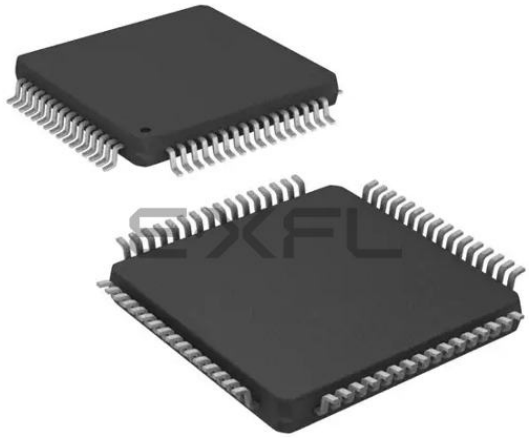


Welcome to E-XFL.COM

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, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	24
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	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f061r

List of Tables

1. System Overview

The C8051F06x family of devices are fully integrated mixed-signal System-on-a-Chip MCUs with 59 digital I/O pins (C8051F060/2/4/6) or 24 digital I/O pins (C8051F061/3/5/7), and two integrated 16-bit 1 Msps ADCs. Highlighted features are listed below; refer to Table 1.1 for specific product feature selection.

- High-Speed pipelined 8051-compatible CIP-51 microcontroller core (up to 25 MIPS)
- Two 16-bit 1 Msps ADCs with a Direct Memory Access controller
- Controller Area Network (CAN 2.0B) Controller with 32 message objects, each with its own indentifier mask (C8051F060/1/2/3)
- In-system, full-speed, non-intrusive debug interface on-chip
- 10-bit 200 ksps ADC with PGA and 8-channel analog multiplexer (C8051F060/1/2/3)
- Two 12-bit DACs with programmable update scheduling (C8051F060/1/2/3)
- 64 kB (C8051F060/1/2/3/4/5) or 32 kB (C8051F066/7) of in-system programmable Flash memory
- 4352 (4096 + 256) bytes of on-chip RAM
- External Data Memory Interface with 64 kB direct address space (C8051F060/2/4/6)
- SPI, SMBus/I2C, and (2) UART serial interfaces implemented in hardware
- Five general purpose 16-bit Timers
- Programmable Counter/Timer Array with six capture/compare modules
- On-chip Watchdog Timer, VDD Monitor, and Temperature Sensor

With on-chip VDD monitor, Watchdog Timer, and clock oscillator, the C8051F06x family of devices are truly stand-alone System-on-a-Chip solutions. All analog and digital peripherals are enabled/disabled and configured by user firmware. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware.

On-board JTAG debug circuitry allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, Run and Halt commands. All analog and digital peripherals are fully functional while debugging using JTAG.

Each MCU is specified for 2.7 to 3.6 V operation over the industrial temperature range (-45 to +85 °C). The C8051F060/2/4/6 are available in a 100-pin TQFP package and the C8051F061/3/5/7 are available in a 64-pin TQFP package (see block diagrams in Figure 1.1, Figure 1.2, Figure 1.3 and Figure 1.4).

1.3. JTAG Debug and Boundary Scan

The C8051F06x family has on-chip JTAG boundary scan and debug circuitry that provides *non-intrusive, full speed, in-circuit debugging using the production part installed in the end application*, via the four-pin JTAG interface. The JTAG port is fully compliant to IEEE 1149.1, providing full boundary scan for test and manufacturing purposes.

Silicon Laboratories' debugging system supports inspection and modification of memory and registers, breakpoints, watchpoints, a stack monitor, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADCs and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized with instruction execution.

The C8051F060DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F06x MCUs. The kit includes a Windows (95 or later) development environment, a serial adapter for connecting to the JTAG port, and a target application board with a C8051F060 MCU installed. All of the necessary communication cables and a wall-mount power supply are also supplied with the development kit. Silicon Labs' debug environment is a vastly superior configuration for developing and debugging embedded applications compared to standard MCU emulators, which use on-board "ICE Chips" and target cables and require the MCU in the application board to be socketed. Silicon Labs' debug environment both increases ease of use and preserves the performance of the precision, on-chip analog peripherals.

