



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

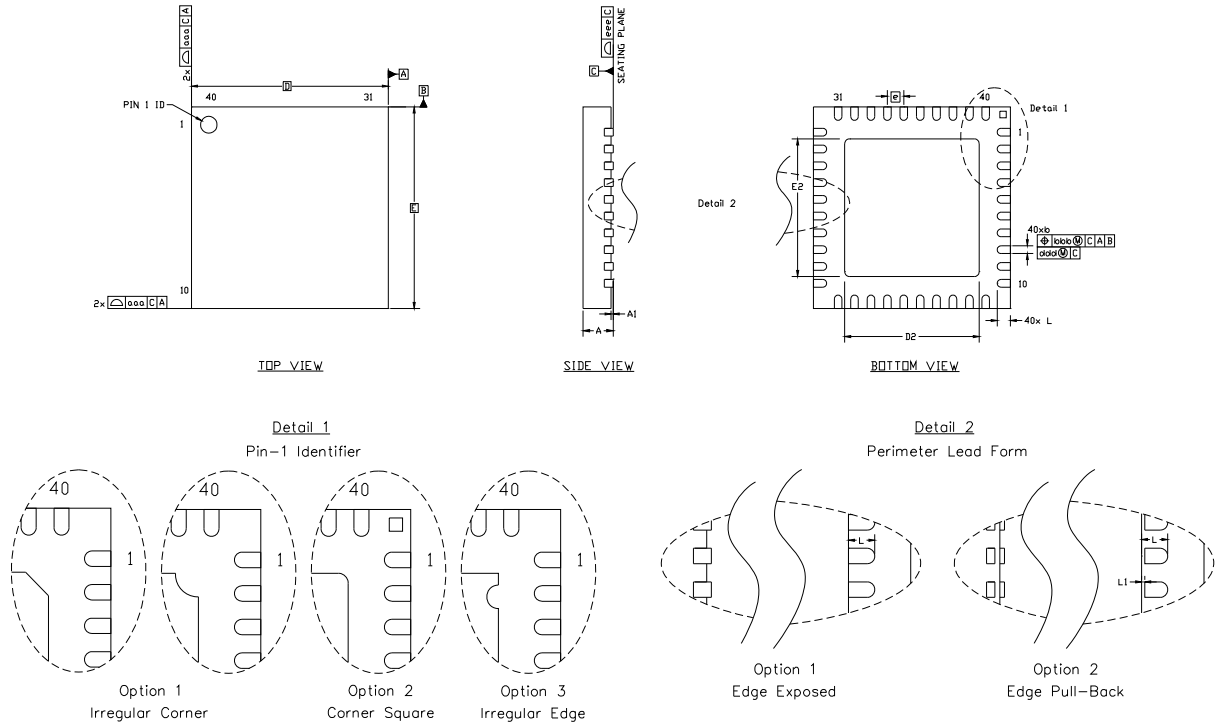
#### Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f562-imr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f562-imr</a>

