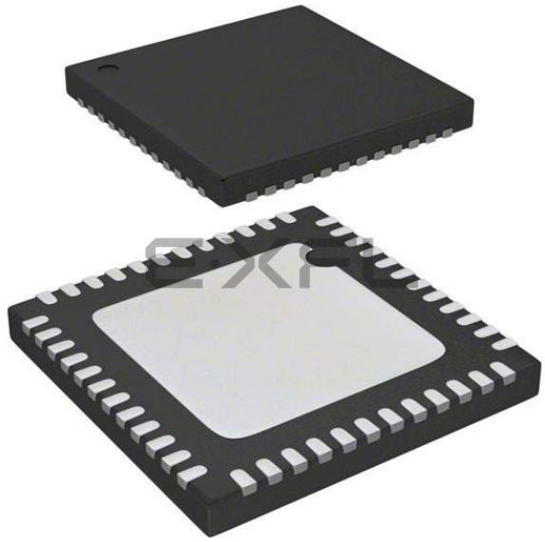

Welcome to E-XFL.COM

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	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f585-imr

4.3. QFN-40 Package Specifications

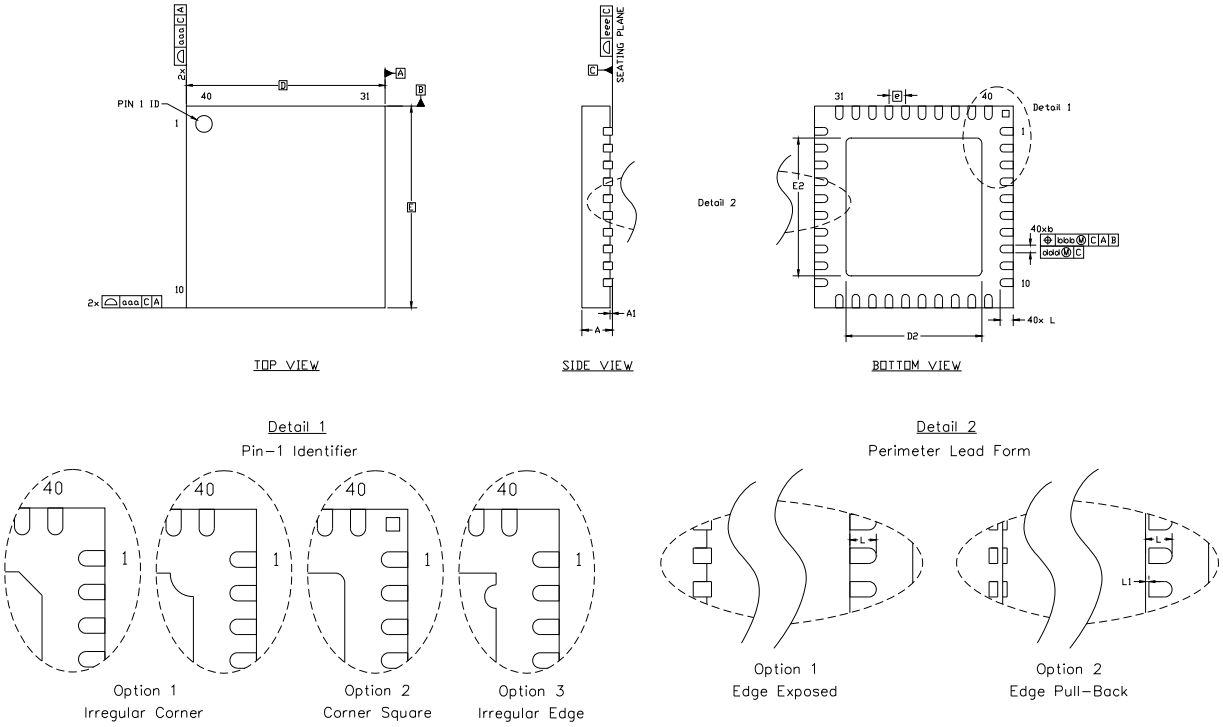

Figure 4.5. Typical QFN-40 Package Drawing

Table 4.5. QFN-40 Package Dimensions

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.80	0.85	0.90	E2	4.00	4.10	4.20
A1	0.00		0.05	L	0.35	0.40	0.45
b	0.18	0.23	0.28	L1			0.10
D	6.00 BSC			aaa			0.10
D2	4.00	4.10	4.20	bbb			0.10
e	0.50 BSC			ddd			0.05
E	6.00 BSC			eee			0.08

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation VJJD-5, except for features A, D2, and E2 which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

6.2. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code. When the repeat count is set to 1, conversion codes are represented in 12-bit unsigned integer format and the output conversion code is updated after each conversion. Inputs are measured from 0 to V_{REF} x 4095/4096. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.2). Unused bits in the ADC0H and ADC0L registers are set to 0. Example codes are shown below for both right-justified and left-justified data.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 4095/4096	0x0FFF	0xFFF0
VREF x 2048/4096	0x0800	0x8000
VREF x 2047/4096	0x07FF	0x7FF0
0	0x0000	0x0000

When the ADC0 Repeat Count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, or 16 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0CF register. The value must be right-justified (AD0LJST = 0), and unused bits in the ADC0H and ADC0L registers are set to 0. The following example shows right-justified codes for repeat counts greater than 1. Notice that accumulating 2^n samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 8	Repeat Count = 16
V_{REF} x 4095/4096	0x3FFC	0x7FF8	0xFFF0
V_{REF} x 2048/4096	0x2000	0x4000	0x8000
V_{REF} x 2047/4096	0x1FFC	0x3FF8	0x7FF0
0	0x0000	0x0000	0x0000

6.2.1. Settling Time Requirements

A minimum tracking time is required before an accurate conversion is performed. This tracking time is determined by any series impedance, including the AMUX0 resistance, the ADC0 sampling capacitance, and the accuracy required for the conversion.

