



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 ² C), CANbus, 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-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f561-iqr

Table 5.3. Port I/O DC Electrical Characteristics

$V_{DD} = 1.8$ to 2.75 V, -40 to $+125$ °C unless otherwise specified.

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	$I_{OH} = -3$ mA, Port I/O push-pull $I_{OH} = -10$ μ A, Port I/O push-pull $I_{OH} = -10$ mA, Port I/O push-pull	$V_{IO} - 0.4$ $V_{IO} - 0.02$ —	— — $V_{IO} - 0.7$	— — —	V
Output Low Voltage	$V_{IO} = 1.8$ V: $I_{OL} = 70$ μ A $I_{OL} = 8.5$ mA $V_{IO} = 2.7$ V: $I_{OL} = 70$ μ A $I_{OL} = 8.5$ mA $V_{IO} = 5.25$ V: $I_{OL} = 70$ μ A $I_{OL} = 8.5$ mA	— — — — — —	— — — — — —	50 750 45 550 40 400	mV
Input High Voltage	$V_{REGIN} = 5.25$ V	$0.7 \times V_{IO}$	—		V
Input Low Voltage	$V_{REGIN} = 2.7$ V	—	—	$0.3 \times V_{IO}$	V
Input Leakage Current	Weak Pullup Off Weak Pullup On, $V_{IO} = 2.1$ V, $V_{IN} = 0$ V, $V_{DD} = 1.8$ V Weak Pullup On, $V_{IO} = 2.6$ V, $V_{IN} = 0$ V, $V_{DD} = 2.6$ V Weak Pullup On, $V_{IO} = 5.0$ V, $V_{IN} = 0$ V, $V_{DD} = 2.6$ V	— — — —	— 7 17 49	± 2 9 22 115	μ A

6.5. ADC0 Analog Multiplexer

ADC0 includes an analog multiplexer to enable multiple analog input sources. Any of the following may be selected as an input: P0.0–P3.7, the on-chip temperature sensor, the core power supply (V_{DD}), or ground (GND). **ADC0 is single-ended and all signals measured are with respect to GND.** The ADC0 input channels are selected using the ADC0MX register as described in SFR Definition 6.13.

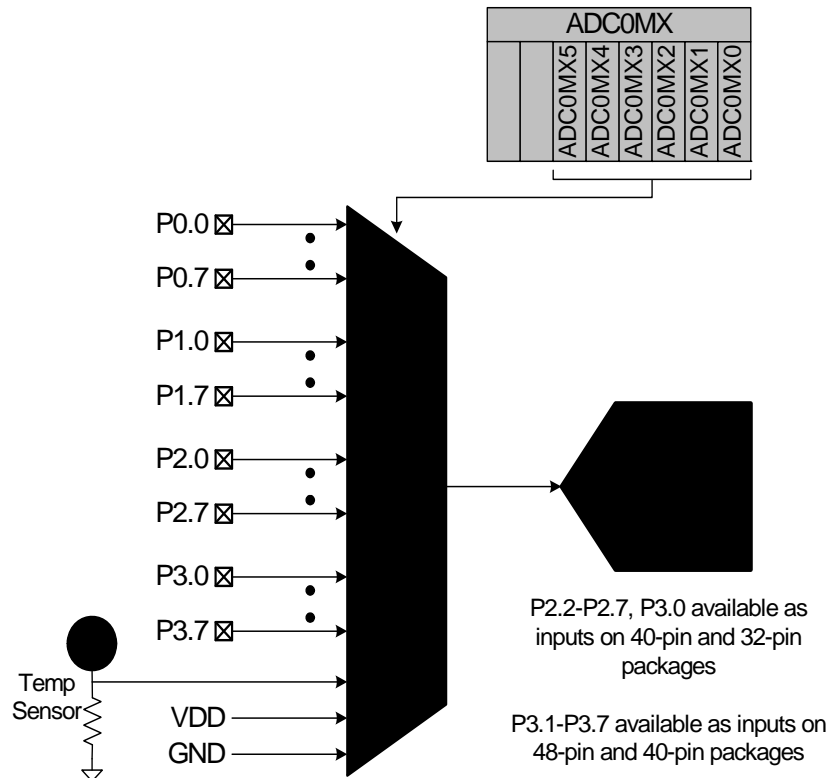


Figure 6.8. ADC0 Multiplexer Block Diagram

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN. To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section “19. Port Input/Output” on page 169 for more Port I/O configuration details.

