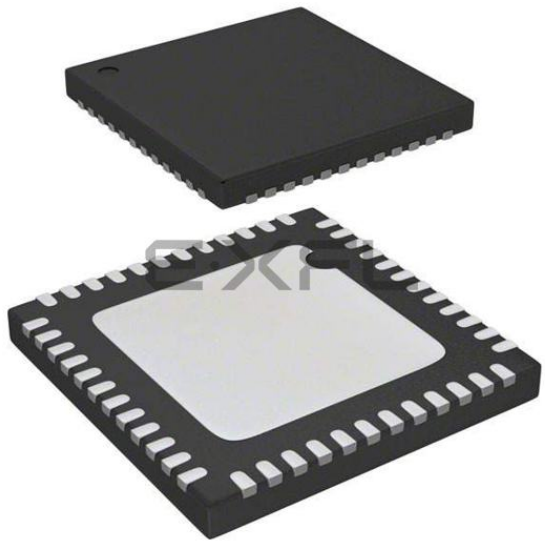

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f580-imr

4.4. QFP-32 Package Specifications

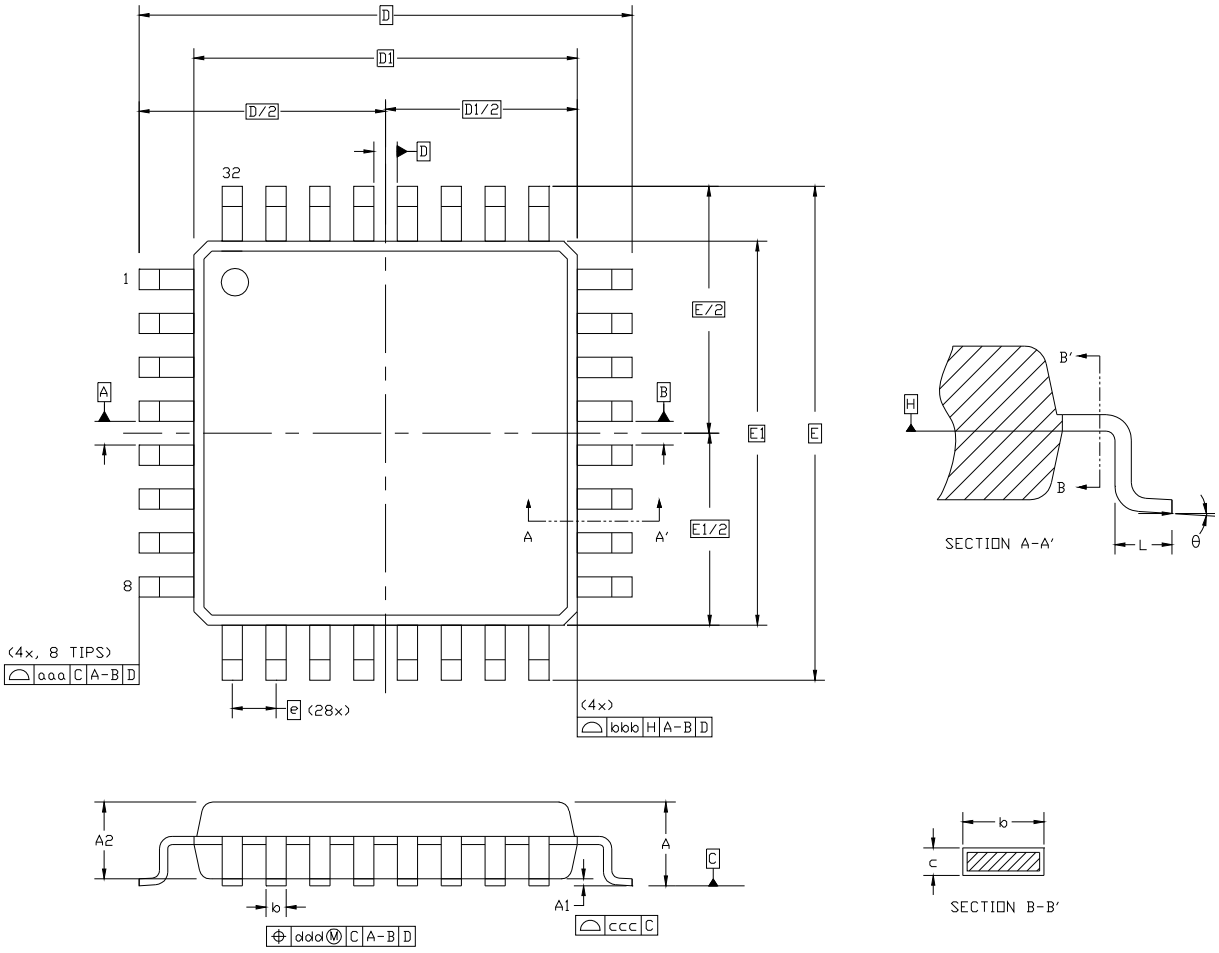

Figure 4.7. QFP-32 Package Drawing

Table 4.7. QFP-32 Package Dimensions

Dimension	Min	Typ	Max
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
c	0.09	—	0.20
D	9.00 BSC.		
D1	7.00 BSC.		
e	0.80 BSC.		

Dimension	Min	Typ	Max
E	9.00 BSC.		
E1	7.00 BSC.		
L	0.45	0.60	0.75
aaa	0.20		
bbb	0.20		
ccc	0.10		
ddd	0.20		
θ	0°	3.5°	7°

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC outline MS-026, variation BBA.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Table 5.2. Global Electrical Characteristics (Continued)
–40 to +125 °C, 24 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)					
I_{DD}[4]	V_{DD} = 2.1 V, F = 200 kHz	—	130	—	µA
	V_{DD} = 2.1 V, F = 1.5 MHz	—	440	—	µA
	V_{DD} = 2.1 V, F = 25 MHz	—	5.8	8.0	mA
	V_{DD} = 2.1 V, F = 50 MHz	—	11	16	mA
I_{DD}[4]	V_{DD} = 2.6 V, F = 200 kHz	—	170	—	µA
	V_{DD} = 2.6 V, F = 1.5 MHz	—	570	—	µA
	V_{DD} = 2.6 V, F = 25 MHz	—	7.3	15	mA
	V_{DD} = 2.6 V, F = 50 MHz	—	15	25	mA
I_{DD} Supply Sensitivity[4]	F = 25 MHz	—	53	—	%/V
	F = 1 MHz	—	60	—	
I_{DD} Frequency Sensitivity [4.6]	V_{DD} = 2.1V, F ≤ 12.5 MHz, T = 25 °C	—	0.28	—	mA/MHz
	V_{DD} = 2.1V, F > 12.5 MHz, T = 25 °C	—	0.28	—	
	V_{DD} = 2.6V, F ≤ 12.5 MHz, T = 25 °C	—	0.35	—	
	V_{DD} = 2.6V, F > 12.5 MHz, T = 25 °C	—	0.35	—	