Figure 6.5 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 6.1. When measuring the Temperature Sensor output, use the tracking time specified in Table 5.11 on page 52. When measuring V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 5.10 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 6.1. ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).

6.3.2. Setting the Gain Value

The three programmable gain registers are accessed indirectly using the ADC0H and ADC0L registers when the GAINEN bit (ADC0CF.0) bit is set. ADC0H acts as the address register, and ADC0L is the data register. The programmable gain registers can only be written to and cannot be read. See Gain Register Definition 6.1, Gain Register Definition 6.2, and Gain Register Definition 6.3 for more information.

The gain is programmed using the following steps:

1. Set the GAINEN bit (ADC0CF.0)
2. Load the ADC0H with the ADC0GNH, ADC0GNL, or ADC0GNA address.
3. Load ADC0L with the desired value for the selected gain register.
4. Reset the GAINEN bit (ADC0CF.0)

Notes:

1. An ADC conversion should not be performed while the GAINEN bit is set.
2. Even with gain enabled, the maximum input voltage must be less than V_{REGIN} and the maximum voltage of the signal after gain must be less than or equal to V_{REF}.

In code, changing the value to 0.44 gain from the previous example looks like:

```
// in 'C':
ADC0CF |= 0x01;          // GAINEN = 1
ADC0H = 0x04;            // Load the ADC0GNH address
ADC0L = 0x6C;            // Load the upper byte of 0x6CA to ADC0GNH
ADC0H = 0x07;            // Load the ADC0GNL address
ADC0L = 0xA0;            // Load the lower nibble of 0x6CA to ADC0GNL
ADC0H = 0x08;            // Load the ADC0GNA address
ADC0L = 0x01;            // Set the GAINADD bit
ADC0CF &= ~0x01;         // GAINEN = 0

; in assembly
ORL ADC0CF,#01H          ; GAINEN = 1
MOV ADC0H,#04H           ; Load the ADC0GNH address
MOV ADC0L,#06CH          ; Load the upper byte of 0x6CA to ADC0GNH
MOV ADC0H,#07H           ; Load the ADC0GNL address
MOV ADC0L,#0A0H          ; Load the lower nibble of 0x6CA to ADC0GNL
MOV ADC0H,#08H           ; Load the ADC0GNA address
MOV ADC0L,#01H           ; Set the GAINADD bit
ANL ADC0CF,#0FEH         ; GAINEN = 0
```

