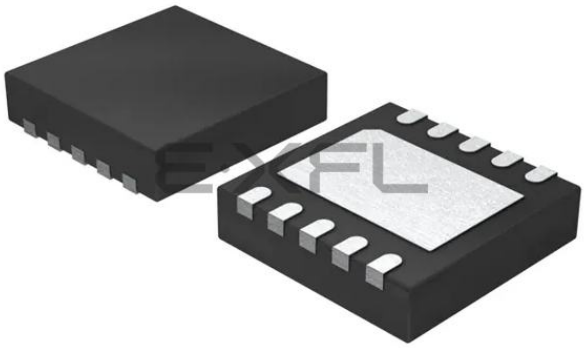


Welcome to E-XFL.COM

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	25MHz
Connectivity	LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	6
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 6x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	10-VFDFN Exposed Pad
Supplier Device Package	10-DFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f526a-imr

1.1. Ordering Information

The following features are common to all devices in this family:

- 25 MHz system clock and 25 MIPS throughput (peak)
- 256 bytes of internal RAM
- Enhanced SPI peripheral
- Enhanced UART peripheral
- Three Timers
- Three Programmable Counter Array channels
- Internal 24.5 MHz oscillator
- Internal Voltage Regulator
- 12-bit, 200 ksps ADC
- Internal Voltage Reference and Temperature Sensor
- One Analog Comparator

Table 1.1 shows the features that differentiate the devices in this family.

Table 1.1. Product Selection Guide (Recommended for New Designs)

Ordering Part Number	Flash Memory (kB)	Port I/Os	LIN	Package	Ordering Part Number	Flash Memory (kB)	Port I/Os	LIN	Package
C8051F520-C-IM	8	6	✓	DFN-10	C8051F534-C-IM	4	16	—	QFN-20
C8051F521-C-IM	8	6	—	DFN-10	C8051F536-C-IM	2	16	✓	QFN-20
C8051F523-C-IM	4	6	✓	DFN-10	C8051F537-C-IM	2	16	—	QFN-20
C8051F524-C-IM	4	6	—	DFN-10	C8051F530-C-IT	8	16	✓	TSSOP-20
C8051F526-C-IM	2	6	✓	DFN-10	C8051F531-C-IT	8	16	—	TSSOP-20
C8051F527-C-IM	2	6	—	DFN-10	C8051F533-C-IT	4	16	✓	TSSOP-20
C8051F530-C-IM	8	16	✓	QFN-20	C8051F534-C-IT	4	16	—	TSSOP-20
C8051F531-C-IM	8	16	—	QFN-20	C8051F536-C-IT	2	16	✓	TSSOP-20
C8051F533-C-IM	4	16	✓	QFN-20	C8051F537-C-IT	2	16	—	TSSOP-20

All devices in Table 1.1 are also available in an automotive version. For the automotive version, the -I in the ordering part number is replaced with -A. For example, the automotive version of the C8051F520-C-IM is the C8051F520-C-AM.

The -AM and -AT devices receive full automotive quality production status, including AEC-Q100 qualification (fault coverage report available upon request), registration with International Material Data System (IMDS) and Part Production Approval Process (PPAP) documentation. PPAP documentation is available at www.silabs.com with a registered NDA and approved user account. The -AM and -AT devices enable high volume automotive OEM applications with their enhanced testing and processing. Please contact Silicon Labs sales for more information regarding -AM and -AT devices for your automotive project.