Digital Supply Current[4] (Stop or Suspend Mode)	Oscillator not running, V_{DD} Monitor Disabled				µA
	Temp = 25 °C	—	230	—	
	Temp = 60 °C	—	230	—	
	Temp= 125 °C	—	330	—	

Notes:
1. Given in Table 5.4 on page 48.
2. V_{IO} should not be lower than the V_{DD} voltage.
3. SYSCLK must be at least 32 kHz to enable debugging.
4. Based on device characterization data; Not production tested. Does not include oscillator supply current.
5. IDD can be estimated for frequencies ≤ 15 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} for >15 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F = 20 MHz, I_{DD} = 21 mA - (50 MHz - 20 MHz) * 0.46 mA/MHz = 7.2 mA.
6. Idle IDD can be estimated for frequencies ≤ 1 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate Idle I_{DD} for >1 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number.
   For example: V_{DD} = 2.6 V; F = 5 MHz, Idle I_{DD} = 19 mA – (50 MHz – 5 MHz) x 0.38 mA/MHz = 1.9 mA.

6.3.2. Setting the Gain Value

The three programmable gain registers are accessed indirectly using the ADC0H and ADC0L registers when the GAINEN bit (ADC0CF.0) bit is set. ADC0H acts as the address register, and ADC0L is the data register. The programmable gain registers can only be written to and cannot be read. See Gain Register Definition 6.1, Gain Register Definition 6.2, and Gain Register Definition 6.3 for more information.

