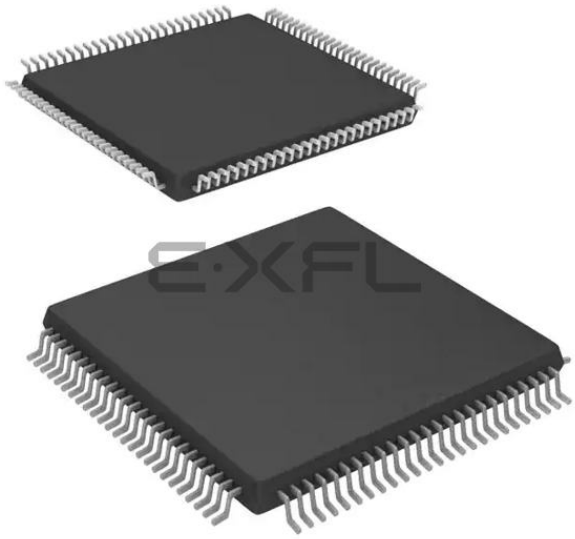


Welcome to E-XFL.COM

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x16b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f060

1.6. Controller Area Network

The C8051F060/1/2/3 devices feature a Controller Area Network (CAN) controller that implements serial communication using the CAN protocol. The CAN controller 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 C8051 RAM), a message handler state machine, and control registers.

The CAN controller can operate at bit rates up to 1 Mbit/second. Silicon Labs CAN has 32 message objects each having its own identifier mask used for acceptance filtering of received messages. Incoming data, message objects and 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 C8051 MCU. In this way, minimal CPU bandwidth is used for CAN communication. The C8051 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFR) in the C8051.

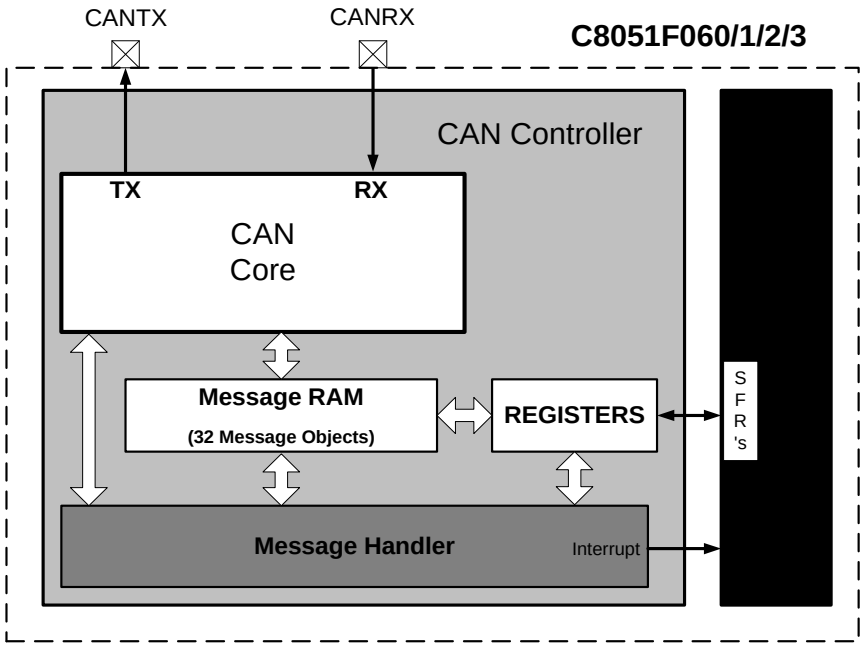


Figure 1.11. CAN Controller Overview

5.3. ADC Modes of Operation

ADC0 and ADC1 have a maximum conversion speed of 1 Msps. The conversion clocks for the ADCs are derived from the system clock. The ADCnSC bits in the ADCnCF register determine how many system clocks (from 1 to 16) are used for each conversion clock.

5.3.1. Starting a Conversion

For ADC0, conversions can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1, AD0CM0) in ADC0CN. For ADC0, conversions may be initiated by:

1. Writing a '1' to the AD0BUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR0;
4. A Timer 2 overflow (i.e. timed continuous conversions).