SFR Definition 9.5. CPT2CN: Comparator2 Control

Bit	7	6	5	4	3	2	1	0
Name	CP2EN	CP2OUT	CP2RIF	CP2FIF	CP2HYP[1:0]		CP2HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9A; SFR Page = 0x10

Bit	Name	Function
7	CP2EN	**Comparator2 Enable Bit.** 0: Comparator2 Disabled. 1: Comparator2 Enabled.
6	CP2OUT	**Comparator2 Output State Flag.** 0: Voltage on CP2+ < CP2–. 1: Voltage on CP2+ > CP2–.
5	CP2RIF	**Comparator2 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator2 Rising Edge has occurred since this flag was last cleared. 1: Comparator2 Rising Edge has occurred.
4	CP2FIF	**Comparator2 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator2 Falling-Edge has occurred since this flag was last cleared. 1: Comparator2 Falling-Edge has occurred.
3:2	CP2HYP[1:0]	**Comparator2 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP2HYN[1:0]	**Comparator2 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

SFR Definition 9.9. CPT2MX: Comparator2 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name	CMX2N[3:0]				CMX2P[3:0]			
Type	R/W				R/W			
Reset	0	1	1	1	0	1	1	1

SFR Address = 0x9C; SFR Page = 0x10

Bit	Name	Function
7:4	CMX2N[3:0]	**Comparator2 Negative Input MUX Selection.** 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 1010: P2.5 1011: P2.7 1100–1111: None
3:0	CMX2P[3:0]	**Comparator2 Positive Input MUX Selection.** 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 1010: P2.4 1011: P2.6 1100–1111: None

Table 11.1. CIP-51 Instruction Set Summary (Prefetch-Enabled) (Continued)

Mnemonic	Description	Bytes	Clock Cycles
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/(4-6)*
JNC rel	Jump if Carry is not set	2	2/(4-6)*
JB bit, rel	Jump if direct bit is set	3	3/(5-7)*
JNB bit, rel	Jump if direct bit is not set	3	3/(5-7)*
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/(5-7)*
Program Branching			
ACALL addr11	Absolute subroutine call	2	4-6*
LCALL addr16	Long subroutine call	3	5-7*
RET	Return from subroutine	1	6-8*
RETI	Return from interrupt	1	6-8*
AJMP addr11	Absolute jump	2	4-6*
LJMP addr16	Long jump	3	5-7*
SJMP rel	Short jump (relative address)	2	4-6*
JMP @A+DPTR	Jump indirect relative to DPTR	1	3-5*
JZ rel	Jump if A equals zero	2	2/(4-6)*
JNZ rel	Jump if A does not equal zero	2	2/(4-6)*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	4/(6-8)*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/(6-8)*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/(5-7)*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/(6-8)*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/(4-6)*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/(5-7)*
NOP	No operation	1	1

Note: Certain instructions take a variable number of clock cycles to execute depending on instruction alignment and the FLRT setting (SFR Definition 15.3).

14.3. External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “27.1. Timer 0 and Timer 1” on page 287) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 14.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT0}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “20.3. Priority Crossbar Decoder” on page 192 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

18.4. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

18.4.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 18.1.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the Q outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time $\overline{RD}$ or $\overline{WR}$ is asserted.

See Section "18.6.2. Multiplexed Mode" on page 172 for more information.

