bit enables the SMBus1 interface when set to 1. When enabled, the interface constantly monitors the SDA1 and SCL1 pins.
6	INH1	**SMBus1 Slave Inhibit.** When this bit is set to logic 1, the SMBus1 does not generate an interrupt when slave events occur. This effectively removes the SMBus1 slave from the bus. Master Mode interrupts are not affected.
5	BUSY1	**SMBus1 Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD1	**SMBus1 Setup and Hold Time Extension Enable.** This bit controls the SDA1 setup and hold times according to Table 28.2. 0: SDA1 Extended Setup and Hold Times disabled. 1: SDA1 Extended Setup and Hold Times enabled.
3	SMB1TOE	**SMBus1 SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus1 forces Timer 4 to reload while SCL1 is high and allows Timer 4 to count when SCL1 goes low. If Timer 4 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL1 is high. Timer 4 should be programmed to generate interrupts at 25 ms, and the Timer 4 interrupt service routine should reset SMBus1 communication.
2	SMB1FTE	**SMBus1 Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL1 and SDA1 remain high for more than 10 SMBus clock source periods.
1:0	SMB1CS[1:0]	**SMBus1 Clock Source Selection.** These two bits select the SMBus1 clock source, which is used to generate the SMBus1 bit rate. The selected device should be configured according to Equation 28.1. 00: Timer 0 Overflow 01: Timer 5 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

SFR Definition 30.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

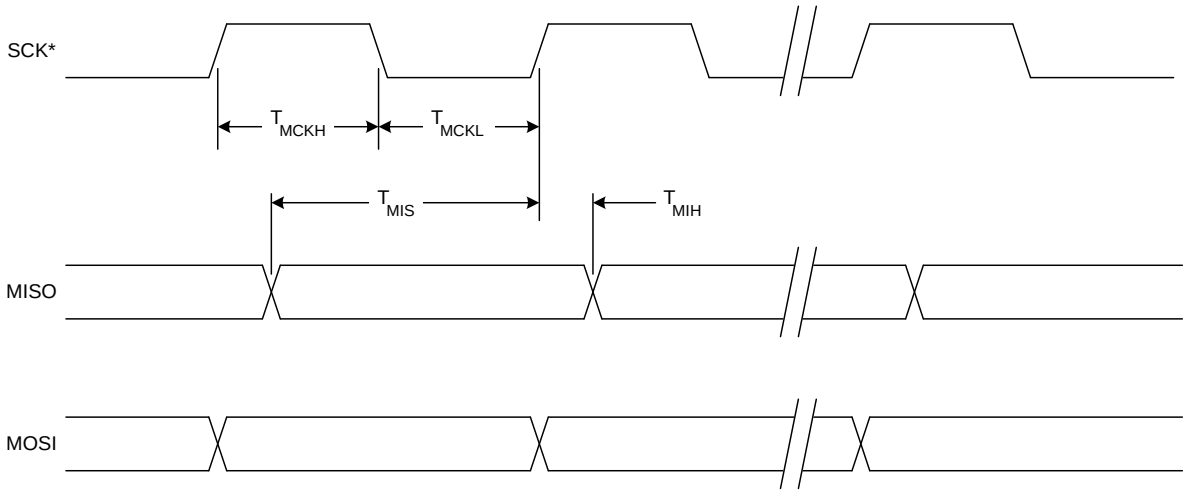
SFR Address = 0xA3; SFR Page = All Pages

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.



