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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	32
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f230-gq

List of Figures and Tables

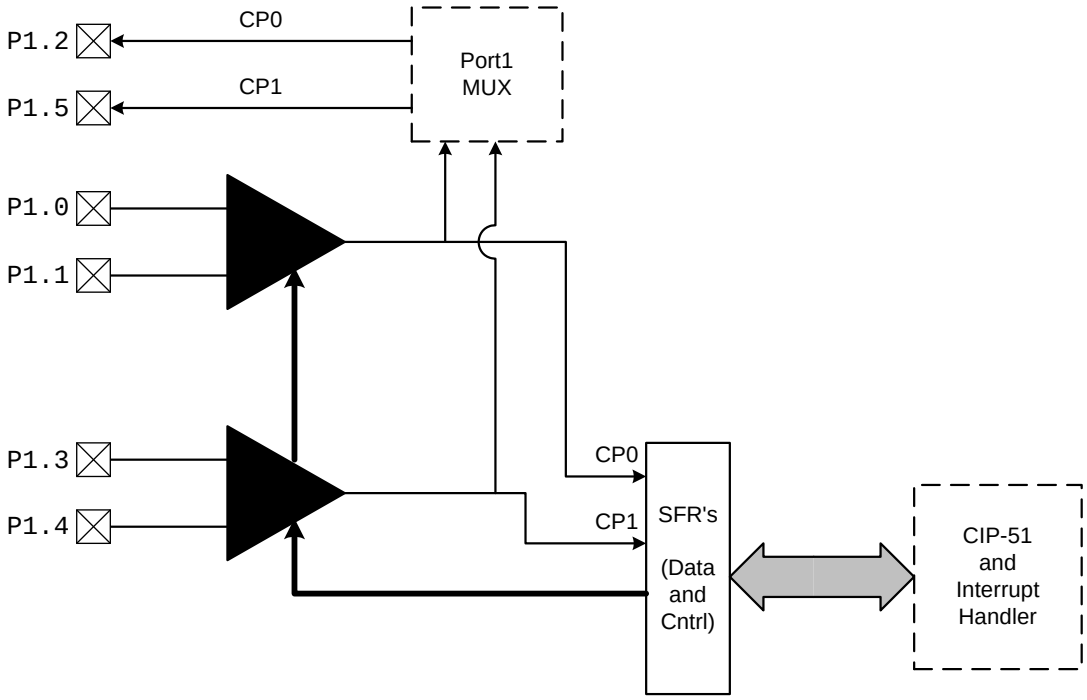
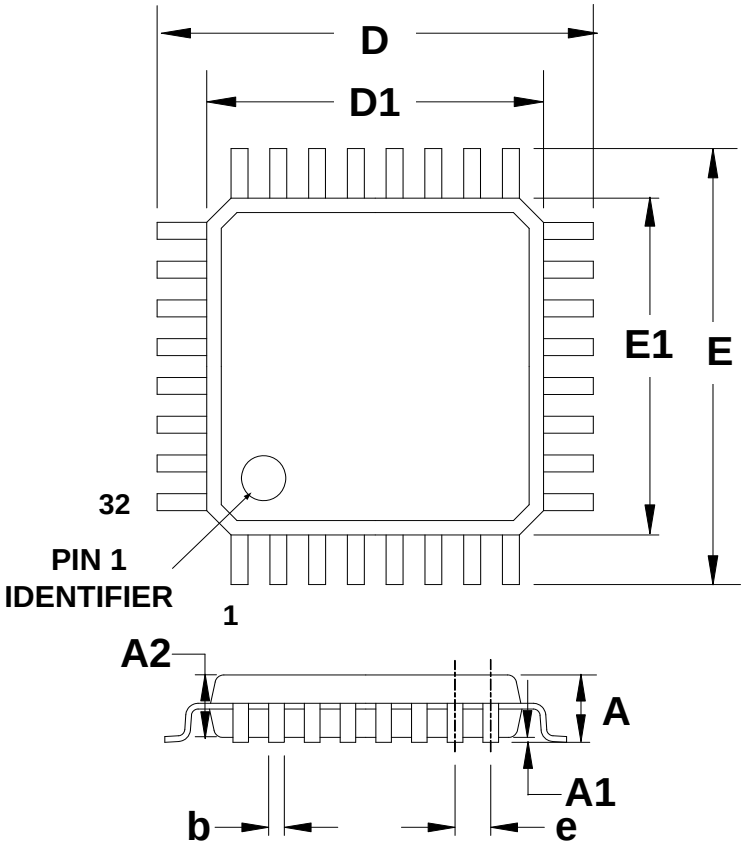