The gain is programmed using the following steps:

1. Set the GAINEN bit (ADC0CF.0)
2. Load the ADC0H with the ADC0GNH, ADC0GNL, or ADC0GNA address.
3. Load ADC0L with the desired value for the selected gain register.
4. Reset the GAINEN bit (ADC0CF.0)

Notes:

1. An ADC conversion should not be performed while the GAINEN bit is set.
2. Even with gain enabled, the maximum input voltage must be less than V_{REGIN} and the maximum voltage of the signal after gain must be less than or equal to V_{REF}.

In code, changing the value to 0.44 gain from the previous example looks like:

```
// in 'C':
ADC0CF |= 0x01;          // GAINEN = 1
ADC0H = 0x04;            // Load the ADC0GNH address
ADC0L = 0x6C;            // Load the upper byte of 0x6CA to ADC0GNH
ADC0H = 0x07;            // Load the ADC0GNL address
ADC0L = 0xA0;            // Load the lower nibble of 0x6CA to ADC0GNL
ADC0H = 0x08;            // Load the ADC0GNA address
ADC0L = 0x01;            // Set the GAINADD bit
ADC0CF &= ~0x01;         // GAINEN = 0

; in assembly
ORL ADC0CF,#01H          ; GAINEN = 1
MOV ADC0H,#04H           ; Load the ADC0GNH address
MOV ADC0L,#06CH          ; Load the upper byte of 0x6CA to ADC0GNH
MOV ADC0H,#07H           ; Load the ADC0GNL address
MOV ADC0L,#0A0H          ; Load the lower nibble of 0x6CA to ADC0GNL
MOV ADC0H,#08H           ; Load the ADC0GNA address
MOV ADC0L,#01H           ; Set the GAINADD bit
ANL ADC0CF,#0FEH         ; GAINEN = 0
```

Table 11.1. CIP-51 Instruction Set Summary (Prefetch-Enabled) (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	2
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	4-7*
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2

Note: Certain instructions take a variable number of clock cycles to execute depending on instruction alignment and the FLRT setting (SFR Definition 15.3).

Table 13.2. Special Function Register (SFR) Memory Map for Page 0x0C

	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8			CAN0IF2DA2L	CAN0IF2DA2H	CAN0IF2DB1L	CAN0IF2DB1H	CAN0IF2DB2L	CAN0IF2DB2H
F0	B (All Pages)		CAN0IF2A2L	CAN0IF2A2H			CAN0IF2DA1L	CAN0IF2DA1H
E8			CAN0IF2M1L	CAN0IF2M1H	CAN0IF2M2L	CAN0IF2M2H	CAN0IF2A1L	CAN0IF2A1H
E0	ACC (All Pages)		CAN0IF2CML	CAN0IF2CMH			EIE1 (All Pages)	EIE2 (All Pages)
D8			CAN0IF1DB1L	CAN0IF1DB1H	CAN0IF1DB2L	CAN0IF1DB2H	CAN0IF2CRL	CAN0IF2CRH
D0	PSW (All Pages)		CAN0IF1MCL	CAN0IF1MCH	CAN0IF1DA1L	CAN0IF1DA1H	CAN0IF1DA2L	CAN0IF1DA2H
C8			CAN0IF1A1L	CAN0IF1A1H	CAN0IF1A2L	CAN0IF1A2H	CAN0IF2MCL	CAN0IF2MCH
C0	CAN0CN		CAN0IF1CML	CAN0IF1CMH	CAN0IF1M1L	CAN0IF1M1H	CAN0IF1M2L	CAN0IF1M2H
B8	IP (All Pages)		CAN0MV1L	CAN0MV1H	CAN0MV2L	CAN0MV2H	CAN0IF1CRL	CAN0IF1CRH
B0	P3 (All Pages)		CAN0IP2L	CAN0IP2H		P4 (All Pages)	FLSCL (All Pages)	FLKEY (All Pages)
A8	IE (All Pages)		CAN0ND1L	CAN0ND1H	CAN0ND2L	CAN0ND2H	CAN0IP1L	CAN0IP1H
A0	P2 (All Pages)	CAN0BRPE	CAN0TR1L	CAN0TR1H	CAN0TR2L	CAN0TR2H		SFRPAGE (All Pages)
98	SCON0 (All Pages)		CAN0BTL	CAN0BTH	CAN0IIDL	CAN0IIDH	CAN0TST	
90	P1 (All Pages)		CAN0CFG		CAN0STAT		CAN0ERRL	CAN0ERRH
88	TCON (All Pages)	TMOD (All Pages)	TL0 (All Pages)	TL1 (All Pages)	TH0 (All Pages)	TH1 (All Pages)	CKCON (All Pages)	
80	P0 (All Pages)	SP (All Pages)	DPL (All Pages)	DPH (All Pages)		SFRNEXT (All Pages)	SFRLAST (All Pages)	PCON (All Pages)
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