2.2. Electrical Characteristics

Table 2.2. Global DC Electrical Characteristics

–40 to +125 °C, 25 MHz System Clock unless otherwise specified. Typical values are given at 25 °C

Parameter	Conditions	Min	Typ	Max	Units
Supply Input Voltage (V_{REGIN})[1]	Output Current ≤ 1 mA				
	C8051F52x/53x	2.7	—	5.25	V
	C8051F52xA/53xA	1.8[1]	—	5.25	V
	C8051F52x-C/53x-C	2.0[1]	—	5.25	V
Digital Supply Voltage (V_{DD})	C8051F52x/53x	2.0	—	2.7	V
	C8051F52xA/53xA	1.8	—	2.7	V
	C8051F52x-C/53x-C	2.0	—	2.75	V
Core Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[2]		0	—	25	MHz
Specified Operating Temperature Range		–40	—	+125	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[3,4]	**V_{DD} = 2.1 V:**				
	Clock = 32 kHz	—	13	—	µA
	Clock = 200 kHz	—	60	—	µA
	Clock = 1 MHz	—	0.28	—	mA
	Clock = 25 MHz	—	5.1	9	mA
	V_{DD} = 2.6 V:				
	Clock = 32 kHz	—	22	—	µA
	Clock = 200 kHz	—	105	—	µA
	Clock = 1 MHz	—	0.5	—	mA
	Clock = 25 MHz	—	7.3	13	mA
I_{DD} Frequency Sensitivity[3,5]	T = 25 °C:				
	V_{DD} = 2.1 V, F ≤ 12 MHz	—	0.276	—	mA/MHz
	V_{DD} = 2.1 V, F > 12 MHz	—	0.140	—	mA/MHz
	V_{DD} = 2.6 V, F ≤ 12 MHz	—	0.424	—	mA/MHz
	V_{DD} = 2.6 V, F > 12 MHz	—	0.184	—	mA/MHz

Notes:

1. For more information on V_{REGIN} characteristics, see Table 2.6 on page 30.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization data; Not production tested.
4. Does not include internal oscillator or internal regulator supply current.
5. I_{DD} can be estimated for frequencies <= 12 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} > 12 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F= 20 MHz, I_{DD} = 7.3 mA – (25 MHz – 20 MHz) x 0.184 mA/MHz = 6.38 mA.
6. Idle I_{DD} can be estimated for frequencies <= 1 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} > 1 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F= 5 MHz, Idle I_{DD} = 3 mA – (25 MHz– 5 MHz) x 118 µA/MHz = 0.64 mA.

Table 2.3. ADC0 Electrical Characteristics

V_{DD} = 2.1 V, V_{REF} = 1.5 V (REFSL=0), –40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			12		bits
Integral Nonlinearity		—	—	±3	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	—	±1	LSB
Offset Error[1]		–10	±1	+10	LSB
Full Scale Error		–20	±1	+20	LSB
Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		60	66	—	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	74	—	dB
Spurious-Free Dynamic Range		—	88	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	3	MHz
Burst Mode Oscillator		—	—	27	MHz
Conversion Time in SAR Clocks[2]		—	13	—	clocks
Track/Hold Acquisition Time[3,6]		1	—	—	µs
Throughput Rate[4]		—	—	200	ksps
Analog Inputs					
ADC Input Voltage Range[5]	gain = 1.0 (default) gain = n	0 0	— —	V_{REF} V_{REF} / n	V
Absolute Pin Voltage wrt to GND		0	—	V_{REGIN}	V
Sampling Capacitance		—	24	—	pF
Input Multiplexer Impedance		—	1.5	—	kΩ
Power Specifications					
Power Supply Current (from VDD)	Operating Mode, 200 ksps	—	1050	1400	µA
Burst Mode (Idle)		—	930	—	µA
Power-on Time		—	5	—	µs
Power Supply Rejection		—	1	—	mV/V

Notes:

1. Represents one standard deviation from the mean. Offset and full-scale error can be removed through calibration.
2. An additional 2 FCLK cycles are required to start and complete a conversion.
3. Additional tracking time may be required depending on the output impedance connected to the ADC input. See Section “4.3.6. Settling Time Requirements” on page 60.
4. An increase in tracking time will decrease the ADC throughput.
5. See Section “4.4. Selectable Gain” on page 60 for more information about setting the gain.
6. Additional tracking time might be needed ifVDD < 2.0 V; See Section “11.2.1. VDD Monitor Thresholds and Minimum VDD” on page 108 for minimum V_{DD} requirements.