ADC1 conversions can be initiated in five different ways, according to the ADC1 Start of Conversion Mode bits (AD1CM2-AD1CM0) in ADC1CN. For ADC1, conversions may be initiated by:

1. Writing a '1' to the AD1BUSY bit of ADC1CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR1;
4. A Timer 2 overflow (i.e. timed continuous conversions);
5. Writing a '1' to the AD0BUSY bit of ADC0CN.

The ADnBUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of ADnBUSY triggers an interrupt (when enabled) and sets the ADnINT interrupt flag (ADCnCN.5). In single-ended mode, the converted data for ADCn is available in the ADCn data word MSB and LSB registers, ADCnH, ADCnL. In differential mode, the converted data (combined from ADC0 and ADC1) is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L.

When initiating conversions by writing a '1' to ADnBUSY, the ADnINT bit should be polled to determine when a conversion has completed (ADCn interrupts may also be used). The recommended polling procedure is shown below.

Step 1. Write a '0' to ADnINT;
Step 2. Write a '1' to ADnBUSY;
Step 3. Poll ADnINT for '1';
Step 4. Process ADCn data.

When an external start-of-conversion source is required in differential mode the two pins (CNVSTR0 and CNVSTR1) should be tied together.

5.3.2. Tracking Modes

The ADnTM bit in register ADCnCN controls the ADCn track-and-hold mode. When the ADC is enabled, the ADC input is continuously tracked when a conversion is not in progress. When the ADnTM bit is logic 1, each conversion is preceded by a tracking period (after the start-of-conversion signal). When the CNVSTRn signal is used to initiate conversions, the ADC will track until a rising edge occurs on the CNVSTRn pin (see Figure 5.4 and Table 5.1 for conversion timing parameters). Setting ADnTM to 1 can be useful to ensure that settling time requirements are met when an external multiplexer is used on the analog input (see Section "5.3.3. Settling Time Requirements" on page 56).

Figure 5.8. ADC1CF: ADC1 Configuration Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD1SC3	AD1SC2	AD1SC1	AD1SC0	AD1SCAL	AD1GCAL	AD1LCAL	AD1OCAL	11110000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBC
SFR Page: 1

Bits 7-4: AD1SC3-0: ADC1 SAR Conversion Clock Period Bits.
SAR Conversion clock is divided down from the system clock according to the AD1SC bits (AD1SC3-0). The number of system clocks used for each SAR conversion clock is equal to AD1SC + 1. (Note: the ADC1 SAR Conversion Clock should be less than or equal to 25 MHz). See Table 5.1 for conversion timing details.

Bit 3: AD1SCAL: System Calibration Enable.
0: Internal ground and reference voltage are used for offset and gain calibration.
1: External voltages can be used for offset and gain calibration.

Bit 2: AD1GCAL: Gain Calibration.
Read:
0: Gain Calibration is completed or not yet started.
1: Gain Calibration is in progress.
Write:
0: No Effect.
1: Initiates a gain calibration if ADC1 is idle.

Bit 1: AD1LCAL: Linearity Calibration
Read
0: Linearity Calibration is completed or not yet started
1: Linearity Calibration is in progress
Write
0: No Effect
1: Initiates a linearity calibration if ADC1 is idle

Bit 0: AD1OCAL: Offset Calibration.
Read:
0: Offset Calibration is completed or not yet started.
1: Offset Calibration is in progress.
Write:
0: No Effect.
1: Initiates an offset calibration if ADC1 is idle.

Figure 5.16. ADC1H: ADC1 Data Word MSB Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBF
SFR Page: 1

Bits 7-0: ADC1 Data Word High-Order Bits.

Figure 5.17. ADC1L: ADC1 Data Word LSB Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBE
SFR Page: 1

Bits 7-0: ADC1 Data Word Low-Order Bits.

Figure 5.18. ADC1 Data Word Example

16-bit ADC1 Data Word appears in the ADC1 Data Word Registers as follows:

Example: ADC1 Data Word Conversion Map, AIN1 Input in Single-Ended Mode (AMX1SL = 0x00)

AIN1-AIN1G (Volts)	**ADC1H:ADC1L**
VREF * (65535/65536)	0xFFFF
VREF / 2	0x8000
VREF * (32767/65536)	0x7FFF
0	0x0000

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n \quad \text{; 'n' = 16}$$

For differential mode, the differential data word appears in ADC0H and ADC0L. The single-ended ADC1 results are always present in ADC1H and ADC1L, regardless of the operating mode.

