Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	48MHz
Connectivity	SmartCard, SPI, UART/USART, USB
Peripherals	LED, POR, WDT
Number of I/O	46
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c5122d-altum

Pinout

High Pin Count Package Description

AT8xC5122/AT83R5122 version

Figure 1. VQFP64 Package Pinout

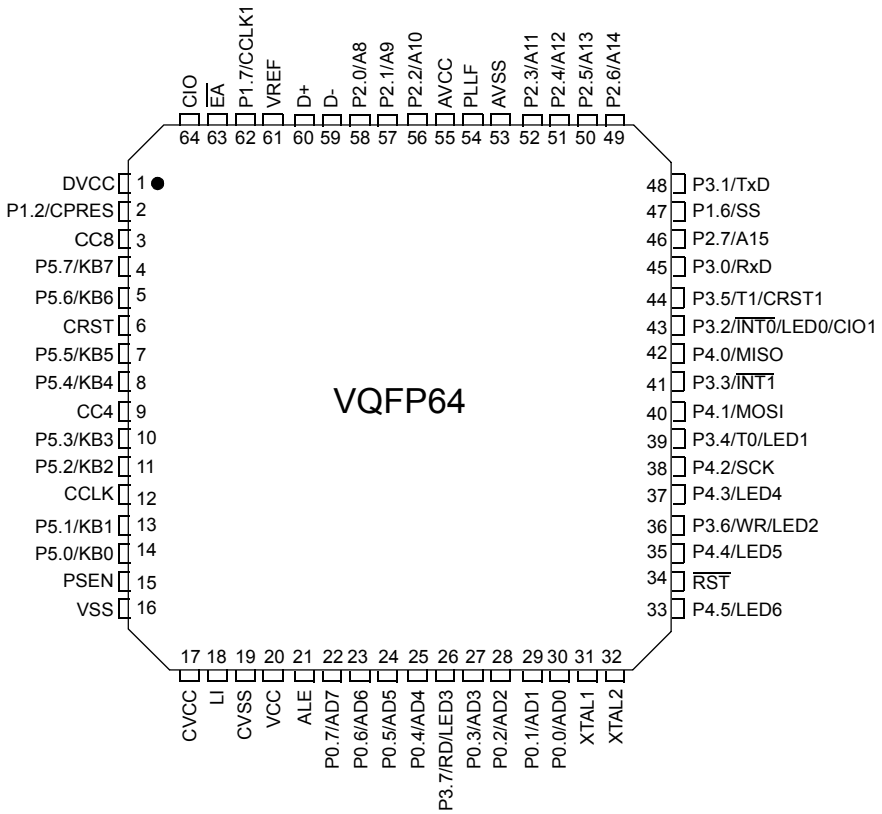
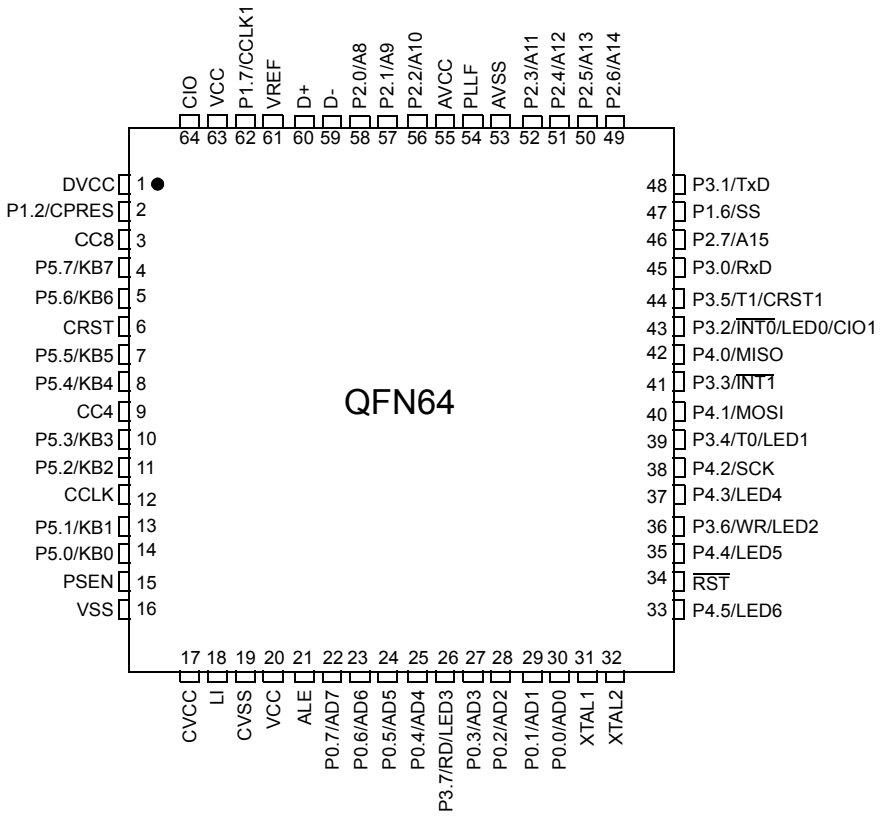