17.6. Timing .....	151
17.6.1. Multiplexed Mode .....	153
<b>18. Oscillators and Clock Selection .....</b>	<b>157</b>
18.1. System Clock Selection .....	157
18.2. Programmable Internal Oscillator .....	159
18.2.1. Internal Oscillator Suspend Mode .....	159
18.3. Clock Multiplier .....	162
18.4. External Oscillator Drive Circuit .....	164
18.4.1. External Crystal Example .....	166
18.4.2. External RC Example .....	167
18.4.3. External Capacitor Example .....	167
<b>19. Port Input/Output .....</b>	<b>169</b>
19.1. Port I/O Modes of Operation .....	170
19.1.1. Port Pins Configured for Analog I/O .....	170
19.1.2. Port Pins Configured For Digital I/O .....	170
19.1.3. Interfacing Port I/O in a Multi-Voltage System .....	171
19.2. Assigning Port I/O Pins to Analog and Digital Functions .....	171
19.2.1. Assigning Port I/O Pins to Analog Functions .....	171
19.2.2. Assigning Port I/O Pins to Digital Functions .....	171
19.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions ...	172
19.3. Priority Crossbar Decoder .....	172
19.4. Port I/O Initialization .....	174
19.5. Port Match .....	179
19.6. Special Function Registers for Accessing and Configuring Port I/O .....	183
<b>20. Local Interconnect Network (LIN0) .....</b>	<b>193</b>
20.1. Software Interface with the LIN Controller .....	194
20.2. LIN Interface Setup and Operation .....	194
20.2.1. Mode Definition .....	194
20.2.2. Baud Rate Options: Manual or Autobaud .....	194
20.2.3. Baud Rate Calculations: Manual Mode .....	194
20.2.4. Baud Rate Calculations—Automatic Mode .....	196
20.3. LIN Master Mode Operation .....	197
20.4. LIN Slave Mode Operation .....	198
20.5. Sleep Mode and Wake-Up .....	199
20.6. Error Detection and Handling .....	199
20.7. LIN Registers .....	200
20.7.1. LIN Direct Access SFR Registers Definitions .....	200
20.7.2. LIN Indirect Access SFR Registers Definitions .....	202
<b>21. Controller Area Network (CAN0) .....</b>	<b>210</b>
21.1. Bosch CAN Controller Operation .....	211
21.1.1. CAN Controller Timing .....	211
21.1.2. CAN Register Access .....	212
21.1.3. Example Timing Calculation for 1 Mbit/Sec Communication .....	212
21.2. CAN Registers .....	214
21.2.1. CAN Controller Protocol Registers .....	214

## 4. Package Specifications

### 4.1. QFN-40 Package Specifications



**Figure 4.1. QFN-40 Package Drawing**

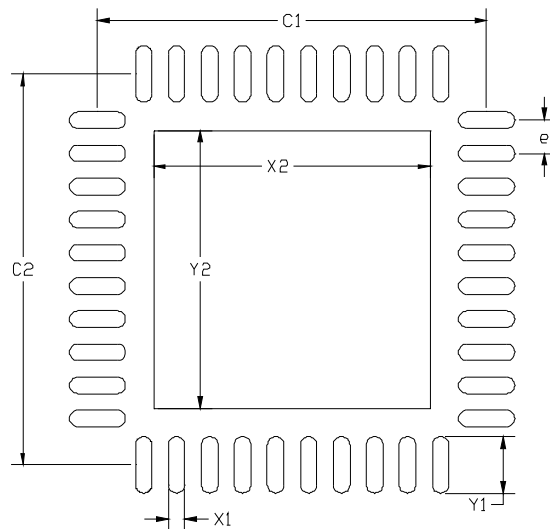
**Table 4.1. QFN-40 Package Dimensions**

Dimension	Min	Typ	Max
A	0.80	0.85	0.90
A1	0.00		0.05
b	0.18	0.23	0.28
D	6.00 BSC		
D2	4.00	4.10	4.20
e	0.50 BSC		
E	6.00 BSC		

Dimension	Min	Typ	Max
E2	4.00	4.10	4.20
L	0.35	0.40	0.45
L1			0.10
aaa			0.10
bbb			0.10
ddd			0.05
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.



**Figure 4.2. QFN-40 Landing Diagram**

**Table 4.2. QFN-40 Landing Diagram Dimensions**

Dimension	Min	Max	Dimension	Min	Max
C1	5.80	5.90	X2	4.10	4.20
C2	5.80	5.90	Y1	0.75	0.85
e	0.50 BSC		Y2	4.10	4.20
X1	0.15	0.25			

**Notes:**

**General**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimension and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-SM-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

**Solder Mask Design**

5. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60  $\mu$ m minimum, all the way around the pad.

**Stencil Design**

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 4x4 array of 0.80 mm square openings on a 1.05 mm pitch should be used for the center ground pad.