7. 10-Bit ADC (ADC2, C8051F060/1/2/3)

The ADC2 subsystem for the C8051F060/1/2/3 consists of an analog multiplexer (referred to as AMUX2), and a 200 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 7.1). The AMUX2, data conversion modes, and window detector can all be configured from within software via the Special Function Registers shown in Figure 7.1. ADC2 operates in both Single-ended and Differential modes, and may be configured to measure any of the pins on Port 1, or the Temperature Sensor output. The ADC2 subsystem is enabled only when the AD2EN bit in the ADC2 Control register (ADC2CN) is set to logic 1. The ADC2 subsystem is in low power shutdown when this bit is logic 0.

Figure 7.1. ADC2 Functional Block Diagram

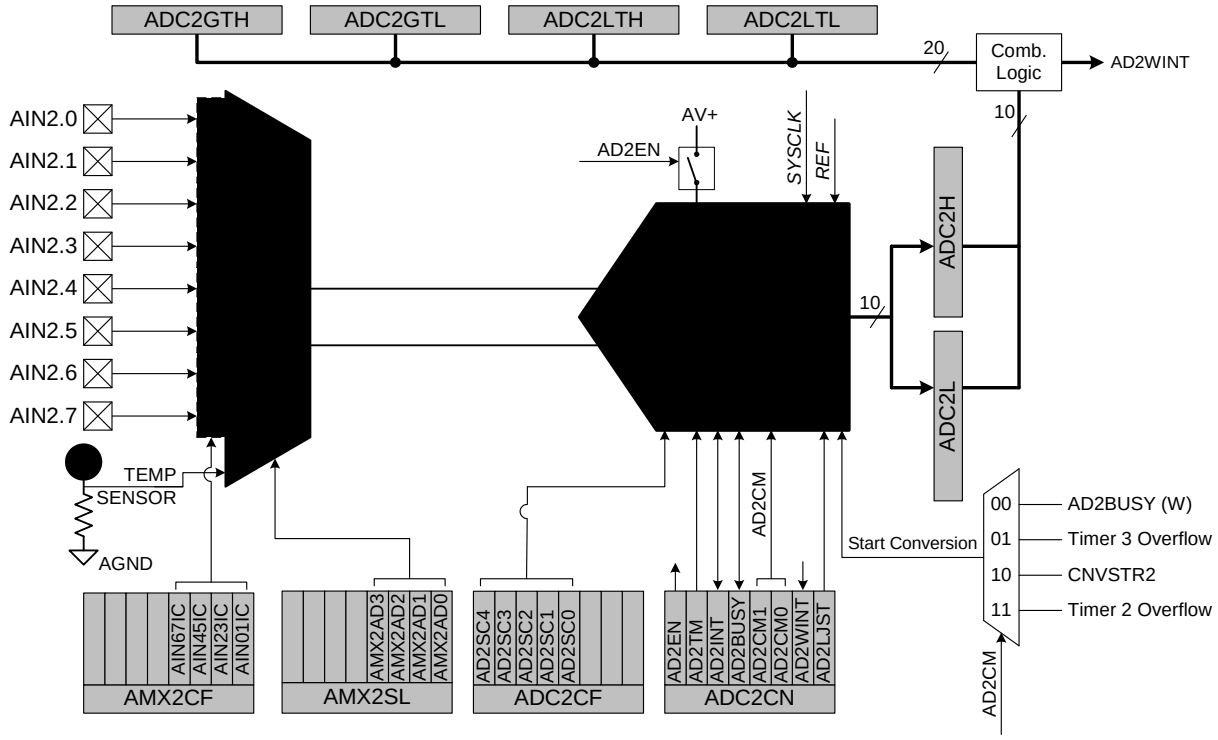


Figure 7.13. ADC2LTH: ADC2 Less-Than Data High Byte Register

SFR Page: 2
SFR Address: 0xC7

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	

Bits7-0: High byte of ADC2 Less-Than Data Word.