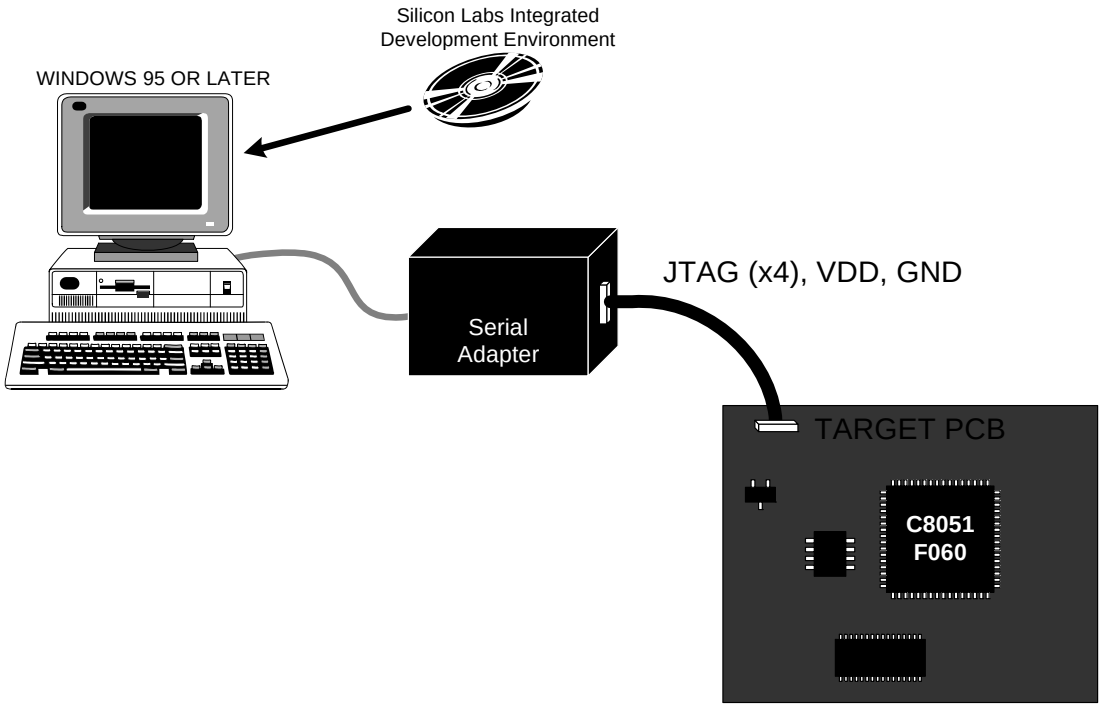


Figure 1.8. Development/In-System Debug Diagram

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.