**Card Assembly**

10. A No-Clean, Type-3 solder paste is recommended.
11. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

**Table 5.9. ADC0 Electrical Characteristics**

VDDA = 1.8 to 2.75 V, -40 to +125 °C, VREF = 1.5 V (REFSL=0) unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
<b>DC Accuracy</b>					
Resolution			12		bits
Integral Nonlinearity		—	±0.5	±3	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Offset Error <sup>1</sup>		-10	3.0	10	LSB
Full Scale Error		-20	5.7	20	LSB
Offset Temperature Coefficient		—	7.7	—	ppm/°C
<b>Dynamic performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, 200 ksps)</b>					
Signal-to-Noise Plus Distortion		63	65	—	dB
Total Harmonic Distortion	Up to the 5th harmonic;	—	80	—	dB
Spurious-Free Dynamic Range		—	-82	—	dB
<b>Conversion Rate</b>					
SAR Conversion Clock		—	—	3.6	MHz
Conversion Time in SAR Clocks <sup>2</sup>		13	—	—	clocks
Track/Hold Acquisition Time <sup>3</sup>	VDDA ≥ 2.0 V	1.5	—	—	μs
	VDDA < 2.0 V	3.5	—	—	
Throughput Rate <sup>4</sup>	VDDA ≥ 2.0 V	—	—	200	ksps
<b>Analog Inputs</b>					
ADC Input Voltage Range <sup>5</sup>	gain = 1.0 (default)	0	—	VREF	V
	gain = n	0	—	VREF / n	
Absolute Pin Voltage with respect to GND		0	—	V <sub>IO</sub>	V
Sampling Capacitance		—	31	—	pF
Input Multiplexer Impedance		—	3	—	kΩ
<b>Power Specifications</b>					
Power Supply Current (VDDA supplied to ADC0)	Operating Mode, 200 ksps	—	1100	1500	μA
Burst Mode (Idle)		—	1100	1500	μA
Power-On Time		5	—	—	μs
Power Supply Rejection		—	-60	—	mV/V
<b>Notes:</b>					
<ol style="list-style-type: none"> <li>1. Represents one standard deviation from the mean. Offset and full-scale error can be removed through calibration.</li> <li>2. An additional 2 FCLK cycles are required to start and complete a conversion</li> <li>3. Additional tracking time may be required depending on the output impedance connected to the ADC input. See Section “6.2.1. Settling Time Requirements” on page 52.</li> <li>4. An increase in tracking time will decrease the ADC throughput.</li> <li>5. See Section “6.3. Selectable Gain” on page 53 for more information about the setting the gain.</li> </ol>					

## 6. 12-Bit ADC (ADC0)

The ADC0 on the C8051F55x/56x/57x consists of an analog multiplexer (AMUX0) with 33, 25, or 18 total input selections and a 200 kps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, programmable window detector, programmable attenuation (1:2), and hardware accumulator. The ADC0 subsystem has a special Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 6.1. ADC0 inputs are single-ended and may be configured to measure P0.0-P3.7, the Temperature Sensor output,  $V_{DD}$ , or GND with respect to GND. The voltage reference for ADC0 is selected as described in Section “6.6. Temperature Sensor” on page 67. ADC0 is enabled when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1, or when performing conversions in Burst Mode. ADC0 is in low power shutdown when AD0EN is logic 0 and no Burst Mode conversions are taking place.