For example, the initial example in this section requires a gain of 0.44 to convert 5 V full scale to 2.2 V full scale. Using Equation 4.3:

$$GAIN = \left(0.44 - GAINADD \times \left(\frac{1}{64}\right)\right) \times 4096$$

If GAINADD is set to 1, this makes the equation:

$$GAIN = \left(0.44 - 1 \times \left(\frac{1}{64}\right)\right) \times 4096 = 0.424 \times 4096 = 1738 = 0x06CA$$

The actual gain from setting GAINADD to 1 and ADC0GNH and ADC0GNL to 0x6CA is 0.4399. A similar gain can be achieved if GAINADD is set to 0 with a different value for ADC0GNH and ADC0GNL.

4.4.2. 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 4.1, Gain Register Definition 4.2, and Gain Register Definition 4.3 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 4.4. ADC0MX: ADC0 Channel Select

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	AD0MX					00011111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xBB

Bits7–5: **UNUSED**. Read = 000b; Write = don't care.
Bits4–0: **AD0MX4–0**: AMUX0 Positive Input Selection

AD0MX4–0	ADC0 Input Channel
00000	P0.0
00001	P0.1
00010	P0.2
00011	P0.3
00100	P0.4
00101	P0.5
00110	P0.6*
00111	P0.7*
01000	P1.0*
01001	P1.1*
01010	P1.2*
01011	P1.3*
01100	P1.4*
01101	P1.5*
01110	P1.6*
01111	P1.7*
11000	Temp Sensor
11001	V_{DD}
11010 - 11111	GND

Note: Only applies to C8051F53x/C8051F53xA parts.

4.5.1. Window Detector In Single-Ended Mode

Figure 4.7 shows two example window comparisons for right-justified data with ADC0LTH:ADC0LTL = 0x0200 (512d) and ADC0GTH:ADC0GTL = 0x0100 (256d). The input voltage can range from 0 to V_{REF} x (4095/4096) with respect to GND, and is represented by a 12-bit unsigned integer value. The repeat count is set to one. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0100 < ADC0H:ADC0L < 0x0200). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0100 or ADC0H:ADC0L > 0x0200). Figure 4.8 shows an example using left-justified data with the same comparison values.