SFR Definition 14.7. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE4; SFR Page = 0x0F

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7

SFR Definition 15.3. FLSCL: Flash Scale

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	FLRT	Reserved	Reserved	FLEWT	Reserved
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xB6; SFR Page = All Pages

Bit	Name	Function
7:5	Reserved	Must Write 000b.
4	FLRT	**Flash Read Time Control.** This bit should be programmed to the smallest allowed value, according to the system clock speed. 0: SYSCLK ≤ 25 MHz (Flash read strobe is one system clock). 1: SYSCLK > 25 MHz (Flash read strobe is two system clocks).
3:2	Reserved	Must Write 00b.
1	FLEWT	**Flash Erase Write Time Control.** This bit should be set to 1b before Writing or Erasing Flash. 0: Short Flash Erase / Write Timing. 1: Extended Flash Erase / Write Timing.
0	Reserved	Must Write 0b.

Table 18.2. EMIF Pinout (C8051F588/9-F590/1)

Multiplexed Mode	
Signal Name	**Port Pin**
$\overline{RD}$	P1.6
$\overline{WR}$	P1.7
ALE	P1.5
D0/A0	P3.0
D1/A1	P3.1
D2/A2	P3.2
D3/A3	P3.3
D4/A4	P3.4
D5/A5	P3.5
D6/A6	P3.6
D7/A7	P3.7
A8	P2.0
A9	P2.1
A10	P2.2
A11	P2.3
A12	P2.4
A13	P2.5
A14	P2.6
A15	P2.7

20.1. Port I/O Modes of Operation

Port pins P0.0–P3.7 use the Port I/O cell shown in Figure 20.2. Each of these Port I/O cells can be configured by software for analog I/O or digital I/O using the PnMDIN registers. P4.0-P4.7 use a similar cell, except that they can only be configured as digital I/O pins and do not have a corresponding PnMDIN or PnSKIP register. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled until the Crossbar is enabled (XBARE = 1).

20.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC inputs, external oscillator inputs, or VREF should be configured for analog I/O (PnMDIN.n = 0). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog I/O will always read back a value of 0.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

20.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the VIO or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the VIO supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

