E·XFL



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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	13
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.154", 3.90mm Width)
Supplier Device Package	16-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f809-gs

Email: info@E-XFL.COM

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

SFR	Definition	21.2.	RSTSRC: Reset Source	128
SFR	Definition	22.1.	CLKSEL: Clock Select	130
SFR	Definition	22.2.	OSCICL: Internal H-F Oscillator Calibration	131
SFR	Definition	22.3.	OSCICN: Internal H-F Oscillator Control	132
SFR	Definition	22.4.	OSCXCN: External Oscillator Control	134
SFR	Definition	23.1.	XBR0: Port I/O Crossbar Register 0	148
SFR	Definition	23.2.	XBR1: Port I/O Crossbar Register 1	149
SFR	Definition	23.3.	P0MASK: Port 0 Mask Register	151
SFR	Definition	23.4.	P0MAT: Port 0 Match Register	151
SFR	Definition	23.5.	P1MASK: Port 1 Mask Register	152
SFR	Definition	23.6.	P1MAT: Port 1 Match Register	152
SFR	Definition	23.7.	P0: Port 0	153
SFR	Definition	23.8.	P0MDIN: Port 0 Input Mode	154
SFR	Definition	23.9.	P0MDOUT: Port 0 Output Mode	154
SFR	Definition	23.10). P0SKIP: Port 0 Skip	155
SFR	Definition	23.11	I. P1: Port 1	155
SFR	Definition	23.12	2. P1MDIN: Port 1 Input Mode	156
SFR	Definition	23.13	B. P1MDOUT: Port 1 Output Mode	156
SFR	Definition	23.14	I. P1SKIP: Port 1 Skip	157
SFR	Definition	23.15	5. P2: Port 2	157
SFR	Definition	23.16	6. P2MDOUT: Port 2 Output Mode	158
SFR	Definition	24.1.	CRC0CN: CRC0 Control	163
SFR	Definition	24.2.	CRC0IN: CRC Data Input	164
SFR	Definition	24.3.	CRC0DATA: CRC Data Output	164
SFR	Definition	24.4.	CRC0AUTO: CRC Automatic Control	165
SFR	Definition	24.5.	CRC0CNT: CRC Automatic Flash Sector Count	165
SFR	Definition	24.6.	CRC0FLIP: CRC Bit Flip	166
SFR	Definition	25.1.	SPI0CFG: SPI0 Configuration	174
SFR	Definition	25.2.	SPI0CN: SPI0 Control	175
SFR	Definition	25.3.	SPI0CKR: SPI0 Clock Rate	176
SFR	Definition	25.4.	SPI0DAT: SPI0 Data	176
SFR	Definition	26.1.	SMB0CF: SMBus Clock/Configuration	186
SFR	Definition	26.2.	SMB0CN: SMBus Control	188
SFR	Definition	26.3.	SMB0ADR: SMBus Slave Address	191
SFR	Definition	26.4.	SMB0ADM: SMBus Slave Address Mask	191
SFR	Definition	26.5.	SMB0DAT: SMBus Data	192
SFR	Definition	27.1.	SCON0: Serial Port 0 Control	206
SFR	Definition	27.2.	SBUF0: Serial (UART0) Port Data Buffer	207
SFR	Definition	28.1.	CKCON: Clock Control	210
SFR	Definition	28.2.	TCON: Timer Control	215
SFR	Definition	28.3.	TMOD: Timer Mode	216
SFR	Definition	28.4.	TL0: Timer 0 Low Byte	217
SFR	Definition	28.5.	TL1: Timer 1 Low Byte	217
SFR	Definition	28.6.	TH0: Timer 0 High Byte	218
SFR	Definition	28.7.	TH1: Timer 1 High Byte	218





Figure 1.2. C8051F801, C8051F807, C8051F813, C8051F819 Block Diagram



8.1. Output Code Formatting

The ADC measures the input voltage with reference to GND. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit. Conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

8.2. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode. In 8-bit mode, only the 8 MSBs of data are converted, and the ADC0H register holds the results. The AD0LJST bit is ignored for 8-bit mode. 8-bit conversions take two fewer SAR clock cycles than 10-bit conversions, so the conversion is completed faster, and a 500 ksps sampling rate can be achieved with a slower SAR clock.

8.3. Modes of Operation

ADC0 has a maximum conversion speed of 500 ksps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register.