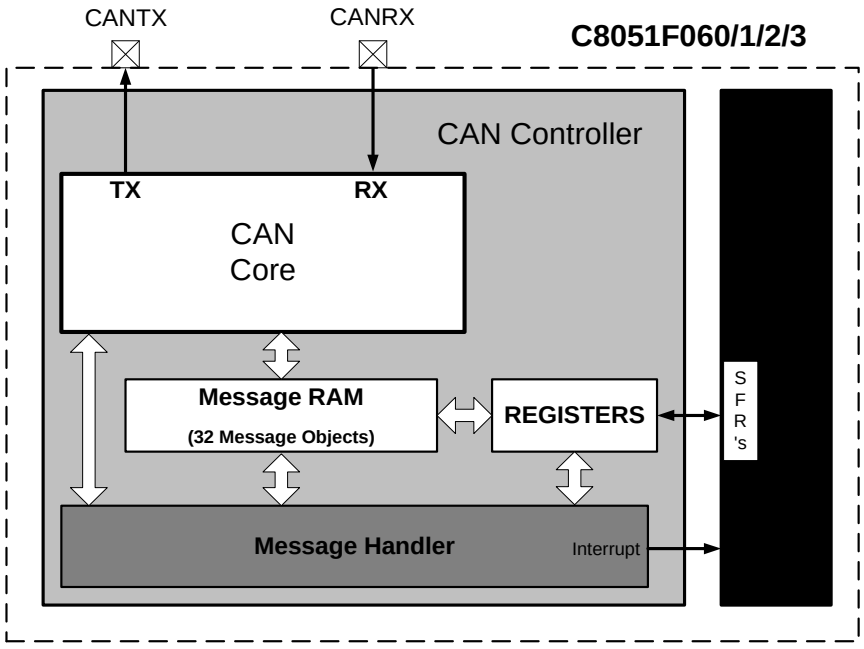


Figure 1.11. CAN Controller Overview

2. Absolute Maximum Ratings

Table 2.1. Absolute Maximum Ratings*

Parameter	Conditions	Min	Typ	Max	Units
Ambient temperature under bias		-55		125	°C
Storage Temperature		-65		150	°C
Voltage on any pin (except VDD, AV+, AVDD, and Port 0) with respect to DGND		-0.3		VDD + 0.3	V
Voltage on any Port 0 Pin with respect to DGND.		-0.3		5.8	V
Voltage on VDD, AV+, or AVDD with respect to DGND		-0.3		4.2	V
Maximum Total current through VDD, AV+, AVDD, DGND, and AGND				800	mA
Maximum output current sunk by any Port pin				100	mA
Maximum output current sunk by any other I/O pin				50	mA
Maximum output current sourced by any Port pin				100	mA
Maximum output current sourced by any other I/O pin				50	mA

* Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 4.1. Pin Definitions (Continued)

Name	Pin Numbers				Type	Description
	F060	F061	F064	F065		
	F062	F063	F066	F067		
P2.5	41	36	41	36	D I/O	Port 2.5. See Port Input/Output section for complete description.
P2.6	40	35	40	35	D I/O	Port 2.6. See Port Input/Output section for complete description.
P2.7	39	34	39	34	D I/O	Port 2.7. See Port Input/Output section for complete description.
P3.0	54		54		D I/O	Port 3.0. See Port Input/Output section for complete description.
P3.1	53		53		D I/O	Port 3.1. See Port Input/Output section for complete description.
P3.2	52		52		D I/O	Port 3.2. See Port Input/Output section for complete description.
P3.3	51		51		D I/O	Port 3.3. See Port Input/Output section for complete description.
P3.4	50		50		D I/O	Port 3.4. See Port Input/Output section for complete description.
P3.5	49		49		D I/O	Port 3.5. See Port Input/Output section for complete description.
P3.6	48		48		D I/O	Port 3.6. See Port Input/Output section for complete description.
P3.7	47		47		D I/O	Port 3.7. See Port Input/Output section for complete description.
P4.5/ALE	93		93		D I/O	Port 4.5. See Port Input/Output section for complete description. ALE Strobe for External Memory Address Bus (Multiplexed mode).
P4.6/$\overline{RD}$	92		92		D I/O	Port 4.6. See Port Input/Output section for complete description. /RD Strobe for External Memory Address Bus.
P4.7/$\overline{WR}$	91		91		D I/O	Port 4.7. See Port Input/Output section for complete description. /WR Strobe for External Memory Address Bus.
P5.0/A8	88		88		D I/O	Port 5.0. See Port Input/Output section for complete description. Bit 8 External Memory Address Bus (Non-multiplexed mode).
P5.1/A9	87		87		D I/O	Port 5.1. See Port Input/Output section for complete description.