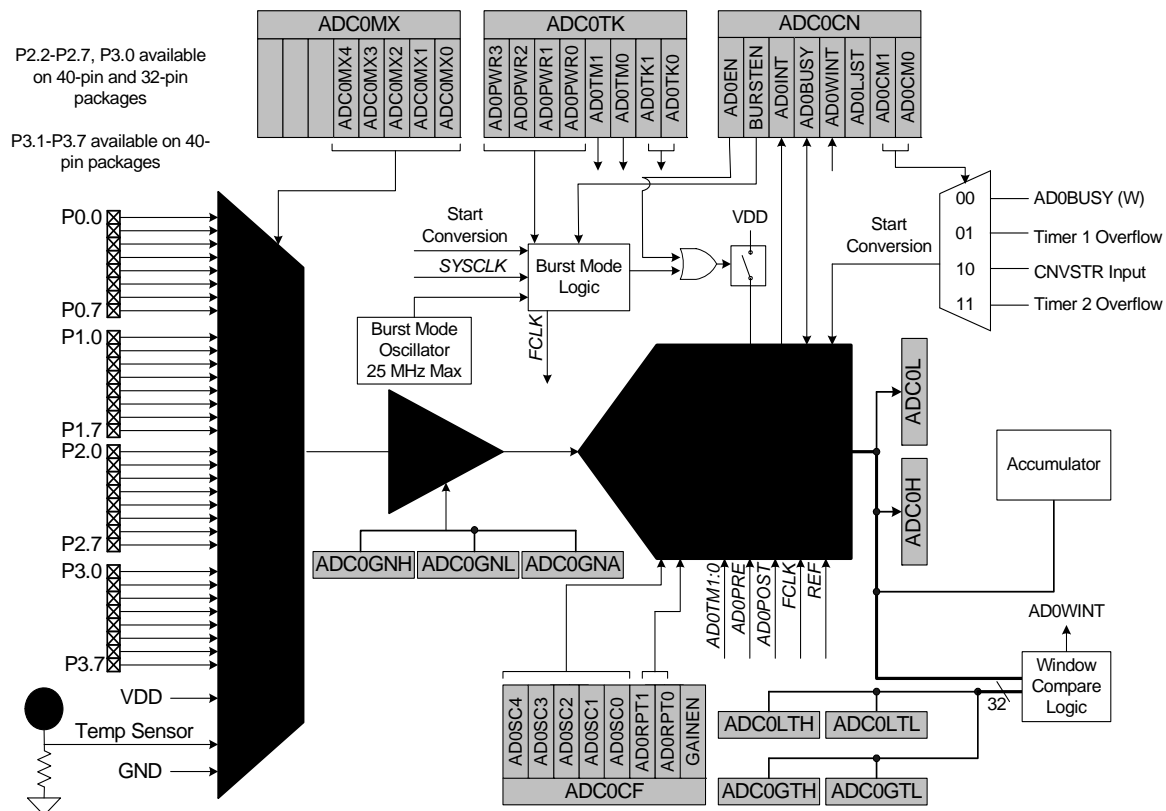


Figure 6.1. ADC0 Functional Block Diagram

# C8051F55x/56x/57x

## SFR Definition 6.4. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD0RPT[1:0]		GAINEN
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	0

SFR Address = 0xBC; SFR Page = 0x00

Bit	Name	Function
7:3	AD0SC[4:0]	<b>ADC0 SAR Conversion Clock Period Bits.</b> SAR Conversion clock is derived from system clock by the following equation, where AD0SC refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification table BURSTEN = 0: FCLK is the current system clock BURSTEN = 1: FCLK is a maximum of 30 MHz, independent of the current system clock.. $AD0SC = \frac{FCLK}{CLK_{SAR}} - 1$ <b>Note:</b> Round up the result of the calculation for AD0SC
2:1	A0RPT[1:0]	<b>ADC0 Repeat Count.</b> Controls the number of conversions taken and accumulated between ADC0 End of Conversion (ADCINT) and ADC0 Window Comparator (ADCWINT) interrupts. A convert start is required for each conversion unless Burst Mode is enabled. In Burst Mode, a single convert start can initiate multiple self-timed conversions. Results in both modes are accumulated in the ADC0H:ADC0L register. <b>When AD0RPT1–0 are set to a value other than '00', the AD0LJST bit in the ADC0CN register must be set to '0' (right justified).</b> 00: 1 conversion is performed. 01: 4 conversions are performed and accumulated. 10: 8 conversions are performed and accumulated. 11: 16 conversions are performed and accumulated.
0	GAINEN	<b>Gain Enable Bit.</b> Controls the gain programming. Refer to Section “6.3. Selectable Gain” on page 53 for information about using this bit.