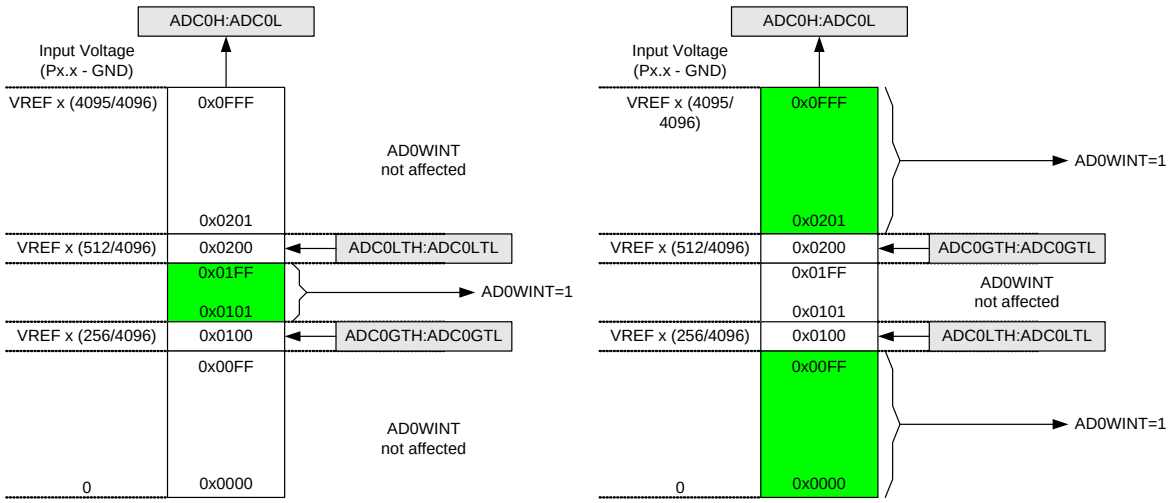


Figure 4.7. ADC Window Compare Example: Right-Justified Single-Ended Data

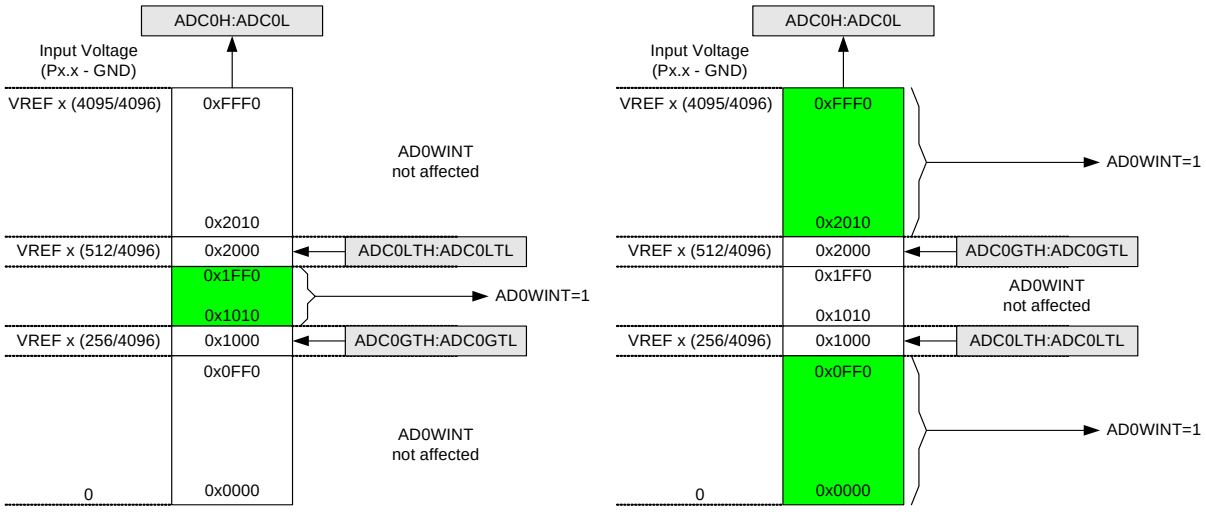


Figure 4.8. ADC Window Compare Example: Left-Justified Single-Ended Data

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
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	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
Program Branching			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	4/5
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

SFR Definition 8.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xD0

Bit7: **CY**: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit6: **AC**: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit5: **F0**: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: **RS1–RS0**: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00–0x07
0	1	1	0x08–0x0F
1	0	2	0x10–0x17
1	1	3	0x18–0x1F

Bit2: **OV**: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: **F1**: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: **PARITY**: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 9.1 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 9.2, for a detailed description of each register.

Table 9.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0			VDDMON
F0	B	P0MDIN	P1MDIN				EIP1	
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2			RSTSRC
E0	ACC	XBR0	XBR1		IT01CF		EIE1	
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	PSW	REF0CN			P0SKIP	P1SKIP		P0MAT
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H		P1MAT
C0				ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P0MASK
B8	IP		ADC0TK	ADC0MX	ADC0CF	ADC0L	ADC0	P1MASK
B0	OSCIFIN	OSCXCN	OSCICN	OSCICL				FLKEY
A8	IE	CLKSEL						
A0		SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT		
98	SCON0	SBUF0		CPT0CN		CPT0MD		CPT0MX
90	P1		LINADDR	LINDATA		LINCF		
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

10.4. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 10.1. IE: Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable
							SFR Address:	0xA8

Bit7: **EA**: Global Interrupt Enable.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit6: **ESPI0**: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit5: **ET2**: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit4: **ES0**: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit3: **ET1**: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit2: **EX1**: Enable External Interrupt 1.
This bit sets the masking of the external interrupt 1.
0: Disable external interrupt 1.
1: Enable extern interrupt 1 requests.