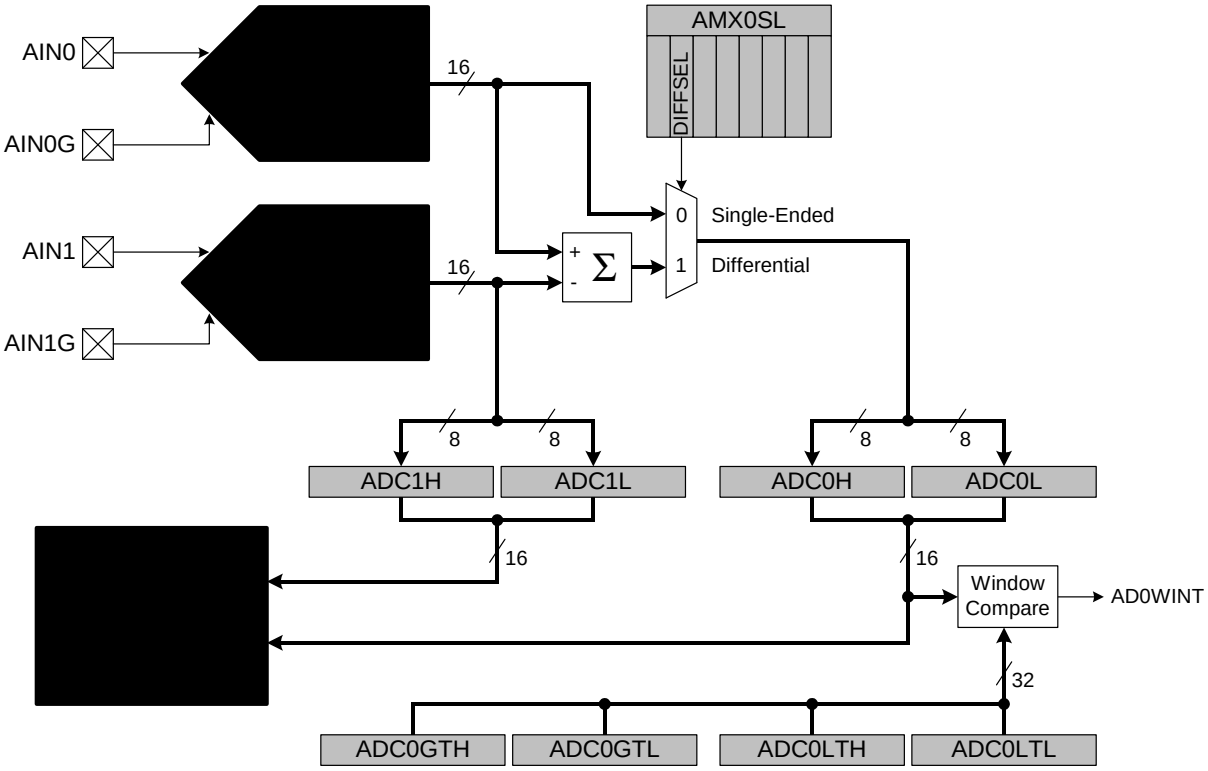


Figure 5.2. 16-bit ADC0 and ADC1 Data Path Diagram

5.1. Single-Ended or Differential Operation

ADC0 and ADC1 can be programmed to operate independently as single-ended ADCs, or together to accept a differential input. In single-ended mode, the ADCs can be configured to sample simultaneously, or to use different conversion speeds. In differential mode, ADC1 is a slave to ADC0, and its configuration is based on ADC0 settings, except during offset or gain calibrations. The DIFFSEL bit in the Channel Select Register AMX0SL (Figure 5.6) selects between single-ended and differential mode.

5.1.1. Pseudo-Differential Inputs

The inputs to the ADCs are pseudo-differential. The actual voltage measured by each ADC is equal to the voltage between the AINn pin and the AINnG pin. AINnG must be a DC signal between -0.2 and 0.6 V. In most systems, AINnG will be connected to AGND. If not tied to AGND, the AINnG signal can be used to negate a limited amount of fixed offset, but it is recommended that the internal offset calibration features of the device be used for this purpose. When operating in differential mode, AIN0G and AIN1G should be tied together. AINn must remain above AINnG in both modes for accurate conversion results.

5.4. Calibration

The ADCs are calibrated for linearity, offset, and gain in production. ADC0 and ADC1 can also be independently calibrated for each of these parameters in-system. Calibrations are initiated using bits in the ADC0 or ADC1 Configuration Register. The calibration coefficients can be accessed using the ADC Calibration Pointer Register (ADC0CPT, Figure 5.22) and the ADC Calibration Coefficient Register (ADC0CCF, Figure 5.23). The CPTR bits in ADC0CPT allow the ADC0CCF register to read and write specific calibration coefficients. Figure 5.19 shows the Calibration Coefficient locations.

Figure 5.19. Calibration Coefficient Locations

	ADC0CCF							
ADC0CPT Bits 5-0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x00	Linearity Calibration Coefficients (locations 0x00 through 0x12)							
. .								
0x12								
0x13	Offset7	Offset6	Offset5	Offset4	Offset3	Offset2	Offset1	Offset0
0x14			Offset13	Offset12	Offset11	Offset10	Offset9	Offset8
0x15	Gain7	Gain6	Gain5	Gain4	Gain3	Gain2	Gain1	Gain0
0x16				Gain12	Gain11	Gain10	Gain9	Gain8

The ADCs are calibrated for linearity in production. Under normal circumstances, no additional linearity calibration is necessary. If linearity calibrations are desired, they can be initiated by setting the ADCnLCAL bit to ‘1’. When the calibration is finished, the ADCnLCAL bit will be set to ‘0’ by the hardware. Linearity Calibration Coefficients are stored in the locations shown in Figure 5.19.