## Notes on Registers, Operands and Addressing Modes:

**Rn**—Register R0–R7 of the currently selected register bank.

**@Ri**—Data RAM location addressed indirectly through R0 or R1.

**rel**—8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

**direct**—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

**#data**—8-bit constant

**#data16**—16-bit constant

**bit**—Direct-accessed bit in Data RAM or SFR

**addr11**—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

**addr16**—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

All mnemonics copyrighted © Intel Corporation 1980.

## 10.3. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic 1. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.



## SFR Definition 12.2. SFRPAGE: SFR Page

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

SFR Address = 0xA7; SFR Page = All Pages

Bit	Name	Function
7:0	SFRPAGE[7:0]	<p><b>SFR Page Bits.</b></p> <p>Represents the SFR Page the C8051 core uses when reading or modifying SFRs.</p> <p>Write: Sets the SFR Page.</p> <p>Read: Byte is the SFR page the C8051 core is using.</p> <p>When enabled in the SFR Page Control Register (SFR0CN), the C8051 core will automatically switch to the SFR Page that contains the SFRs of the corresponding peripheral/function that caused the interrupt, and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt). SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and not by reading/writing to the SFRPAGE register)</p>

**SFR Definition 13.1. IE: Interrupt Enable**

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA8; Bit-Addressable; SFR Page = All Pages

Bit	Name	Function
7	EA	<b>Enable All Interrupts.</b> Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	<b>Enable Serial Peripheral Interface (SPI0) Interrupt.</b> This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	<b>Enable Timer 2 Interrupt.</b> 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.
4	ES0	<b>Enable UART0 Interrupt.</b> This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	<b>Enable Timer 1 Interrupt.</b> 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.
2	EX1	<b>Enable External Interrupt 1.</b> This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	<b>Enable Timer 0 Interrupt.</b> 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.
0	EX0	<b>Enable External Interrupt 0.</b> This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

## 15. Power Management Modes

The C8051F55x/56x/57x devices have three software programmable power management modes: Idle, Stop, and Suspend. Idle mode and Stop mode are part of the standard 8051 architecture, while Suspend mode is an enhanced power-saving mode implemented by the high-speed oscillator peripheral.

Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Suspend mode is similar to Stop mode in that the internal oscillator and CPU are halted, but the device can wake on events such as a Port Match or Comparator low output. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode and Suspend mode consume the least power because the majority of the device is shut down with no clocks active. SFR Definition 15.1 describes the Power Control Register (PCON) used to control the C8051F55x/56x/57x devices' Stop and Idle power management modes. Suspend mode is controlled by the SUSPEND bit in the OSCICN register (SFR Definition 18.2).

Although the C8051F55x/56x/57x has Idle, Stop, and Suspend modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off oscillators lowers power consumption considerably, at the expense of reduced functionality.

### 15.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the hardware to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

**Note:** If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in 'C':
PCON |= 0x01;           // set IDLE bit
PCON = PCON;           // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h          ; set IDLE bit
MOV PCON, PCON          ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “16.6. PCA Watchdog Timer Reset” on page 142 for more information on the use and configuration of the WDT.

The CAN controller clock must be less than or equal to 25 MHz. If the CIP-51 system clock is above 25 MHz, the divider in the CAN0CFG register must be set to divide the CAN controller clock down to an appropriate speed.

## 21.1.2. CAN Register Access

The CAN controller clock divider selected in the CAN0CFG SFR affects how the CAN registers can be accessed. If the divider is set to 1, then a CAN SFR can immediately be read after it is written. If the divider is set to a value other than 1, then a read of a CAN SFR that has just been written must be delayed by a certain number of cycles. This delay can be performed using a NOP or some other instruction that does not attempt to read the register. This access limitation applies to read and read-modify-write instructions that occur immediately after a write. The full list of affected instructions is ANL, ORL, MOV, XCH, and XRL.