C8051F55x/56x/57x

SFR Definition 7.1. REF0CN: Reference Control

Bit	7	6	5	4	3	2	1	0
Name			ZTCEN	REFLV	REFSL	TEMPE	BIASE	REFBE
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD1; SFR Page = 0x00

Bit	Name	Function
7:6	Unused	Read = 00b; Write = don't care.
5	ZTCEN	Zero Temperature Coefficient Bias Enable Bit. This bit must be set to 1b before entering oscillator suspend mode. 0: ZeroTC Bias Generator automatically enabled when required. 1: ZeroTC Bias Generator forced on.
4	REFLV	Voltage Reference Output Level Select. This bit selects the output voltage level for the internal voltage reference 0: Internal voltage reference set to 1.5 V. 1: Internal voltage reference set to 2.20 V.
3	REFSL	Voltage Reference Select. This bit selects the ADCs voltage reference. 0: V_{REF} pin used as voltage reference. 1: V_{DD} used as voltage reference. If V_{DD} is selected as the voltage reference and the ADC is enabled in the ADC0CN register, the P0.0/VREF pin cannot operate as a general purpose I/O pin in open-drain mode. With the above settings, this pin can operate in push-pull output mode or as an analog input.
2	TEMPE	Temperature Sensor Enable Bit. 0: Internal Temperature Sensor off. 1: Internal Temperature Sensor on.
1	BIASE	Internal Analog Bias Generator Enable Bit. 0: Internal Bias Generator off. 1: Internal Bias Generator on.
0	REFBE	On-chip Reference Buffer Enable Bit. 0: On-chip Reference Buffer off. 1: On-chip Reference Buffer on. Internal voltage reference driven on the V_{REF} pin.

SFR Definition 8.3. CPT1CN: Comparator1 Control

Bit	7	6	5	4	3	2	1	0
Name	CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP[1:0]		CP1HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9D; SFR Page = 0x00

Bit	Name	Function
7	CP1EN	Comparator1 Enable Bit. 0: Comparator1 Disabled. 1: Comparator1 Enabled.
6	CP1OUT	Comparator1 Output State Flag. 0: Voltage on CP1+ < CP1−. 1: Voltage on CP1+ > CP1−.
5	CP1RIF	Comparator1 Rising-Edge Flag. Must be cleared by software. 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.
4	CP1FIF	Comparator1 Falling-Edge Flag. Must be cleared by software. 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.
3:2	CP1HYP[1:0]	Comparator1 Positive Hysteresis Control Bits. 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP1HYN[1:0]	Comparator1 Negative Hysteresis Control Bits. 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

C8051F55x/56x/57x

SFR Definition 13.2. IP: Interrupt Priority

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; Bit-Addressable; SFR Page = All Pages

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	Serial Peripheral Interface (SPI0) Interrupt Priority Control. This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	Timer 2 Interrupt Priority Control. This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	UART0 Interrupt Priority Control. This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	Timer 1 Interrupt Priority Control. This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	External Interrupt 1 Priority Control. This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	Timer 0 Interrupt Priority Control. This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	External Interrupt 0 Priority Control. This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

C8051F55x/56x/57x

SFR Definition 13.6. EIP2: Extended Interrupt Priority Enabled 2

Bit	7	6	5	4	3	2	1	0
Name						PMAT	PCAN0	PREG0
Type	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF7; SFR Page = 0x00 and 0x0F

Bit	Name	Function
7:3	Unused	Read = 00000b; Write = Don't Care.
2	PMAT	Port Match Interrupt Priority Control. This bit sets the priority of the Port Match interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.
1	PCAN0	CAN0 Interrupt Priority Control. This bit sets the priority of the CAN0 interrupt. 0: CAN0 interrupt set to low priority level. 1: CAN0 interrupt set to high priority level.
0	PREG0	Voltage Regulator Dropout Interrupt Priority Control. This bit sets the priority of the Voltage Regulator Dropout interrupt. 0: Voltage Regulator Dropout interrupt set to low priority level. 1: Voltage Regulator Dropout interrupt set to high priority level.