Offset and gain calibrations can be performed using either internal or external voltages as calibration sources. The ADCnSCAL bit determines whether the internal or external voltages are used in the calibration process. To ensure accuracy, offset calibration should be done prior to a gain calibration. The offset and gain calibration coefficients are decoded in Figure 5.20. Offset calibration is initiated by setting the ADCnOCAL bit to ‘1’. When the calibration is finished, the ADCnOCAL bit will be set to ‘0’ by the hardware. Offset calibration can compensate for offset errors of approximately ±3.125% of full scale. The offset value is added to the AINnG input prior to digitization by the ADC. Gain calibration is initiated by setting the ADCnGCAL bit to ‘1’. When the calibration is finished, the ADCnGCAL bit will be set to ‘0’ by the hardware. Gain calibration can compensate for slope errors of approximately ±3.125%. The gain value is added to the ADC’s VREF path to change the slope of the converter’s transfer function. Figure 5.21 shows how the offset and gain values affect the analog signals used by the ADC.

Figure 12.3. CPTnCN: Comparator 0, 1, and 2 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CPnEN	CPnOUT	CPnRIF	CPnFIF	CPnHYP1	CPnHYP0	CPnHYN1	CPnHYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: CPT0CN: 0x88; CPT1CN: 0x88; CPT2CN: 0x88
SFR Pages: CPT0CN: page 1; CPT1CN: page 2; CPT2CN: page 3

Bit7: CPnEN: Comparator Enable Bit. **(Please see note below.)**
0: Comparator Disabled.
1: Comparator Enabled.

Bit6: CPnOUT: Comparator Output State Flag.
0: Voltage on CPn+ < CPn-.
1: Voltage on CPn+ > CPn-.

Bit5: CPnRIF: Comparator Rising-Edge Interrupt Flag.
0: No Comparator Rising Edge Interrupt has occurred since this flag was last cleared.
1: Comparator Rising Edge Interrupt has occurred. Must be cleared by software.

Bit4: CPnFIF: Comparator Falling-Edge Interrupt Flag.
0: No Comparator Falling-Edge Interrupt has occurred since this flag was last cleared.
1: Comparator Falling-Edge Interrupt has occurred. Must be cleared by software.

Bits3-2: CPnHYP1-0: Comparator Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 20 mV.

Bits1-0: CPnHYN1-0: Comparator Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 20 mV.

NOTE: Upon enabling a comparator, the output of the comparator is not immediately valid. Before using a comparator as an interrupt or reset source, software should wait for a minimum of the specified “Power-up time” as specified in Table 12.1, “Comparator Electrical Characteristics,” on page 122.

Table 13.2. Special Function Register (SFR) Memory Map

ADDRESS	SFR PAGE	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 1 2 3 F	SPI0CN CAN0CN DMA0CF P7	PCA0L DMA0CTL	PCA0H DMA0CTH	PCA0CPL0 DMA0CSL	PCA0CPH0 DMA0CSH	PCA0CPL1 DMA0BND	PCA0CPH1 DMA0ISW	WDTCN **(ALL PAGES)**
F0	0 1 2 3 F	B **(ALL PAGES)**						EIP1 **(ALL PAGES)**	EIP2 **(ALL PAGES)**
E8	0 1 2 3 F	ADC0CN ADC1CN ADC2CN P6	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	PCA0CPL4	PCA0CPH4	RSTSRC
E0	0 1 2 3 F	ACC **(ALL PAGES)**	PCA0CPL5 XBR0	PCA0CPH5 XBR1	XBR2	XBR3		EIE1 **(ALL PAGES)**	EIE2 **(ALL PAGES)**
D8	0 1 2 3 F	PCA0CN CAN0DATL DMA0CN P5	PCA0MD CAN0DATH DMA0DAL	PCA0CPM0 CAN0ADR DMA0DAH	PCA0CPM1 CAN0TST DMA0DSL	PCA0CPM2 DMA0DSH	PCA0CPM3 DMA0IPT	PCA0CPM4 DMA0IDT	PCA0CPM5
D0	0 1 2 3 F	PSW **(ALL PAGES)**	REF0CN REF1CN REF2CN	DAC0L DAC1L	DAC0H DAC1H	DAC0CN DAC1CN			
C8	0 1 2 3 F	TMR2CN TMR3CN TMR4CN P4	TMR2CF TMR3CF TMR4CF	RCAP2L RCAP3L RCAP4L	RCAP2H RCAP3H RCAP4H	TMR2L TMR3L TMR4L	TMR2H TMR3H TMR4H		SMB0CR
C0	0 1 2 3 F	SMB0CN CAN0STA	SMB0STA	SMB0DAT	SMB0ADR	ADC0GTL ADC2GTL	ADC0GTH ADC2GTH	ADC0LTL ADC2LTL	ADC0LTH ADC2LTH
B8	0 1 2 3 F	IP **(ALL PAGES)**	SADEN0	AMX2CF ADC0CPT	AMX0SL AMX2SL ADC0CCF	ADC0CF ADC1CF ADC2CF		ADC0L ADC1L ADC2L	ADC0H ADC1H ADC2H
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Figure 13.20. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS0	PT1	PX1	PT0	PX0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xB8
SFR Page: All Pages