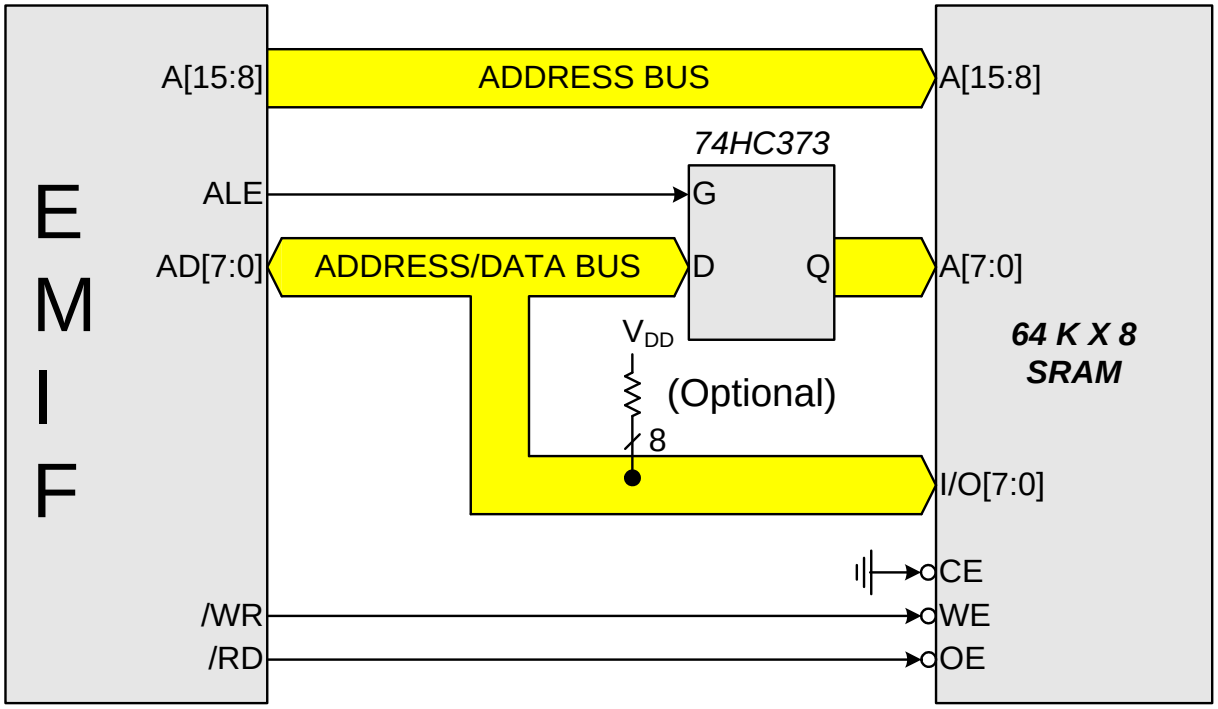

Figure 18.1. Multiplexed Configuration Example

SFR Definition 19.3. OSCICRS: Internal Oscillator Coarse Calibration

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

SFR Address = 0xA2; SFR Page = 0x0F;

Bit	Name	Function
7	Unused	Read = 0; Write = Don't Care
6:0	OSCICRS[6:0]	**Internal Oscillator Coarse Calibration Bits.** These bits determine the internal oscillator period. When set to 0000000b, the internal oscillator operates at its slowest setting. When set to 1111111b, the internal oscillator operates at its fastest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 19.4. OSCIFIN: Internal Oscillator Fine Calibration

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

SFR Address = 0x9E; SFR Page = 0x0F;

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care
5:0	OSCIFIN[5:0]	**Internal Oscillator Fine Calibration Bits.** These bits are fine adjustment for the internal oscillator period. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 20.9. P2MASK: Port 2 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P2MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xB2; SFR Page = 0x00

Bit	Name	Function
7:0	P2MASK[7:0]	**Port 2 Mask Value.** Selects P2 pins to be compared to the corresponding bits in P2MAT. 0: P2.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P2.n pin logic value is compared to P2MAT.n.