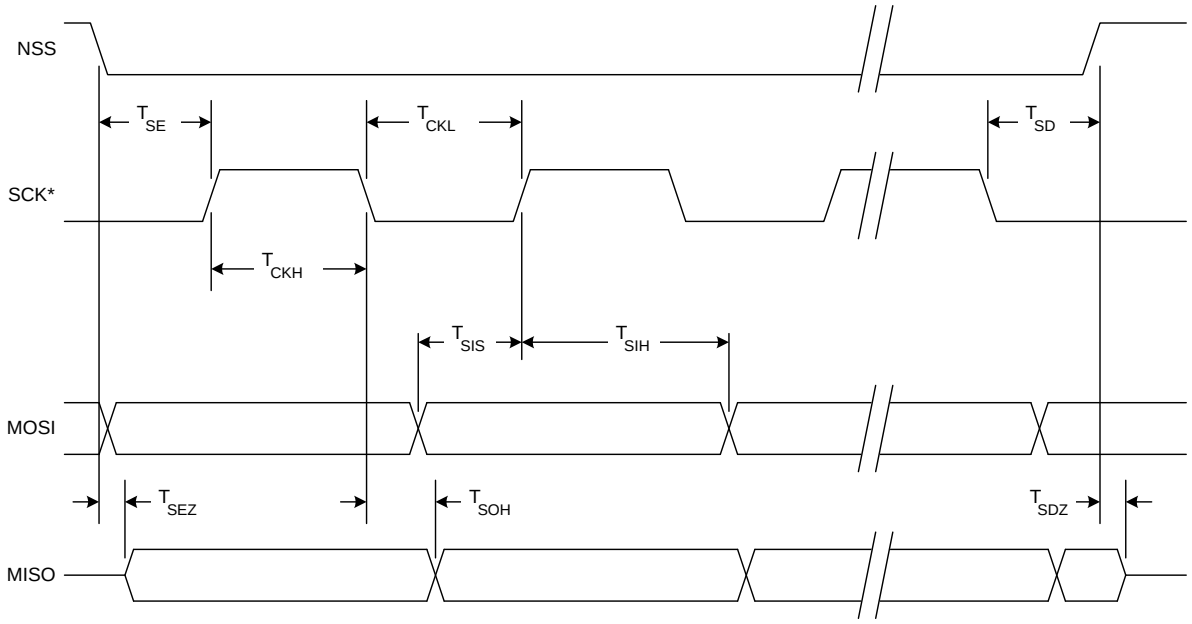
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.8. SPI Master Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.9. SPI Master Timing (CKPHA = 1)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.10. SPI Slave Timing (CKPHA = 0)

31.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “20.3. External Interrupts INT0 and INT1” on page 128 for details on the external input signals INT0 and INT1).

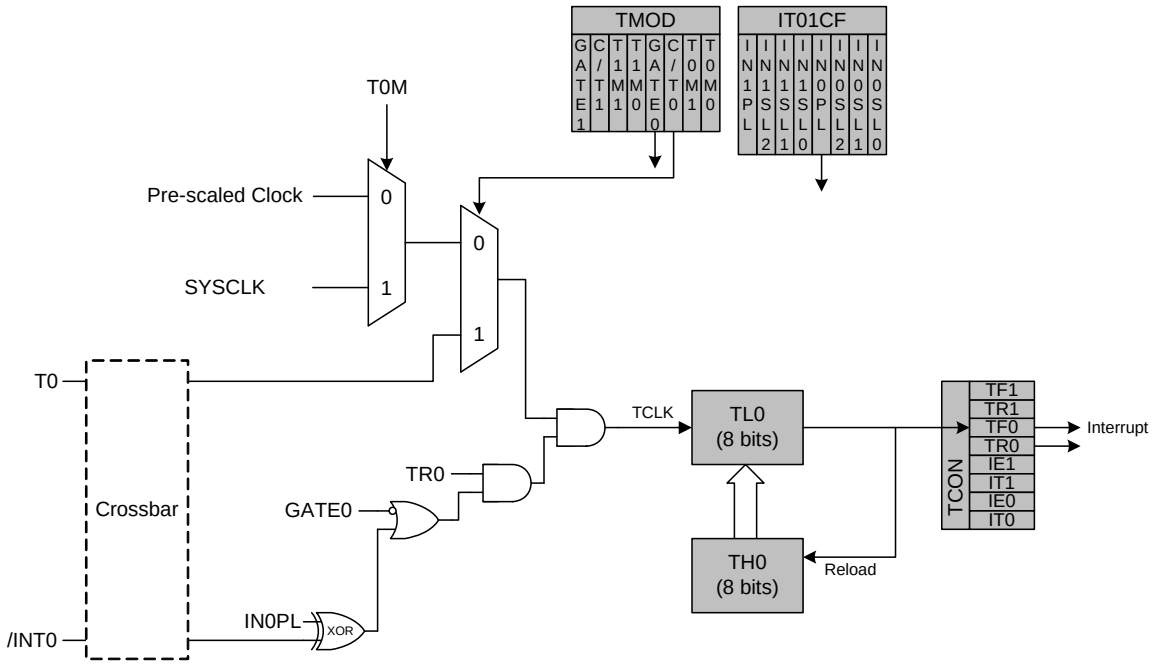


Figure 31.2. T0 Mode 2 Block Diagram

31.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 31.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