Figure 1.11. Comparator Diagram

4. Pinout and Package Definitions

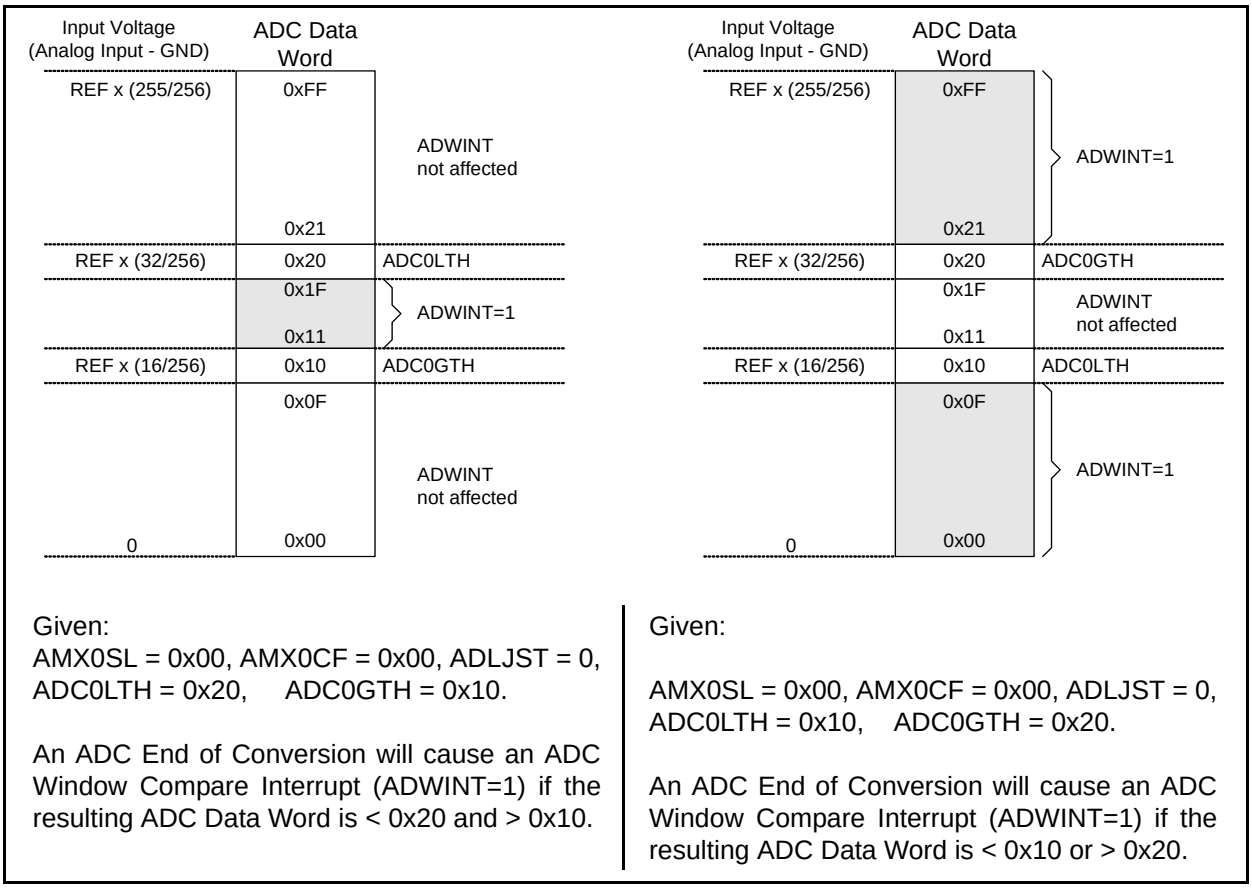
Table 4.1. Pin Definitions

Name	'F206, F220, 226, 230, 236	'F221, 231	Type	Description
	48-Pin	32-Pin		
V_{DD}	11,31	8		Digital Voltage Supply.
GND	5,6,8, 13,32	9		Ground. (**Note:** Pins 5,6, and 8 on the 48-pin package are not connected (NC), but it is recommended that they be connected to ground.)
MONEN	12		D In	Monitor Enable (on 48 pin package ONLY). Enables reset voltage monitor function when pulled high (logic "1").
TCK	25	17	D In	JTAG Test Clock with internal pull-up.
TMS	26	18	D In	JTAG Test-Mode Select with internal pull-up.
TDI	28	20	D In	JTAG Test Data Input with internal pull-up. TDI is latched on a rising edge of TCK.
TDO	27	19	D Out	JTAG Test Data Output. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver.
XTAL1	9	6	A In	Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock.
XTAL2	10	7	A Out	Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator.
$\overline{RST}$	14	10	D I/O	Chip Reset. Open-drain output of internal Voltage Supply monitor. Is driven low when V_{DD} is < 2.7V and MONEN=1, or when a '1'is written to PORSF. An external source can force a system reset by driving this pin low.
VREF	7	5	A I/O	Voltage Reference. When configured as an input, this pin is the voltage reference for the ADC. Otherwise, V_{DD} will be the reference. NOTE: this pin is Not Connected (NC) on 'F230/1/6.
CP0+	4	4	A In	Comparator 0 Non-Inverting Input.
CP0-	3	3	A In	Comparator 0 Inverting Input.
CP0	2	2	D Out	Comparator 0 Output
CP1+	1	1	A In	Comparator 1 Non-Inverting Input.
CP1-	48	32	A In	Comparator 1 Inverting Input.
CP1	47	31	D Out	Comparator 1 Output
P0.0/TX	40	28	D I/O A In	Port0 Bit0. (See the Port I/O Sub-System section for complete description).
P0.1/RX	39	27	D I/O A In	Port0 Bit1. (See the Port I/O Sub-System section for complete description).
P0.2/INT0	38	26	D I/O A In	Port0 Bit2. (See the Port I/O Sub-System section for complete description).