SFR Definition 20.10. P2MAT: Port 2 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P2MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB1; SFR Page = 0x00

Bit	Name	Function
7:0	P2MAT[7:0]	**Port 2 Match Value.** Match comparison value used on Port 2 for bits in P2MAT which are set to 1. 0: P2.n pin logic value is compared with logic LOW. 1: P2.n pin logic value is compared with logic HIGH.

Table 21.2. Manual Baud Rate Parameters Examples

	Baud (bits/sec)														
	20 K			19.2 K			9.6 K			4.8 K			1 K		
SYSCLK (MHz)	Mult.	Pres.	Div.	Mult.	Pres.	Div.	Mult.	Pres.	Div.	Mult.	Pres.	Div.	Mult.	Pres.	Div.
25	0	1	312	0	1	325	1	1	325	3	1	325	19	1	312
24.5	0	1	306	0	1	319	1	1	319	3	1	319	19	1	306
24	0	1	300	0	1	312	1	1	312	3	1	312	19	1	300
22.1184	0	1	276	0	1	288	1	1	288	3	1	288	19	1	276
16	0	1	200	0	1	208	1	1	208	3	1	208	19	1	200
12.25	0	0	306	0	0	319	1	0	319	3	0	319	19	0	306
12	0	0	300	0	0	312	1	0	312	3	0	312	19	0	300
11.0592	0	0	276	0	0	288	1	0	288	3	0	288	19	0	276
8	0	0	200	0	0	208	1	0	208	3	0	208	19	0	200

21.2.4. Baud Rate Calculations—Automatic Mode

If the LIN controller is configured for slave mode, only the prescaler and divider need to be calculated:

$$\text{prescaler} = \ln\left[\frac{\text{SYSCLK}}{4000000}\right] \times \frac{1}{\ln 2} - 1$$

$$\text{divider} = \frac{\text{SYSCLK}}{2^{(\text{prescaler}+1)} \times 20000}$$

The following example calculates the values of these variables for a 24 MHz system clock:

$$\text{prescaler} = \ln\left[\frac{24000000}{4000000}\right] \times \frac{1}{\ln 2} - 1 = 1.585 \cong 1$$

$$\text{divider} = \frac{24000000}{2^{(1+1)} \times 20000} = 300$$

Table 21.3 presents some typical values of system clock and baud rate along with their factors.

22.2. CAN Registers

CAN registers are classified as follows:

1. **CAN Controller Protocol Registers**: CAN control, interrupt, error control, bus status, test modes.
2. **Message Object Interface Registers**: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The CIP-51 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. **Message Handler Registers**: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).

For the registers other than CAN0CFG, refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

22.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes.

The registers are: CAN Control Register (CAN0CN), CAN Clock Configuration (CAN0CFG), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register.

22.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers.

22.2.3. Message Handler Registers

The Message Handler Registers are read only registers. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

