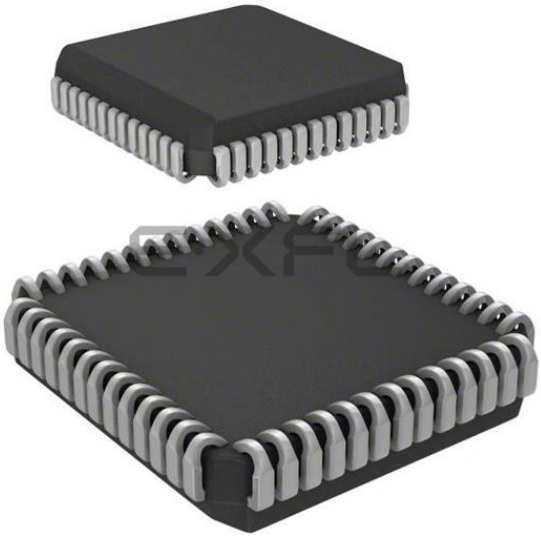


Welcome to E-XFL.COM

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	80C51
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	LED, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.15x19.15)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c5131a-s3sum

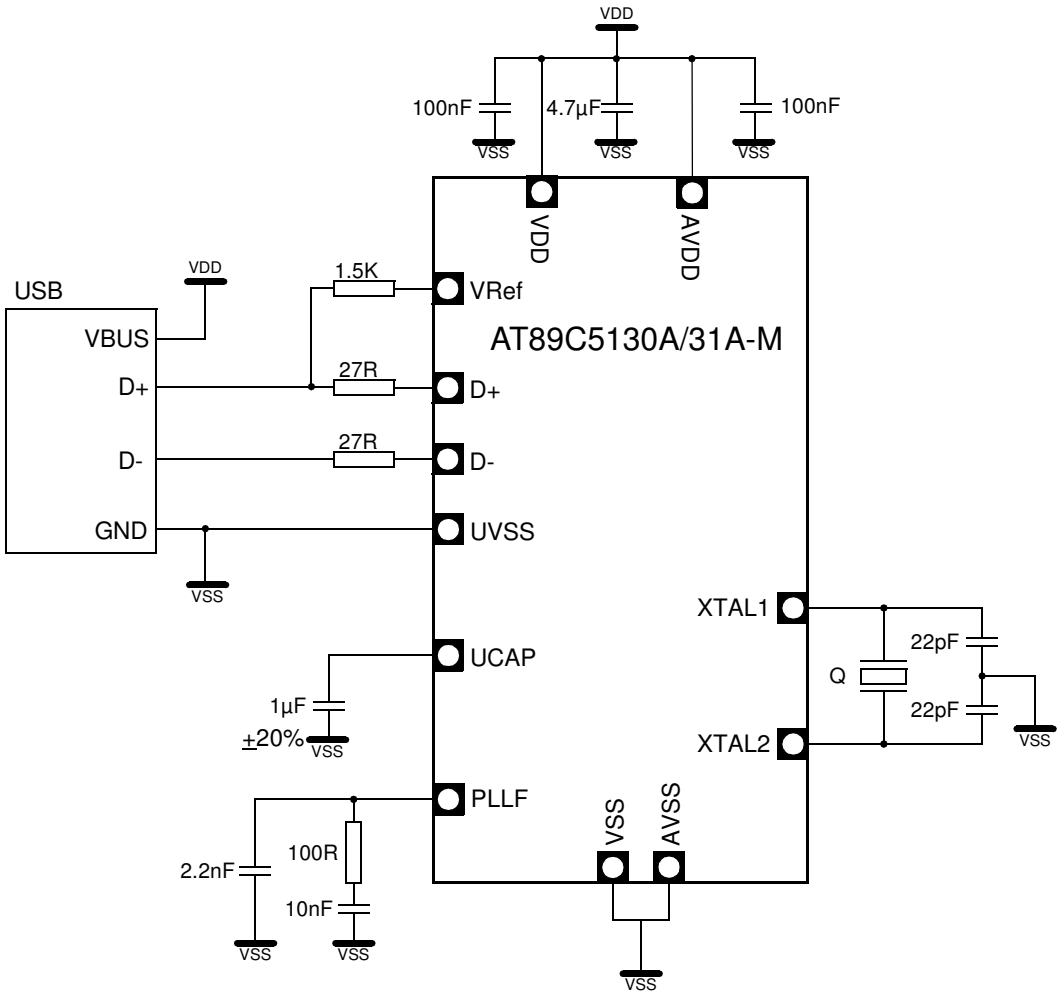
4. Typical Application

4.1 Recommended External components

All the external components described in the figure below must be implemented as close as possible from the microcontroller package.