Bits7-6: UNUSED. Read = 11b, Write = don't care.
Bit5: 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.
Bit4: 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.
Bit3: 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.
Bit2: 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.
Bit1: 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.
Bit0: 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.

17.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of four steps:

1. Enable the EMIF on the High Ports (P7, P6, P5, and P4).
2. Configure the Output Modes of the port pins as either push-pull or open-drain (push-pull is most common).
3. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic ‘1’).
4. Select Multiplexed mode or Non-multiplexed mode.
5. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
6. Set up timing to interface with off-chip memory or peripherals.

Each of these four steps is explained in detail in the following sections. The Port enable bit, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in Figure 17.2.

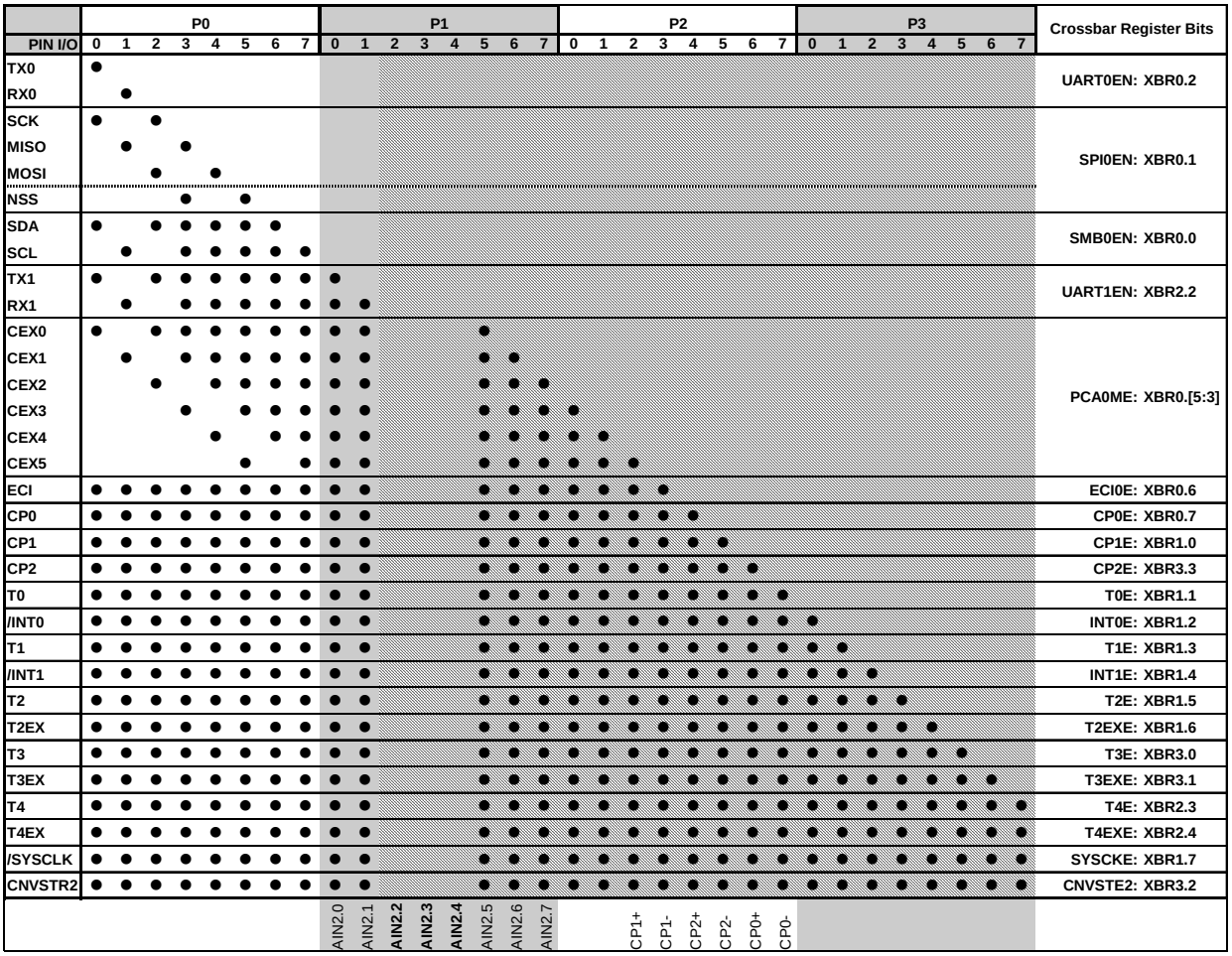
17.3. Port Selection and Configuration

When enabled, the External Memory Interface appears on Ports 7, 6, 5, and 4 in non-multiplexed mode, or Ports 7, 6, and 4 in multiplexed mode.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches. See Section “18. Port Input/Output” on page 203 for more information about the Port operation and configuration. **The Port latches should be explicitly configured to ‘park’ the External Memory Interface pins in a dormant state when not in use, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. See Section “18. Port Input/Output” on page 203 for more information about Port output mode configuration.