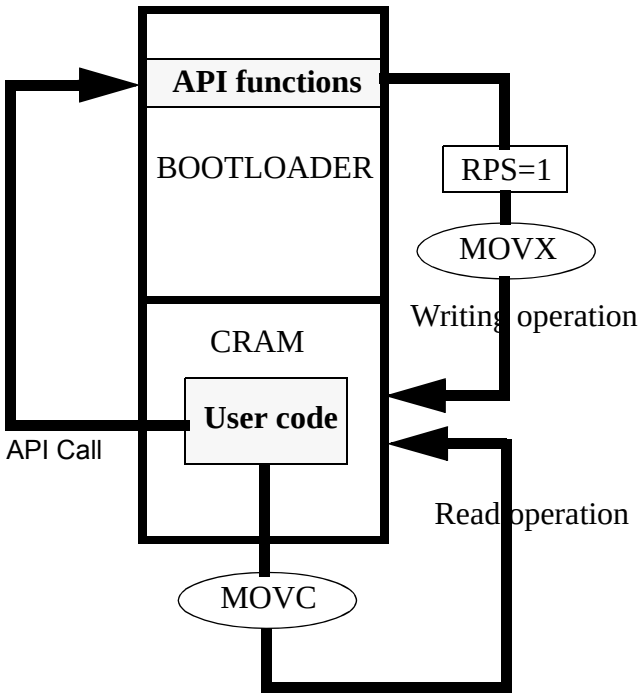
Figure 5. QFN64 Package Pinout



Using CRAM Memory

The CRAM is a read / write volatile memory that is mapped in the program memory space. Then when the power is switched off the code is lost and needs to be reload at each power up. In return, the CRAM enables a lot of flexibility in the code development as it can be programmed indefinitely. The user code running in the CRAM can perform read operations in CRAM itself by means of MOVC instructions like any C51 microcontroller does. Although the writing operations in CRAM are usually handled by the bootloader, it is possible for the user code to handle its own writing operations in CRAM as well. The user code must call API functions provided by the bootloader in the ROM memory. Refer to bootloader datasheet for further details about the use of these API functions. These API functions use a mechanism provided by the AT8xC5122 microcontroller. When the bit RPS is set in RCON register (Table 8 on page 24), the MOVX intructions are configured to write in CRAM instead of XRAM memory. However, due to C51 architecture, it is not possible for the user code to write directly in CRAM when it is itself running in CRAM. This is why the API functions must be called in order to have the code executing in ROM while the CRAM is written.

Figure 11. Read / Write Mechanisms in CRAM Memory



AT8xC5122 Version

		Bit addressable	Not bit addressable					
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h		UEPINT 0000 0000							
F0h		B 0000 0000	LEDCON0 0000 0000						
E8h		P5 1111 1111							
E0h		ACC 0000 0000	LEDCON1 XX00 0000	UBYCTX 0000 0000					
D8h									
D0h		PSW 0000 0000	RCON XXXX 0XXX			UEPCONX 1000 0000	UEPRST 0000 0000		
C8h								UEPSTAX 0000 0000	UEPDATX 0000 0000
SCRS 1	C0h	P4 1111 1111	SCICLK (1) 0X10 1111	UEPIEN 0000 0000	SPCON 0001 0100	SPSTA 0000 0000	SPDAT 1111 1111	USBADDR 1000 0000	UEPNUM 0000 0000
SCRS 0	C0h		SCWT3 (1) 0000 0000						
B8h		IPL0 X000 000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEN 0000 0000	DCCKPS 0000 0000
SCRS 1	B0h	P3 1111 1111	IEN1 XXXX X000	IPL1 00XX 00X0	IPH1 00XX 00X0	SCGT0 (1) 0000 1100	SCGT1(1) XXXX XXX0	SCICR (1) 0000 0000	IPH0 X000 0000
SCRS 0	B0h					SCWT0(1) 1000 0000	SCWT1 (1) 0010 0101	SCWT2 (1) 0000 0000	
SCRS 1	A8h	IEN0 0000 0000	SADDR 0000 0000	SCIBUF XXXX XXXX	SCSR X000 1000	SCETU0 (1) 0111 0100	SCETU1 (1) XXXX X001	SCIER (1) 0X00 0000	
SCRS 0	A8h					SCCON (1) 0000 0000	SCISR (1) 10X0 0000	SCIIR (1) 0X00 0000	
A0h		P2 1111 1111	ISEL 0000 0100	AUXR1 XX1X 0XX0	PLLCON XXXX X000	PLLDIV 0000 0000		WDTRST XXXX XXXX	WDTPRG XXXX X000
98h		SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	KBLS 0000 0000	KBE 0000 0000	KBF 0000 0000	
90h		P1 1111 1111	PMOD0(2) 0000 0000						CKRL XXXX 1111
88h		TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR 0XXX X000	CKCON0 X0X0 X000
80h		P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000	PMOD1 0000 0000	CKSEL XXXX XXX0		PCON 00X1 0000

Notes: 1. Mapping is done using SCRS bit in SCSR register.
2. Grey areas : do not write in.

Table 22. USB SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
UEPIEN	C2h	USB Endpoint Interrupt Enable		EP6INTE	EP5INTE	EP4INTE	EP3INTE	EP2INTE	EP1INTE	EP0INTE
UEPDATX	CFh	USB Endpoint X Fifo Data	FDAT7 - 0							
UBYCTX	E2h	USB Byte Counter Low (EPX)		BYCT6-0						
UFNUML	BAh	USB Frame Number Low	FNUM7 - 0							
UFNUMH	BBh	USB Frame Number High			CRCOK	CRCERR		FNUM10-8		

Table 23. LED SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
LEDCON0	F1h	LED Control 0	LED3		LED2		LED1		LED0	
LEDCON1(1)	E1h	LED Control 1			LED6		LED5		LED4	

Note: 1. Only for AT8xC5122

Table 26. Clock Configuration Register 0 - CKCON0 (S:8Fh)

7	6	5	4	3	2	1	0
-	WDX2	-	SIX2	-	T1X2	T0X2	X2

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	WDX2	**Watchdog clock** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the 1/2 prescaler. Set to select the 1/2 output for this peripheral.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	SIX2	**Enhanced UART clock (Mode 0 and 2)** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the 1/2 prescaler. Set to select the 1/2 output for this peripheral.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	T1X2	**Timer 1 clock** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the 1/2 prescaler. Set to select the 1/2 output for this peripheral.
1	T0X2	**Timer 0 clock** This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect. Cleared to bypass the 1/2 prescaler. Set to select the 1/2 output for this peripheral.
0	X2	**System clock Control bit** Cleared to select the PRT output for CPU and all the peripherals . Set to bypass the PRT prescaler and to enable the individual peripherals ‘X2’ bits.