The following figure represents the typical wiring schematic.

Figure 4-1. Typical Application

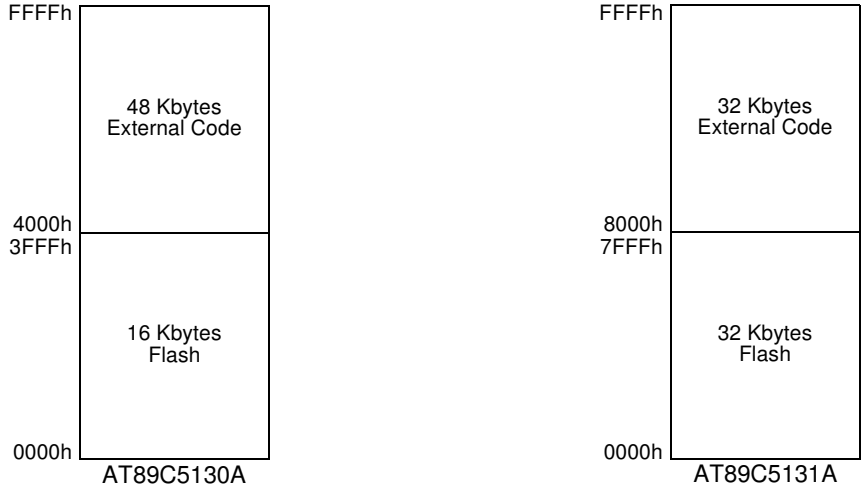


8. Program/Code Memory

The AT89C5130A/31A-M implement 16/ 32 Kbytes of on-chip program/code memory. Figure 8-1 shows the split of internal and external program/code memory spaces depending on the product.

The Flash memory increases EPROM and ROM functionality by in-circuit electrical erasure and programming. Thanks to the internal charge pump, the high voltage needed for programming or erasing Flash cells is generated on-chip using the standard V_{DD} voltage. Thus, the Flash Memory can be programmed using only one voltage and allows In- application Software Programming commonly known as IAP. Hardware programming mode is also available using specific programming tool.

Figure 8-1. Program/Code Memory Organization



Note: If the program executes exclusively from on-chip code memory (not from external memory), beware of executing code from the upper byte of on-chip memory (3FFFh/7FFFh) and thereby disrupting I/O Ports 0 and 2 due to external prefetch. Fetching code constant from this location does not affect Ports 0 and 2.

8.1 External Code Memory Access

8.1.1 Memory Interface

The external memory interface comprises the external bus (Port 0 and Port 2) as well as the bus control signals ($\overline{PSEN}$, and ALE).

Figure 8-2 shows the structure of the external address bus. P0 carries address A7:0 while P2 carries address A15:8. Data D7:0 is multiplexed with A7:0 on P0. Table 8-1 describes the external memory interface signals.

10. EEPROM Data Memory

10.1 Description

The 1-Kbyte on-chip EEPROM memory block is located at addresses 0000h to 03FFh of the ERAM memory space and is selected by setting control bits in the EECON register.

A read in the EEPROM memory is done with a MOVX instruction.

A physical write in the EEPROM memory is done in two steps: write data in the column latches and transfer of all data latches into an EEPROM memory row (programming).

The number of data written on the page may vary from 1 to 128 bytes (the page size). When programming, only the data written in the column latch is programmed and a ninth bit is used to obtain this feature. This provides the capability to program the whole memory by bytes, by page or by a number of bytes in a page. Indeed, each ninth bit is set when the writing the corresponding byte in a row and all these ninth bits are reset after the writing of the complete EEPROM row.