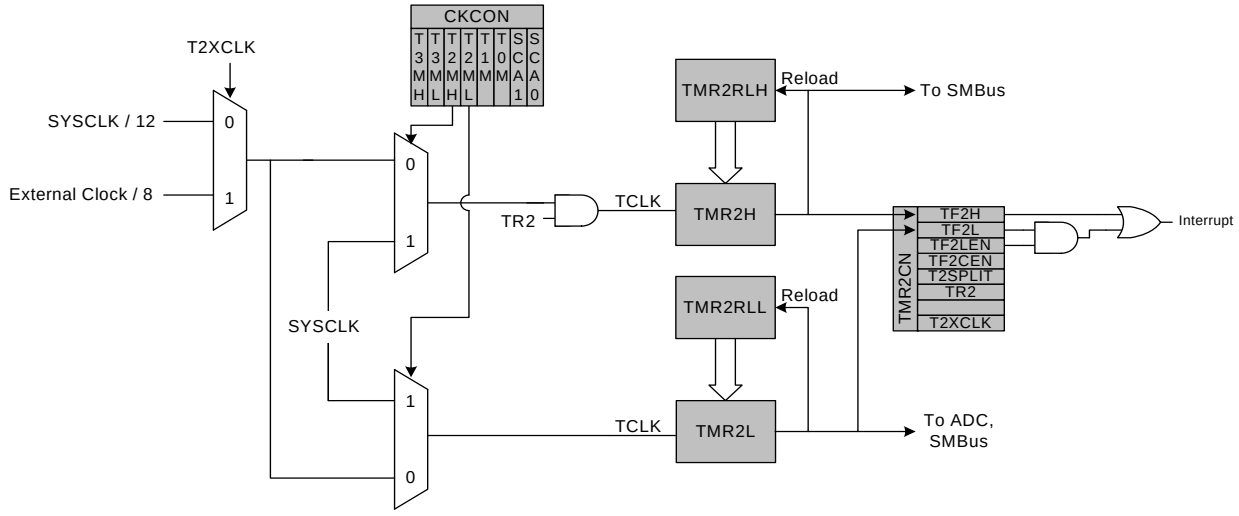


Figure 31.5. Timer 2 8-Bit Mode Block Diagram

SFR Definition 31.13. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD; SFR Page = 0

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

SFR Definition 31.25. TMR5RLL: Timer 5 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCA; SFR Page = F

Bit	Name	Function
7:0	TMR5RLL[7:0]	**Timer 5 Reload Register Low Byte.** TMR5RLL holds the low byte of the reload value for Timer 5.

SFR Definition 31.26. TMR5RLH: Timer 5 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCB; SFR Page = F

Bit	Name	Function
7:0	TMR5RLH[7:0]	**Timer 5 Reload Register High Byte.** TMR5RLH holds the high byte of the reload value for Timer 5.

SFR Definition 31.27. TMR5L: Timer 5 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC; SFR Page = F

Bit	Name	Function
7:0	TMR5L[7:0]	**Timer 5 Low Byte.** In 16-bit mode, the TMR5L register contains the low byte of the 16-bit Timer 5. In 8-bit mode, TMR5L contains the 8-bit low byte timer value.

32.3.5.2. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an "Auto-Reload" Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module's capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 32.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module's auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 32.2, where N is the number of bits in the PWM cycle.