Reset Value = X0X0 X000b

Table 27. Clock Configuration Register 1 - CKCON1 (S:AFh) only for AT8xC5122

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	SPIX2

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	SPIX2	**SPI clock** This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect. Cleared to bypass the 1/2 prescaler. Set to select the 1/2 output for this peripheral.

Reset Value = XXXX XXX0b

Table 28. PLL Control Register - PLLCON (S:A3h)

7	6	5	4	3	2	1	0
-	-	-	-	-	EXT48	PLLEN	PLOCK

Bit Number	Bit Mnemonic	Description
7 - 3	-	**Reserved** The value read from these bits is always 0. Do not set this bits.
2	EXT48	**External 48 MHz Enable Bit** Set this bit to select XTAL1 as USB clock. Clear this bit to select PLL as USB clock. SCIB clock is controlled by EXT48 bit and XTSCS bit.
1	PLLEN	**PLL Enable bit** Set to enable the PLL. Clear to disable the PLL.
0	PLOCK	**PLL Lock Indicator** Set by hardware when PLL is locked Clear by hardware when PLL is unlocked

Reset Value = 0000 0000b

Table 29. PLL Divider Register - PLLDIV (S:A4h)

7	6	5	4	3	2	1	0
R3	R2	R1	R0	N3	N2	N1	N0

Bit Number	Bit Mnemonic	Description
7 - 4	R3:0	PLL R Divider Bits
3 - 0	N3:0	PLL N Divider Bits

Reset Value = 0000 0000b

Port 4

Port 4 has the following functions:

- Default function: Port 4 is an 6-bit I/O port.
- Alternate functions: see table below

Port 4 has the following configurations:

- Default configuration: Pseudo bi-directional “Port51” digital input/output with internal pull-ups.
- Alternate configurations: See Table 33.

Table 33. Port 4 Description

Port	Alternate Functions		Configurations		
	Signal	Description	Mode 1	Mode 2	Mode 3
P4.0	MISO	SPI Master In Slave Out I/O			
P4.1	MOSI	SPI Master Out Slave In I/O			
P4.2	SCK	SPI clock			
P4.3			Push-pull	KB_OUT	Input MPU
P4.4			Push-pull	KB_OUT	Input MPU
P4.5			Push-pull	KB_OUT	Input MPU

Port 5

Port 5 has the following functions:

- Default function: Port 5 is an 8-bit I/O port.
- Alternate function 1: Port 5 is an 8-bit keyboard port KB0 to KB7.

Port 5 has the following configurations:

- Default configuration: Pseudo bi-directional “Port51” digital input/output with internal pull-ups.
- Alternate configuration: see Table 34.

Table 34. Port 5 Description

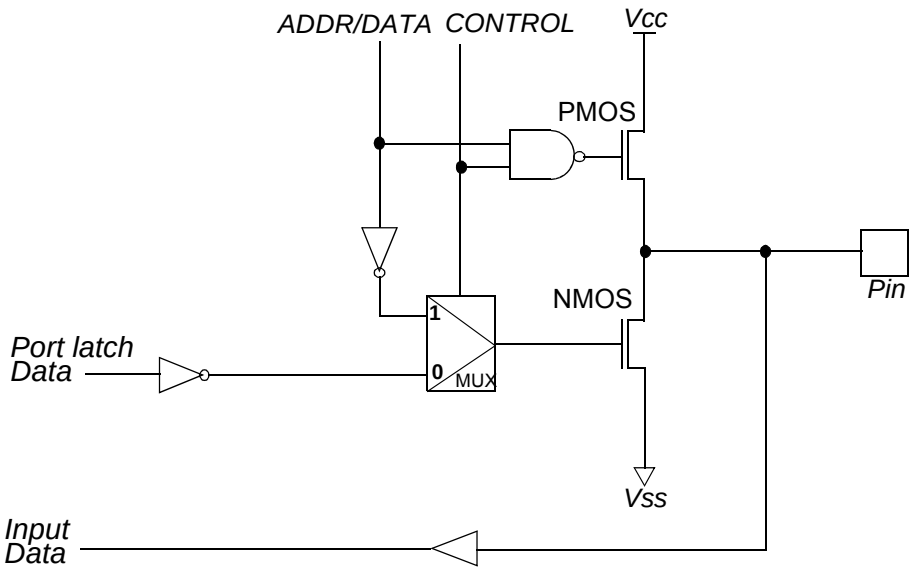
Port	Configurations			
	Mode 1	Mode 2	Mode 3	Comments
P5.0	Push-pull	Input MPU	Input WPU	First cluster
P5.1	Push-pull	Input MPU	Input WPU	
P5.2	Push-pull	Input MPU	Input WPU	
P5.3	Push-pull	Input WPD	Input WPU	Second cluster
P5.4	Push-pull	Input WPD	Input WPU	
P5.5	Push-pull	Input WPD	Input WPU	
P5.6	Push-pull	Input WPD	Input WPU	Third cluster
P5.7	Push-pull	Input WPD	Input WPU	

Port Configuration

Standard I/O P0

The P0 port is described in Figure 22.