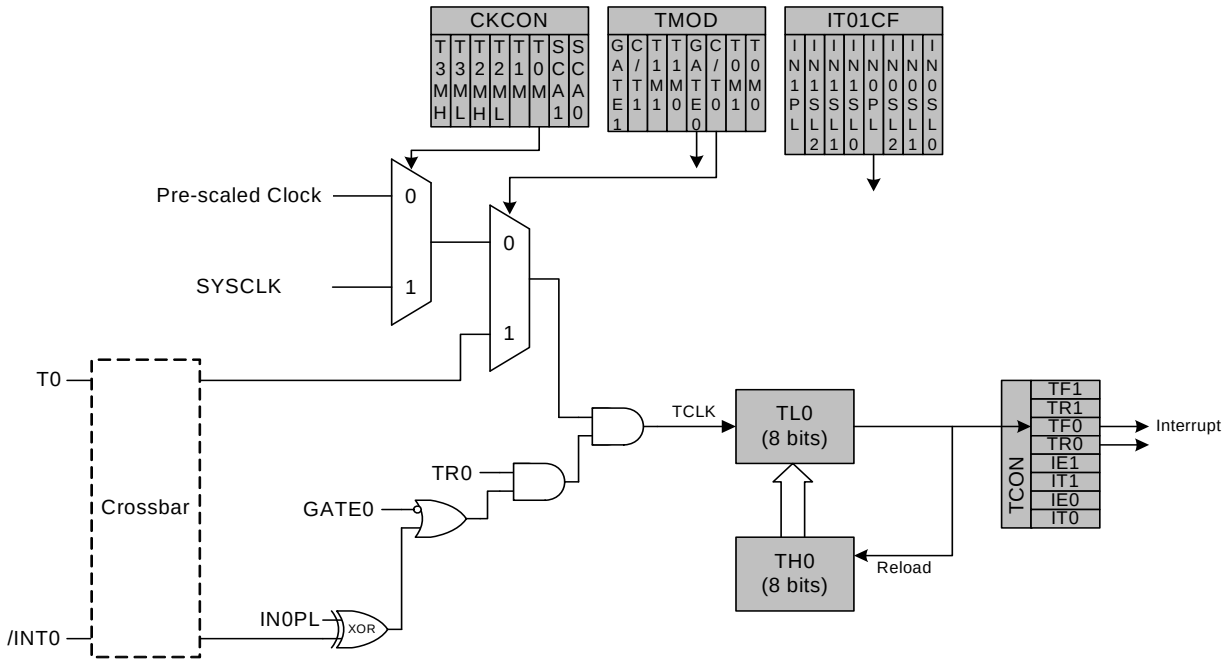

Figure 27.2. T0 Mode 2 Block Diagram

27.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

SFR Definition 27.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Capture Mode Enable.** 0: Timer 2 Capture Mode is disabled. 1: Timer 2 Capture Mode is enabled.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Read = 0b; Write = Don't Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