Figure 7.14. ADC2LTL: ADC2 Less-Than Data Low Byte Register

SFR Page: 2
SFR Address: 0xC6

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	

Bits7-0: Low byte of ADC2 Less-Than Data Word.

Programming and Debugging Support

A JTAG-based serial interface is provided for in-system programming of the Flash program memory and communication with on-chip debug support logic. The re-programmable Flash can also be read and changed a single byte at a time by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints and watch points, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debug is completely non-intrusive and non-invasive, requiring no RAM, Stack, timers, or other on-chip resources.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) which interfaces to the CIP-51 via its JTAG port to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

13.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set; standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

13.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 13.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

13.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and writing to on-chip program Flash memory. The Flash access feature provides a mechanism for user software to update program code and use the program memory space for non-volatile data storage (see Section “16. Flash Memory” on page 177). The External Memory Interface provides a fast access to off-chip XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to Section “17. External Data Memory Interface and On-Chip XRAM” on page 187 for details.

Figure 13.16. PSW: Program Status Word

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

SFR Address: 0xD0
SFR Page: All Pages

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.

Figure 13.24. EIP2: Extended Interrupt Priority 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PDMA0	PS1	PCAN0	PADC2	PWADC2	PT4	PADC1	PT3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF7
SFR Page: All Pages

Bit7: PDMA0: DMA0 Interrupt Priority Control.
This bit sets the priority of the DMA0 interrupt.
0: DMA0 interrupt set to low priority.
1: DMA0 interrupt set to high priority.

Bit6: PS1: UART1 Interrupt Priority Control.
This bit sets the priority of the UART1 interrupt.
0: UART1 interrupt set to low priority.
1: UART1 interrupt set to high priority.

Bit5: PCAN0: CAN Interrupt Priority Control.
This bit sets the priority of the CAN Interrupt.
0: CAN Interrupt set to low priority level.
1: CAN Interrupt set to high priority level.

Bit4: PADC2: ADC2 End Of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC2 End of Conversion interrupt.
0: ADC2 End of Conversion interrupt set to low priority.
1: ADC2 End of Conversion interrupt set to high priority.

Bit3: PWADC2: ADC2 Window Comparator Interrupt Priority Control.
0: ADC2 Window interrupt set to low priority.
1: ADC2 Window interrupt set to high priority.

Bit2: PT4: Timer 4 Interrupt Priority Control.
This bit sets the priority of the Timer 4 interrupt.
0: Timer 4 interrupt set to low priority.
1: Timer 4 interrupt set to high priority.

Bit1: PADC1: ADC End of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC1 End of Conversion Interrupt.
0: ADC1 End of Conversion interrupt set to low priority level.
1: ADC1 End of Conversion interrupt set to high priority level.

Bit0: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupts.
0: Timer 3 interrupt set to low priority level.
1: Timer 3 interrupt set to high priority level.

Table 14.1. Reset Electrical Characteristics

-40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
/RST Output Low Voltage	I_{OL} = 8.5 mA, VDD = 2.7 V to 3.6 V			0.6	V
/RST Input High Voltage		0.7 x VDD			V
/RST Input Low Voltage				0.3 x VDD	
/RST Input Leakage Current	/RST = 0.0 V		50		µA
VDD for /RST Output Valid		1.0			V
AV+ for /RST Output Valid		1.0			V
VDD POR Threshold (V_{RST})		2.40	2.55	2.70	V
Minimum /RST Low Time to Generate a System Reset		10			ns
Reset Time Delay	/RST rising edge after VDD crosses V_{RST} threshold	80	100	120	ms
Missing Clock Detector Time-out	Time from last system clock to reset initiation	100	220	500	µs

Figure 15.5. OSCXCN: External Oscillator Control Register

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8C
SFR Page: F

Bit7: XTLVLD: Crystal Oscillator Valid Flag.
(**Valid 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 (External CMOS Clock input on XTAL1 pin).
011: External CMOS Clock Mode with divide by 2 stage (External CMOS Clock input on XTAL1 pin).
10x: RC/C Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: Unused. Read = 0, Write = don't care.