Figure 18.4. Crossbar Example:
(P1MDIN = 0xE3; XBR0 = 0x3D; XBR1 = 0x14; XBR2 = 0x40)



19.1. Bosch CAN Controller Operation

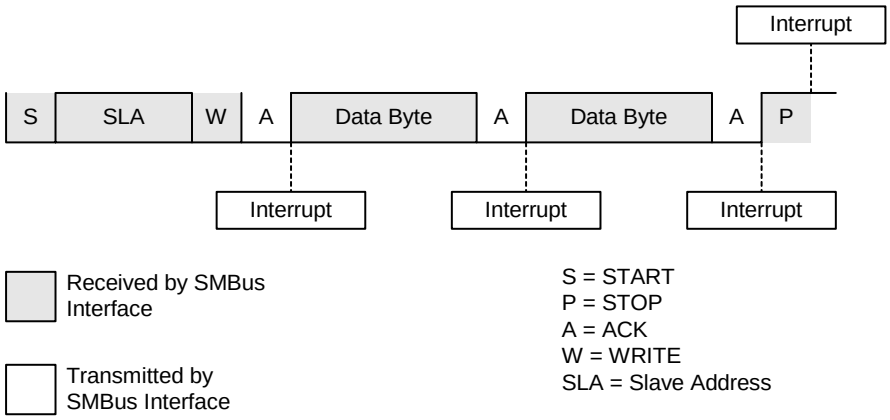
The CAN Controller featured in the C8051F060/1/2/3 devices is a full implementation of Bosch's full CAN module and fully complies with CAN specification 2.0B.

The function and use of the CAN Controller is detailed in the *Bosch CAN User's Guide*. The User's Guide should be used as a reference to configure and use the CAN controller. This Silicon Labs datasheet describes how to access the CAN controller.

The CAN Control Register (CAN0CN), CAN Test Register (CAN0TST), and CAN Status Register (CAN0STA) in the CAN controller can be accessed directly or indirectly via CIP-51 SFRs. All other CAN registers must be accessed via an indirect indexing method. See "Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers" on page 229.

Mode after receiving a STOP condition from the master.

Figure 20.7. Typical Slave Receiver Sequence



22.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit and the built-in UART0 address recognition hardware. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0. UART0 will recognize as “valid” (i.e., capable of causing an interrupt) **two** types of addresses: (1) a *masked* address and (2) a *broadcast* address **at any given time**. Both are described below.