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
D	-	9.00	-
D1	-	7.00	-
e	-	0.80	-
E	-	9.00	-
E1	-	7.00	-

Figure 4.4. LQFP-32 Package Drawing



Given:
AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 0,
ADC0LTH = 0x20, ADC0GTH = 0x10.

An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x20 and > 0x10.

Given:

AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 0,
ADC0LTH = 0x10, ADC0GTH = 0x20.

An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x10 or > 0x20.

Figure 5.3. 8-Bit ADC Window Interrupt Examples

SFR Definition 6.2. ADC0CF: ADC Configuration (‘F220/1/6 and ‘F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	01100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7–5: ADCSC2–0: ADC SAR Conversion Clock Period Bits
000: SAR Conversion Clock = 1 System Clock
001: SAR Conversion Clock = 2 System Clocks
010: SAR Conversion Clock = 4 System Clocks
011: SAR Conversion Clock = 8 System Clocks
1xx: SAR Conversion Clock = 16 Systems Clocks
NOTE: SAR conversion clock should be less than or equal to 2MHz.

Bits4–3: UNUSED. Read = 00b; Write = don't care

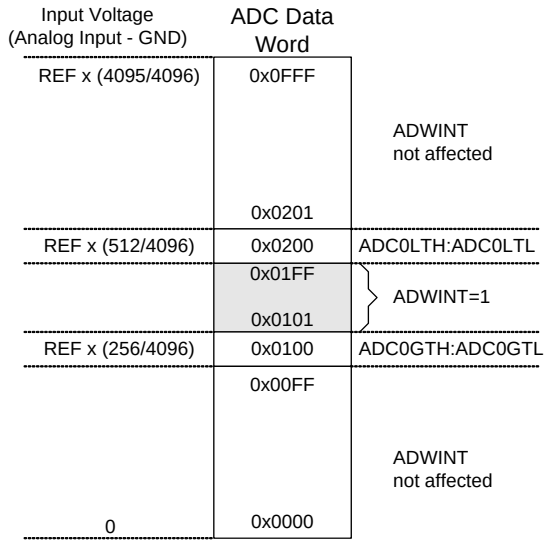
Bits2–0: AMPGN2–0: ADC Internal Amplifier Gain
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
1x: Gain = 0.5

SFR Definition 6.9. ADC0LTL: ADC Less-Than Data Low Byte (C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC4

Bits7–0:
These bits are the low byte of the ADC Less-Than Data Word.

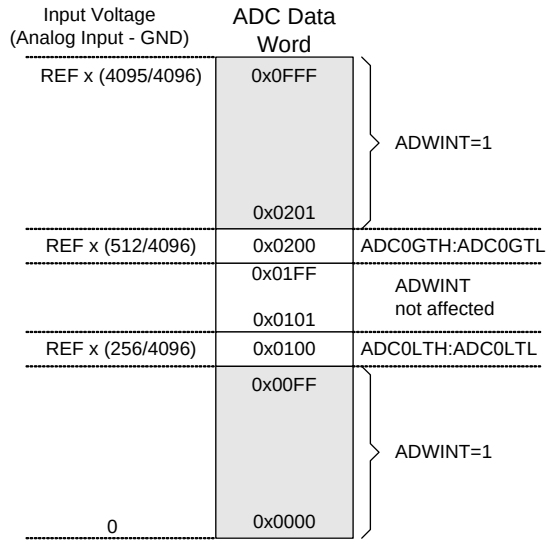
Definition:
ADC Less-Than Data Word = ADC0LTH:ADC0LTL



Given:

AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 0, ADC0LTH:ADC0LTL = 0x0200, ADC0GTH:ADC0GTL = 0x0100.

An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x0200 and > 0x0100.



Given:

AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 0, ADC0LTH:ADC0LTL = 0x0100, ADC0GTH:ADC0GTL = 0x0200.

An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x0100 or > 0x0200.

Figure 6.3. 12-Bit ADC Window Interrupt Examples, Right Justified Data

9.2.5. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP, 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

The MCU also has built-in hardware for a stack record. The stack record is a 32-bit shift register, where each Push or increment SP pushes one record bit onto the register, and each Call pushes two record bits onto the register. (A Pop or decrement SP pops one record bit, and a Return pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the 32-bit shift register, and can notify the emulator software even with the MCU running full-speed debug.