Figure 22. Standard Input/Output Port



Quasi Bi-directional Port

The default port output configuration for standard I/O ports is the quasi-bi-directional output that is common on the 80C51 and most of its derivatives. The “Port51” output type can be used as both an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low.

When the port outputs a logic low state, it is driven strongly and is able to sink a fairly large current.

These features are somewhat similar to an open-drain output except that there are three pull-up transistors in the quasi-bi-directional output that serve different purposes.

One of these pull-ups, called the weak pull-up, is turned on whenever the port latch for the pin contains a logic 1. The weak pull-up sources a very small current that will pull the pin high if it is left floating. The weak pull-up can be turned off by the DPU bit in AUXR register.

A second pull-up, called the medium pull-up, is turned on when the port latch for the pin contains a logic 1 and the pin itself is also at a logic 1 level. This pull-up provides the primary source current for a quasi-bi-directional pin that is outputting a 1. If a pin that has a logic 1 on it is pulled low by an external device, the medium pull-up turns off, and only the weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to overpower the medium pull-up and take the voltage on the port pin below its input threshold.

Note: for CIO, CC4, CC8 ports of SCIB interface , in input mode when the ICC (smart card) is driving the port pin :

- if 0 < Vin < CVCC/2 : weak pull-up is active (~100KOhm)
- if CVCC/2 < Vin < CVCC : weak (~100KOhm) and medium (~12KOhm) pull-up’s are active

Registers

Table 36. Port Mode Register 0 - PMOD0 (91h) for AT8xC5122

7	6	5	4	3	2	1	0
P3C1	P3C0	P2C1	P2C0	CPRESRES	-	P0C1	P0C0

Bit Number	Bit Mnemonic	Description
7 - 6	P3C1-P3C0	**Port 3 Configuration bits (Applicable to P3.0, P3.1, P3.3, P3.4 only)** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-up
5-4	P2C1-P2C0	**Port 2 Configuration bits** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-down
3	CPRESRES	**Card Presence Pull-up resistor** Cleared to connect the internal 100K pull-up Set to disconnect the internal pull-up
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1-0	P0C1-P0C0	**Port 0 Configuration bits** **00** C51 Standard P0 **01** Reserved **10** Output Low Speed **11** Push-pull

Reset Value = 0000 0x00b

Table 37. Port Mode Register 0 - PMOD0 (91h) for AT83C5123

7	6	5	4	3	2	1	0
P3C1	P3C0	-	-	CPRESRES	-	-	-

Bit Number	Bit Mnemonic	Description
7 - 6	P3C1-P3C0	**Port 3 Configuration bits (Applicable to P3.0, P3.1, P3.3, P3.4 only)** **00** Quasi bi-directional **01** Push-pull **10** Output Low Speed **11** Input with weak pull-up
5-4		**Reserved** The value read from these bits are indeterminate. Do not set these bit.
3	CPRESRES	**Card Presence Pull-up resistor** Cleared to connect the internal 100K pull-up Set to disconnect the internal pull-up
2-0	-	**Reserved** The value read from these bits are indeterminate. Do not set these bit.

Reset Value = 00xx 0xxxb

Table 40. LED Port Control Register 0 - LEDCON0 (F1h)

7	6	5	4	3	2	1	0
LED3.1	LED3.0	LED2.1	LED2.0	LED1.1	LED1.0	LED0.1	LED0.0

Bit Number	Bit Mnemonic	Description
7 - 6	LED3	**Port LED3 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.7 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.7 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.7 is configured as Quasi-bidirect. mode
5 - 4	LED2	**Port LED2 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.6 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.6 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.6 is configured as Quasi-bidirect. mode
3 - 2	LED1	**Port LED1 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.4 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.4 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.4 is configured as Quasi-bidirect. mode
1 - 0	LED0	**Port LED0 Configuration bits** **00** LED control disabled **01** 2 mA current source when P3.2 is configured as Quasi-bi-directional mode **10** 4 mA current source when P3.2 is configured as Quasi-bi-directional mode **11** 10 mA current source when P3.2 is configured as Quasi-bidirect. mode

Reset Value = 0000 0000b

Table 41. LED Port Control Register 1- LEDCON1 (F1h) only for AT8xC5122

7	6	5	4	3	2	1	0
-	-	LED6.1	LED6.0	LED5.1	LED5.0	LED4.1	LED4.0

Bit Number	Bit Mnemonic	Description
7 - 6		**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5 - 4	LED6	**Port LED6 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.5 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.5 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.5 is configured as Quasi-bidirect. mode
3 - 2	LED5	**Port LED5 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.4 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.4 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.4 is configured as Quasi-bidirect. mode
1 - 0	LED4	**Port LED0 Configuration bits** **00** LED control disabled **01** 2 mA current source when P4.3 is configured as Quasi-bi-directional mode **10** 4 mA current source when P4.3 is configured as Quasi-bi-directional mode **11** 10 mA current source when P4.3 is configured as Quasi-bidirect. mode

Reset Value = 0000 0000b

Table 56. Smart Card Character/Block Waiting Time Register 2
SCWT2 (S:B6h, SCRS=0)

7	6	5	4	3	2	1	0
WT23	WT22	WT21	WT20	WT19	WT18	WT17	WT16

Bit Number	Bit Mnemonic	Description
7 - 0	WT[23:16]	**Waiting Time Byte2** Used together with WT[31:24] and WT[15:0] in registers SCWT3,SCWT1, SCWT0 (see Table 58).