17. External Data Memory Interface and On-Chip XRAM

For C8051F55x/56x/57x devices, 2 kB of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F568-9 and 'F570-5 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 17.1).

Note: The MOVX instruction can also be used for writing to the Flash memory. See Section “14. Flash Memory” on page 124 for details. The MOVX instruction accesses XRAM by default.

17.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

17.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h      ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR          ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

17.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h      ; load high byte of address into EMI0CN
MOV    R0, #34h          ; load low byte of address into R0 (or R1)
MOVX   a, @R0            ; load contents of 0x1234 into accumulator A
```


18.4.1. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 18.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 18.6 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b and a 32.768 kHz Watch Crystal requires an XFCN setting of 001b. After an external 32.768 kHz oscillator is stabilized, the XFCN setting can be switched to 000 to save power. It is recommended to enable the missing clock detector before switching the system clock to any external oscillator source.

When the crystal oscillator is first enabled, the oscillator amplitude detection circuit requires a settling time to achieve proper bias. Introducing a delay of 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

1. Force XTAL1 and XTAL2 to a high state. This involves enabling the Crossbar and writing 1 to the port pins associated with XTAL1 and XTAL2.
2. Configure XTAL1 and XTAL2 as analog inputs using.
3. Enable the external oscillator.
4. Wait at least 1 ms.
5. Poll for XTLVLD => 1.
6. Enable the Missing Clock Detector.
7. Switch the system clock to the external oscillator.

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The desired load capacitance depends upon the crystal and the manufacturer. Refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 18.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 18.3.

Port	P0								P1								P2								P3								P4
																P2.2-P2.7, P3.0 available on 40-pin and 32-pin packages								P3.1-P3.7, P4.0 available on 40-pin packages									
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
UART_TX																																	
UART_RX																																	
CAN_TX																																	
CAN_RX																																	
SCK																																	
MISO																																	
MOSI																																	
NSS																																	
SDA																																	
SCL																																	
CP0																																	
CP0A																																	
CP1																																	
CP1A																																	
SYSCLK																																	
CEX0																																	
CEX1																																	
CEX2																																	
CEX3																																	
CEX4																																	
CEX5																																	
ECI																																	
T0																																	
T1																																	
LIN_TX																																	
LIN_RX																																	

Figure 19.3. Peripheral Availability on Port I/O Pins

Registers XBR0, XBR1, and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); and similarly when the UART, CAN or LIN are selected, the Crossbar assigns both pins associated with the peripheral (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.4; UART RX0 is always assigned to P0.5. CAN0 pin assignments are fixed to P0.6 for CAN_TX and P0.7 for CAN_RX. Standard Port I/Os appear contiguously after the prioritized functions have been assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, pending the state of the NSS-MD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