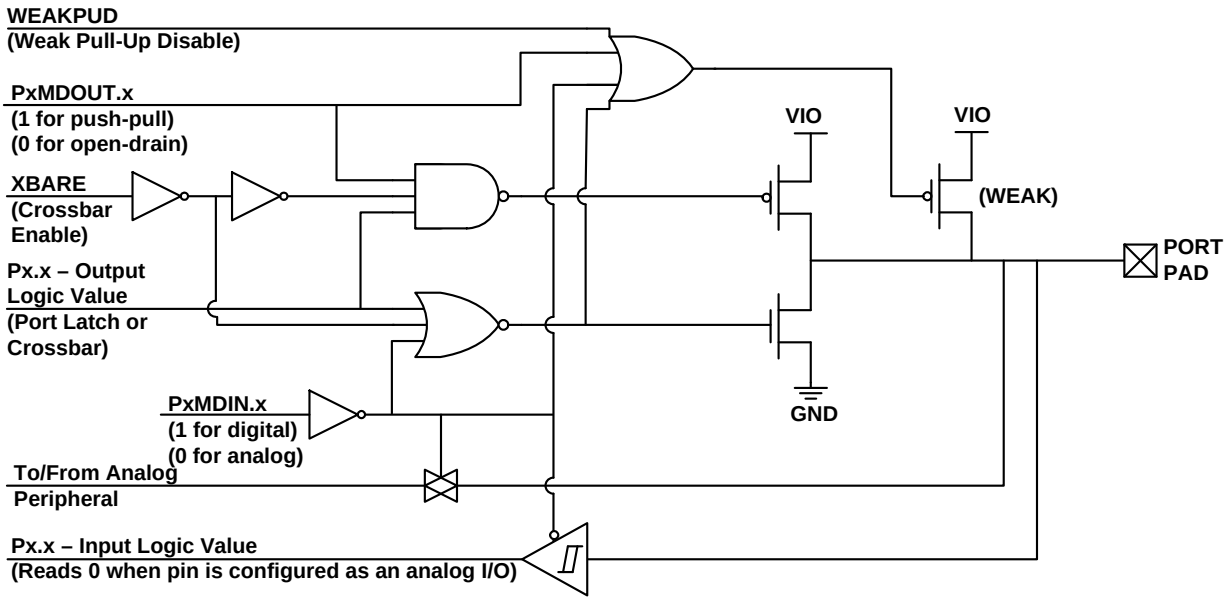

Figure 20.2. Port I/O Cell Block Diagram

SFR Definition 20.7. P1MASK: Port 1 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P1MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = 0x00

Bit	Name	Function
7:0	P1MASK[7:0]	**Port 1 Mask Value.** Selects P1 pins to be compared to the corresponding bits in P1MAT. 0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P1.n pin logic value is compared to P1MAT.n.

SFR Definition 20.8. P1MAT: Port 1 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P1MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF3; SFR Page = 0x00

Bit	Name	Function
7:0	P1MAT[7:0]	**Port 1 Match Value.** Match comparison value used on Port 1 for bits in P1MAT which are set to 1. 0: P1.n pin logic value is compared with logic LOW. 1: P1.n pin logic value is compared with logic HIGH.

SFR Definition 20.26. P3MDIN: Port 3 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF4; SFR Page = 0x0F

Bit	Name	Function
7:0	P3MDIN[7:0]	**Analog Configuration Bits for P3.7–P3.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P3MDOUT register. 0: Corresponding P3.n pin is configured for analog mode. 1: Corresponding P3.n pin is not configured for analog mode.
Note: Port P3.1–P3.7 are only available on the 48-pin and 40-pin packages.		

SFR Definition 20.27. P3MDOUT: Port 3 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAE; SFR Page = 0x0F

Bit	Name	Function
7:0	P3MDOUT[7:0]	**Output Configuration Bits for P3.7–P3.0 (respectively).** These bits are ignored if the corresponding bit in register P3MDIN is logic 0. 0: Corresponding P3.n Output is open-drain. 1: Corresponding P3.n Output is push-pull.
Note: Port P3.1–P3.7 are only available on the 48-pin and 40-pin packages.		

SFR Definition 20.28. P3SKIP: Port 3Skip

Bit	7	6	5	4	3	2	1	0
Name	P3SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD7; SFR Page = 0x0F

Bit	Name	Function
7:0	P3SKIP[7:0]	**Port 3 Crossbar Skip Enable Bits.** These bits select Port 3 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P3.n pin is not skipped by the Crossbar. 1: Corresponding P3.n pin is skipped by the Crossbar.
Note: Port P3.1–P3.7 are only available on the 48-pin and 40-pin packages.		

SFR Definition 20.29. P4: Port 4

Bit	7	6	5	4	3	2	1	0
Name	P4[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB5; SFR Page = All Pages

Bit	Name	Description	Write	Read
7:0	P4[7:0]	**Port 4 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: P4.n Port pin is logic LOW. 1: P4.n Port pin is logic HIGH.
Note: Port 4.0 is only available on the 48-pin and 40-pin packages.; P4.1-P4.7 is only available on the 48-pin packages.				