Table 9.3. Special Function Registers

SFR's are listed in alphabetical order.

Address	Register	Description	Page No.
0xE0	ACC	Accumulator	73
0xBC	ADC0CF	ADC Configuration	35
0xE8	ADC0CN	ADC Control	36
0xC5	ADC0GTH[1]	ADC Greater-Than Data Word (High Byte)	37
0xC4	ADC0GTL[4]	ADC Greater-Than Data Word (Low Byte)	46
0xBF	ADC0H[1]	ADC Data Word (High Byte)	37
0xBE	ADC0L[4]	ADC Data Word (Low Byte)	45
0xC7	ADC0LTH[1]	ADC Less-Than Data Word (High Byte)	46
0xCE	ADC0LTL[4]	ADC Less-Than Data Word (Low Byte)	47
0xBB	AMX0SL	ADC MUX Channel Selection	34
0xF0	B	B Register	73
0x8E	CKCON	Clock Control	131
0x9E	CPT0CN	Comparator 0 Control	55
0x9F	CPT1CN	Comparator 1 Control	56
0x83	DPH	Data Pointer (High Byte)	71
0x82	DPL	Data Pointer (Low Byte)	71
0xE6	EIE1	Extended Interrupt Enable 1	79
0xE7	EIE2	Extended Interrupt Enable 2	80
0xF6	EIP1	External Interrupt Priority 1	81
0xF7	EIP2	External Interrupt Priority 2	82
0xAF	EMI0CN[3]	External Memory Interface Control	90
0xB7	FLACL	Flash Memory Read Limit	89
0xB6	FLSCL	Flash Memory Timing Prescaler	89
0xA8	IE	Interrupt Enable	77
0xB8	IP	Interrupt Priority Control	78
0xB2	OSCICN	Internal Oscillator Control	98
0xB1	OSCXCN	External Oscillator Control	99
0x80	P0	Port 0 Latch	105
0x90	P1	Port 1 Latch	106
0xA0	P2	Port 2 Latch	107
0xB0	P3	Port 3 Latch	108
0xF1	P0MODE	Port0 Digital/Analog Output Mode	106
0xF2	P1MODE	Port1 Digital/Analog Output Mode	107
0xF3	P2MODE	Port2 Digital/Analog Output Mode	108
0xF4	P3MODE[2]	Port3 Digital/Analog Output Mode	109
0x87	PCON	Power Control	84

10. Flash Memory

This MCU includes 8 k + 128 bytes of on-chip, re-programmable Flash memory for program code and non-volatile data storage. The Flash memory can be programmed in-system, a single byte at a time, through the JTAG interface or by software using the MOVX instruction. Once cleared to 0, a Flash bit must be erased to set it back to 1. The bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution. Data polling to determine the end of the write/erase operation is not required. The Flash memory is designed to withstand at least 20,000 write/erase cycles. Refer to Table 10.1 for the electrical characteristics of the Flash memory.

10.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the JTAG interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the JTAG commands to program Flash memory, see Section 18.1.

The Flash memory can be programmed by software using the MOVX instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, flash write operations must be enabled by setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1. Writing to Flash remains enabled until the PSWE bit is cleared by software.

To ensure the contents of the Flash contents, it is strongly recommended that the on-chip V_{DD} monitor be enabled (by tieing the MONEN pin 'high') in any application that writes and/or erases Flash memory from software.

Writes to Flash memory can clear bits but cannot set them. Only an erase operation can set bits in Flash. The byte location to be programmed must be erased before a new value can be written. The 8kbyte Flash memory is organized in 512-byte sectors. The erase operation applies to an entire sector (setting all bytes in the sector to 0xFF). Setting the PSEE Program Store Erase Enable bit (PSCTL.1) and PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 and then using the MOVX command to write a data byte to any byte location within the sector will erase an entire 512-byte sector. The data byte written can be of any value because it is not actually written to the Flash. Flash erasure remains enabled until the PSEE bit is cleared by software. The following sequence illustrates the algorithm for programming the Flash memory by software:

1. Disable interrupts.
2. Enable Flash Memory write/erase in FLSCL Register using FLASCL bits.
3. Set PSEE (PSCTL.1) to enable Flash sector erase.
4. Set PSWE (PSCTL.0) to enable Flash writes.
5. Use MOVX to write a data byte to any location within the 512-byte sector to be erased.
6. Clear PSEE to disable Flash sector erase.
7. Use MOVX to write a data byte to the desired byte location within the erased 512-byte sector. Repeat until finished. (Any number of bytes can be written from a single byte to and entire sector.)
8. Clear the PSWE bit to disable Flash writes.

