# E·XFL

#### Silicon Labs - C8051F557-IMR Datasheet



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	18
Program Memory Size	16KB (16K 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 18x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f557-imr

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

#### Table 5.10. Temperature Sensor Electrical Characteristics

VDDA = 1.8 to 2.75 V, -40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Тур	Мах	Units			
Linearity			±0.1		°C			
Slope			3.33		mV/°C			
Slope Error*			88		µV/°C			
Offset	Temp = 0 °C		856	_	mV			
Offset Error*	Temp = 0 °C		±14		mV			
Power Supply Current			18		μA			
Tracking Time		12	_	_	μs			
*Note: Represents one standard deviation from the mean.								

## Table 5.11. Voltage Reference Electrical Characteristics

VDDA = 1.8 to 2.75 V, -40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Тур	Max	Units			
nternal Reference (REFBE = 1)								
Output Voltage	25 °C ambient (REFLV = 0)	1.45	1.50	1.55	N/			
	25 °C ambient (REFLV = 1), $V_{DD}$ = 2.6 V	2.15	2.20	2.25	V			
VREF Short-Circuit Current		_	5	10	mA			
VREF Temperature Coefficient		_	38	_	ppm/°C			
Power Consumption	Internal	_	30	50	μA			
Load Regulation	Load = 0 to 200 µA to AGND	_	3	_	μV/μA			
VREF Turn-on Time 1	4.7 $\mu$ F tantalum and 0.1 $\mu$ F bypass	_	1.5	_	ms			
VREF Turn-on Time 2	0.1 μF bypass	_	46	_	μs			
Power Supply Rejection		_	1.2	_	mV/V			
External Reference (REFBI	Ē = 0)							
Input Voltage Range		1.5		$V_{DDA}$	V			
Input Current	Sample Rate = 200 ksps; VREF = 1.5 V	_	2.1	_	μA			
Power Specifications				•				
Reference Bias Generator	REFBE = 1 or TEMPE = 1		21	40	μA			



## SFR Definition 6.4. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name			AD0SC[4:0]	ADORI	PT[1:0]	GAINEN		
Туре			R/W		R/W	R/W	R/W	
Reset	1	1	1	1	1	0	0	0

SFR Address = 0xBC; SFR Page = 0x00

Bit	Name	Function
7:3	AD0SC[4:0]	ADC0 SAR Conversion Clock Period Bits.
		SAR Conversion clock is derived from system clock by the following equation, where <i>ADOSC</i> refers to the 5-bit value held in bits ADOSC4–0. SAR Conversion clock requirements are given in the ADC specification table BURSTEN = 0: FCLK is the current system clock BURSTEN = 1: FCLK is a maximum of 30 MHz, independent of the current system clock
		$ADOSC = \frac{10DR}{CLK_{SAR}} - 1$
		Note: Round up the result of the calculation for AD0SC
2:1	A0RPT[1:0]	ADC0 Repeat Count.
		Controls the number of conversions taken and accumulated between ADC0 End of Conversion (ADCINT) and ADC0 Window Comparator (ADCWINT) interrupts. A con- vert start is required for each conversion unless Burst Mode is enabled. In Burst Mode, a single convert start can initiate multiple self-timed conversions. Results in both modes are accumulated in the ADC0H:ADC0L register. When AD0RPT1–0 are set to a value other than '00', the AD0LJST bit in the ADC0CN register must be set to '0' (right justified). 00: 1 conversion is performed. 01: 4 conversions are performed and accumulated. 10: 8 conversions are performed and accumulated. 11: 16 conversions are performed and accumulated.
0	GAINEN	Gain Enable Bit.
		Controls the gain programming. Refer to Section "6.3. Selectable Gain" on page 53 for information about using this bit.



## 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]	
Туре	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	<ul> <li>Comparator1 Rising-Edge Flag. Must be cleared by software.</li> <li>0: No Comparator1 Rising Edge has occurred since this flag was last cleared.</li> <li>1: Comparator1 Rising Edge has occurred.</li> </ul>
4	CP1FIF	<ul> <li>Comparator1 Falling-Edge Flag. Must be cleared by software.</li> <li>0: No Comparator1 Falling-Edge has occurred since this flag was last cleared.</li> <li>1: Comparator1 Falling-Edge has occurred.</li> </ul>
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.



#### SFR Definition 10.1. DPL: Data Pointer Low Byte

7	6	5	4	3	2	1	0	
DPL[7:0]								
R/W								
0	0	0	0	0	0	0	0	
	7 0	7         6           0         0	7         6         5           0         0         0	7         6         5         4           DPL         DPL           0         0         0	7         6         5         4         3           DPL[7:0]           R/W           0         0         0         0         0	7         6         5         4         3         2           DPL[7:0]           R/W           0         0         0         0         0	7         6         5         4         3         2         1           DPL[7:0]           R/W           0         0         0         0         0         0         0	

SFR Address = 0x82; SFR Page = All Pages

Bit	Name	Function
7:0	DPL[7:0]	Data Pointer Low.
		The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indi- rectly addressed Flash memory or XRAM.

## SFR Definition 10.2. DPH: Data Pointer High Byte

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

SFR Address = 0x83; SFR Page = All Pages

Bit	Name	Function
7:0	DPH[7:0]	Data Pointer High.
		The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indi- rectly addressed Flash memory or XRAM.



### **10.4.** Serial Number Special Function Registers (SFRs)

The C8051F55x/56x/57x devices include four SFRs, SN0 through SN3, that are pre-programmed during production with a unique, 32-bit serial number. The serial number provides a unique identification number for each device and can be read from the application firmware. If the serial number is not used in the application, these four registers can be used as general purpose SFRs.

### SFR Definition 10.7. SNn: Serial Number n

Bit	7	6	5	4	3	2	1	0		
Nam	e	SERNUMn[7:0]								
Тур	9	R/W								
Rese	et		Va	aries—Uniqu	ie 32-bit val	ue				
SFR /	Addresses: SN0	= 0xF9; SI	N1 = 0xFA; S	SN2 = 0xFB;	SN3 = 0xFC	C; SFR Page	= 0x0F;			
Bit	Name				Functior	า				
7:0	SERNUMn[7:0	)] Serial N	Serial Number Bits.							
		The four most sig	<sup>-</sup> serial numb Inificant byte	er registers and SN0 as	form a 32-bi the least si	it serial numb gnificant byte	per, with SN3 e.	as the		



## SFR Definition 12.4. SFRLAST: SFR Last

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

SFR Address = 0xA7; SFR Page = All Pages

Bit	Name	Function
7:0	SFRLAST[7:0]	SFR Page Stack Bits.
		This is the value that will go to the SFRNEXT register upon a return from inter- rupt.
		Write: Sets the SFR Page in the last entry of the SFR Stack. This will cause the SFRNEXT SFR to have this SFR page value upon a return from interrupt.
		Read: Returns the value of the SFR page contained in the last entry of the SFR stack.
		SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to "push" or "pop". Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.



#### Table 13.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Тор	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPIO (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
ADC0 Window Com- pare	0x0043	8	ADOWINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.1)	PWADC0 (EIP1.1)
ADC0 Conversion Complete	0x004B	9	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.2)	PADC0 (EIP1.2)
Programmable Counter Array	0x0053	10	CF (PCA0CN.7) CCFn (PCA0CN.n) COVF (PCA0PWM.6)	Y	N	EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
Comparator0	0x005B	11	CPOFIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.4)	PCP0 (EIP1.4)
Comparator1	0x0063	12	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.5)	PCP1 (EIP1.5)
Timer 3 Overflow	0x006B	13	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.6)	PT3 (EIP1.6)
LINO	0x0073	14	LIN0INT (LINST.3)	N	N*	ELIN0 (EIE1.7)	PLIN0 (EIP1.7)
Voltage Regulator Dropout	0x007B	15	N/A	N/A	N/A	EREG0 (EIE2.0)	PREG0 (EIP2.0)
CAN0	0x0083	16	CAN0INT (CAN0CN.7)	N	Y	ECAN0 (EIE2.1)	PCAN0 (EIP2.1)
Port Match	0x008B	17	None	N/A	N/A	EMAT (EIE2.2)	PMAT



## SFR Definition 13.7. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL		IN1SL[2:0]		IN0PL	IN0SL[2:0]		
Туре	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	INT1 Polarity. 0: INT1 input is active low.
		1: INT1 input is active high.
6:4	IN1SL[2:0]	INT1 Port Pin Selection Bits. These bits select which Port pin is assigned to INT1. Note that this pin assignment is independent of the Crossbar; INT1 will monitor the assigned Port pin without disturb- ing 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	INOPL	INTO Polarity. 0: INTO input is active low. 1: INTO input is active high.
2:0	IN0SL[2:0]	INTO Port Pin Selection Bits. These bits select which Port pin is assigned to INTO. Note that this pin assignment is independent of the Crossbar; INTO will monitor the assigned Port pin without disturb- ing 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



#### 14.1.4. Flash Write Optimization

The Flash write procedure includes a block write option to optimize the time to perform consecutive byte writes. When block write is enabled by setting the CHBLKW bit (CCH0CN.0), writes to two consecutive bytes in Flash require the same amount of time as a single byte write. This is performed by caching the first byte that is written to Flash and then committing both bytes to Flash when the second byte is written. When block writes are enabled, if the second write does not occur, the first data byte written is not actually written to Flash. Flash bytes with block write enabled are programmed by software with the following sequence:

- 1. Disable interrupts (recommended).
- 2. Erase the 512-byte Flash page containing the target location, as described in Section 14.1.2.
- 3. Set the FLEWT bit (register FLSCL).
- 4. Set the CHBLKW bit (register CCH0CN).
- 5. Set the PSWE bit (register PSCTL).
- 6. Clear the PSEE bit (register PSCTL).
- 7. Write the first key code to FLKEY: 0xA5.
- 8. Write the second key code to FLKEY: 0xF1.
- 9. Using the MOVX instruction, write the first data byte to the desired location within the 512-byte sector.
- 10. Write the first key code to FLKEY: 0xA5.
- 11. Write the second key code to FLKEY: 0xF1.
- 12. Using the MOVX instruction, write the second data byte to the desired location within the 512-byte sector. The location of the second byte must be the next higher address from the first data byte.
- 13.Clear the PSWE bit.
- 14.Clear the CHBLKW bit.

### 16.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a 1 to the CORSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to CORSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0–), the device is put into the reset state. After a Comparator0 reset, the CORSEF flag (RSTSRC.5) will read 1 signifying Comparator0 as the reset source; otherwise, this bit reads 0. The state of the RST pin is unaffected by this reset.

#### 16.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section "26.4. Watchdog Timer Mode" on page 291; the WDT is enabled and clocked by SYSCLK/12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to 1. The state of the RST pin is unaffected by this reset.

#### 16.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to 1 and a MOVX write operation targets an address in or above the reserved space.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address in or above the reserved space.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address in or above the reserved space.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section "14.3. Security Options" on page 127).
- A Flash read, write, or erase is attempted when the VDD Monitor is not enabled to the high threshold and set as a reset source.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the  $\overline{RST}$  pin is unaffected by this reset.

#### 16.8. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the RST pin is unaffected by this reset.



#### **17.5. Memory Mode Selection**

The external data memory space can be configured in one of four modes, shown in Figure 17.2, based on the EMIF Mode bits in the EMIOCF register (SFR Definition 17.2). These modes are summarized below. More information about the different modes can be found in Section "17.6. Timing" on page 151.



Figure 17.2. EMIF Operating Modes

#### 17.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 2 kB boundaries. As an example, the addresses 0x800 and 0x1000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

#### 17.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and offchip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is onchip or off-chip. However, in the "No Bank Select" mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with "Split Mode with Bank Select" described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.



- 3. The LIN controller does not directly support LIN Version 1.3 Extended Frames. If the application detects an unknown identifier (e.g. extended identifier), it has to write a 1 to the STOP bit (LIN0CTRL.7) instead of setting the DTACK (LIN0CTRL.4) bit. At that time, steps 2 through 5 can then be skipped. In this situation, the LIN controller stops the processing of LIN communication until the next SYNC BREAK is received.
- 4. Changing the configuration of the checksum during a transaction will cause the interface to reset and the transaction to be lost. To prevent this, the checksum should not be configured while a transaction is in progress. The same applies to changes in the LIN interface mode from slave mode to master mode and from master mode to slave mode.

#### 20.5. Sleep Mode and Wake-Up

To reduce the system's power consumption, the LIN Protocol Specification defines a Sleep Mode. The message used to broadcast a Sleep Mode request must be transmitted by the LIN master application in the same way as a normal transmit message. The LIN slave application must decode the Sleep Mode Frame from the Identifier and data bytes. After that, it has to put the LIN slave node into the Sleep Mode by setting the SLEEP bit (LIN0CTRL.6).

If the SLEEP bit (LINOCTRL.6) of the LIN slave application is not set and there is no bus activity for four seconds (specified bus idle timeout), the IDLTOUT bit (LINOST.6) is set and an interrupt request is generated. After that the application may assume that the LIN bus is in Sleep Mode and set the SLEEP bit (LINOCTRL.6).

Sending a wake-up signal from the master or any slave node terminates the Sleep Mode of the LIN bus. To send a wake-up signal, the application has to set the WUPREQ bit (LIN0CTRL.1). After successful transmission of the wake-up signal, the DONE bit (LIN0ST.0) of the master node is set and an interrupt request is generated. The LIN slave does not generate an interrupt request after successful transmission of the wake-up signal but it generates an interrupt request if the master does not respond to the wake-up signal within 150 milliseconds. In that case, the ERROR bit (LIN0ST.2) and TOUT bit (LIN0ERR.2) are set. The application then has to decide whether or not to transmit another wake-up signal.

All LIN nodes that detect a wake-up signal will set the WAKEUP (LIN0ST.1) and DONE bits (LIN0ST.0) and generate an interrupt request. After that, the application has to clear the SLEEP bit (LIN0CTRL.6) in the LIN slave.

#### 20.6. Error Detection and Handling

The LIN controller generates an interrupt request and stops the processing of the current frame if it detects an error. The application has to check the type of error by processing LIN0ERR. After that, it has to reset the error register and the ERROR bit (LIN0ST.2) by writing a 1 to the RSTERR bit (LIN0CTRL.2). Starting a new message with the LIN controller selected as master or sending a Wakeup signal with the LIN controller selected as a master or slave is possible only if the ERROR bit (LIN0ST.2) is set to 0.



### 21.1. Bosch CAN Controller Operation

The CAN Controller featured in the C8051F550/1/4/5, 'F560/1/4/5/8/9, and 'F572/3 devices is a full implementation of Bosch's full CAN module and fully complies with CAN specification 2.0B. A block diagram of the CAN controller is shown in Figure 21.2. The CAN Core provides shifting (CANTX and CANRX), serial/ parallel conversion of messages, and other protocol related tasks such as transmission of data and acceptance filtering. The message RAM stores 32 message objects which can be received or transmitted on a CAN network. The CAN registers and message handler provide an interface for data transfer and notification between the CAN controller and the CIP-51.

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 data sheet describes how to access the CAN controller.

All of the CAN controller registers are located on SFR Page 0x0C. Before accessing any of the CAN registers, the SFRPAGE register must be set to 0x0C.

The CAN Controller is typically initialized using the following steps:

- 1. Set the SFRPAGE register to the CAN registers page (page 0x0C).
- 2. Set the INIT and the CCE bits to 1 in CAN0CN. See the CAN User's Guide for bit definitions.
- 3. Set timing parameters in the Bit Timing Register and the BRP Extension Register.
- 4. Initialize each message object or set its MsgVal bit to NOT VALID.
- 5. Reset the INIT bit to 0.





#### 21.1.1. CAN Controller Timing

The CAN controller's clock (fsys) is derived from the CIP-51 system clock (SYSCLK). The internal oscillator is accurate to within 0.5% of 24 MHz across the entire temperature range and for VDD voltages greater than or equal to the minimum output of the on-chip voltage regulator, so an external oscillator is not required for CAN communication for most systems.

Refer to Section "4.10.4 Oscillator Tolerance Range" in the Bosch CAN User's Guide for further information regarding this topic.



overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

#### 22.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. Note that a clock source is required for free timeout detection, even in a slave-only implementation.

#### 22.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

SMBus interrupts are generated for each data byte or slave address that is transferred. The point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e. sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e. receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See Section 22.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section 22.4.2; Table 22.4 provides a quick SMB0CN decoding reference.

#### 22.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).



### 24.2. SPI0 Master Mode Operation

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

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSS-MD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 24.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 24.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 24.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.



## SFR Definition 24.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0	
Nam	e SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0] TXBMT S		SPIEN		
Туре	R/W	R/W	R/W	R/W	R/W R R/W				
Rese	et O	0	0	0	0	1	1	0	
SFR Address = 0xF8; Bit-Addressable; SFR Page = 0x00									
Bit	Name				Function	Ì			
7	SPIF	SPI0 Inte This bit is enabled, s tine. This	<b>SPI0 Interrupt Flag.</b> This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by bardware. It must be cleared by setting						
		ware.							
6	WCOL	Write Col This bit is write to the It must be	Write Collision Flag. This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.						
5	MODF	Mode Fau	Mode Fault Flag.						
		This bit is ter mode This bit is	This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, $MSTEN = 1$ , and $NSSMD[1:0] = 01$ ). This bit is not automatically cleared by hardware. It must be cleared by software.						
4	RXOVRN	Receive (	Overrun Fla	g (valid in s	lave mode	only).			
		This bit is receive bu current tra cleared by	set to logic uffer still holo ansfer is shif / hardware.	1 by hardwa Is unread da ted into the S It must be cl	re (and gen ta from a pr SPI0 shift re eared by so	erates a SP evious trans gister. This ftware.	10 interrupt) v sfer and the la bit is not auto	when the ast bit of the omatically	
3:2	NSSMD[1:0]	Slave Sel	ect Mode.						
		Selects be (See Sect 00: 3-Wire 01: 4-Wire 1x: 4-Wire device an	<ul> <li>Selects between the following NSS operation modes:</li> <li>(See Section 24.2 and Section 24.3).</li> <li>00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin.</li> <li>01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device.</li> <li>1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.</li> </ul>						
1	TXBMT	Transmit	Buffer Emp	oty.					
		This bit wi When dat be set to l	This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.						
0	SPIEN	SPI0 Ena 0: SPI dis 1: SPI ena	<b>ble.</b> abled. abled.						





Figure 25.1. T0 Mode 0 Block Diagram

#### 25.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

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

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

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



This mode allows software to determine the external oscillator frequency when an RC network or capacitor is used to generate the clock source.



Figure 25.6. Timer 2 External Oscillator Capture Mode Block Diagram



## SFR Definition 25.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Туре	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]							
Туре	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.





Figure 26.4. PCA Capture Mode Diagram

**Note:** The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

#### 26.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer 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.