For example, with the CAN0CFG divider set to 1, the CAN0CN SFR can be accessed as follows:

```
MOV CAN0CN, #041      ; Enable access to Bit Timing Register
MOV R7, CAN0CN         ; Copy CAN0CN to R7
```

With the CAN0CFG divider set to /2, the same example code requires an additional NOP:

```
MOV CAN0CN, #041      ; Enable access to Bit Timing Register
NOP                   ; Wait for write to complete
MOV R7, CAN0CN         ; Copy CAN0CN to R7
```

The number of delay cycles required is dependent on the divider setting. With a divider of 2, the read must wait for 1 system clock cycle. With a divider of 4, the read must wait 3 system clock cycles, and with the divider set to 8, the read must wait 7 system clock cycles. The delay only needs to be applied when reading the same register that was written. The application can write and read other CAN SFRs without any delay.

## 21.1.3. Example Timing Calculation for 1 Mbit/Sec Communication

This example shows how to configure the CAN controller timing parameters for a 1 Mbit/Sec bit rate. Table 21.1 shows timing-related system parameters needed for the calculation.

**Table 21.1. Background System Information**

Parameter	Value	Description
CIP-51 system clock (SYSCLK)	24 MHz	Internal Oscillator Max
CAN controller clock ( $f_{sys}$ )	24 MHz	CAN0CFG divider set to 1
CAN clock period ( $t_{sys}$ )	41.667 ns	Derived from $1/f_{sys}$
CAN time quantum ( $t_q$ )	41.667 ns	Derived from $t_{sys} \times BRP^{1,2}$
CAN bus length	10 m	5 ns/m signal delay between CAN nodes
Propagation delay time <sup>3</sup>	400 ns	2 x (transceiver loop delay + bus line delay)
<b>Notes:</b> <ol style="list-style-type: none"> <li>1. The CAN time quantum is the smallest unit of time recognized by the CAN controller. Bit timing parameters are specified in integer multiples of the time quantum.</li> <li>2. The Baud Rate Prescaler (BRP) is defined as the value of the BRP Extension Register plus 1. The BRP extension register has a reset value of 0x0000. The BRP has a reset value of 1.</li> <li>3. Based on an ISO-11898 compliant transceiver. CAN does not specify a physical layer.</li> </ol>		

Each bit transmitted on a CAN network has 4 segments (Sync\_Seg, Prop\_Seg, Phase\_Seg1, and Phase\_Seg2), as shown in Figure 18.3. The sum of these segments determines the CAN bit time (1/bit rate). In this example, the desired bit rate is 1 Mbit/sec; therefore, the desired bit time is 1000 ns.

**SFR Definition 22.1. SMB0CF: SMBus Clock/Configuration**

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0x00

Bit	Name	Function
7	ENSMB	<b>SMBus Enable.</b> This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	<b>SMBus Slave Inhibit.</b> When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	<b>SMBus Busy Indicator.</b> This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	<b>SMBus Setup and Hold Time Extension Enable.</b> This bit controls the SDA setup and hold times according to Table 22.2. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	<b>SMBus SCL Timeout Detection Enable.</b> This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	<b>SMBus Free Timeout Detection Enable.</b> When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	<b>SMBus Clock Source Selection.</b> These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 22.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

### 22.5.2. 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 22.6 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.