Important Note About PCA0CPHn and PCA0CPLn Registers: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(2^N - PCA0CPn)}{2^N}$$

Equation 32.3. 9, 10, and 11-Bit PWM Duty Cycle

A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

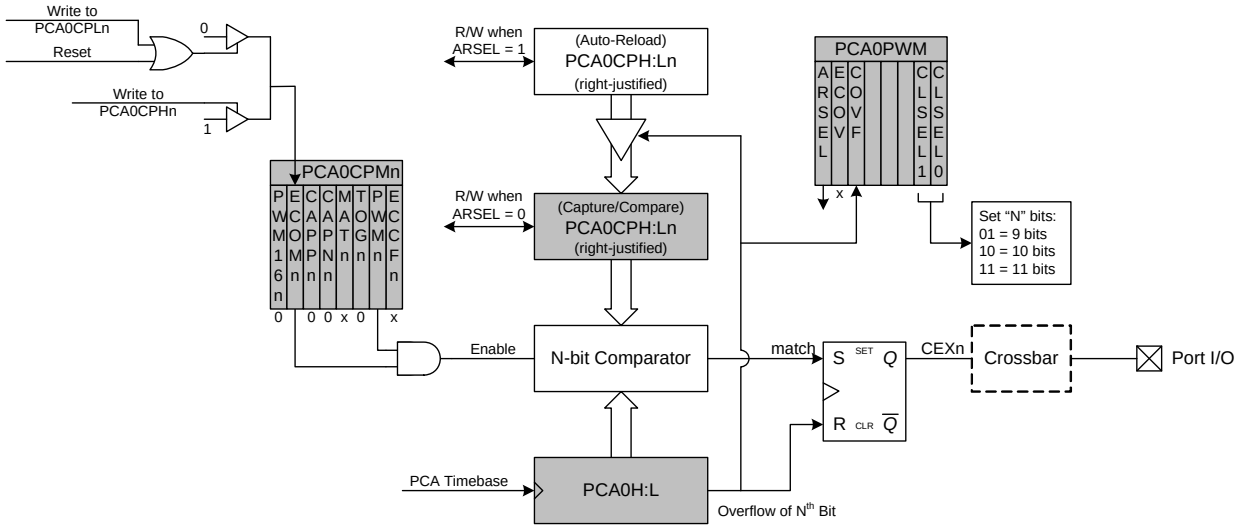


Figure 32.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

33.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (RST) and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 33.1.

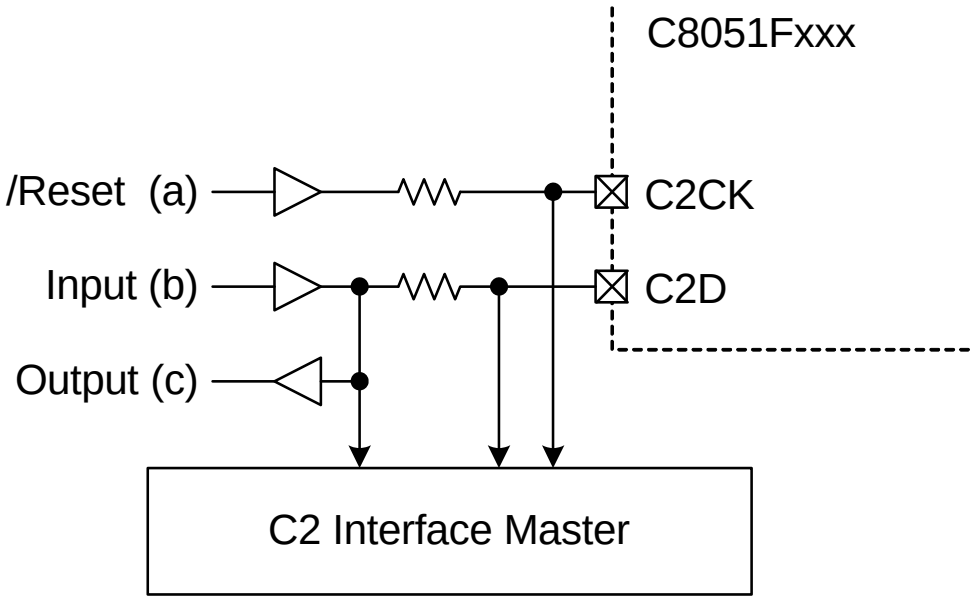


Figure 33.1. Typical C2 Pin Sharing

The configuration in Figure 33.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.