Write/Erase timing is automatically controlled by hardware based on the prescaler value held in the Flash Memory Timing Prescaler register (FLSCL). The 4-bit prescaler value FLASCL determines the time interval for write/erase operations. The FLASCL value required for a given system clock is shown in SFR Definition 10.2, along with the formula used to derive the FLASCL values. When FLASCL is set to 1111b, the

SFR Definition 12.2. RSTSRC: Reset Source

	R	R/W	R/W	R	R	R/W	R	Reset Value
-		C0RSEF	SWRSEF	WDTRSF	MCDRSF	PORSF	PINRSF	xxxxxxxx
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xEF

(Note: Do not use read-modify-write operations on this register.)

Bit7: RESERVED.
Bit6: Not Used. Read only 0b.
Bit5: C0RSEF: Comparator 0 Reset Enable and Flag
Write
0: Comparator 0 is not a reset source
1: Comparator 0 is a reset source (active low)
Read
Note: The value read from C0RSEF is not defined if Comparator 0 has not been enabled as a reset source.
0: Source of prior reset was not from Comparator 0
1: Source of prior reset was from Comparator 0
Bit4: SWRSF: Software Reset Force and Flag
Write
0: No Effect
1: Forces an internal reset. $\overline{\text{RST}}$ pin is not affected.
Read
0: Prior reset source was not from write to the SWRSF bit.
1: Prior reset source was from write to the SWRSF bit.
Bit3: WDTRSF: Watchdog Timer Reset Flag (Read only)
0: Source of prior reset was not from WDT timeout.
1: Source of prior reset was from WDT timeout.
Bit2: MCDRSF: Missing Clock Detector Flag (Read only)
0: Source of prior reset was not from Missing Clock Detector timeout.
1: Source of prior reset was from Missing Clock Detector timeout.
Bit1: PORSF: Power-On Reset Force and Flag
Write
0: No effect
1: Forces a Power-On Reset. $\overline{\text{RST}}$ is driven low.
Read
0: Source of prior reset was not from POR.
1: Source of prior reset was from POR.
Bit0: PINRSF: HW Pin Reset Flag
0: Source of prior reset was not from $\overline{\text{RST}}$ pin.
1: Source of prior reset was from $\overline{\text{RST}}$ pin.

15.2. Operation

Only a SPI master device can initiate a data transfer. The SPI is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.1). Writing a byte of data to the SPI data register (SPI0DAT) when in Master Mode starts a data transfer. The SPI master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. The SPI master can be configured to shift in/out from one to eight bits in a transfer operation in order to accommodate slave devices with different word lengths. The SPIFRS bits in the SPI Configuration Register (SPI0CFG.[2:0]) are used to select the number of bits to shift in/out in a transfer operation.

While the SPI master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. The data byte received from the slave replaces the data in the master's data register. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data transfer in both directions is synchronized with the serial clock generated by the master. Figure 15.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

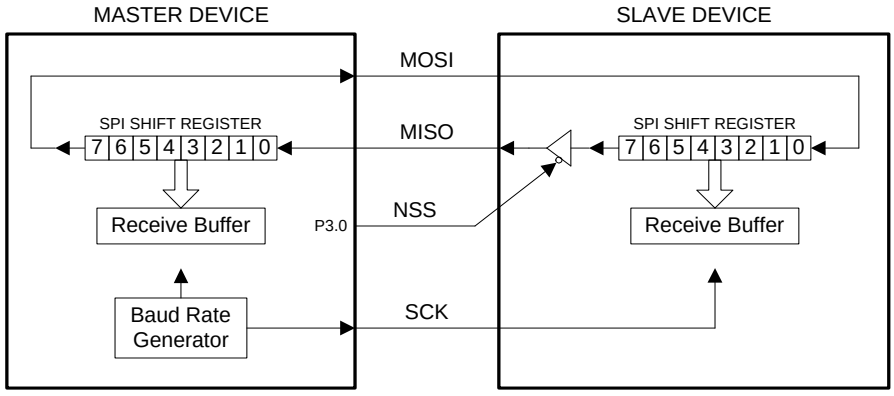


Figure 15.3. Full Duplex Operation

The SPI data register is double buffered on reads, but not on a write. If a write to SPI0DAT is attempted during a data transfer, the WCOL flag (SPI0CN.6) will be set to logic 1 and the write is ignored. The current data transfer will continue uninterrupted. A read of the SPI data register by the system controller actually reads the receive buffer. If the receive buffer still holds unread data from a previous transfer when the last bit of the current transfer is shifted into the SPI shift register, a receive overrun occurs and the RXOVRN flag (SPI0CN.4) is set to logic 1. The new data is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte causing the overrun is lost.

When the SPI is enabled and not configured as a master, it will operate as an SPI slave. Another SPI device acting as a master will initiate a transfer by driving the NSS signal low. The master then shifts data out of the shift register on the MOSI pin using the its serial clock. The SPIF flag is set to logic 1 at the end of a data transfer (when the NSS signal goes high). The slave can load its shift register for the next data transfer by writing to the SPI data register. The slave must make the write to the data register at least one SPI serial clock cycle before the master starts the next transmission. Otherwise, the byte of data already in the slave's shift register will be transferred.

