# E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I <sup>2</sup> C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K 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-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f500-iq

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Figure 25.5. Master Made Data/Clack Timing	057
Figure 25.5. Master Mode Data/Clock Timing	257
Figure 25.6. Slave Mode Data/Clock Timing (CKPHA = 0)	258
Figure 25.7. Slave Mode Data/Clock Timing (CKPHA = 1)	258
Figure 25.8. SPI Master Timing (CKPHA = 0)	262
Figure 25.9. SPI Master Timing (CKPHA = 1)	262
Figure 25.10. SPI Slave Timing (CKPHA = 0)	263
Figure 25.11. SPI Slave Timing (CKPHA = 1)	263
Figure 26.1. T0 Mode 0 Block Diagram	268
Figure 26.2. T0 Mode 2 Block Diagram	269
Figure 26.3. T0 Mode 3 Block Diagram	270
Figure 26.4. Timer 2 16-Bit Mode Block Diagram	275
Figure 26.5. Timer 2 8-Bit Mode Block Diagram	276
Figure 26.6. Timer 2 External Oscillator Capture Mode Block Diagram	277
Figure 26.7. Timer 3 16-Bit Mode Block Diagram	281
Figure 26.8. Timer 3 8-Bit Mode Block Diagram	282
Figure 26.9. Timer 3 External Oscillator Capture Mode Block Diagram	283
Figure 27.1. PCA Block Diagram	287
Figure 27.2. PCA Counter/Timer Block Diagram	288
Figure 27.3. PCA Interrupt Block Diagram	289
Figure 27.4. PCA Capture Mode Diagram	291
Figure 27.5. PCA Software Timer Mode Diagram	292
Figure 27.6. PCA High-Speed Output Mode Diagram	293
Figure 27.7. PCA Frequency Output Mode	294
Figure 27.8. PCA 8-Bit PWM Mode Diagram	295
Figure 27.9. PCA 9, 10 and 11-Bit PWM Mode Diagram	296
Figure 27.10. PCA 16-Bit PWM Mode	297
Figure 27.11. PCA Module 2 with Watchdog Timer Enabled	298
Figure 28.1. Typical C2 Pin Sharing	309



## 3. Pin Definitions

Name	Pin 'F500/1/4/5	Pin F508/9- F510/1 (40-pip)	Pin 'F502/3/6/7	Туре	Description
	(48-pin)	(40-pin)	(32-pin)		
VDD	4	4	4		Digital Supply Voltage. Must be connected.
GND	6	6	6		Digital Ground. Must be connected.
VDDA	5	5	5		Analog Supply Voltage. Must be connected.
GNDA	7	7	7		Analog Ground. Must be connected.
VREGIN	3	3	3		Voltage Regulator Input
VIO	2	2	2		Port I/O Supply Voltage. Must be connected.
RST/	12	10	10	d I/O	Device Reset. Open-drain output of internal POR or $V_{DD}$ Monitor. An external source can initiate a system reset by driving this pin low.
C2CK				D I/O	Clock signal for the C2 Debug Interface.
C2D	11	—	_	d I/O	Bi-directional data signal for the C2 Debug Interface.
P4.0/	—	9	—	D I/O or A In	Port 4.0. See SFR Definition 20.29 for a description.
C2D				d I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0/	—		9	D I/O or A In	Port 3.0. See SFR Definition 20.24 for a description.
C2D				d I/O	Bi-directional data signal for the C2 Debug Interface.
P0.0	8	8	8	D I/O or A In	Port 0.0. See SFR Definition 20.12 for a description.
P0.1	1	1	1	D I/O or A In	Port 0.1
P0.2	48	40	32	D I/O or A In	Port 0.2
P0.3	47	39	31	D I/O or A In	Port 0.3
P0.4	46	38	30	D I/O or A In	Port 0.4
P0.5	45	37	29	D I/O or A In	Port 0.5

Table 3.1. Pin Definitions for the C8051F50x/F51x





### Figure 4.7. QFP-32 Package Drawing

### Table 4.7. QFP-32 Package Dimensions

Dimension	Min	Тур	Max		Dimension	Min	Тур	Max		
A	_	_	1.60		E		9.00 BSC.			
A1	0.05	—	0.15		E1		7.00 BSC.			
A2	1.35	1.40	1.45		L	0.45	0.60	0.75		
b	0.30	0.37	0.45		aaa		0.20			
С	0.09	—	0.20		bbb		0.20			
D		9.00 BSC.			CCC		0.10			
D1	7.00 BSC.				ddd		0.20			
е		0.80 BSC.			θ	0°	3.5°	7°		

#### 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 the JEDEC outline MS-026, variation BBA.
- **4.** Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.



## SFR Definition 6.5. ADC0H: ADC0 Data Word MSB

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

SFR Address = 0xBE; SFR Page = 0x00

Bit	Name	Function
7:0	ADC0H[7:0]	ADC0 Data Word High-Order Bits.
		For AD0LJST = 0 and AD0RPT as follows:
		00: Bits 3–0 are the upper 4 bits of the 12-bit result. Bits 7–4 are 0000b.
		01: Bits 4–0 are the upper 5 bits of the 14-bit result. Bits 7–5 are 000b.
		10: Bits 5–0 are the upper 6 bits of the 15-bit result. Bits 7–6 are 00b.
		11: Bits 7–0 are the upper 8 bits of the 16-bit result.
		For AD0LJST = 1 (AD0RPT must be 00): Bits 7–0 are the most-significant bits of the
		ADC0 12-bit result.