LIN Register Definition 21.6. LIN0ST: LIN0 Status Register

Bit	7	6	5	4	3	2	1	0
Name	ACTIVE	IDLTOUT	ABORT	DTREQ	LININT	ERROR	WAKEUP	DONE
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x09

Bit	Name	Function
7	ACTIVE	**LIN Active Indicator Bit.** 0: No transmission activity detected on the LIN bus. 1: Transmission activity detected on the LIN bus.
6	IDLT	**Bus Idle Timeout Bit. (slave mode only)** 0: The bus has not been idle for four seconds. 1: No bus activity has been detected for four seconds, but the bus is not yet in Sleep mode.
5	ABORT	**Aborted Transmission Bit. (slave mode only)** 0: The current transmission has not been interrupted or stopped. This bit is reset to 0 after receiving a SYNCH BREAK that does not interrupt a pending transmission. 1: New SYNCH BREAK detected before the end of the last transmission or the STOP bit (LIN0CTRL.7) has been set.
4	DTREQ	**Data Request Bit. (slave mode only)** 0: Data identifier has not been received. 1: Data identifier has been received.
3	LININT	**Interrupt Request Bit.** 0: An interrupt is not pending. This bit is cleared by setting RSTINT (LIN0CTRL.3) 1: There is a pending LIN0 interrupt.
2	ERROR	**Communication Error Bit.** 0: No error has been detected. This bit is cleared by setting RSTERR (LIN0CTRL.2) 1: An error has been detected.
1	WAKEUP	**Wakeup Bit.** 0: A wakeup signal is not being transmitted and has not been received. 1: A wakeup signal is being transmitted or has been received
0	DONE	**Transmission Complete Bit.** 0: A transmission is not in progress or has not been started. This bit is cleared at the start of a transmission. 1: The current transmission is complete.

LIN Register Definition 21.11. LIN0ID: LIN0 Identifier Register

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

Indirect Address = 0x0E

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5:0	ID[5:0]	**LIN Identifier Bits.** These bits form the data identifier. If the LINSIZE bits (LIN0SIZE[3:0]) are 1111b, bits ID[5:4] are used to determine the data size and are interpreted as follows: 00: 2 bytes 01: 2 bytes 10: 4 bytes 11: 8 bytes

22. Controller Area Network (CAN0)

Important Documentation Note: The Bosch CAN Controller is integrated in the C8051F580/2/4/6/8-F590 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, refer to Bosch's C_CAN User's Manual as an accompanying manual to the Silicon Labs' data sheet.

The C8051F580/2/4/6/8-F590 devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN 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 CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 22.1 shows an example typical configuration on a CAN bus.

Silicon Labs CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their 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 CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFRs) in the CIP-51.