Multiple masters may reside on the same bus. A Mode Fault flag (MODF, SPI0CN.5) is set to logic 1 when the SPI is configured as a master (MSTEN = 1) and its slave select signal NSS is pulled low. When the Mode Fault flag is set, the MSTEN and SPIEN bits of the SPI control register are cleared by hardware, thereby placing the SPI module in an "off-line" state. In a multiple-master environment, the system controller should check the state of the SLVSEL flag (SPI0CN.2) to ensure the bus is free before setting the MSTEN bit and initiating a data transfer.

15.2. Serial Clock Timing

As shown in Figure 15.4, four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.7) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.6) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. Note: the SPI should be disabled (by clearing the SPIEN bit, SPI0CN.0) while changing the clock phase and polarity.

The SPI Clock Rate Register (SPI0CKR) as shown in SFR Definition 15.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode.

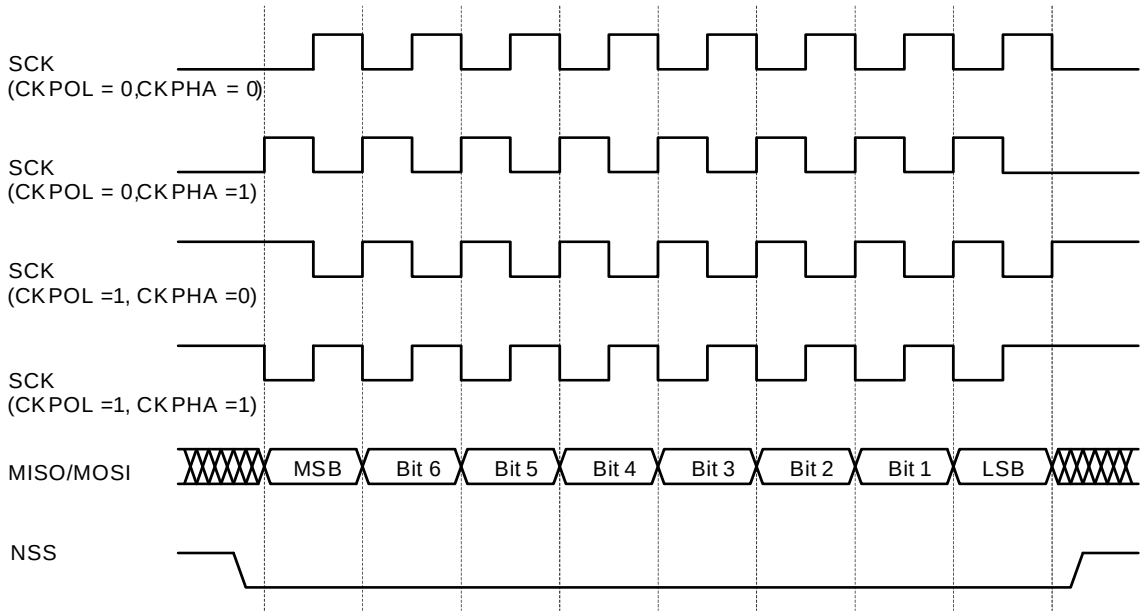


Figure 15.4. Full Duplex Operation

15.3. SPI Special Function Registers

The SPI is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI Bus are described in the following section.

SFR Definition 15.3. SPI0CKR: SPI Clock Rate Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7–0: SCR7–SCR0: SPI Clock Rate
These bits determine the frequency of the SCK output when the SPI module is configured for master mode operation. The SCK clock frequency is a divided down version of the system clock, and is given in the following equations:

fSCK = 0.5 x fSYSCLK / (SPI0CKR + 1), for 0 ≤ SPI0CKR ≤ 255,

SFR Definition 15.4. SPI0DAT: SPI Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9B

Bits7–0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI data. Writing data to SPI0DAT places the data immediately into the shift register and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

Table 16.2. Oscillator Frequencies for Standard Baud Rates (Continued)

Oscillator Frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate**
7.3728	64	0xFC	115200
5.5296	48	0xFD	115200
3.6864	32	0xFE	115200
1.8432	16	0xFF	115200
24.576	320	0xEC	76800
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	848	0xCB	28800 (28921)
24.0	833	0xCC	28800 (28846)
23.592	819	0xCD	28800 (28911)
22.1184	768	0xD0	28800
18.432	640	0xD8	28800
16.5888	576	0xDC	28800
14.7456	512	0xE0	28800
12.9024	448	0xE4	28800
11.0592	348	0xE8	28800
9.216	320	0xEC	28800
7.3728	256	0xF0	28800
5.5296	192	0xF4	28800
3.6864	128	0xF8	28800
1.8432	64	0xFC	28800

SFR Definition 16.1. SBUF: Serial (UART) Data Buffer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x99

Bits7–0: SBUF.[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF is what initiates the transmission. When data is moved from SBUF, it comes from the receive buffer.

TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
X = Don't Care			

Setting TR0 does not reset the timer register. The timer register should be initialized to the desired value before enabling the timer.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0.