Bit1: **ET0**: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit0: **EX0**: Enable External Interrupt 0.
This bit sets the masking of the external interrupt 0.
0: Disable external interrupt 0.
1: Enable extern interrupt 0 requests.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

13.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
4. Assign Port pins to desired peripherals using the XBRn registers.
5. Enable the Crossbar (XBARE = 1).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a 1 indicates a digital input, and a 0 indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 13.4 for the PnMDIN register details.

Important Note: Port 0 and Port 1 pins are 5.25 V tolerant across the operating range of V_{REGIN}.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. When the WEAKPUD bit in XBR1 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a 0 and for pins configured for analog input mode to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to 1 enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. **Port output drivers are disabled while the Crossbar is disabled.**

SFR Definition 13.1. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP0AE	CP0E	SYSCKE	LINE	SPI0E	URT0E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE1

Bit7–6: **RESERVED**. Read = 00b; Must write 00b.
Bit5: **CP0AE**: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.
Bit4: **CP0E**: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.
Bit3: **SYSCKE**: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK output routed to Port pin.
Bit2: **LINE**. Lin Output Enable
Bit1: **SPI0E**: SPI I/O Enable
0: SPI I/O unavailable at Port pins.
1: SPI I/O routed to Port pins. Note that the SPI can be assigned either 3 or 4 GPIO pins.
Bit0: **URT0E**: UART I/O Output Enable
0: UART I/O unavailable at Port pin.
1: UART TX0, RX0 routed to Port pins (P0.3 and P0.4) or (P0.4 and P0.5).*

Note: Refer to Section “20. Device Specific Behavior” on page 210.

SFR Definition 14.4. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	Reserved	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
							SFR Address:	0xB1

Bit7: **XTLVLD**: Crystal Oscillator Valid Flag. (Read only when XOSCMD = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: **XOSCMD2–0**: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode.
101: Capacitor Oscillator Mode.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: **RESERVED**. Read = 0b; Must write 0b.

Bits2–0: **XFCN2–0**: External Oscillator Frequency Control Bits.
000-111: See table below:

XFCN	Crystal (XOSCMD = 11x)	RC (XOSCMD = 10x)	C (XOSCMD = 10x)
000	f ≤ 20 kHz	f ≤ 25 kHz	K Factor = 0.87
001	20 kHz < f ≤ 58 kHz	25 kHz < f ≤ 50 kHz	K Factor = 2.6
010	58 kHz < f ≤ 155 kHz	50 kHz < f ≤ 100 kHz	K Factor = 7.7
011	155 kHz < f ≤ 415 kHz	100 kHz < f ≤ 200 kHz	K Factor = 22
100	415 kHz < f ≤ 1.1 MHz	200 kHz < f ≤ 400 kHz	K Factor = 65
101	1.1 MHz < f ≤ 3.1 MHz	400 kHz < f ≤ 800 kHz	K Factor = 180
110	3.1 MHz < f ≤ 8.2 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 664
111	8.2 MHz < f ≤ 25 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590

Crystal Mode (Circuit from Figure 14.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal or resonator frequency.