10.2 Write Data in the Column Latches

Data is written by byte to the column latches as for an external RAM memory. Out of the 11 address bits of the data pointer, the 4 MSBs are used for page selection (row) and 7 are used for byte selection. Between two EEPROM programming sessions, all the addresses in the column latches must stay on the same page, meaning that the 4 MSB must not be changed.

The following procedure is used to write to the column latches:

- Set bit EEE of EECON register
- Load DPTR with the address to write
- Store A register with the data to be written
- Execute a MOVX @DPTR, A
- If needed, loop the three last instructions until the end of a 128 bytes page

10.3 Programming

The EEPROM programming consists on the following actions:

- Writing one or more bytes of one page in the column latches. Normally, all bytes must belong to the same page; if not, the first page address will be latched and the others discarded.
- Launching programming by writing the control sequence (52h followed by A2h) to the EECON register.
- EEBUSY flag in EECON is then set by hardware to indicate that programming is in progress and that the EEPROM segment is not available for reading.
- The end of programming is indicated by a hardware clear of the EEBUSY flag.

10.4 Read Data

The following procedure is used to read the data stored in the EEPROM memory:

- Set bit EEE of EECON register
- Stretch the MOVX to accommodate the slow access time of the column latch (Set bit M0 of AUXR register)
- Load DPTR with the address to read
- Execute a MOVX A, @DPTR

Table 13-1. T2CON Register

T2CON - Timer 2 Control Register (C8h)

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 overflow Flag** Must be cleared by software. Set by hardware on Timer 2 overflow, if RCLK = 0 and TCLK = 0.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2 = 1. When set, causes the CPU to vector to Timer 2 interrupt routine when Timer 2 interrupt is enabled. Must be cleared by software. EXF2 doesn't cause an interrupt in Up/down counter mode (DCEN = 1).
5	RCLK	**Receive Clock bit** Cleared to use Timer 1 overflow as receive clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as receive clock for serial port in mode 1 or 3.
4	TCLK	**Transmit Clock bit** Cleared to use Timer 1 overflow as transmit clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as transmit clock for serial port in mode 1 or 3.
3	EXEN2	**Timer 2 External Enable bit** Cleared to ignore events on T2EX pin for Timer 2 operation. Set to cause a capture or reload when a negative transition on T2EX pin is detected, if Timer 2 is not used to clock the serial port.
2	TR2	**Timer 2 Run control bit** Cleared to turn off Timer 2. Set to turn on Timer 2.
1	C/T2#	**Timer/Counter 2 select bit** Cleared for timer operation (input from internal clock system: $F_{CLK\ PERIPH}$). Set for counter operation (input from T2 input pin, falling edge trigger). Must be 0 for clock out mode.
0	CP/RL2#	**Timer 2 Capture/Reload bit** If RCLK = 1 or TCLK = 1, CP/RL2# is ignored and timer is forced to Auto-reload on Timer 2 overflow. Cleared to Auto-reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2 = 1. Set to capture on negative transitions on T2EX pin if EXEN2 = 1.

Reset Value = 0000 0000b

Bit addressable

15. Serial I/O Port

The serial I/O port in the AT89C5130A/31A-M is compatible with the serial I/O port in the 80C52. It provides both synchronous and asynchronous communication modes. It operates as an Universal Asynchronous Receiver and Transmitter (UART) in three full-duplex modes (modes 1, 2 and 3). Asynchronous transmission and reception can occur simultaneously and at different baud rates.

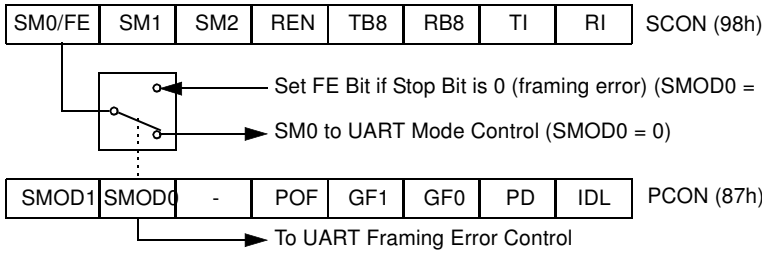
Serial I/O port includes the following enhancements:

- Framing error detection
- Automatic address recognition

15.1 Framing Error Detection

Framing bit error detection is provided for the three asynchronous modes (modes 1, 2 and 3). To enable the framing bit error detection feature, set SMOD0 bit in PCON register (see Figure 15-1).