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 ≤ 32 kHz	f ≤ 25 kHz	K Factor = 0.87
001	32 kHz < f ≤ 84 kHz	25 kHz < f ≤ 50 kHz	K Factor = 2.6
010	84 kHz < f ≤ 225 kHz	50 kHz < f ≤ 100 kHz	K Factor = 7.7
011	225 kHz < f ≤ 590 kHz	100 kHz < f ≤ 200 kHz	K Factor = 22
100	590 kHz < f ≤ 1.5 MHz	200 kHz < f ≤ 400 kHz	K Factor = 65
101	1.5 MHz < f ≤ 4 MHz	400 kHz < f ≤ 800 kHz	K Factor = 180
110	4 MHz < f ≤ 10 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 664
111	10 MHz < f ≤ 30 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590

CRYSTAL MODE (Circuit from Figure 15.1, Option 1; XOSCMD = 11x).
Choose XFCN value to match crystal frequency.

RC MODE (Circuit from Figure 15.1, Option 2; XOSCMD = 10x).
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R * C)$, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pull-up resistor value in kΩ

C MODE (Circuit from Figure 15.1, Option 3; XOSCMD = 10x).
Choose K Factor (KF) for the oscillation frequency desired:
f = KF / (C * VDD), where
f = frequency of oscillation in MHz
C = capacitor value on XTAL1, XTAL2 pins in pF
VDD = Power Supply on MCU in volts

20.4.3. Data Register

The SMBus0 Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software can read or write to this register while the SI flag is set to logic 1; software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag reads logic 0 since the hardware may be in the process of shifting a byte of data in or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. Therefore, SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in SMB0DAT.

Figure 20.10. SMB0DAT: SMBus0 Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC2
SFR Page: 0

Bits7-0: SMB0DAT: SMBus0 Data.
The SMB0DAT register contains a byte of data to be transmitted on the SMBus0 serial interface or a byte that has just been received on the SMBus0 serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.3) is set to logic 1. When the SI flag is not set, the system may be in the process of shifting data and the CPU should not attempt to access this register.

20.4.4. Address Register

The SMB0ADR Address register holds the slave address for the SMBus0 interface. In slave mode, the seven most-significant bits hold the 7-bit slave address. The least significant bit (Bit0) is used to enable the recognition of the general call address (0x00). If Bit0 is set to logic 1, the general call address will be recognized. Otherwise, the general call address is ignored. The contents of this register are ignored when

21.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 21.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 21.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

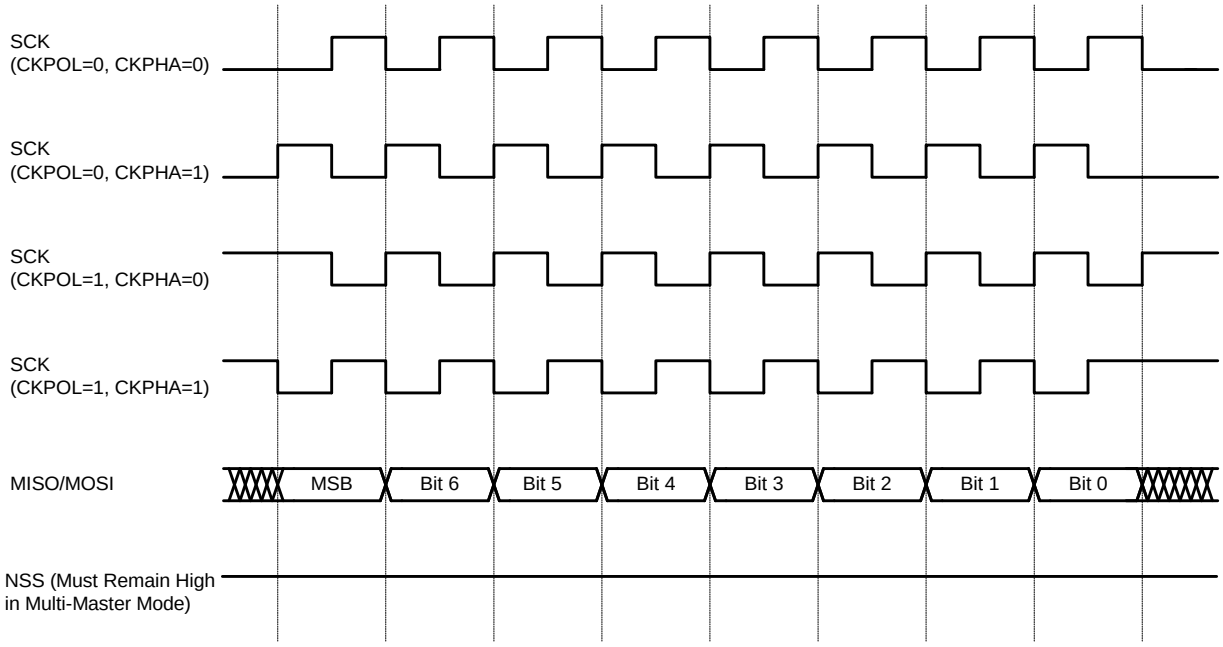
4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 21.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