Reset Value = 0000 0000b

Table 57. Smart Card Character/Block Waiting Time Register 1
SCWT1 (S:B5h, SCRS=0)

7	6	5	4	3	2	1	0
WT15	WT14	WT13	WT12	WT11	WT10	WT9	WT8

Bit Number	Bit Mnemonic	Description
7 - 0	WT[15:8]	**Waiting Time Byte 1** Used together with WT[31:16] and WT[7:0] in registers SCWT3,SCWT2, SCWT0 (see Table 55).

Reset Value = 0010 0101b

Table 58. Smart Card Character/Block Waiting Time Register 0
SCWT0 (S:B4h, SCRS=0)

7	6	5	4	3	2	1	0
WT7	WT6	WT5	WT4	WT3	WT2	WT1	WT0

Bit Number	Bit Mnemonic	Description
7 - 0	WT[7:0]	**Waiting Time Byte 0** WT[31:0] is the reload value of the Waiting Time Counter (WTC). The WTC is a general-purpose timer. It is using the ETU clock and is controlled by the WTEN bit (see Table 44 on page 77 and Section “Waiting Time (WT) Counter”, page 67). When UART bit of Registers is set, the WTC is automatically reloaded at each start bit of the UART. It is used to check the maximum time between to consecutive start bits.

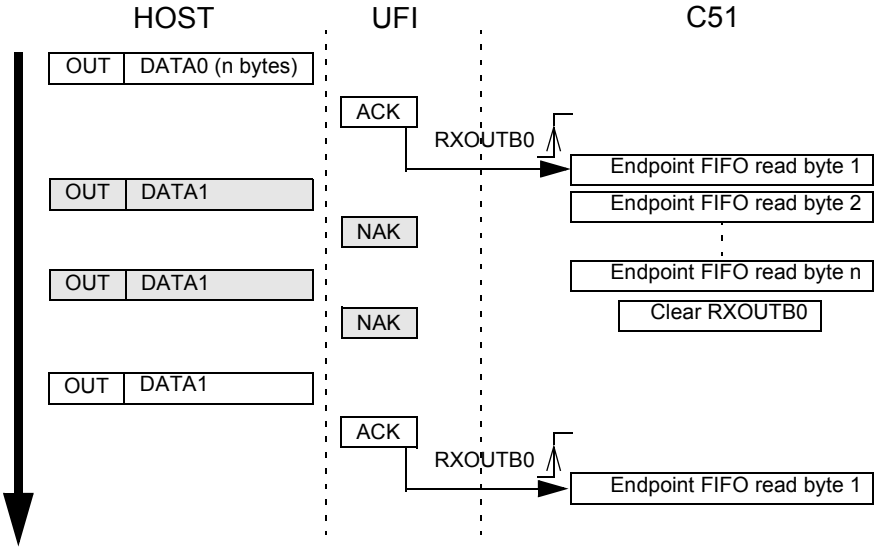
Reset Value = 1000 0000b

Bulk / Interrupt Transactions

Bulk and Interrupt transactions are managed in the same way.

Bulk/Interrupt OUT Transactions in Standard Mode

Figure 54. Bulk/Interrupt OUT transactions in Standard Mode



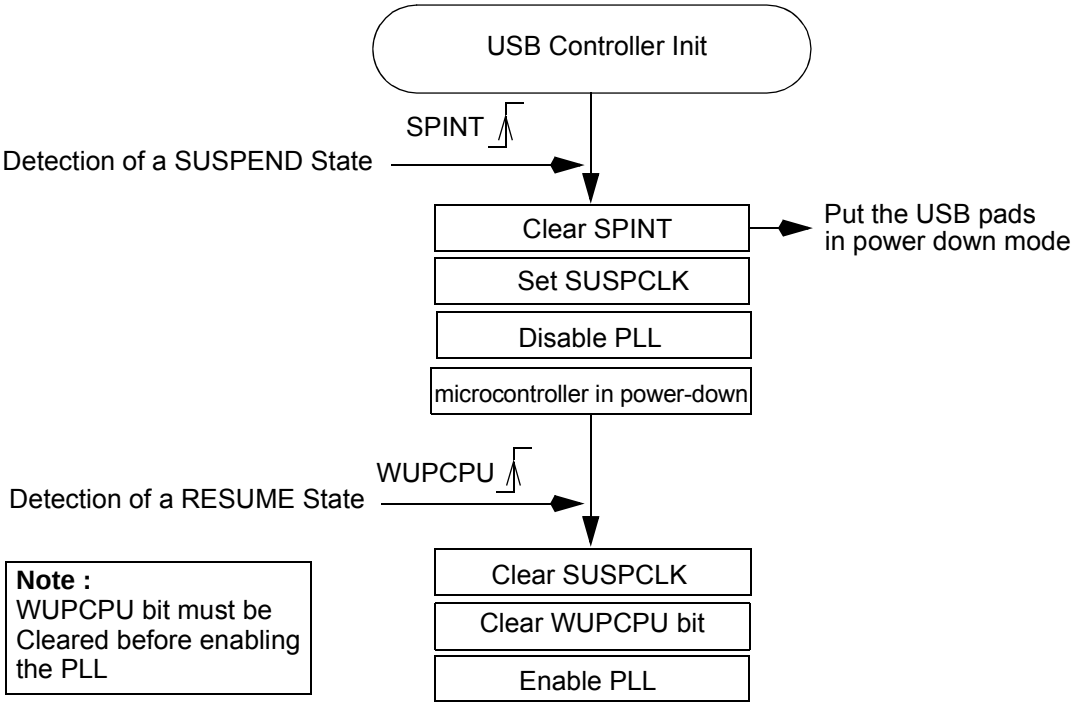
An endpoint should be first enabled and configured before being able to receive Bulk or Interrupt packets.