Figure 15-1. Framing Error Block Diagram



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 CPUs. If a valid stop bit is not found, the Framing Error bit (FE) in SCON register (See Table 15-1) bit is set.

Software may examine FE bit after each reception to check for data errors. Once set, only software or a reset can clear FE bit. Subsequently received frames with valid stop bits cannot clear FE bit. When FE feature is enabled, RI rises on stop bit instead of the last data bit (See Figure 15-2 and Figure 15-3).

Figure 15-2. UART Timings in Mode 1

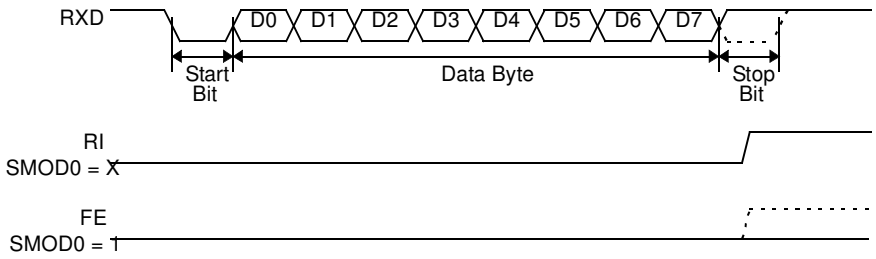


Table 15-3. PCON Register

PCON - Power Control Register (87h)

7	6	5	4	3	2	1	0
SMOD1	**SMOD0**	**-**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1 for UART** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	Serial port Mode bit 0 for UART Cleared to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-Off Flag** Cleared to recognize next reset type. Set by hardware when V_{CC} rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General-purpose Flag** Cleared by user for general-purpose usage. Set by user for general-purpose usage.
2	GF0	**General-purpose Flag** Cleared by user for general-purpose usage. Set by user for general-purpose usage.
1	PD	**Power-down Mode Bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle Mode Bit** Cleared by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b

Not bit addressable

Power-off flag reset value will be 1 only after a power on (cold reset). A warm reset doesn't affect the value of this bit.

Table 15-4. BDRCON Register

BDRCON - Baud Rate Control Register (9Bh)

7	6	5	4	3	2	1	0
-	**-**	**-**	**BRR**	**TBCK**	**RBCK**	**SPD**	**SRC**

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	BRR	**Baud Rate Run Control bit** Cleared to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK	**Transmission Baud rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
2	RBCK	**Reception Baud Rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
1	SPD	**Baud Rate Speed Control bit for UART** Cleared to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC	**Baud Rate Source select bit in Mode 0 for UART** Cleared to select $F_{OSC}/12$ as the Baud Rate Generator ($F_{CLK\ PERIPH}/6$ in X2 mode). Set to select the internal Baud Rate Generator for UARTs in mode 0.

Reset Value = XXX0 0000b

Not bit addressable

Bit Number	Bit Mnemonic	Description
7	EA	**Enable All interrupt bit** Cleared to disable all interrupts. Set to enable all interrupts.
6	EC	**PCA interrupt enable bit** Cleared to disable. Set to enable.
5	ET2	**Timer 2 overflow interrupt Enable bit** Cleared to disable Timer 2 overflow interrupt. Set to enable Timer 2 overflow interrupt.
4	ES	**Serial port Enable bit** Cleared to disable serial port interrupt. Set to enable serial port interrupt.
3	ET1	**Timer 1 overflow interrupt Enable bit** Cleared to disable Timer 1 overflow interrupt. Set to enable Timer 1 overflow interrupt.
2	EX1	**External interrupt 1 Enable bit** Cleared to disable external interrupt 1. Set to enable external interrupt 1.
1	ET0	**Timer 0 overflow interrupt Enable bit** Cleared to disable timer 0 overflow interrupt. Set to enable timer 0 overflow interrupt.
0	EX0	**External interrupt 0 Enable bit** Cleared to disable external interrupt 0. Set to enable external interrupt 0.