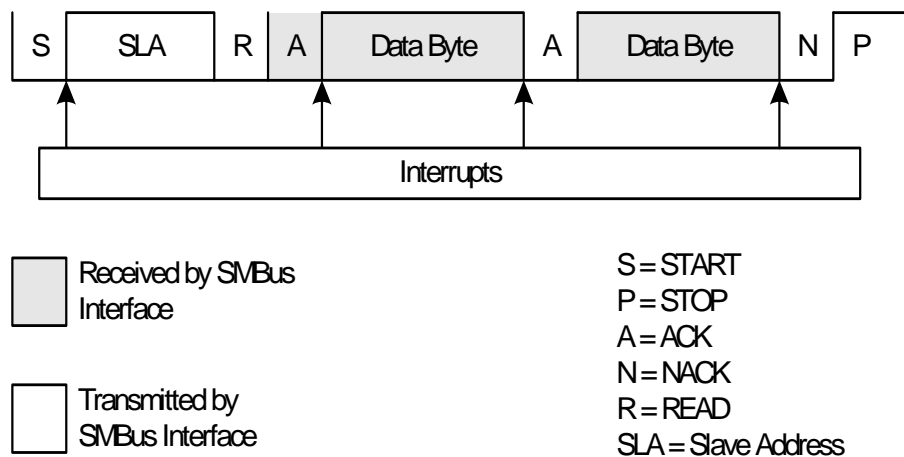
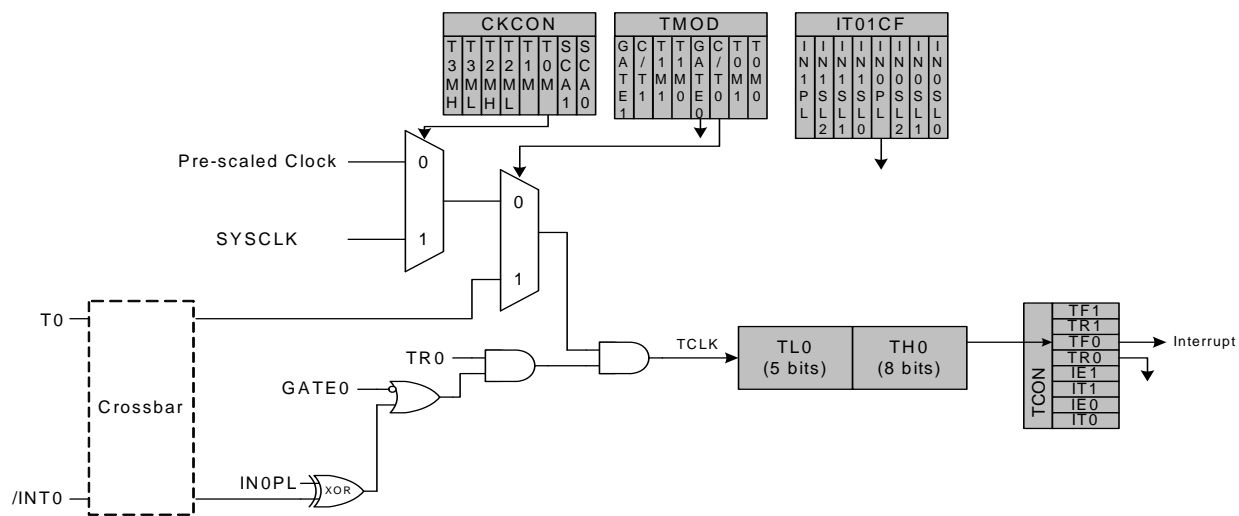


Figure 22.6. Typical Master Read Sequence



**Figure 25.1. T0 Mode 0 Block Diagram**

## 25.1.2. 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.

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

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “13.3. External Interrupts INT0 and INT1” on page 122 for details on the external input signals INT0 and INT1).

# C8051F55x/56x/57x

## SFR Definition 25.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8C; SFR Page = All Pages

Bit	Name	Function
7:0	TH0[7:0]	<b>Timer 0 High Byte.</b> The TH0 register is the high byte of the 16-bit Timer 0.

## SFR Definition 25.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8D; SFR Page = All Pages

Bit	Name	Function
7:0	TH1[7:0]	<b>Timer 1 High Byte.</b> The TH1 register is the high byte of the 16-bit Timer 1.



# C8051F55x/56x/57x

## SFR Definition 25.8. 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	<b>Timer 2 High Byte Overflow Flag.</b> 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	<b>Timer 2 Low Byte Overflow Flag.</b> 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	<b>Timer 2 Low Byte Interrupt Enable.</b> 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	<b>Timer 2 Capture Mode Enable.</b> 0: Timer 2 Capture Mode is disabled. 1: Timer 2 Capture Mode is enabled.
3	T2SPLIT	<b>Timer 2 Split Mode Enable.</b> 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	<b>Timer 2 Run Control.</b> 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	<b>Timer 2 External Clock Select.</b> 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).