When a valid OUT packet is received on an endpoint, the RXOUTB0 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware has to select the corresponding endpoint, store the number of data bytes by reading the UBYCTX register. If the received packet is a ZLP (Zero Length Packet), the UBYCTX register value is equal to 0 and no data has to be read.

When all the endpoint FIFO bytes have been read, the firmware should clear the RXOUTB0 bit to allow the USB controller to accept the next OUT packet on this endpoint. Until the RXOUTB0 bit has been cleared by the firmware, the USB controller will answer a NAK handshake for each OUT requests.

If the Host sends more bytes than supported by the endpoint FIFO, the overflow data won't be stored, but the USB controller will consider that the packet is valid if the CRC is correct and the endpoint byte counter contains the number of bytes sent by the Host.

Figure 58. Example of a Suspend/Resume Management



Warning: The core must be switched in external clock mode before disabling the PLL.

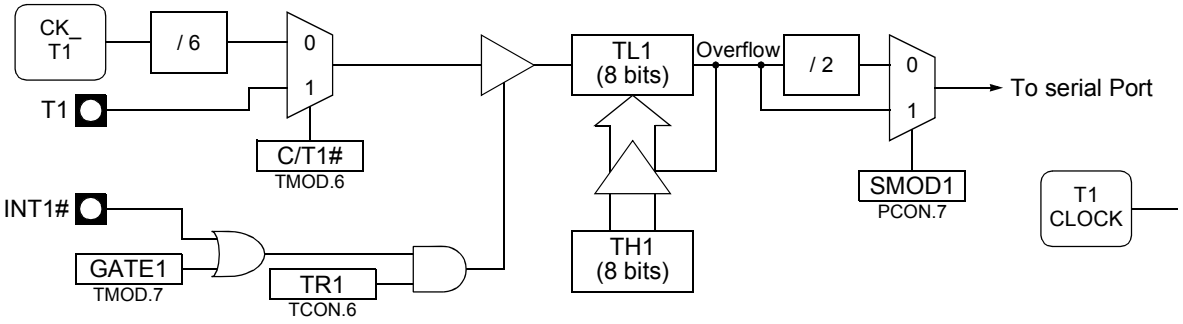
Upstream Resume

A USB device can be allowed by the Host to send an upstream resume for Remote Wake-up purpose.

When the USB controller receives the SET_FEATURE request: DEVICE_REMOTE_WAKEUP, the firmware should set to 1 the RMWUPE bit in the USBCON register to enable this function. RMWUPE value should be 0 in the other cases.

If the device is in SUSPEND mode, the USB controller can send an upstream resume by clearing first the SPINT bit in the USBINT register and by setting then to 1 the SDRMWUP bit in the USBCON register. The USB controller sets to 1 the UPRSM bit in the USBCON register. All clocks must be enabled first. The Remote Wake is sent only if the USB bus was in Suspend state for at least 5 ms. When the upstream resume is completed, the UPRSM bit is reset to 0 by hardware. The firmware should then clear the SDRMWUP bit.

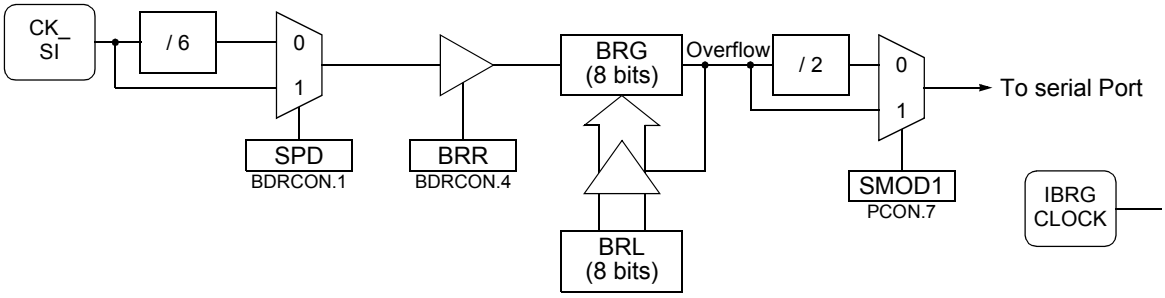
Figure 67. Timer 1 Baud Rate Generator Block Diagram



Internal Baud Rate Generator

When using the Internal Baud Rate Generator, the Baud Rate is derived from the overflow of the timer. As shown in Figure 68 the Internal Baud Rate Generator is an 8-bit auto-reload timer feed by the peripheral clock or by the peripheral clock divided by 6 depending on the SPD bit in BDRCON register (see Table 82 on page 134). The Internal Baud Rate Generator is enabled by setting BRR bit in BDRCON register. SMOD1 bit in PCON register allows doubling of the generated baud rate.

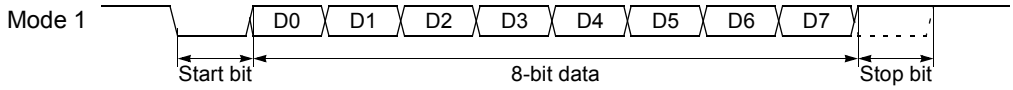
Figure 68. Internal Baud Rate Generator Block Diagram



Synchronous Mode (Mode 0)

Mode 0 is a half-duplex, synchronous mode, which is commonly used to expand the I/0 capabilities of a device with shift registers. The transmit data (TXD) pin outputs a set of eight clock pulses while the receive data (RXD) pin transmits or receives a byte of data. The 8-bit data are transmitted and received least-significant bit (LSB) first. Shifts occur at a fixed Baud Rate (see Section “Baud Rate Selection (Mode 0)”). Figure 69 shows the serial port block diagram in Mode 0.