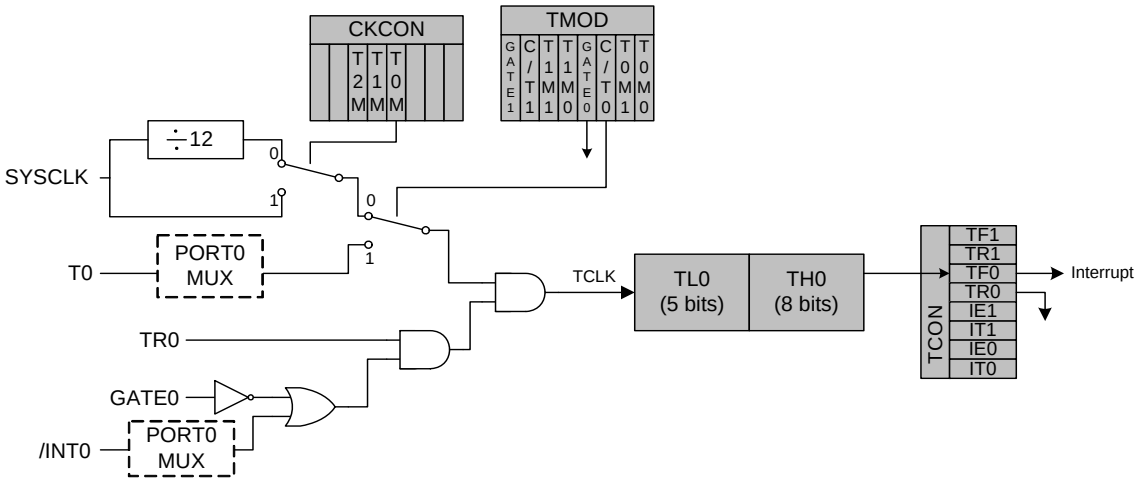


Figure 17.1. T0 Mode 0 Block Diagram

17.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

17.1.3. 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. The 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 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If 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.

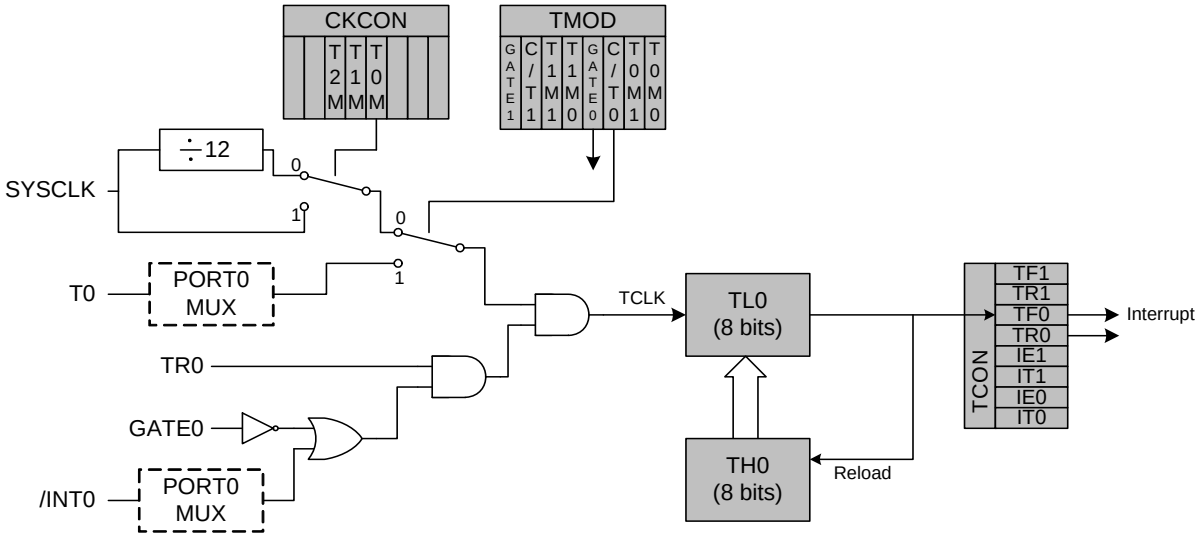


Figure 17.2. T0 Mode 2 Block Diagram

SFR Definition 17.3. CKCON: Clock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	T2M	T1M	T0M	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8E

Bits7–6: UNUSED. Read = 00b, Write = don't care.

Bit5: T2M: Timer 2 Clock Select.
This bit controls the division of the system clock supplied to Timer 2. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T2 = 1).
0: Timer 2 uses the system clock divided by 12.
1: Timer 2 uses the system clock.

Bit4: T1M: Timer 1 Clock Select.
This bit controls the division of the system clock supplied to Timer 1.
0: Timer 1 uses the system clock divided by 12.
1: Timer 1 uses the system clock.