RC Mode (Circuit from Figure 14.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R \times C)$, where
f = frequency of clock in MHz
C = capacitor value in pF
R = Pullup resistor value in kΩ

C Mode (Circuit from Figure 14.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF / (C \times V_{DD})$, where
f = frequency of clock in MHz
C = capacitor value the XTAL2 pin in pF
V_{DD} = Power Supply on MCU in volts

SFR Definition 15.2. SBUF0: Serial (UART0) Port 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: **SBUF0[7:0]**: Serial Data Buffer Bits 7–0 (MSB–LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Table 15.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	**Baud Rate % Error**	**Oscillator Divide Factor**	**Timer Clock Source**	**SCA1–SCA0 (pre-scale select)***	**T1M***	**Timer 1 Reload Value (hex)**
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK / 4	01	0	0x96
	14400	0.15%	1704	SYSCLK / 12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK / 12	00	0	0x96
	2400	–0.32%	10176	SYSCLK / 48	10	0	0x96
	1200	0.15%	20448	SYSCLK / 48	10	0	0x2B

X = Don't care

Note: SCA1–SCA0 and T1M bit definitions can be found in Section 18.1.

SFR Definition 16.3. SPI0CKR: SPI0 Clock Rate

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: 0xA2

Bits7–0: **SCR7–SCR0**: SPI0 Clock Rate.
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

in progress. The same applies to changes in the LIN interface mode from slave mode to master mode and from master mode to slave mode.

17.5. Sleep Mode and Wake-Up

To reduce the system's power consumption, the LIN Protocol Specification defines a Sleep Mode. The message used to broadcast a Sleep Mode request must be transmitted by the LIN master application in the same way as a normal transmit message. The LIN slave application must decode the Sleep Mode Frame from the Identifier and data bytes. After that, the LIN slave node must be put into the Sleep Mode by setting the SLEEP bit (LIN0CTRL.6).

If the SLEEP bit (LIN0CTRL.6) of the LIN slave application is not set and there is no bus activity for four seconds (specified bus idle timeout), the IDLTOUT bit (LIN0ST.6) is set and an interrupt request is generated. After that the application may assume that the LIN bus is in Sleep Mode and set the SLEEP bit (LIN0CTRL.6).

Sending a Wakeup signal from the master or any slave node terminates the Sleep Mode of the LIN bus. To send a Wakeup signal, the application has to set the WUPREQ bit (LIN0CTRL.1). After successful transmission of the wakeup signal, the DONE bit (LIN0ST.0) of the master node is set and an interrupt request is generated. The LIN slave does not generate an interrupt request after successful transmission of the Wakeup signal but it generates an interrupt request if the master does not respond to the Wakeup signal within 150 milliseconds. In that case, the ERROR bit (LIN0ST.2) and TOUT bit (LIN0ERR.2) are set. The application then has to decide whether or not to transmit another Wakeup signal.

All LIN nodes that detect a wakeup signal will set the WAKEUP (LIN0ST.1) and DONE bits (LIN0ST.0) and generate an interrupt request. After that, the application has to clear the SLEEP bit (LIN0CTRL.6) in the LIN slave.

17.6. Error Detection and Handling

The LIN peripheral generates an interrupt request and stops the processing of the current frame if it detects an error. The application has to check the type of error by processing LIN0ERR. After that, it has to reset the error register and the ERROR bit (LIN0ST.2) by writing a 1 to the RSTERR bit (LIN0CTRL.2). Starting a new message with the LIN peripheral selected as master or sending a Wakeup signal with the LIN peripheral selected as a master or slave is possible only if ERROR bit (LIN0ST.2) is set to 0.

SFR Definition 17.12. LIN0CTRL: LIN0 Control Register

W	W	W	R/W	R/W	R/W	R/W	R/W	Reset Value
STOP	SLEEP	TXRX	DTACK	RSTINT	RSTERR	WUPREQ	STREQ	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Address: 0x08 (indirect)

Bit7: **STOP**: Stop Communication Processing Bit (**slave mode only**).
This bit is to be set by the application to block the processing of the LIN Communications until the next SYNCH BREAK signal. It is used when the application is handling a data request interrupt and cannot use the frame content with the received identifier (always reads 0).