As an example configuration, if CAN0, SPI0 in 4-wire mode, and PCA0 Modules 0, 1, and 2 are enabled on the crossbar with P0.1, P0.2, and P0.5 skipped, the registers should be set as follows: XBR0 = 0x06 (CAN0 and SPI0 enabled), XBR1 = 0x0C (PCA0 modules 0, 1, and 2 enabled), XBR2 = 0x40 (Crossbar enabled), and P0SKIP = 0x26 (P0.1, P0.2, and P0.5 skipped). The resulting crossbar would look as shown in Figure 19.4.

19.6. Special Function Registers for Accessing and Configuring Port I/O

All Port I/O are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable, except for P4 which is only byte addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

Ports 0–3 have a corresponding PnSKIP register which allows its individual Port pins to be assigned to digital functions or skipped by the Crossbar. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

The Port input mode of the I/O pins is defined using the Port Input Mode registers (PnMDIN). Each Port cell can be configured for analog or digital I/O. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is P4, which can only be used for digital I/O.

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. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings.

SFR Definition 19.12. P0: Port 0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x80; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	Port 0 Data. Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.

C8051F55x/56x/57x

LIN Register Definition 20.9. LIN0DIV: LIN0 Divider Register

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

Indirect Address = 0x0C

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

LIN Register Definition 20.10. LIN0MUL: LIN0 Multiplier Register

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

Indirect Address = 0x0D

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

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 ³	400 ns	2 x (transceiver loop delay + bus line delay)
Notes: <ol style="list-style-type: none"> 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. 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. 3. Based on an ISO-11898 compliant transceiver. CAN does not specify a physical layer. 		

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.

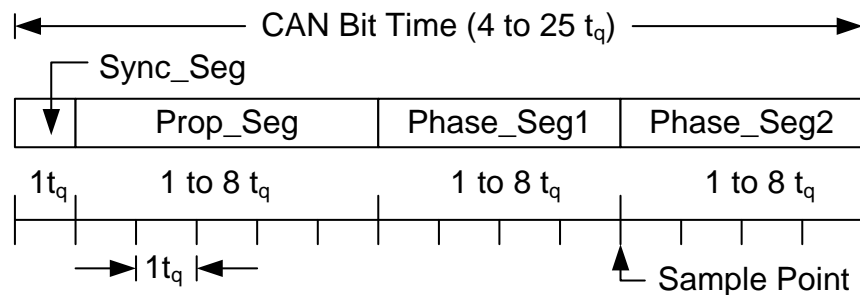


Figure 21.3. Four segments of a CAN Bit

The length of the 4 bit segments must be adjusted so that their sum is as close as possible to the desired bit time. Since each segment must be an integer multiple of the time quantum (t_q), the closest achievable bit time is $24 t_q$ (1000.008 ns), yielding a bit rate of 0.999992 Mbit/sec. The Sync_Seg is a constant $1 t_q$. The Prop_Seg must be greater than or equal to the propagation delay of 400 ns and so the choice is $10 t_q$ (416.67 ns).

The remaining time quanta (13 t_q) in the bit time are divided between Phase_Seg1 and Phase_Seg2 as shown in. Based on this equation, Phase_Seg1 = 6 t_q and Phase_Seg2 = 7 t_q .

$$\text{Phase_Seg1} + \text{Phase_Seg2} = \text{Bit_Time} - (\text{Sync_Seg} + \text{Prop_Seg})$$

1. If Phase_Seg1 + Phase_Seg2 is even, then Phase_Seg2 = Phase_Seg1. If the sum is odd, Phase_Seg2 = Phase_Seg1 + 1.
2. Phase_Seg2 should be at least 2 t_q .

Equation 21.1. Assigning the Phase Segments

The Synchronization Jump Width (SJW) timing parameter is defined by. It is used for determining the value written to the Bit Timing Register and for determining the required oscillator tolerance. Since we are using a quartz crystal as the system clock source, an oscillator tolerance calculation is not needed.

$$\text{SJW} = \text{minimum}(4, \text{Phase_Seg1})$$

Equation 21.2. Synchronization Jump Width (SJW)

The value written to the Bit Timing Register can be calculated using Equation 18.3. The BRP Extension register is left at its reset value of 0x0000.

$$\text{BRPE} = \text{BRP} - 1 = \text{BRP Extension Register} = 0x0000$$

$$\text{SJWp} = \text{SJW} - 1 = \text{minimum}(4, 6) - 1 = 3$$