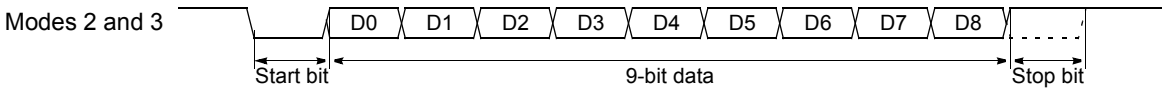
Figure 75. Data Frame Format (Mode 1)



Modes 2 and 3

Modes 2 and 3 are full-duplex, asynchronous modes. The data frame (see Figure 76) consists of 11 bits: one start bit, eight data bits (transmitted and received LSB first), one programmable ninth data bit and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. On receive, the ninth bit is read from RB8 bit in SCON register. On transmit, the ninth data bit is written to TB8 bit in SCON register. Alternatively, you can use the ninth bit as a command/data flag.

Figure 76. Data Frame Format (Modes 2 and 3)



Transmission (Modes 1, 2 and 3)

To initiate a transmission, write to SCON register, setting SM0 and SM1 bits according to Figure 69 on page 128, and setting the ninth bit by writing to TB8 bit. Then, writing the byte to be transmitted to SBUF register starts the transmission.

Reception (Modes 1, 2 and 3)

To prepare for a reception, write to SCON register, setting SM0 and SM1 bits according to Figure 69 on page 128, and setting REN bit. The actual reception is then initiated by a detected high-to-low transition on the RXD pin.

Framing Error Detection (Modes 1, 2 and 3)

Framing error detection is provided for the three asynchronous modes. To enable the framing bit error detection feature, set SMOD0 bit in PCON register as shown in Figure 77.

When this feature is enabled, the receiver checks each incoming data frame for a valid stop bit. An invalid stop bit may result from noise on the serial lines or from simultaneous transmission by two devices. If a valid stop bit is not found, the software sets FE bit in SCON register.

Software may examine FE bit after each reception to check for data errors. Once set, only software or a chip reset clear FE bit. Subsequently received frames with valid stop bits cannot clear FE bit. When the framing error detection feature is enabled, RI rises on stop bit instead of the last data bit as detailed in Figure 75 and Figure 76.

Figure 77. Framing Error Block Diagram

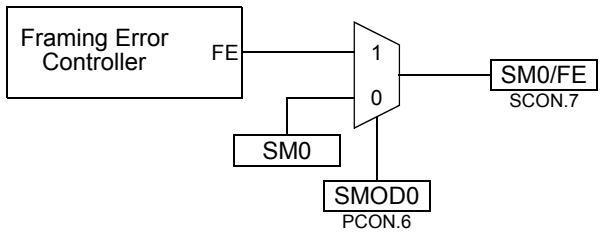
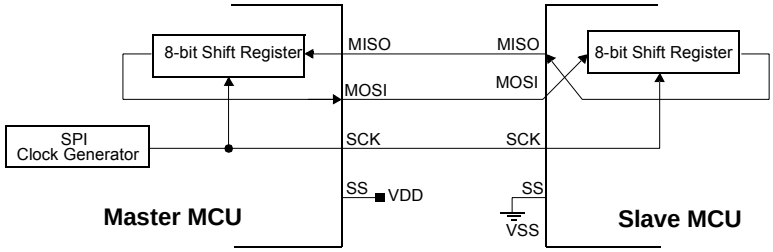


Figure 84. Full-duplex Master-Slave Interconnection



Master Mode

The SPI operates in Master mode when the Master bit, MSTR [3], in the SPCON register is set. Only one Master SPI device can initiate transmissions. Software begins the transmission from a Master SPI module by writing to the Serial Peripheral Data Register (SPDAT). If the shift register is empty, the byte is immediately transferred to the shift register. The byte begins shifting out on MOSI pin under the control of the serial clock, SCK. Simultaneously, another byte shifts in from the Slave on the Master's MISO pin. The transmission ends when the Serial Peripheral transfer data flag, SPIF, in SPSTA becomes set. At the same time that SPIF becomes set, the received byte from the Slave is transferred to the receive data register in SPDAT. Software clears SPIF by reading the Serial Peripheral Status register (SPSTA) with the SPIF bit set, and then reading the SPDAT.

When the pin SS is pulled down during a transmission, the data is interrupted and when the transmission is established again, the data present in the SPDAT is resent.

Slave Mode

The SPI operates in Slave mode when the Master bit, MSTR [4], in the SPCON register is cleared. Before a data transmission occurs, the Slave Select pin, $\overline{SS}$, of the Slave device must be set to '0'. $\overline{SS}$ must remain low until the transmission is complete.

In a Slave SPI module, data enters the shift register under the control of the SCK from the Master SPI module. After a byte enters the shift register, it is immediately transferred to the receive data register in SPDAT, and the SPIF bit is set. To prevent an overflow condition, Slave software must then read the SPDAT before another byte enters the shift register [5]. A Slave SPI must complete the write to the SPDAT (shift register) at least one bus cycle before the Master SPI starts a transmission. If the write to the data register is late, the SPI transmits the data already in the shift register from the previous transmission.

Transmission Formats

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPCON: the Clock Polarity (CPOL [6]) and the Clock Phase (CPHA[4]). CPOL defines the default SCK line level in idle state. It has no significant effect on the transmission format. CPHA defines the edges on which the input data are sampled and the edges on which the output data are shifted (Figure 85 and Figure 86). The clock phase and polarity should be identical for the Master SPI device and the communicating Slave device.