Bit6: **SLEEP**: Sleep Mode Warning.
This bit is to be set by the application to warn the peripheral that a Sleep Mode Frame was received and that the Bus is in sleep mode or if a Bus Idle timeout interrupt is requested. The application must reset it when a Wake-Up interrupt is requested.

Bit5: **TXRX**: Transmit/Receive Selection Bit.
This bit determines if the current frame is a transmit frame or a receive frame.
0: Current frame is a receive operation.
1: Current frame is a transmit operation.

Bit4: **DTACK**: Data acknowledge bit (**slave mode only**).
Set to 1 after handling a data request interrupt to acknowledge the transfer. The bit will automatically be cleared to 0 by the LIN controller.

Bit3: **RSTINT**: Interrupt Reset bit.
This bit always reads as 0.
0: No effect.
1: Reset the LININT bit (LIN0ST.3).

Bit2: **RSTERR**: Error Reset Bit.
This bit always reads as 0.
0: No effect.
1: Reset the error bits in LIN0ST and LIN0ERR.

Bit1: **WUPREQ**: Wake-Up Request Bit.
Set to 1 to terminate sleep mode by sending a wakeup signal. The bit will automatically be cleared to 0 by the LIN controller.

Bit0: **STREQ**: Start Request Bit (**master mode only**).
1: Start a LIN transmission. This should be set only after loading the identifier, data length and data buffer if necessary.
The bit is reset to 0 upon transmission completion or error detection.

19.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are 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 ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare 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.