21.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 21.5. For slave mode, the clock and data relationships are shown in Figure 21.6 and Figure 21.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in Figure 21.10 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

Figure 21.5. Master Mode Data/Clock Timing



22.1. UART0 Operational Modes

UART0 provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON0 register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 22.1.

Table 22.1. UART0 Modes

Mode	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	Synchronous	SYSCLK / 12	8	None
1	Asynchronous	Timer 1, 2, 3, or 4 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK / 32 or SYSCLK / 64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1, 2, 3, or 4 Overflow	9	1 Start, 1 Stop

22.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX0 pin. The TX0 pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions (see the interconnect diagram in Figure 22.3).

Data transmission begins when an instruction writes a data byte to the SBUF0 register. Eight data bits are transferred LSB first (see the timing diagram in Figure 22.2), and the TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the eighth bit time. Data reception begins when the REN0 Receive Enable bit (SCON0.4) is set to logic 1 and the RI0 Receive Interrupt Flag (SCON0.0) is cleared. One cycle after the eighth bit is shifted in, the RI0 flag is set and reception stops until software clears the RI0 bit. An interrupt will occur if enabled when either TI0 or RI0 are set.

23. UART1

UART1 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “23.1. Enhanced Baud Rate Generation” on page 278). Received data buffering allows UART1 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART1 has two associated SFRs: Serial Control Register 1 (SCON1) and Serial Data Buffer 1 (SBUF1). The single SBUF1 location provides access to both transmit and receive registers. Reading SBUF1 accesses the buffered Receive register; writing SBUF1 accesses the Transmit register.

With UART1 interrupts enabled, an interrupt is generated each time a transmit is completed (TI1 is set in SCON1), or a data byte has been received (RI1 is set in SCON1). The UART1 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART1 interrupt (transmit complete or receive complete).