Bit3: T0M: Timer 0 Clock Select.
This bit controls the division of the system clock supplied to Counter/Timer 0.
0: Counter/Timer uses the system clock divided by 12.
1: Counter/Timer uses the system clock.

Bits2–0: UNUSED. Read = 000b, Write = don't care.

17.2.1. Mode 0: 16-bit Counter/Timer with Capture

In this mode, Timer 2 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T2EX input pin causes the 16-bit value in Timer 2 (TH2, TL2) to be loaded into the capture registers (RCAP2H, RCAP2L).

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the external T2 input pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/T2 bit (T2CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set to logic 1, a high-to-low transition at the T2 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF2 timer overflow flag (T2CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL2 (T2CON.0) and the Timer 2 Run Control bit TR2 (T2CON.2) to logic 1. The Timer 2 External Enable EXEN2 (T2CON.3) must also be set to logic 1 to enable a capture. If EXEN2 is cleared, transitions on T2EX will be ignored.

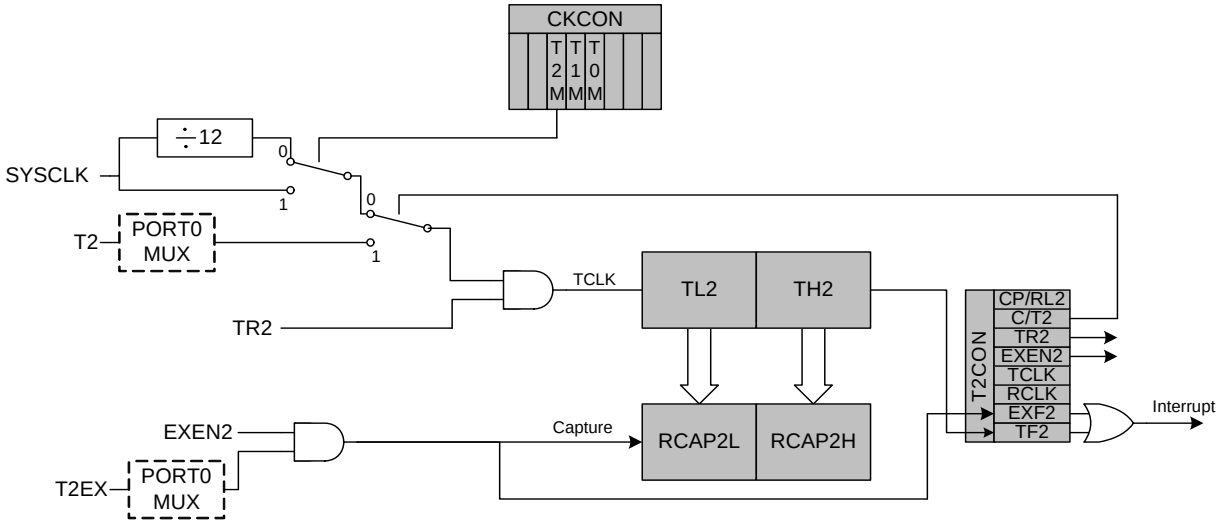


Figure 17.4. T2 Mode 0 Block Diagram

JTAG Register Definition 18.2. FLASHCON: JTAG Flash Control

WRMD3	WRMD2	WRMD1	WRMD0	RDMD3	RDMD2	RDMD1	RDMD0	Reset Value
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bits7–4: WRMD3–0: Write Mode Select Bits.
The Write Mode Select Bits control how the interface logic responds to writes to the FLASHDAT Register per the following values:
- 0000: A FLASHDAT write replaces the data in the FLASHDAT register, but is otherwise ignored.
- 0001: A FLASHDAT write initiates a write of FLASHDAT into the memory address selected by the FLASHADR register. FLASHADR is incremented by one when complete.
- 0010: A FLASHDAT write initiates an erasure (sets all bytes to 0xFF) of the Flash page containing the address in FLASHADR. FLASHDAT must be 0xA5 for the erase to occur. FLASHADR is not affected. If FLASHADR = 0x1DFE – 0x1DFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x1E00 – 0x1FFF).

(All other values for WRMD3–0 are reserved.)

Bits3–0: RDMD3–0: Read Mode Select Bits.
The Read Mode Select Bits control how the interface logic responds to reads to the FLASHDAT Register per the following values:
- 0000: A FLASHDAT read provides the data in the FLASHDAT register, but is otherwise ignored.
- 0001: A FLASHDAT read initiates a read of the byte addressed by the FLASHADR register if no operation is currently active. This mode is used for block reads.
- 0010: A FLASHDAT read initiates a read of the byte addressed by FLASHADR only if no operation is active and any data from a previous read has already been read from FLASHDAT. This mode allows single bytes to be read (or the last byte of a block) without initiating an extra read.

(All other values for RDMD3–0 are reserved.)

JTAG Register Definition 18.3. FLASHADR: JTAG Flash Address

																Reset Value
																0x0000
Bit15															Bit0	

This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15–0: Flash Operation 16-bit Address.