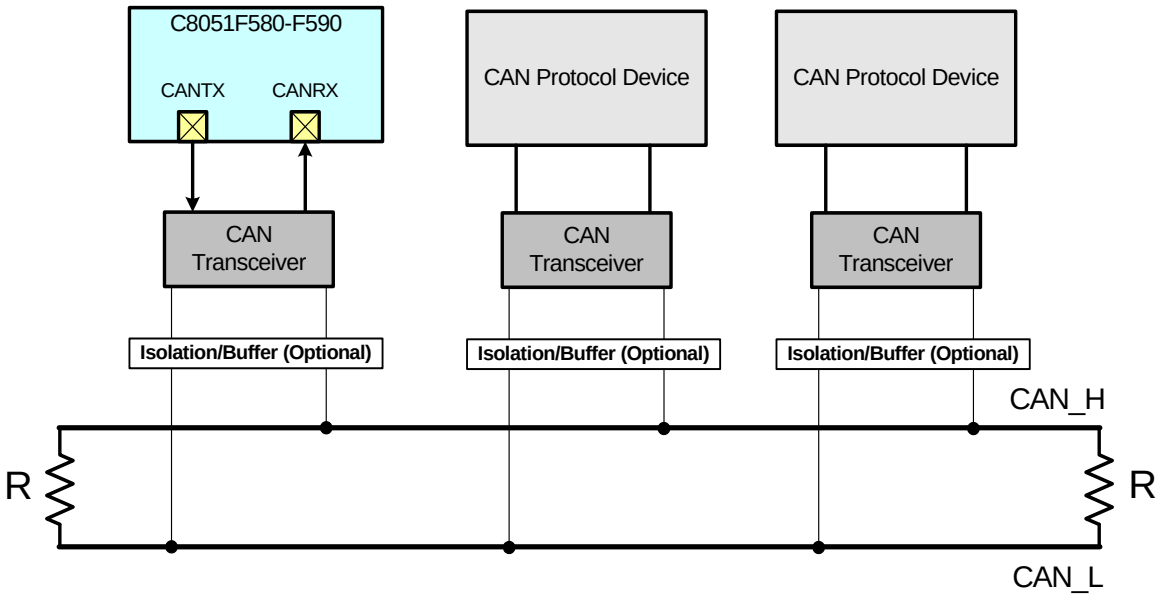

Figure 22.1. Typical CAN Bus Configuration

Table 22.2. Standard CAN Registers and Reset Values (Continued)

CAN Addr.	Name	SFR Name (High)	SFR Addr.	SFR Name (Low)	SFR Addr.	16-bit SFR	Reset Value
0x50	IF2 Data A 2	CAN0IF2DA2H	0xFB	CAN0IF2DA2L	0xFA	CAN0IF2DA2	0x0000
0x52	IF2 Data B 1	CAN0IF2DB1H	0xFD	CAN0IF2DB1L	0xFC	CAN0IF2DB1	0x0000
0x54	IF2 Data B 2	CAN0IF2DB2H	0xFF	CAN0IF2DB2L	0xFE	CAN0IF2DB2	0x0000
0x80	Transmission Request 1[1]	CAN0TR1H	0xA3	CAN0TR1L	0xA2	CAN0TR1	0x0000
0x82	Transmission Request 2[1]	CAN0TR2H	0xA5	CAN0TR2L	0xA4	CAN0TR2	0x0000
0x90	New Data 1[1]	CAN0ND1H	0xAB	CAN0ND1L	0xAA	CAN0ND1	0x0000
0x92	New Data 2[1]	CAN0ND2H	0xAD	CAN0ND2L	0xAC	CAN0ND2	0x0000
0xA0	Interrupt Pending 1[1]	CAN0IP1H	0xAF	CAN0IP1L	0xAE	CAN0IP1	0x0000
0xA2	Interrupt Pending 2 [1]	CAN0IP2H	0xB3	CAN0IP2L	0xB2	CAN0IP2	0x0000
0xB0	Message Valid 1[1]	CAN0MV1H	0xBB	CAN0MV1L	0xBA	CAN0MV1	0x0000
0xB2	Message Valid 2[1]	CAN0MV2H	0xBD	CAN0MV2L	0xBC	CAN0MV2	0x0000

Notes:
1. Read-only register.
2. Write-enabled by CCE.
3. The reset value of CAN0TST could also be r0000000b, where r signifies the value of the CAN RX pin.
4. Write-enabled by Test.

23.4.3. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into 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. 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 or address in SMB0DAT.

SFR Definition 23.3. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC2; SMB0DAT = 0x00

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

23.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. As a receiver, the interrupt for an ACK occurs **before** the ACK. As a transmitter, interrupts occur **after** the ACK.

23.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter 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 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 23.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the 'data byte transferred' interrupts occur **after** the ACK cycle in this mode.

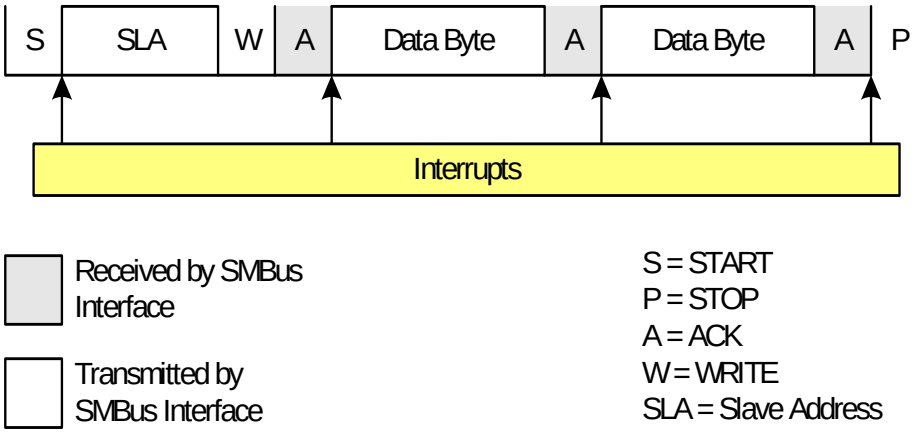

Figure 23.5. Typical Master Write Sequence

26. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

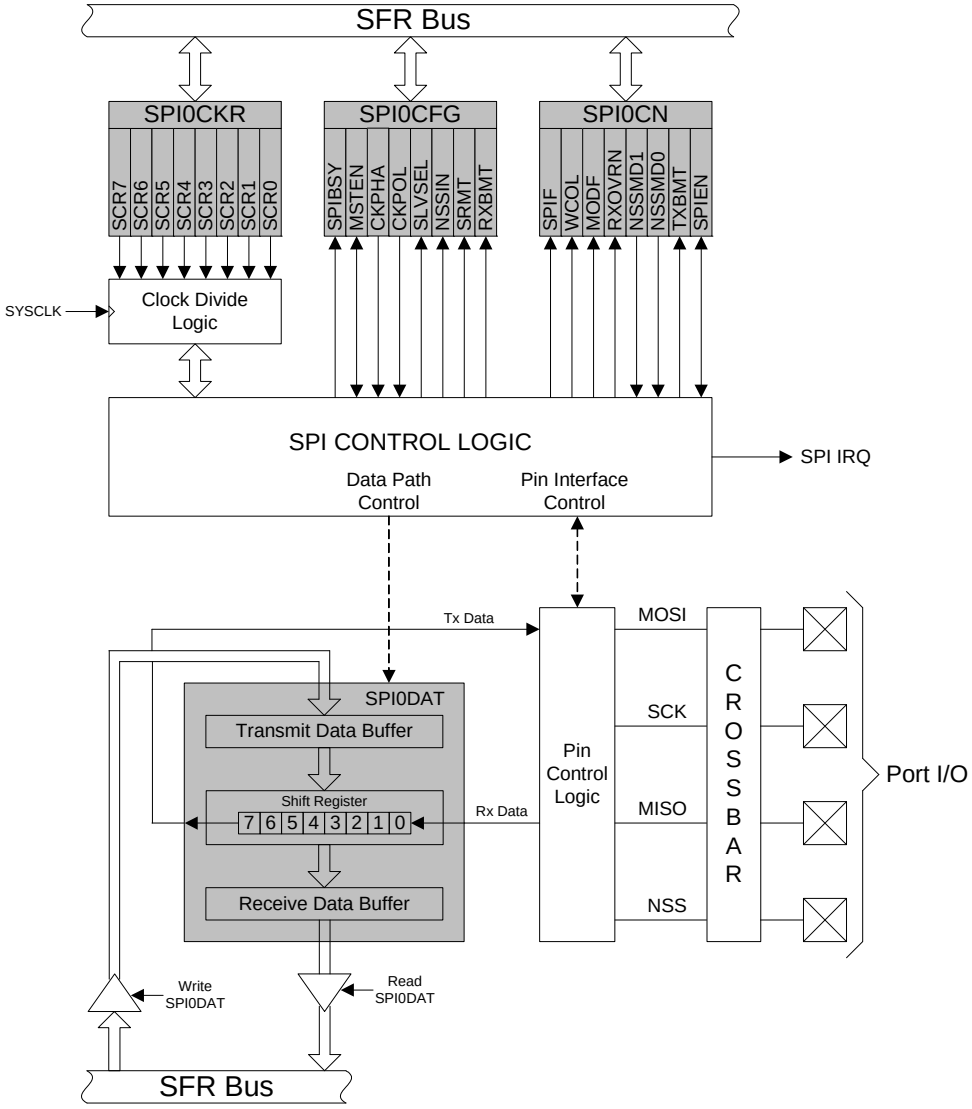

Figure 26.1. SPI Block Diagram

SFR Definition 27.22. TMRnL: Timer 4 and 5 Low Byte

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

TMR4L SFR Address = 0xCC; TMR5L SFR Address = 0x94; SFR Page = 0x10

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

SFR Definition 27.23. TMRnH Timer 4 and 5 High Byte

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

TMR4H SFR Address = 0xCD; TMR5H SFR Address = 0x95; SFR Page = 0x10

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