22.2.1. Configuration of a Masked Address

The UART0 address is configured via two SFRs: SADDR0 (Serial Address) and SADEN0 (Serial Address Enable). SADEN0 sets the bit mask for the address held in SADDR0: bits set to logic 1 in SADEN0 correspond to bits in SADDR0 that are checked against the received address byte; bits set to logic 0 in SADEN0 correspond to “don’t care” bits in SADDR0.

Example 1, SLAVE #1	
SADDR0	= 00110101
SADEN0	= 00001111
UART0 Address	= xxxx0101

Example 2, SLAVE #2	
SADDR0	= 00110101
SADEN0	= 11110011
UART0 Address	= 0011xx01

Example 3, SLAVE #3	
SADDR0	= 00110101
SADEN0	= 11000000
UART0 Address	= 00xxxxxx

Setting the SM20 bit (SCON0.5) configures UART0 such that when a stop bit is received, UART0 will generate an interrupt only if the ninth bit is logic 1 (RB80 = ‘1’) and the received data byte matches the UART0 slave address. Following the received address interrupt, the slave will clear its SM20 bit to enable interrupts on the reception of the following data byte(s). Once the entire message is received, the addressed slave resets its SM20 bit to ignore all transmissions until it receives the next address byte. While SM20 is logic 1, UART0 ignores all bytes that do not match the UART0 address and include a ninth bit that is logic 1.

22.2.2. Broadcast Addressing

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The broadcast address is the logical OR of registers SADDR0 and SADEN0, and ‘0’s of the result are treated as “don’t cares”. Typically a broadcast address of 0xFF (hexadecimal) is acknowledged by all slaves, assuming “don’t care” bits as ‘1’s. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s)..

Example 4, SLAVE #1	
SADDR0	= 00110101
SADEN0	= 00001111
Broadcast Address	= 00111111

Example 5, SLAVE #2	
SADDR0	= 00110101
SADEN0	= 11110011
Broadcast Address	= 11110111

Example 6, SLAVE #3	
SADDR0	= 00110101
SADEN0	= 11000000
Broadcast Address	= 11110101

Where all ZEROES in the Broadcast address are don’t cares.

Note in the above examples 4, 5, and 6, each slave would recognize as “valid” an address of 0xFF as a broadcast address. Also note that examples 4, 5, and 6 uses the same SADDR0 and SADEN0 register values as shown in the examples 1, 2, and 3 respectively (slaves #1, 2, and 3). Thus, a master could address each slave device individually using a masked address, and also broadcast to all three slave devices. For example, if a Master were to send an address “11110101”, only slave #1 would recognize the

Table 23.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator

	Frequency: 24.5 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 Internal Osc.	230400	-0.32%	106	SYSCLK	XX	1	0xCB
	115200	-0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	-0.32%	848	SYSCLK / 4	01	0	0x96
	14400	0.15%	1704	SYSCLK / 12	00	0	0xB9
	9600	-0.32%	2544	SYSCLK / 12	00	0	0x96
	2400	-0.32%	10176	SYSCLK / 48	10	0	0x96
	1200	0.15%	20448	SYSCLK / 48	10	0	0x2B

X = Don't care

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

Table 23.2. Timer Settings for Standard Baud Rates Using an External Oscillator

	Frequency: 25.0 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	-0.47%	108	SYSCLK	XX	1	0xCA
	115200	0.45%	218	SYSCLK	XX	1	0x93
	57600	-0.01%	434	SYSCLK	XX	1	0x27
	28800	0.45%	872	SYSCLK / 4	01	0	0x93
	14400	-0.01%	1736	SYSCLK / 4	01	0	0x27
	9600	0.15%	2608	EXTCLK / 8	11	0	0x5D
	2400	0.45%	10464	SYSCLK / 48	10	0	0x93
	1200	-0.01%	20832	SYSCLK / 48	10	0	0x27
SYSCLK from Internal Osc.	57600	-0.47%	432	EXTCLK / 8	11	0	0xE5
	28800	-0.47%	864	EXTCLK / 8	11	0	0xCA
	14400	0.45%	1744	EXTCLK / 8	11	0	0x93
	9600	0.15%	2608	EXTCLK / 8	11	0	0x5D

X = Don't care

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

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.

25.2.3. High Speed Output Mode

In High Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn) Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

Figure 25.6. PCA High Speed Output Mode Diagram