8.3.1. Starting a Conversion

A conversion can be initiated in one of six ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

- 1. Writing a 1 to the AD0BUSY bit of register ADC0CN
- 2. A Timer 0 overflow (i.e., timed continuous conversions)
- 3. A Timer 2 overflow
- 4. A Timer 1 overflow
- 5. A rising edge on the CNVSTR input signal

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "ondemand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. When Timer 2 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2 is in 16-bit mode. See Section "28. Timers" on page 209 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as a Port I/O pin. When the CNVSTR input is used as the ADC0 conversion source, the associated pin should be skipped by the Digital Crossbar. See Section "23. Port Input/Output" on page 138 for details on Port I/O configuration.



8.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

SFR Definition 8.5. ADC0GTH: ADC0 Greater-Than Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0GTH[7:0]							
Туре	R/W							
Reset	1	1	1	1	1	1	1	1
SFR Address = 0xC4								

Bit	Name	Function
7:0	ADC0GTH[7:0]	ADC0 Greater-Than Data Word High-Order Bits.

SFR Definition 8.6. ADC0GTL: ADC0 Greater-Than Data Low Byte

Bit	7	6	5	4	3	2	1	0
Nam	Name ADC0GTL[7:0]							
Туре	pe R/W							
Rese	et 1	1	1	1	1	1	1	1
SFR A	SFR Address = 0xC3							
Bit	Name		Function					
7:0	7:0 ADC0GTL[7:0] ADC0 Greater-Than Data Word Low-Order Bits.							



13.6. Capacitive Sense Multiplexer

The input multiplexer can be controlled through two methods. The CSOMX register can be written to through firmware, or the register can be configured automatically using the modules auto-scan functionality (see "13.3. Automatic Scanning").



Figure 13.3. CS0 Multiplexer Block Diagram



SFR Definition 14.5. B: B Register

Bit	7	6	5	4	3	2	1	0
Nam	е	B[7:0]						
Туре	pe R/W							
Rese	et O	0	0	0	0	0	0	0
SFR A	Address = 0xF	0; Bit-Addres	sable					
Bit	Name		Function					
7:0	B[7:0]	B Register.						

This register convo	a aa a aaaaad	accumulator for	oortoin c	rithmotic operations
	s as a second		centain	



whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 15.1 illustrates the data memory organization of the C8051F80x-83x.

15.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 14.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

15.2.1.2. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51[™] assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

MOV C, 22.3h

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

15.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.



SFR Definition 16.2. DERIVID: Derivative Identification Byte

Bit	7	6	5	4	3	2	1	0
Name		DERIVID[7:0]						
Туре	R	R	R	R	R	R	R	R
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xAD

Bit	Name	Description
7:0	DERIVID[7:0]	Derivative Identification Byte.
		Shows the C8051F80x-83x derivative being used.
		0xD0: C8051F800; 0xD1: C8051F801; 0xD2: C8051F802; 0xD3: C8051F803
		0xD4: C8051F804; 0xD5: C8051F805; 0xD6: C8051F806; 0xD7: C8051F807
		0xD8: C8051F808; 0xD9: C8051F809; 0xDA: C8051F810; 0xDB: C8051F811
		0xDC: C8051F812; 0xDD: C8051F813; 0xDE: C8051F814; 0xDF: C8051F815
		0xE0: C8051F816; 0xE1: C8051F817; 0xE2: C8051F818; 0xE3: C8051F819
		0xE4: C8051F820; 0xE5: C8051F821; 0xE6: C8051F822; 0xE7: C8051F823
		0xE8: C8051F824; 0xE9: C8051F825; 0xEA: C8051F826; 0xEB: C8051F827
		0xEC: C8051F828; 0xED: C8051F829; 0xEE: C8051F830; 0xEF: C8051F831
		0xF0: C8051F832; 0xF1: C8051F833; 0xF2: C8051F834; 0xF3: C8051F835

SFR Definition 16.3. REVID: Hardware Revision Identification Byte

Bit	7	6	5	4	3	2	1	0
Name		REVID[7:0]						
Туре	R	R	R	R	R	R	R	R
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xB6

Bit	Name	Description
7:0	REVID[7:0]	Hardware Revision Identification Byte.
		Shows the C8051F80x-83x hardware revision being used. For example, 0x00 = Revision A.



18. Interrupts

The C8051F80x-83x includes an extended interrupt system supporting a total of 15 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regard-less of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE–EIE1). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.



21. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the \overrightarrow{RST} pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source. Program execution begins at location 0x0000.



Figure 21.1. Reset Sources



22. Oscillators and Clock Selection

C8051F80x-83x devices include a programmable internal high-frequency oscillator and an external oscillator drive circuit. The internal high-frequency oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 22.1. The system clock can be sourced by the external oscillator circuit or the internal oscillator (default). The internal oscillator offers a selectable post-scaling feature, which is initially set to divide the clock by 8.



Figure 22.1. Oscillator Options

22.1. System Clock Selection

The system clock source for the MCU can be selected using the CLKSEL register. The clock selected as the system clock can be divided by 1, 2, 4, 8, 16, 32, 64, or 128. When switching between two clock divide values, the transition may take up to 128 cycles of the undivided clock source. The CLKRDY flag can be polled to determine when the new clock divide value has been applied. The clock divider must be set to "divide by 1" when entering Suspend mode. The system clock source may also be switched on-the-fly. The switchover takes effect after one clock period of the slower oscillator.



SFR Definition 23.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE		PCAON	/IE[1:0]
Туре	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2

Bit	Name	Function
7	WEAKPUD	Port I/O Weak Pullup Disable.
		0: Weak Pullups enabled (except for Ports whose I/O are configured for analog
		mode).
		1: Weak Pullups disabled.
6	XBARE	Crossbar Enable.
		0: Crossbar disabled.
		1: Crossbar enabled.
5	T1E	T1 Enable.
		0: T1 unavailable at Port pin.
		1: T1 routed to Port pin.
4	T0E	T0 Enable.
		0: T0 unavailable at Port pin.
		1: T0 routed to Port pin.
3	ECIE	PCA0 External Counter Input Enable.
		0: ECI unavailable at Port pin.
		1: ECI routed to Port pin.
2	Unused	Read = 0b; Write = Don't Care.
1:0	PCA0ME[1:0]	PCA Module I/O Enable Bits.
		00: All PCA I/O unavailable at Port pins.
		01: CEX0 routed to Port pin.
		10: CEX0, CEX1 routed to Port pins.
		11: CEX0, CEX1, CEX2 routed to Port pins.



26. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I²C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The SMBus peripheral can be fully driven by software (i.e., software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 26.1.



Figure 26.1. SMBus Block Diagram



Table 26.6. SMBus	Status Decoding	With Hardware ACK	Generation E	nabled (EHACK = 1)
-------------------	-----------------	-------------------	---------------------	--------------------

	Valu	es F	Rea	d			Val V	lues Vrit	e to	tus ected
Mode	Mode Status Vector		Vector ACKRQ ARBLOST ACK		Current SMbus State	Typical Response Options	STA	STO	ACK	Next Sta Vector Exp
	1110	0	0	Х	A master START was gener- ated.	Load slave address + R/W into SMB0DAT.	0	0	Х	1100
					A master data or address byte	Set STA to restart transfer.	1	0	Х	1110
er		0	0	0	was transmitted; NACK received.	Abort transfer.	0	1	Х	_
smitt						Load next data byte into SMB0DAT.	0	0	Х	1100
Iran					A master data or address byte was transmitted: ACK	End transfer with STOP.	0	1	Х	_
aster 1	1100	0	0	1		End transfer with STOP and start another transfer.	1 1 X		_	
Ř		Ŭ	Ŭ		received.	Send repeated START.	1	0	Х	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000
					A master data byte was	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
		0	0	1		Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
er						Initiate repeated START.	1	0	0	1110
er Receiv	1000					Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	Х	1100
aste						Read SMB0DAT; send STOP.	0	1	0	—
Ň					A master data byte was	Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
		0	0	0	received; NACK sent (last	Initiate repeated START.	1	0	0	1110
					byte <i>j</i> .	Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	Х	1100



27.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown in Figure 27.3.



Figure 27.3. UART Interconnect Diagram

27.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.