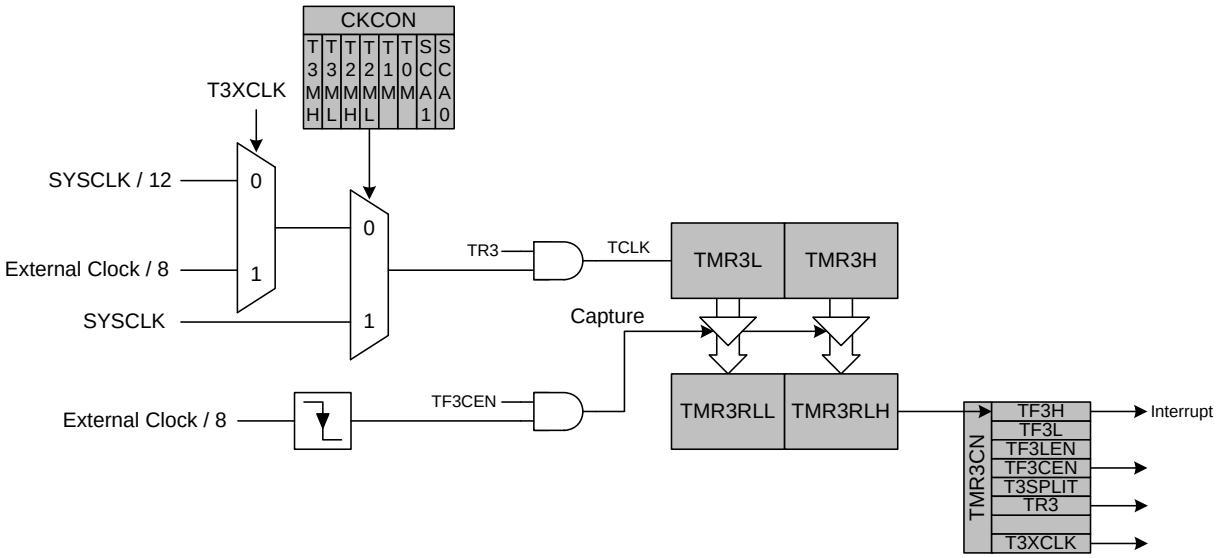

Figure 27.9. Timer 3 External Oscillator Capture Mode Block Diagram

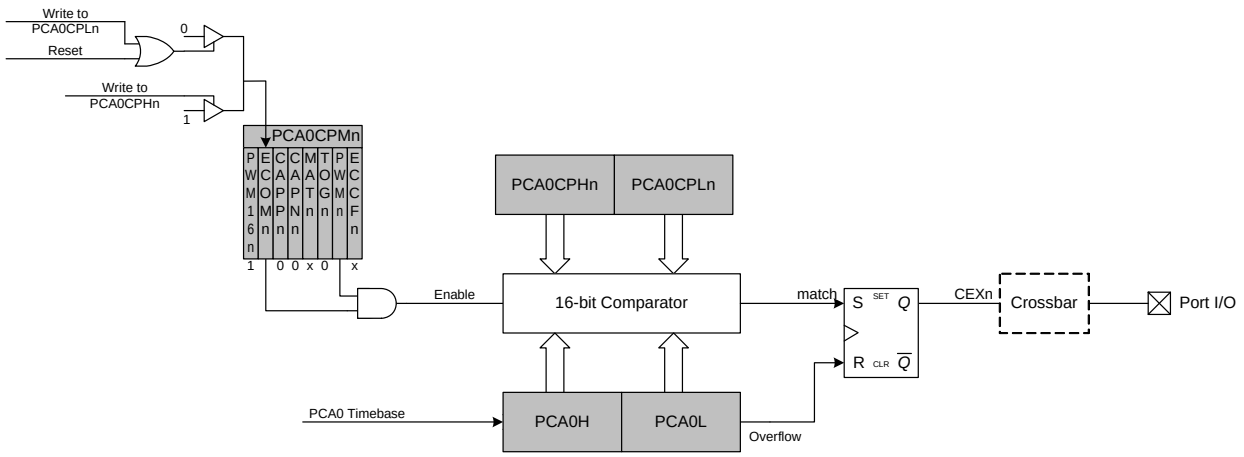

Figure 28.10. PCA0 16-Bit PWM Mode

28.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA0 Module 5. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 5 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA0 counter high byte; the Module 2 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA0 registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

28.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA0 counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA0 clock source bits (CPS2–CPS0) are frozen.
- PCA0 Idle control bit (CIDL) is frozen.
- PCA0 Module 5 is forced into software timer mode.
- Writes to the Module 5 mode register (PCA0CPM5) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA0 counter state; the counter will run until the WDT is disabled. The PCA0 counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA0 counter. If a match occurs between PCA0CPH5 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH5. Upon a PCA0CPH5 write, PCA0H plus the offset held in PCA0CPL5 is loaded into PCA0CPH5 (See Figure 28.11).

29.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA1 Counter and the module's 16-bit capture/compare register (PCA1CPHn and PCA1CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA1CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the TOG1n, MAT1n, and ECOM1n bits in the PCA1CPMn register enables the High-Speed Output mode. If ECOM1n is cleared, the associated pin will retain its state, and not toggle on the next match event.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA1 Capture/Compare registers, the low byte should always be written first. Writing to PCA1CPLn clears the ECOM1n bit to 0; writing to PCA1CPHn sets ECOM1n to 1.