Reset Value = 0000 0000b
Bit addressable

Table 16-3. IPL0 Register

IPL0 - Interrupt Priority Register (B8h)

7	6	5	4	3	2	1	0
-	PPCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L

IEN1 - Interrupt Enable Register (B1h)

7	6	5	4	3	2	1	0
-	EUSB	-	-	-	ESPI	ETWI	EKB

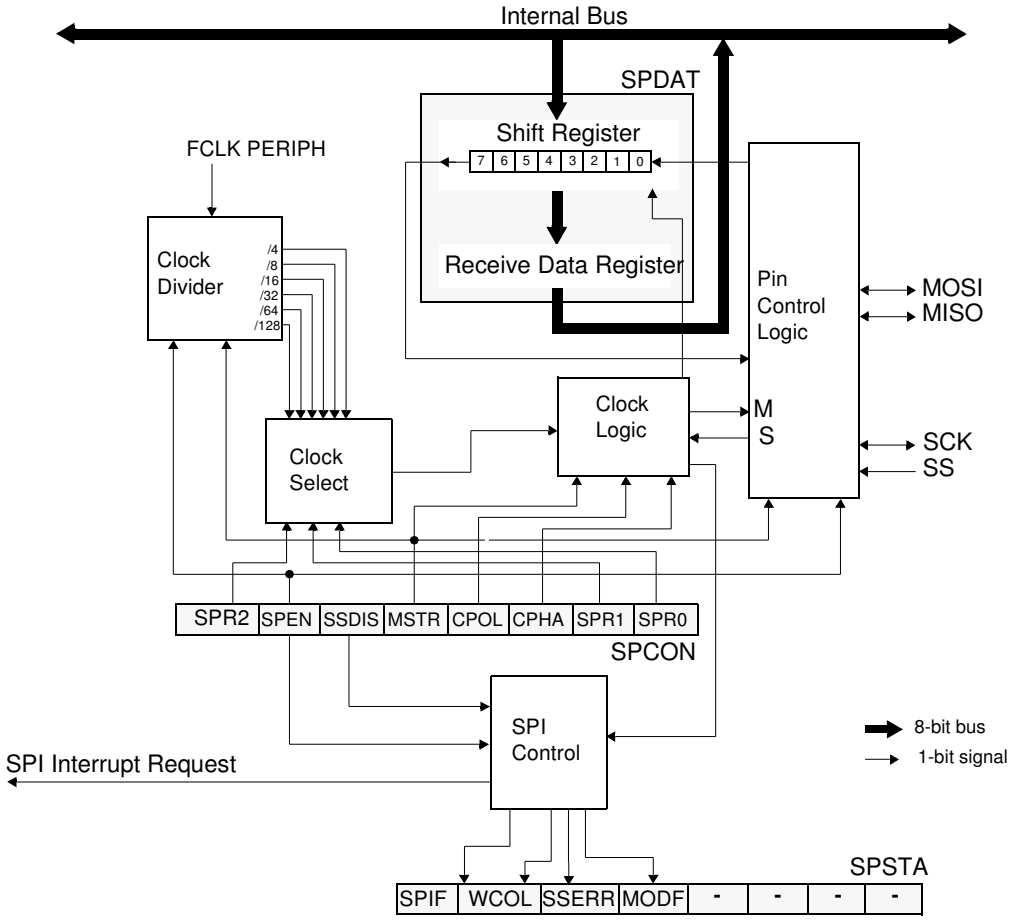
Bit Number	Bit Mnemonic	Description
7	-	Reserved
6	EUSB	USB Interrupt Enable bit Cleared to disable USB interrupt. Set to enable USB interrupt.
5	-	Reserved
4	-	Reserved
3	-	Reserved
2	ESPI	**SPI interrupt Enable bit** Cleared to disable SPI interrupt. Set to enable SPI interrupt.
1	ETWI	**TWI interrupt Enable bit** Cleared to disable TWI interrupt. Set to enable TWI interrupt.
0	EKB	**Keyboard interrupt Enable bit** Cleared to disable keyboard interrupt. Set to enable keyboard interrupt.

SPR2	SPR1	SPR0	Clock Rate	Baud Rate Divisor (BD)
1	1	1	Don't Use	No BRG

19.3 Functional Description

Figure 19-2 shows a detailed structure of the SPI module.