$$\text{TSEG1} = \text{Prop_Seg} + \text{Phase_Seg1} - 1 = 10 + 6 - 1 = 15$$

$$\text{TSEG2} = \text{Phase_Seg2} - 1 = 6$$

$$\text{Bit Timing Register} = (\text{TSEG2} \times 0x1000) + (\text{TSEG1} \times 0x0100)$$

$$\text{Bit Timing Register} = (\text{TSEG2} \times 0x1000) + (\text{TSEG1} \times 0x0100) + (\text{SJWp} \times 0x0040) + \text{BRPE} = 0x6FC0$$

Equation 21.3. Calculating the Bit Timing Register Value

C8051F55x/56x/57x

SFR Definition 21.1. CAN0CFG: CAN Clock Configuration

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

SFR Address = 0x92; SFR Page = 0x0C

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = Don't Care.
1:0	SYSDIV[1:0]	CAN System Clock Divider Bits. The CAN controller clock is derived from the CIP-51 system clock. The CAN controller clock must be less than or equal to 25 MHz. 00: CAN controller clock = System Clock/1. 01: CAN controller clock = System Clock/2. 10: CAN controller clock = System Clock/4. 11: CAN controller clock = System Clock/8.

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.

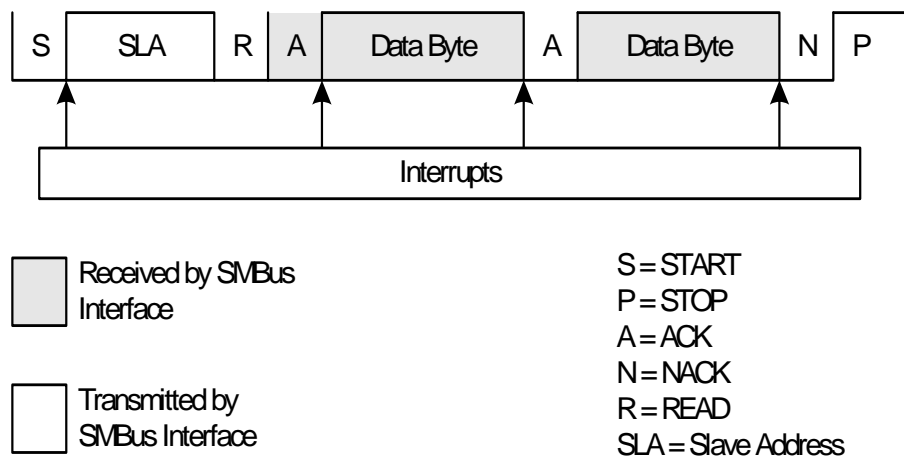


Figure 22.6. Typical Master Read Sequence

the RI0 flag will be set. Note: when MCE0 = 1, RI0 will only be set if the extra bit was equal to 1. Data can be read from the receive FIFO by reading the SBUF0 register. The SBUF0 register represents the oldest byte in the FIFO. After SBUF0 is read, the next byte in the FIFO is immediately loaded into SBUF0, and space is made available in the FIFO for another incoming byte. If enabled, an interrupt will occur when RI0 is set. RI0 can only be cleared to '0' by software when there is no more information in the FIFO. The recommended procedure to empty the FIFO contents is as follows:

1. Clear RI0 to 0.
2. Read SBUF0.
3. Check RI0, and repeat at step 1 if RI0 is set to 1.

If the extra bit function is enabled (XBE0 = 1) and the parity function is disabled (PE0 = 0), the extra bit for the oldest byte in the FIFO can be read from the RBX0 bit (SCON0.2). If the extra bit function is not enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX0. When the parity function is enabled (PE0 = 1), hardware will check the received parity bit against the selected parity type (selected with S0PT[1:0]) when receiving data. If a byte with parity error is received, the PERR0 flag will be set to 1. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

Note: The UART Receive FIFO pointer can be corrupted if the UART receives a byte and firmware reads a byte from the FIFO at the same time. When this occurs, firmware will lose the received byte and the FIFO receive overrun flag (OVR0) will also be set to 1. Systems using the UART Receive FIFO should ensure that the FIFO isn't accessed by hardware and firmware at the same time. In other words, firmware should ensure to read the FIFO before the next byte is received.

24.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 24.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 24.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

24.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

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]	Timer 0 High Byte. 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]	Timer 1 High Byte. The TH1 register is the high byte of the 16-bit Timer 1.

C8051F55x/56x/57x

SFR Definition 25.11. TMR2L: Timer 2 Low Byte

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

SFR Address = 0xCC; SFR Page = 0x00

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

SFR Definition 25.12. TMR2H Timer 2 High Byte

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

SFR Address = 0xCD; SFR Page = 0x00

Bit	Name	Function
7:0	TMR2H[7:0]	Timer 2 High Byte. In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

Table 26.3. Watchdog Timer Timeout Intervals¹

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 ²	255	4194
187,500 ²	128	2114
187,500 ²	32	541
Notes: <ol style="list-style-type: none">1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.2. Internal SYSCLK reset frequency = Internal Oscillator divided by 128.		