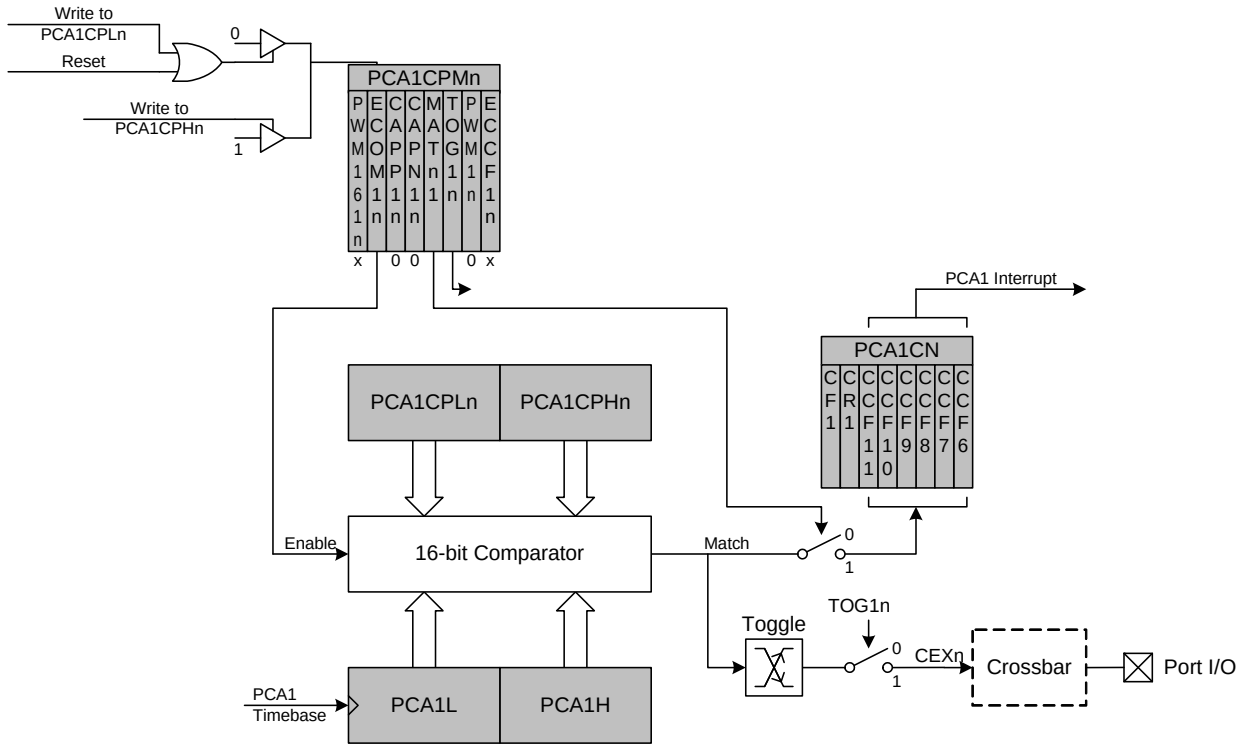

Figure 29.6. PCA1 High-Speed Output Mode Diagram

30. C2 Interface

C8051F58x/F59x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

30.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 30.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function	
7:0	C2ADD[7:0]	**C2 Address.** The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.	
		Address	**Description**
		0x00	Selects the Device ID register for Data Read instructions
		0x01	Selects the Revision ID register for Data Read instructions
		0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
		0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 30.2. DEVICEID: C2 Device ID

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

C2 Address = 0xFD; SFR Address = 0xFD; SFR Page = 0x0F

Bit	Name	Function
7:0	DEVICEID[7:0]	**Device ID.** This read-only register returns the 8-bit device ID: 0x20 (C8051F58x/F59x).

C2 Register Definition 30.3. REVID: C2 Revision ID

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

C2 Address = 0xFE; SFR Address = 0xFE; SFR Page = 0x0F

Bit	Name	Function
7:0	REVID[7:0]	**Revision ID.** This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.

C2 Register Definition 30.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 30.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xB4

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase