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), 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-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f581-iqr

SFR Definition 6.13. ADC0MX: ADC0 Channel Select

Bit	7	6	5	4	3	2	1	0
Name			ADC0MX[5:0]					
Type	R	R	R/W					
Reset	0	0	1	1	1	1	1	1

SFR Address = 0xBB; SFR Page = 0x00;

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5:0	AMX0P[5:0]	**AMUX0 Positive Input Selection.** 000000: P0.0 000001: P0.1 000010: P0.2 000011: P0.3 000100: P0.4 000101: P0.5 000110: P0.6 000111: P0.7 001000: P1.0 001001: P1.1 001010: P1.2 001011: P1.3 001100: P1.4 001101: P1.5 001110: P1.6 001111: P1.7 010000: P2.0 010001: P2.1 010010: P2.2 010011: P2.3 010100: P2.4 010101: P2.5 010110: P2.6 010111: P2.7 011000: P3.0 011001: P3.1 (Available on 48-pin and 40-pin package devices) 011010: P3.2 (Available on 48-pin and 40-pin package devices) 011011: P3.3 (Available on 48-pin and 40-pin package devices) 011100: P3.4 (Available on 48-pin and 40-pin package devices) 011101: P3.5 (Available on 48-pin and 40-pin package devices) 011110: P3.6 (Available on 48-pin and 40-pin package devices) 011111: P3.7 (Available on 48-pin and 40-pin package devices) 100000–101111: Reserved 110000: Temp Sensor 110001: V_{DD} 110010–111111: GND

SFR Definition 14.2. IP: Interrupt Priority

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

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

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

SFR Definition 15.4. CCH0CN: Cache Control

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

SFR Address = 0xE3; SFR Page = 0x0F

Bit	Name	Function
7:6	Reserved	Must Write 00b
5	CHPFEN	**Cache Prefect Enable Bit.** 0: Prefetch engine is disabled. 1: Prefetch engine is enabled.
4:1	Reserved	Must Write 0000b.
0	CHBLKW	**Block Write Enable Bit.** This bit allows block writes to Flash memory from firmware. 0: Each byte of a software Flash write is written individually. 1: Flash bytes are written in groups of two.

SFR Definition 15.5. ONESHOT: Flash Oneshot Period

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

SFR Address = 0xBE; SFR Page = 0x0F

Bit	Name	Function
7:4	Unused	Read = 0000b. Write = don't care.
3:0	PERIOD[3:0]	**Oneshot Period Control Bits.** These bits limit the internal Flash read strobe width as follows. When the Flash read strobe is de-asserted, the Flash memory enters a low-power state for the remainder of the system clock cycle. These bits have no effect when the system clocks is greater than 12.5 MHz and FLRT = 0. $FLASH_{RDMAX} = 5ns + (PERIOD \times 5ns)$

SFR Definition 18.1. EMI0CN: External Memory Interface Control

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

SFR Address = 0xAA; SFR Page = 0x00

Bit	Name	Function
7:0	PGSEL[7:0]	**XRAM Page Select Bits.** The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. 0x00: 0x0000 to 0x00FF 0x01: 0x0100 to 0x01FF ... 0xFE: 0xFE00 to 0xFEFF 0xFF: 0xFF00 to 0xFFFF

18.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 101 or 111

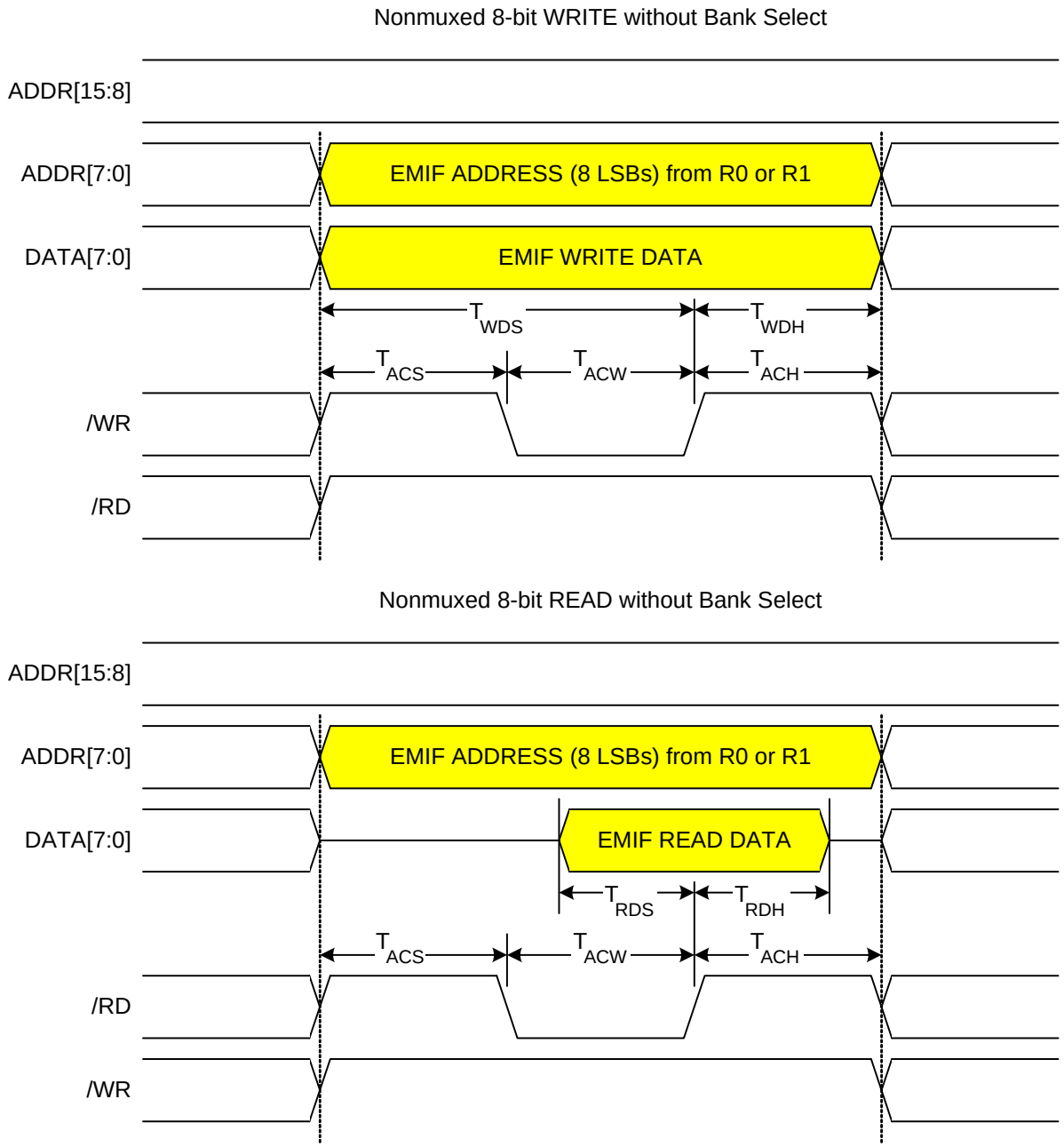

Figure 18.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

SFR Definition 19.3. OSCICRS: Internal Oscillator Coarse Calibration

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

SFR Address = 0xA2; SFR Page = 0x0F;

Bit	Name	Function
7	Unused	Read = 0; Write = Don't Care
6:0	OSCICRS[6:0]	**Internal Oscillator Coarse Calibration Bits.** These bits determine the internal oscillator period. When set to 0000000b, the internal oscillator operates at its slowest setting. When set to 1111111b, the internal oscillator operates at its fastest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 19.4. OSCIFIN: Internal Oscillator Fine Calibration

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

SFR Address = 0x9E; SFR Page = 0x0F;

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care
5:0	OSCIFIN[5:0]	**Internal Oscillator Fine Calibration Bits.** These bits are fine adjustment for the internal oscillator period. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

19.4.1. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 19.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 19.6 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b and a 32.768 kHz Watch Crystal requires an XFCN setting of 001b. After an external 32.768 kHz oscillator is stabilized, the XFCN setting can be switched to 000 to save power. It is recommended to enable the missing clock detector before switching the system clock to any external oscillator source.

Note: Small surface mount crystals can have maximum drive level specifications that are exceeded by the above XFCN recommendations. In these cases, a software-controlled startup sequence may be used to reliably start the crystal using a higher XFCN setting, and then lowering the XFCN setting once the oscillator has started to reduce the drive level and prevent damage or premature aging of the crystal. In all cases, the drive level should be measured to ensure that the crystal is being driven within its operational guidelines as part of robust oscillator system design. Contact technical support for additional details and recommendations if using surface mount crystals with these devices.

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

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

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

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

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

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 19.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 19.3.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.1 if the ADC is configured to use the external conversion start signal (CNVSTR), P0.3 and/or P0.2 if the external oscillator circuit is enabled, and any selected ADC or Comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

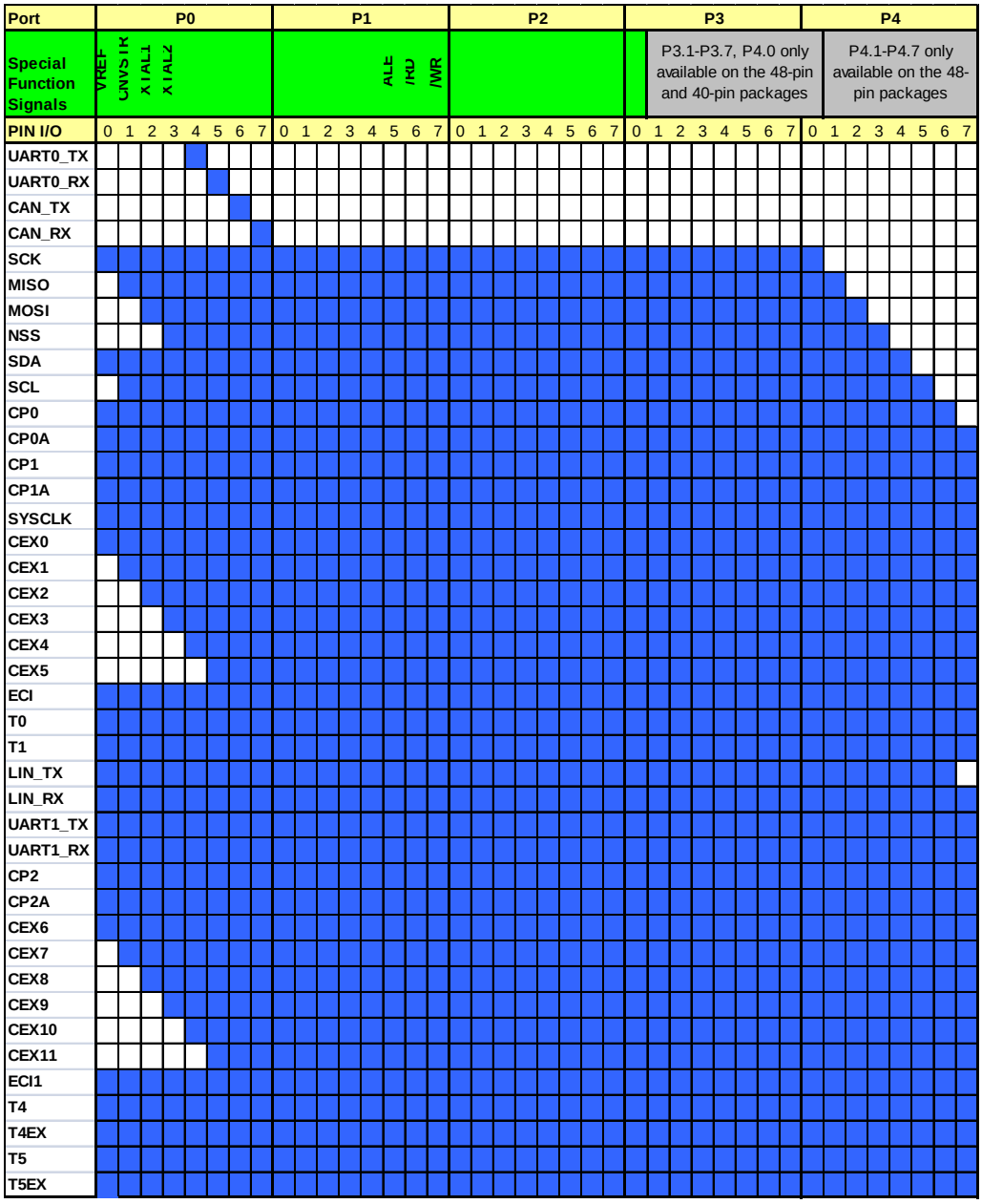

Figure 20.3. Peripheral Availability on Port I/O Pins

Registers XBR0, XBR1, XBR2, and XBR3 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); and similarly when the UART, CAN or LIN are selected, the Crossbar assigns both pins associated with the peripheral (TX and RX).

21.1. Software Interface with the LIN Controller

The selection of the mode (Master or Slave) and the automatic baud rate feature are done though the LIN0 Control Mode (LIN0CF) register. The other LIN registers are accessed indirectly through the two SFRs LIN0 Address (LIN0ADR) and LIN0 Data (LIN0DAT). The LIN0ADR register selects which LIN register is targeted by reads/writes of the LIN0DAT register. The full list of indirectly-accessible LIN registers is given in Table 21.4 on page 223.

21.2. LIN Interface Setup and Operation

The hardware based LIN controller allows for the implementation of both Master and Slave nodes with minimal firmware overhead and complete control of the interface status while allowing for interrupt and polled mode operation.

The first step to use the controller is to define the basic characteristics of the node:

Mode—Master or Slave

Baud Rate—Either defined manually or using the autobaud feature (slave mode only)

Checksum Type—Select between classic or enhanced checksum, both of which are implemented in hardware.

21.2.1. Mode Definition

Following the LIN specification, the controller implements in hardware both the Slave and Master operating modes. The mode is configured using the MODE bit (LIN0CF.6).

21.2.2. Baud Rate Options: Manual or Autobaud

The LIN controller can be selected to have its baud rate calculated manually or automatically. A master node must always have its baud rate set manually, but slave nodes can choose between a manual or automatic setup. The configuration is selected using the ABAUD bit (LIN0CF.5).

Both the manual and automatic baud rate configurations require additional setup. The following sections explain the different options available and their relation with the baud rate, along with the steps necessary to achieve the required baud rate.

21.2.3. Baud Rate Calculations: Manual Mode

The baud rate used by the LIN controller is a function of the System Clock (SYSCLK) and the LIN timing registers according to the following equation:

$$baud\_rate = \frac{SYSCLK}{2^{(prescaler+1)} \times divider \times (multiplier+1)}$$

The prescaler, divider and multiplier factors are part of the LIN0DIV and LIN0MUL registers and can assume values in the following range:

Table 21.1. Baud Rate Calculation Variable Ranges

Factor	Range
prescaler	0…3
multiplier	0…31
divider	200…511

Important Note: The minimum system clock (SYSCLK) to operate the LIN controller is 8 MHz.

Use the following equations to calculate the values for the variables for the baud-rate equation:

LIN Register Definition 21.7. LIN0ERR: LIN0 Error Register

Bit	7	6	5	4	3	2	1	0
Name				SYNCH	PRTY	TOUT	CHK	BITERR
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x0A

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care
4	SYNCH	**Synchronization Error Bit (slave mode only).** 0: No error with the SYNCH FIELD has been detected. 1: Edges of the SYNCH FIELD are outside of the maximum tolerance.
3	PRTY	**Parity Error Bit (slave mode only).** 0: No parity error has been detected. 1: A parity error has been detected.
2	TOUT	**Timeout Error Bit.** 0: A timeout error has not been detected. 1: A timeout error has been detected. This error is detected whenever one of the following conditions is met: • The master is expecting data from a slave and the slave does not respond. • The slave is expecting data but no data is transmitted on the bus. • A frame is not finished within the maximum frame length. • The application does not set the DTACK bit (LIN0CTRL.4) or STOP bit (LIN0CTRL.7) until the end of the reception of the first byte after the identifier.
1	CHK	**Checksum Error Bit.** 0: Checksum error has not been detected. 1: Checksum error has been detected.
0	BITERR	**Bit Transmission Error Bit.** 0: No error in transmission has been detected. 1: The bit value monitored during transmission is different than the bit value sent.

23.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

23.2. SMBus Configuration

Figure 23.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

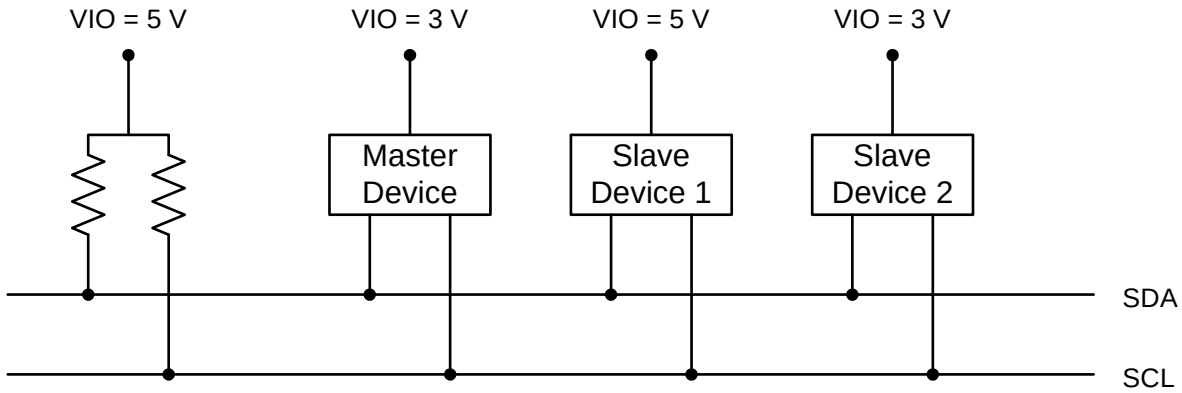

Figure 23.2. Typical SMBus Configuration

23.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. It is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 23.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

Table 23.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	■ A START is generated.	■ A STOP is generated. ■ Arbitration is lost.
TXMODE	■ START is generated. ■ SMB0DAT is written before the start of an SMBus frame.	■ A START is detected. ■ Arbitration is lost. ■ SMB0DAT is not written before the start of an SMBus frame.
STA	■ A START followed by an address byte is received.	■ Must be cleared by software.
STO	■ A STOP is detected while addressed as a slave. ■ Arbitration is lost due to a detected STOP.	■ A pending STOP is generated.
ACKRQ	■ A byte has been received and an ACK response value is needed.	■ After each ACK cycle.
ARBLOST	■ A repeated START is detected as a MASTER when STA is low (unwanted repeated START). ■ SCL is sensed low while attempting to generate a STOP or repeated START condition. ■ SDA is sensed low while transmitting a 1 (excluding ACK bits).	■ Each time SI is cleared.
ACK	■ The incoming ACK value is low (ACKNOWLEDGE).	■ The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	■ A START has been generated. ■ Lost arbitration. ■ A byte has been transmitted and an ACK/NACK received. ■ A byte has been received. ■ A START or repeated START followed by a slave address + R/W has been received. ■ A STOP has been received.	■ Must be cleared by software.

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.

23.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. An interrupt is generated after each received byte.

Software must write the ACK bit at that time to ACK or NACK the received byte. Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 23.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

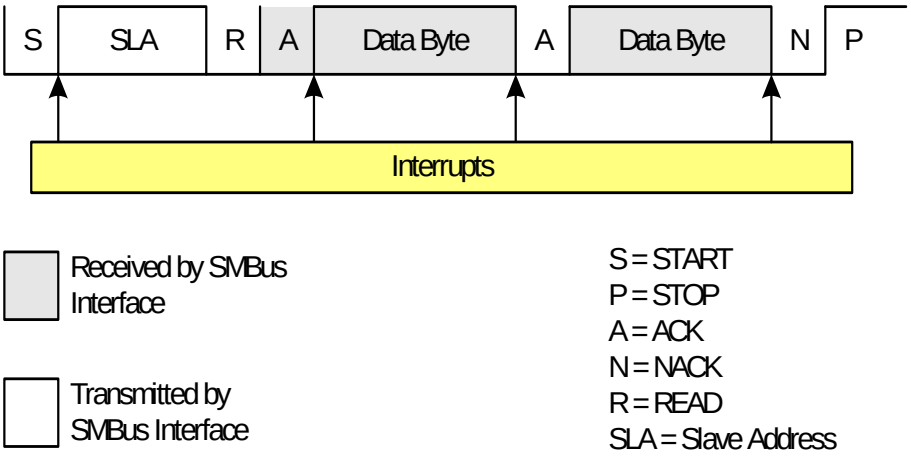

Figure 23.6. Typical Master Read Sequence

SFR Definition 25.2. SBUF1: Serial (UART1) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x99; SFR Page = 0x10

Bit	Name	Function
7:0	SBUF1[7:0]	**Serial Data Buffer Bits 7–0 (MSB–LSB)** This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF1, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF1 initiates the transmission. A read of SBUF1 returns the contents of the receive latch.

SFR Definition 27.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Capture Mode Enable.** 0: Timer 2 Capture Mode is disabled. 1: Timer 2 Capture Mode is enabled.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Read = 0b; Write = Don't Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

SFR Definition 27.11. TMR2L: Timer 2 Low Byte

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

SFR Address = 0xCC; SFR Page = 0x00

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

SFR Definition 27.12. TMR2H Timer 2 High Byte

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

SFR Address = 0xCD; SFR Page = 0x00

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

28.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 28.1. PCA0CN: PCA0 Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
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 = 0xD8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	CF	**PCA0 Counter/Timer Overflow Flag.** Set by hardware when the PCA0 Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA0 Counter/Timer Run Control.** This bit enables/disables the PCA0 Counter/Timer. 0: PCA0 Counter/Timer disabled. 1: PCA0 Counter/Timer enabled.
5	CCF5	**PCA0 Module 5 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF5 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
4	CCF4	**PCA0 Module 4 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF4 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
3	CCF3	**PCA0 Module 3 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF3 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
2	CCF2	**PCA0 Module 2 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	**PCA0 Module 1 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	**PCA0 Module 0 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 28.7. PCA0CPLn: PCA0 Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
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 Addresses: PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB, PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xCE; SFR Page (all registers) = 0x00

Bit	Name	Function
7:0	PCA0CPn[7:0]	**PCA0 Capture Module Low Byte.** The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA0 channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOMn bit to a 0.		

SFR Definition 28.8. PCA0CPHn: PCA0 Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
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 Addresses: PCA0CPH0 = 0xFC, PCA0CPH1 = 0xEA, PCA0CPH2 = 0xEC, PCA0CPH3 = 0xEE, PCA0CPH4 = 0xFE, PCA0CPH5 = 0xCF; SFR Page (all registers) = 0x00

Bit	Name	Function
7:0	PCA0CPn[15:8]	**PCA0 Capture Module High Byte.** The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA0 channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOMn bit to a 1.		

- Added Port 2 Event and Port 3 Events to wake-up sources in “19.2.1. Internal Oscillator Suspend Mode” .
- Updated SFR Definition 20.3 with correct names for bits CP2AE and CP2E.
- Updated “21. Local Interconnect Network (LIN0)” with a voltage range specification for the internal oscillator.
- Updated LIN Register Definitions 21.9 and 21.10 with correct reset values.
- Updated “22. Controller Area Network (CAN0)” with a voltage range specification for the internal oscillator.
- Updated C2 Register Definitions 30.2 and 30.3 with correct C2 and SFR addresses.

Revision 1.2 to Revision 1.3

- Updated the note in “Power-Fail Reset/VDD Monitor” on page 154 to use a larger font.
- Added the note regarding the voltage regulator and VDD monitor in the high setting from “Power-Fail Reset/VDD Monitor” on page 154 to “Voltage Regulator (REG0)” on page 89 and “V_{DD} Maintenance and the V_{DD} monitor” on page 143. Also adjusted the language regarding the solution with the highest system reliability.
- Updated the steps in “V_{DD} Maintenance and the V_{DD} monitor” on page 143 to mention using the VDD monitor in the high setting during flash write/erase operations.
- Updated the SUSPEND bit description in OSCICN (SFR Definition 19.2) to mention that firmware must set the ZTCEN bit in REF0CN (SFR Definition 8.1) before entering suspend.
- Added a note to the IFRDY flag in the OSCICN register (SFR Definition 19.2) that the flag may not accurately reflect the state of the oscillator.
- Added VDD Ramp Time for Power On spec to Table 5.4, “Reset Electrical Characteristics,” on page 48.
- Added a note regarding programming at cold temperatures on –I devices to “Programming The Flash Memory” on page 138 and added Temperature during Programming Operations specification to Table 5.5, “Flash Electrical Characteristics,” on page 48.
- Added a note regarding P0.0/VREF when VDD is used as the reference to Table 20.1, “Port I/O Assignment for Analog Functions,” on page 191 and to the description of the REFSL bit in REF0CN (SFR Definition 8.1).
- Added a note regarding a potential unknown state on GPIO during power up if VIO ramps significantly before VDD to “Port Input/Output” on page 188 and “Reset Sources” on page 152.
- Added steps to set the FLEWT bit in the flash write/erase procedures in “Flash Erase Procedure” on page 139, “Flash Write Procedure” on page 139, and “Flash Write Optimization” on page 140.
- Added the “Reprogramming the VDD Monitor High Threshold” on page 138 section.
- Added a note regarding fast changes on VDD causing the V_{DD} Monitor to trigger to “Power-Fail Reset/VDD Monitor” on page 154.
- Added notes regarding UART TX and RX behavior in “Data Transmission” on page 259 and “Data Reception” on page 259.
- Added a note regarding an issue with /RST low time on some older devices to “Power-On Reset” on page 153.
- Added Table 5.8, “Crystal Oscillator Electrical Characteristics,” on page 50.
- Added a paragraph in “External Crystal Example” on page 185 regarding surface mount crystals and drive current.
- Removed recommendations to introduce a delay after enabling the VDD Monitor before enabling it as a reset source in “Power-Fail Reset/VDD Monitor” on page 154.