### SFR Definition 6.6. ADC0L: ADC0 Data Word LSB

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

SFR Address = 0xBD; SFR Page = 0x00

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 ADC0 Accumulated Result. For AD0LJST = 1 (AD0RPT must be '00'): Bits 7–4 are the lower 4 bits of the 12-bit result. Bits 3–0 are 0000b.



Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed.

### SFR Definition 9.1. CPT0CN: Comparator0 Control

Bit	7	6	5	4	3	2	1	0	
Name	CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0H	CP0HYP[1:0]		CP0HYN[1:0]	
Туре	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 = 0x00

Bit	Name	Function
7	CP0EN	Comparator0 Enable Bit.
		0: Comparator0 Disabled.
		1: Comparator0 Enabled.
6	CP0OUT	Comparator0 Output State Flag.
		0: Voltage on CP0+ < CP0–.
		1: Voltage on CP0+ > CP0
5	CP0RIF	Comparator0 Rising-Edge Flag. Must be cleared by software.
		0: No Comparator0 Rising Edge has occurred since this flag was last cleared.
		1: Comparator0 Rising Edge has occurred.
4	CP0FIF	Comparator0 Falling-Edge Flag. Must be cleared by software.
		0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.
		1: Comparator0 Falling-Edge has occurred.
3:2	CP0HYP[1:0]	Comparator0 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	CP0HYN[1:0]	Comparator0 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 9.5. CPT0MX: Comparator0 MUX Selection

Bit	7	6	5	4	3	2	1	0		
Nam	е	CMX0	N[3:0]	1		CMX0P[3:0]				
Тур	e	R/	W		R/W					
Rese	et 0	1	1	1	0	1	1	1		
SFR A	Address = 0x9	C: SFR Page	e = 0x00							
Bit	Name				Function					
7:4	CMX0N[3:0]	Comparato	r0 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:	Nor	ne						
3:0	CMX0P[3:0]	Comparato	r0 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:	Nor	ne						





Figure 13.3. SFR Page Stack After CAN0 Interrupt Occurs

While in the CAN0 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a high priority interrupt, while the CAN0 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR Page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 0x0C for CAN0) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x00 for SPI0DAT) is pushed down to the SFRLAST register, the "bottom" of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 13.4.



## SFR Definition 14.7. IT01CF: INT0/INT1 Configuration

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

### SFR Address = 0xE4; SFR Page = 0x0F

Bit	Name	Function
7	IN1PL	INT1 Polarity. 0: INT1 input is active low. 1: INT1 input is active high.
6:4	IN1SL[2:0]	INT1 Port Pin Selection Bits. These bits select which Port pin is assigned to INT1. Note that this pin assignment is independent of the Crossbar; INT1 will monitor the assigned Port pin without disturb- ing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7
3	INOPL	INTO Polarity. 0: INTO input is active low. 1: INTO input is active high.
2:0	IN0SL[2:0]	INTO Port Pin Selection Bits. These bits select which Port pin is assigned to INTO. Note that this pin assignment is independent of the Crossbar; INTO will monitor the assigned Port pin without disturb- ing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7



### 18.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 110



Figure 18.6. Non-multiplexed 8-bit MOVX with Bank Select Timing





### Figure 19.3. External 32.768 kHz Quartz Crystal Oscillator Connection Diagram

### 19.4.2. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 19.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation, according to Equation , where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and R = the pull-up resistor value in  $k\Omega$ .

$$f = 1.23 \times 10^3 / (R \times C)$$

### Equation 19.1. RC Mode Oscillator Frequency

For example: If the frequency desired is 100 kHz, let R = 246 k $\Omega$  and C = 50 pF:

f = 1.23(10<sup>3</sup>)/RC = 1.23(10<sup>3</sup>)/[246 x 50] = 0.1 MHz = 100 kHz

Referring to the table in SFR Definition 19.6, the required XFCN setting is 010b.

### 19.4.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 19.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation according to Equation 19.2, where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and V<sub>DD</sub> = the MCU power supply in Volts.



## SFR Definition 20.3. XBR2: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved				LIN0E	
Туре	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

## SFR Address = 0xC7; SFR Page = 0x0F

Bit	Name	Function
7	WEAKPUD	Port I/O Weak Pullup Disable.
		<ul><li>0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode).</li><li>1: Weak Pullups disabled.</li></ul>
6	XBARE	Crossbar Enable.
		0: Crossbar disabled.
		1: Crossbar enabled.
5:1	Reserved	Always Write to 00000b.
0	LIN0E	LIN I/O Output Enable.
		0: LIN I/O unavailable at Port pin.
		1: LIN_TX, LIN_RX routed to Port pins.



## SFR Definition 20.23. P2SKIP: Port 2 Skip

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Туре	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD6; SFR Page = 0x0F

Bit	Name	Function
7:0	P2SKIP[7:0]	Port 2 Crossbar Skip Enable Bits.
		<ul> <li>These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar.</li> <li>0: Corresponding P2.n pin is not skipped by the Crossbar.</li> <li>1: Corresponding P2.n pin is skipped by the Crossbar.</li> </ul>

### SFR Definition 20.24. P3: Port 3

Bit	7	6	5	4	3	2	1	0
Name	P3[7:0]							
Туре	R/W							
Reset	1	1	1	1	1	1	1	1

### SFR Address = 0xB0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read			
7:0	P3[7:0]	<b>Port 3 Data.</b> Sets the Port latch logic value or reads the Port pin logic state in Port cells con- figured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P3.n Port pin is logic LOW. 1: P3.n Port pin is logic HIGH.			
Note:	ote: Port P3.1–P3.7 are only available on the 48-pin and 40-pin packages.						



### LIN Register Definition 21.9. LIN0DIV: LIN0 Divider Register

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

Indirect Address = 0x0C

Bit	Name	Function
7:0	DIVLSB	LIN Baud Rate Divider Least Significant Bits. The 8 least significant bits for the baud rate divider. The 9th and most significant bit is the DIV9 bit (LIN0MUL.0). The valid range for the divider is 200 to 511.

### LIN Register Definition 21.10. LIN0MUL: LIN0 Multiplier Register

Bit	7	6	5	4	3	2	1	0
Name	PRESCL[1:0]			DIV9				
Туре	R/W				R/W			R/W
Reset	1	1	1	1	1	1	1	1

Indirect Address = 0x0D

Bit	Name	Function			
7:6	PRESCL[1:0]	IN Baud Rate Prescaler Bits.			
		These bits are the baud rate prescaler bits.			
5:1	LINMUL[4:0]	LIN Baud Rate Multiplier Bits.			
		These bits are the baud rate multiplier bits. These bits are not used in slave mode.			
0	DIV9	LIN Baud Rate Divider Most Significant Bit.			
		The most significant bit of the baud rate divider. The 8 least significant bits are in LIN0DIV. The valid range for the divider is 200 to 511.			



## 22. Controller Area Network (CAN0)

**Important Documentation Note**: The Bosch CAN Controller is integrated in the C8051F500/2/4/6/8-F510 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, refer to Bosch's C\_CAN User's Manual as an accompanying manual to the Silicon Labs' data sheet.

The C8051F500/2/4/6/8-F510 devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 22.1 shows an example typical configuration on a CAN bus.

Silicon Labs CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFRs) in the CIP-51.



Figure 22.1. Typical CAN Bus Configuration



### 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.



#### 23.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. An interrupt is generated after each received byte.

Software must write the ACK bit at that time to ACK or NACK the received byte. Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 23.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur **before** the ACK cycle in this mode.



Figure 23.6. Typical Master Read Sequence











### 25.6. SPI Special Function Registers

SPI0 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 SPI0 Bus are described in the following figures.



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

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	Х	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 26.5. Timer 2 8-Bit Mode Block Diagram

#### 26.2.3. External Oscillator Capture Mode

Capture Mode allows the external oscillator to be measured against the system clock. Timer 2 can be clocked from the system clock, or the system clock divided by 12, depending on the T2ML (CKCON.4), and T2XCLK bits. When a capture event is generated, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set. A capture event is generated by the falling edge of the clock source being measured, which is the external oscillator / 8. By recording the difference between two successive timer capture values, the external oscillator frequency can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

For example, if T2ML = 1b and TF2CEN = 1b, Timer 2 will clock every SYSCLK and capture every external clock divided by 8. If the SYSCLK is 24 MHz and the difference between two successive captures is 5984, then the external clock frequency is as follows:

24 MHz/(5984/8) = 0.032086 MHz or 32.086 kHz

This mode allows software to determine the external oscillator frequency when an RC network or capacitor is used to generate the clock source.



### SFR Definition 26.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Туре	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC; SFR Page = 0x00

Bit	Name	Function
7:0	TMR2L[7:0]	Timer 2 Low Byte.
		In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8- bit mode, TMR2L contains the 8-bit low byte timer value.

### SFR Definition 26.12. TMR2H Timer 2 High Byte

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

SFR Address = 0xCD; SFR Page = 0x00

Bit	Name	Function
7:0	TMR2H[7:0]	Timer 2 High 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.



System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)				
24,000,000	255	32.8				
24,000,000	128	16.5				
24,000,000	32	4.2				
3,000,000	255	262.1				
3,000,000	128	132.1				
3,000,000	32	33.8				
187,500 <sup>2</sup>	255	4194				
187,500 <sup>2</sup>	128	2114				
187,500 <sup>2</sup>	32	541				
Notes:						
<ol> <li>Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.</li> </ol>						
<ol> <li>Internal SYSCLK reset frequency = Internal Oscillator divided by 128.</li> </ol>						

## Table 27.3. Watchdog Timer Timeout Intervals<sup>1</sup>