Figure 23.1. UART1 Block Diagram

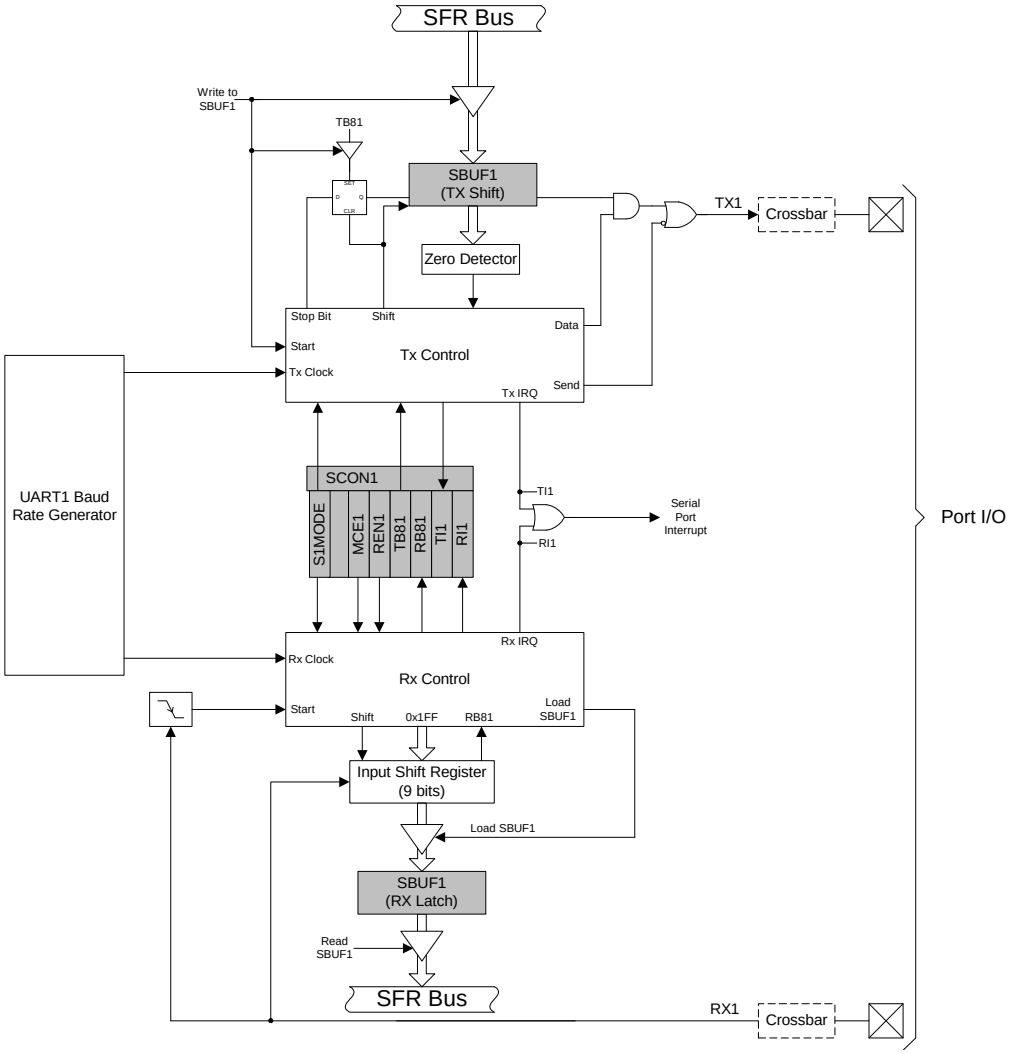


Table 23.5. Timer Settings for Standard Baud Rates Using an External Oscillator

	Frequency: 11.0592 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscilla-tor Divide Factor	Timer Clock Source	SCA1-SCA0 (pre-scale select)†	T1M†	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	48	SYSCLK	XX	1	0xE8
	115200	0.00%	96	SYSCLK	XX	1	0xD0
	57600	0.00%	192	SYSCLK	XX	1	0xA0
	28800	0.00%	384	SYSCLK	XX	1	0x40
	14400	0.00%	768	SYSCLK / 12	00	0	0xE0
	9600	0.00%	1152	SYSCLK / 12	00	0	0xD0
	2400	0.00%	4608	SYSCLK / 12	00	0	0x40
	1200	0.00%	9216	SYSCLK / 48	10	0	0xA0
SYSCLK from Internal Osc.	230400	0.00%	48	EXTCLK / 8	11	0	0xFD
	115200	0.00%	96	EXTCLK / 8	11	0	0xFA
	57600	0.00%	192	EXTCLK / 8	11	0	0xF4
	28800	0.00%	384	EXTCLK / 8	11	0	0xE8
	14400	0.00%	768	EXTCLK / 8	11	0	0xD0
	9600	0.00%	1152	EXTCLK / 8	11	0	0xB8

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

Table 23.6. Timer Settings for Standard Baud Rates Using an External Oscillator

	Frequency: 3.6864 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscilla-tor Divide Factor	Timer Clock Source	SCA1-SCA0 (pre-scale select)†	T1M†	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	16	SYSCLK	XX	1	0xF8
	115200	0.00%	32	SYSCLK	XX	1	0xF0
	57600	0.00%	64	SYSCLK	XX	1	0xE0
	28800	0.00%	128	SYSCLK	XX	1	0xC0
	14400	0.00%	256	SYSCLK	XX	1	0x80
	9600	0.00%	384	SYSCLK	XX	1	0x40
	2400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	1200	0.00%	3072	SYSCLK / 12	00	0	0x80
SYSCLK from Internal Osc.	230400	0.00%	16	EXTCLK / 8	11	0	0xFF
	115200	0.00%	32	EXTCLK / 8	11	0	0xFE
	57600	0.00%	64	EXTCLK / 8	11	0	0xFC
	28800	0.00%	128	EXTCLK / 8	11	0	0xF8
	14400	0.00%	256	EXTCLK / 8	11	0	0xF0
	9600	0.00%	384	EXTCLK / 8	11	0	0xE8