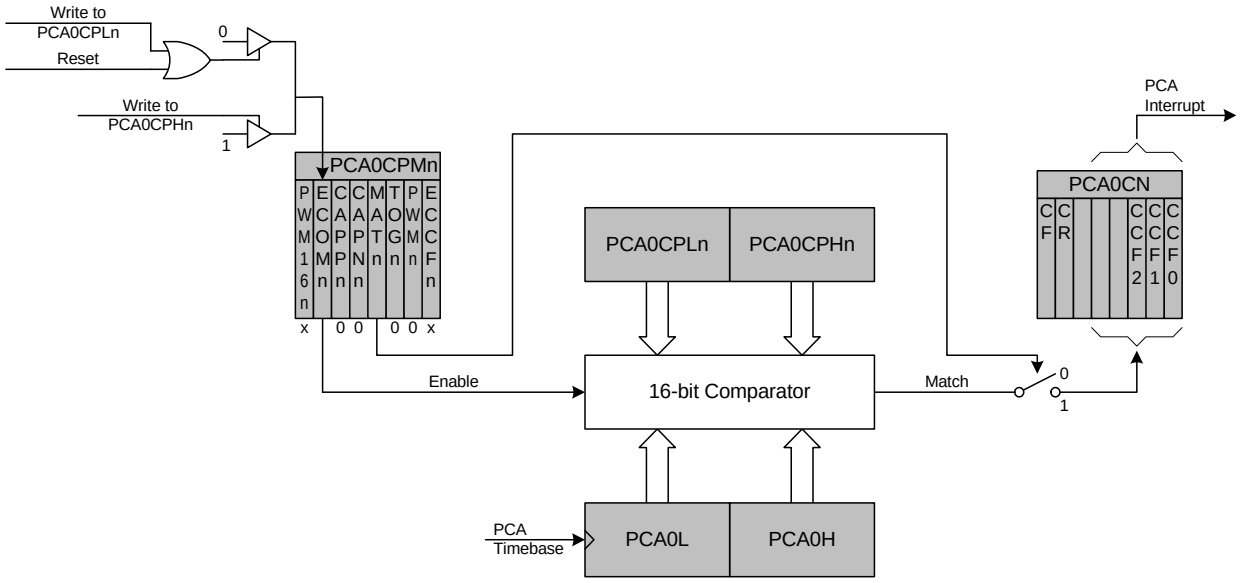
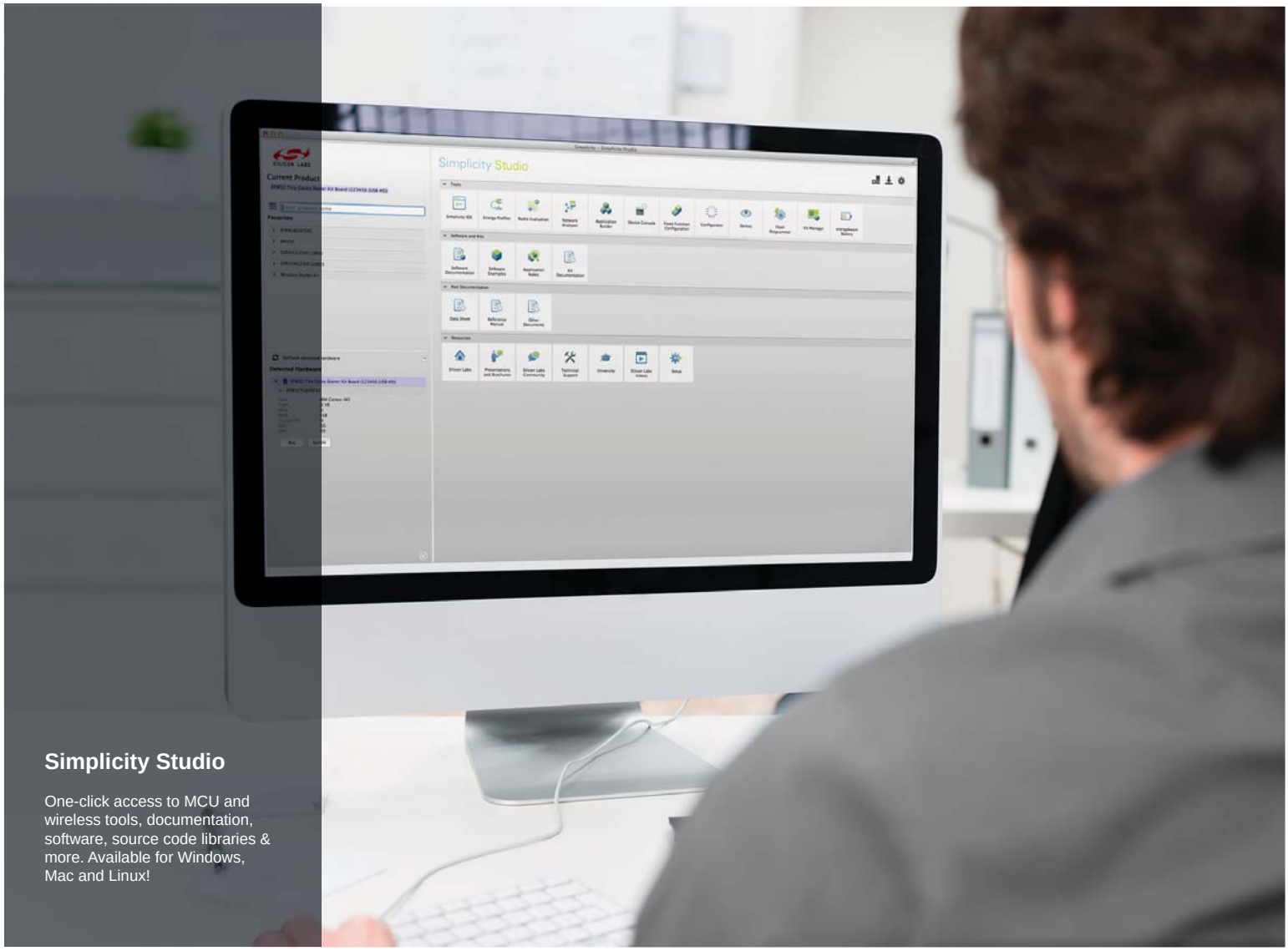


Figure 19.5. PCA Software Timer Mode Diagram



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com