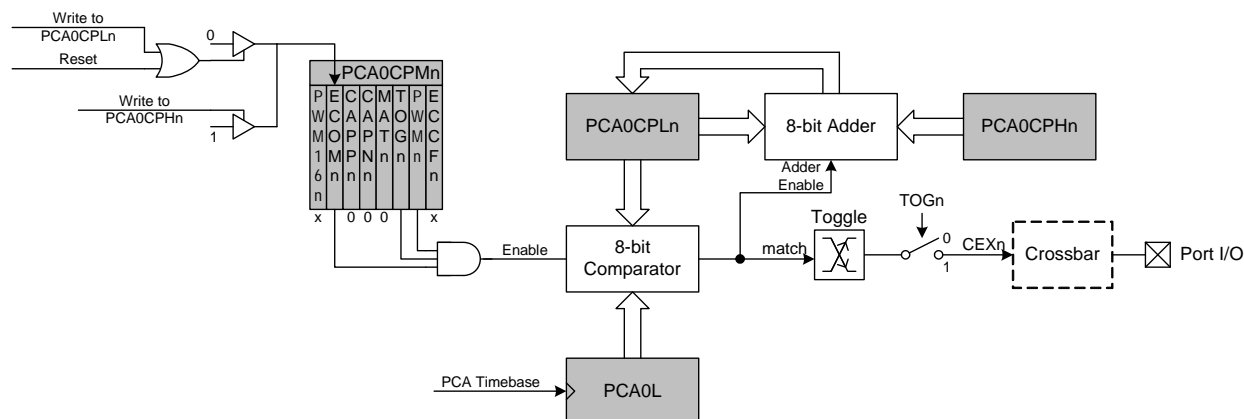
# C8051F55x/56x/57x

## SFR Definition 25.13. TMR3CN: Timer 3 Control

Bit	7	6	5	4	3	2	1	0
Name	TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3		T3XCLK
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 = 0x91; SFR Page = 0x00

Bit	Name	Function
7	TF3H	<b>Timer 3 High Byte Overflow Flag.</b> Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF3L	<b>Timer 3 Low Byte Overflow Flag.</b> Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. TF3L will be set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.
5	TF3LEN	<b>Timer 3 Low Byte Interrupt Enable.</b> When set to 1, this bit enables Timer 3 Low Byte interrupts. If Timer 3 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
4	TF3CEN	<b>Timer 3 Capture Mode Enable.</b> 0: Timer 3 Capture Mode is disabled. 1: Timer 3 Capture Mode is enabled.
3	T3SPLIT	<b>Timer 3 Split Mode Enable.</b> When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.
2	TR3	<b>Timer 3 Run Control.</b> Timer 3 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in split mode.
1	Unused	Read = 0b; Write = Don't Care
0	T3XCLK	<b>Timer 3 External Clock Select.</b> This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 3 clock is the system clock divided by 12. 1: Timer 3 clock is the external clock divided by 8 (synchronized with SYSCLK).



**Figure 26.7. PCA Frequency Output Mode**

## 26.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

### 26.3.5.1. 8-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 26.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module's capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 26.2.

**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.

$$\text{Duty Cycle} = \frac{(256 - \text{PCA0CPHn})}{256}$$

### Equation 26.2. 8-Bit PWM Duty Cycle

Using Equation 26.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

**Table 26.3. Watchdog Timer Timeout Intervals<sup>1</sup>**

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
<b>Notes:</b> <ol style="list-style-type: none"><li>1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.</li><li>2. Internal SYSCLK reset frequency = Internal Oscillator divided by 128.</li></ol>		

# C8051F55x/56x/57x

## C2 Register Definition 27.2. DEVICEID: C2 Device ID

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

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

Bit	Name	Function
7:0	DEVICEID[7:0]	<b>Device ID.</b> This read-only register returns the 8-bit device ID: 0x22 (C8051F55x/56x/57x).

## C2 Register Definition 27.3. 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 = 0xF

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