3. The SPI module should be configured as a Master before it is enabled (SPEN set). Also the Master SPI should be configured before the Slave SPI.
4. The SPI module should be configured as a Slave before it is enabled (SPEN set).
5. The maximum frequency of the SCK for an SPI configured as a Slave is the bus clock speed.
6. Before writing to the CPOL and CPHA bits, the SPI should be disabled (SPEN = '0').

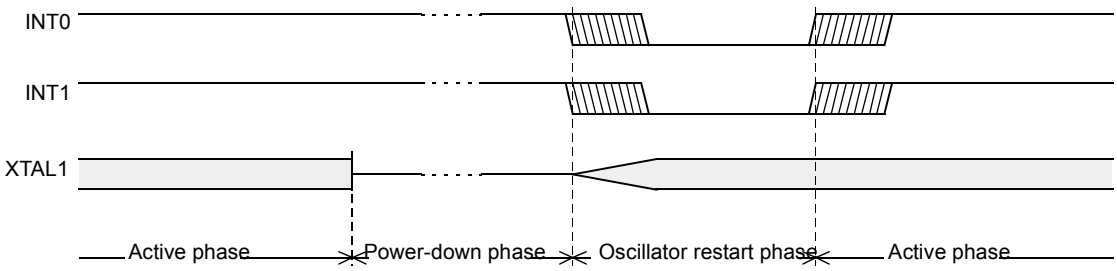
Table 89. Timer/Counter Mode Control Register - TMOD (S:89h)

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select bit** Clear for Timer operation: Timer 1 counts the divided-down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5 4	M11 M01	**Timer 1 Mode Select bits** M11 M01 Operating mode 0 0 Mode 0:8-bit Timer/Counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1:16-bit Timer/Counter. 1 0 Mode 2:8-bit auto-reload Timer/Counter (TL1) reloaded from TH1 at overflow. 1 1 Mode 3:Timer 1 halted. Retains count.
3	GATE0	**Timer 0 Gating Control bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select bit** Clear for Timer operation: Timer 0 counts the divided-down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1 0	M10 M00	**Timer 0 Mode Select bit** M10 M00 Operating mode 0 0 Mode 0:8-bit Timer/Counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1:16-bit Timer/Counter. 1 0 Mode 2:8-bit auto-reload Timer/Counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3:TL0 is an 8-bit Timer/Counter. TH0 is an 8-bit Timer using Timer 1's TR0 and TF0 bits.

Reset Value = 0000 0000b

Figure 108. Power-down Exit Waveform



Exit from power-down by reset redefines all the SFRs, exit from power-down by external interrupt does no affect the SFRs.

Exit from power-down by either reset or external interrupt does not affect the internal RAM content.

Note: If idle mode is activated with power-down mode (IDL and PD bits set), the exit sequence is unchanged, when execution is vectored to interrupt, PD and IDL bits are cleared and idle mode is not entered.

Table shows the state of ports during idle and power-down modes.

Table State of Ports

Mode	Program Memory	ALE	PSEN	P0	P1	P2	P3	P4	P5
Idle	Internal	1	1	Port Data(1)	Port Data	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data	Port Data	Port Data
Power-down	Internal	0	0	Port Dat*	Port Data	Port Data	Port Data	Port Data	Port Data
Power-down	External	0	0	Floating	Port Data	Port Data	Port Data	Port Data	Port Data

Note: 1. Port 0 can force a 0 level. A "one" will leave port floating.

Reduced EMI Mode

The ALE signal is used to demultiplex address and data buses on port 0 when used with external program or data memory. Nevertheless, during internal code execution, ALE signal is still generated. In order to reduce EMI, ALE signal can be disabled by setting AO bit.

The AO bit is located in AUXR register at bit location 0. As soon as AO is set, ALE is no longer output but remains active during MOVX and MOVC instructions and external fetches. During ALE disabling, ALE pin is weakly pulled high.

USB Interface

Suspend

The Suspend state can be detected by the USB controller if all the clocks are enabled and if the USB controller is enabled. The bit SPINT is set by hardware when an idle state is detected for more than 3 ms. This triggers a USB interrupt if enabled.

In order to reduce current consumption, the firmware can put the USB PAD in idle mode, stop the clocks and put the C51 in Idle or Power-down mode. The Resume detection is still active.
The USB PAD is put in idle mode when the firmware clear the SPINT bit. In order to avoid a new suspend detection 3ms later, the firmware has to disable the USB clock input using the SUSPCLK bit in the USBCON Register. The USB PAD automatically exits of idle mode when a wake-up event is detected.

The stop of the 48 MHz clock from the PLL should be done in the following order:

1. Disable of the 48 MHz clock input of the USB controller by setting to 1 the SUSPCLK bit in the USBCON register.
2. If CPU clock is fed from PLL, the on-chip oscillator must be selected to fed the CPU clock.
3. Disable the PLL by clearing the PLLEN bit in the PLLCON register.

Resume

When the USB controller is in Suspend state, the Resume detection is active even if all the clocks are disabled and if the C51 is in Idle or Power-down mode. The WUPCPU bit is set by hardware when a non-idle state occurs on the USB bus. This triggers an interrupt if enabled. This interrupt wakes up the CPU from its Idle or Power-down state and the interrupt function is then executed. The firmware will first enable the 48 MHz generation and then reset to 0 the SUSPCLK bit in the USBCON register if needed.

The firmware has to clear the SPINT bit in the USBINT register before any other USB operation in order to wake up the USB controller from its Suspend mode.

The USB controller is then re-activated.