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

Document Change List

Revision 1.1 to Revision 1.2

- Added four part numbers: C8051F064, C8051F065, C8051F066, and C8051F067.
- Modified all sections to describe functionality of the four new parts.
- Revised and expanded Flash Chapter with clearer descriptions of Flash security features.
- UART0 Chapter, Section 22.3: “FE0 in register SCON0” changed to “FE0 in register SSTA0”.
- UART0 Chapter: Updated and clarified baud rate equations.
- Port I/O Chapter, Section 18.2: Added a note in text body that Port 4-7 registers are all on SFR Page F.
- Comparators Chapter: Updated Table 12.1 “Comparator Electrical Characteristics”.
- CIP51 Chapter: Section 13.4.1: Added note regarding IDLE mode operation.
- ADC2 Chapter: AD2LJST bit removed from ADC2CF register description (AD2LJST is in the ADC2CN register).
- ADC2 Chapter: Updated Table 7.1 “ADC2 Electrical Characteristics” and Figure 7.2 “Temperature Sensor Transfer Function” with temperature sensor information.
- ADC0/ADC1 Chapter: Tracking/Conversion timing when ADnTM = 1 is shown in Figure 5.4 and Table 5.1. References to “18” or “16” SAR clocks of tracking were removed.
- DACs Chapter, Table 8.1 “DAC Electrical Characteristics”: Changed “Gain Error” to “Full-Scale Error”.
- SMBus Chapter, Figure 20.9 SMB0CR: Changed “1.125” to “1.125 * 10^6”.
- PCA Chapter, Figure 25.12 PCA0CPMn: Bit 0 name changed to “ECCFn” (from incorrect “EECFn”).
- JTAG Chapter, Figure 26.3 FLASHCON: Bit 7 description corrected. Bit 7 is SFLE, allowing access to the Scratchpad memory area.
- CAN Chapter: Added text “The CAN controller’s clock (f_{sys}, or CAN_CLK in the C_CAN User’s Guide) is equal to the CIP-51 MCU’s clock (SYSCLK).”
- Table 4.1 “Pin Descriptions”, MONEN: Added text “Recommended configuration is to connect directly to VDD.”
- Timers Chapter: All references to “DCEN” and “DECEN” corrected to “DCENn”.
- Timers Chapter, Equation 24.1: Equation was corrected to “Fsq = Ftclk / (2*(65536-RCAPn))”. This equation is valid for a timer counting up or down.
- Timers Chapter, Figure 24.14 TMRnCF: Corrected Bit 1 description. For square-wave output, CP/RLn = 0, C/Tn = 0, TnOE = 1.
- VREF Chapters: Added VREF Power Supply Current to VREF Electrical Characteristics Tables.
- PCA Chapter: Added Note about writing PCA0CPLn and PCA0CPHn to sections for SW Timer Mode, High-Speed Output Mode, Frequency Output Mode, 8-bit PWM Mode, and 16-bit PWM Mode.
- Oscillators Chapter, Table 15.1 “Internal Oscillator Electrical Characteristics”: Updated typical supply current.
- Table 3.1 “Global DC Electrical Characteristics”, Updated supply current numbers with additional characterization data.
- ADC0/ADC1 Chapter: Table 5.2 “ADC0 and ADC1 Electrical Characteristics”, Updated supply current numbers with additional characterization data.
- ADC0/ADC1 Chapter: Table 5.3 “Voltage Reference 0 and 1 Electrical Characteristics”, Updated Output Voltage numbers with characterization data.
- Figure 4.3 “TQFP-100 Package Drawing”, Added “L” Dimension.
- Figure 4.6 “TQFP-64 Package Drawing”, Added “L” Dimension.