Figure 27.4. 8-Bit UART Timing Diagram



SFR Definition 27.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	SOMODE		MCE0	REN0	TB80	RB80	T10	RI0
Туре	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Address = 0x98; Bit-Addressable

Bit	Name	Function
7	SOMODE	Serial Port 0 Operation Mode. Selects the UART0 Operation Mode.
		0: 8-bit UART with Variable Baud Rate.
		1: 9-bit UART with Variable Baud Rate.
6	Unused	Read = 1b, Write = Don't Care.
5	MCE0	Multiprocessor Communication Enable.
		The function of this bit is dependent on the Serial Port 0 Operation Mode: Mode 0: Checks for valid stop bit.
		0: Logic level of stop bit is ignored.
		1: RIO will only be activated if stop bit is logic level 1.
		Mode 1: Multiprocessor Communications Enable.
		0: Logic level of ninth bit is ignored.
4		Possive Enable
4	RENU	0: LIARTO reception disabled
		1: UART0 reception enabled.
3	TB80	Ninth Transmission Bit.
		The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB80	Ninth Receive Bit.
		RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI0	Transmit Interrupt Flag.
		Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	Receive Interrupt Flag.
		Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.



28.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section "18.2. Interrupt Register Descriptions" on page 104); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section "18.2. Interrupt Register (Section "18.2. Interrupt Register (Section "18.2. Interrupt Register Descriptions" on page 104); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section "18.2. Interrupt Register Descriptions" on page 104). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

28.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 in TCON is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit in the TMOD register selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section "23.3. Priority Crossbar Decoder" on page 143 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit in register CKCON. When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 28.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 18.7). Setting GATE0 to 1 allows the timer to be controlled by the external input signal INT0 (see Section "18.2. Interrupt Register Descriptions" on page 104), facilitating pulse width measurements

TR0	GATE0	INTO	Counter/Timer
0	Х	Х	Disabled
1	0	Х	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't	Care		

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal INT1 is used with Timer 1; the INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 18.7).



SFR Definition 28.2. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0			
Name	e TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0			
Туре	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W R/W			
Rese	t 0	0 0 0 0 0 0 0 0									
SFR A	ddress = 0x8	38: Bit-Addres	sable								
Bit	Name				Function						
7	TF1	Timer 1 Ov	erflow Flag	1							
		Set to 1 by but is autom routine.	Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.								
6	TR1	Timer 1 Ru	n Control.								
		Timer 1 is e	Timer 1 is enabled by setting this bit to 1.								
5	TF0	Timer 0 Ov	erflow Flag								
		Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.									
4	TR0	Timer 0 Ru	n Control.								
		Timer 0 is e	nabled by se	etting this bit	to 1.						
3	IE1	External In	terrupt 1.								
		This flag is s can be clea External Inte	set by hardw red by softwa errupt 1 serv	are when ar are but is au ice routine i	n edge/level tomatically c n edge-trigg	of type defin cleared when ered mode.	ed by IT1 is the CPU ve	detected. It ctors to the			
2	IT1	Interrupt 1	Type Select								
		This bit selects whether the configured /INT1 interrupt will be edge or level sensitive. /INT1 is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 18.7). 0: /INT1 is level triggered. 1: /INT1 is edge triggered.									
1	IE0	External In	terrupt 0.								
		This flag is s can be clea External Inte	set by hardw red by softwa errupt 0 serv	are when ar are but is au ice routine i	n edge/level tomatically c n edge-trigg	of type defin cleared when ered mode.	ed by IT1 is the CPU ve	detected. It ctors to the			
0	IT0	Interrupt 0 Type Select.									

 This bit selects whether the configured INT0 interrupt will be edge or level sensitive.

 INT0 is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 18.7).

 0: INT0 is level triggered.

 1: INT0 is edge triggered.



SFR Definition 28.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Nam	e			TL0	[7:0]			
Туре	Type R/W							
Rese	et 0	0	0	0	0	0	0	0
SFR A	Address = 0x8	A						
Bit	Name		Function					
7:0	TL0[7:0]	Timer 0 Low Byte.						
		The TL0 register is the low byte of the 16-bit Timer 0.						

SFR Definition 28.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Nam	e			TL1	[7:0]			
Туре	•			R/	W			
Rese	et 0	0	0	0	0	0	0	0
SFR A	ddress = 0x8	В						
Bit	Name				Function			
			D (

7:0	TL1[7:0]	Timer 1 Low Byte.
		The TL1 register is the low byte of the 16-bit Timer 1.



DOCUMENT CHANGE LIST

Revision 0.2 to Revision 1.0

- Updated Electrical Specification Tables to reflect production characterization data.
- Added Minimum SYSCLK specification for writing or erasing Flash.
- Added caution for going into suspend with wake source active (Section 20.3)
- Corrected VDM0CN reset values to "Varies".
- Removed mention of IDAC in Pinout table.