Figure 19-2. SPI Module Block Diagram



19.3.1 Operating Modes

The Serial Peripheral Interface can be configured as one of the two modes: Master mode or Slave mode. The configuration and initialization of the SPI module is made through one register:

- The Serial Peripheral CONtrol register (SPCON)

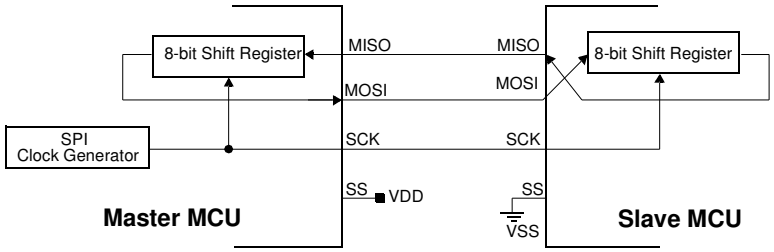
Once the SPI is configured, the data exchange is made using:

- SPCON
- The Serial Peripheral STAtus register (SPSTA)
- The Serial Peripheral DATa register (SPDAT)

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line (SCK) synchronizes shifting and sampling on the two serial data lines (MOSI and MISO). A Slave Select line ($\overline{SS}$) allows individual selection of a Slave SPI device; Slave devices that are not selected do not interfere with SPI bus activities.

When the Master device transmits data to the Slave device via the MOSI line, the Slave device responds by sending data to the Master device via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock (Figure 19-3).

Figure 19-3. Full-duplex Master/Slave Interconnection



19.3.1.1 Master Mode

The SPI operates in Master mode when the Master bit, MSTR [1], 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.

19.3.1.2 Slave Mode

The SPI operates in Slave mode when the Master bit, MSTR [2], 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 [3]. 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.

19.3.2 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 [4]) 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

1. 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.
2. The SPI module should be configured as a Slave before it is enabled (SPEN set).
3. The maximum frequency of the SCK for an SPI configured as a Slave is $F_{CLK\ PERIPH}/2$.
4. Before writing to the CPOL and CPHA bits, the SPI should be disabled (SPEN ='0').

Bit Number	Bit Mnemonic	Description
2	CPHA	**Clock Phase** Cleared to have the data sampled when the SCK leaves the idle state (see CPOL). Set to have the data sampled when the SCK returns to idle state (see CPOL).
1	SPR1	**SPR2 SPR1 SPR0 Serial Peripheral Rate** **000Reserved** **00 1$F_{CLK\ PERIPH}$/4** **010 $F_{CLK\ PERIPH}$/8** **011$F_{CLK\ PERIPH}$/16** **100$F_{CLK\ PERIPH}$/32** **10 1$F_{CLK\ PERIPH}$/64** **110$F_{CLK\ PERIPH}$/128** **1 11Reserved**
0	SPR0	

Reset Value = 0001 0100b

Not bit addressable

19.3.5.2 Serial Peripheral Status Register (SPSTA)

The Serial Peripheral Status Register contains flags to signal the following conditions:

- Data transfer complete
- Write collision
- Inconsistent logic level on $\overline{SS}$ pin (mode fault error)

Table 19-4 describes the SPSTA register and explains the use of every bit in the register.

Table 19-4. SPSTA Register

SPSTA - Serial Peripheral Status and Control register (0C4H)

7	6	5	4	3	2	1	0
SPIF	**WCOL**	SSERR	**MODF**	**-**	**-**	**-**	**-**

Bit Number	Bit Mnemonic	Description
7	SPIF	**Serial Peripheral data transfer flag** Cleared by hardware to indicate data transfer is in progress or has been approved by a clearing sequence. Set by hardware to indicate that the data transfer has been completed.
6	WCOL	**Write Collision flag** Cleared by hardware to indicate that no collision has occurred or has been approved by a clearing sequence. Set by hardware to indicate that a collision has been detected.
5	SSERR	Synchronous Serial Slave Error flag Set by hardware when $\overline{SS}$ is de- asserted before the end of a received data. Cleared by disabling the SPI (clearing SPEN bit in SPCON).

21. USB Controller

.

21.1 Description

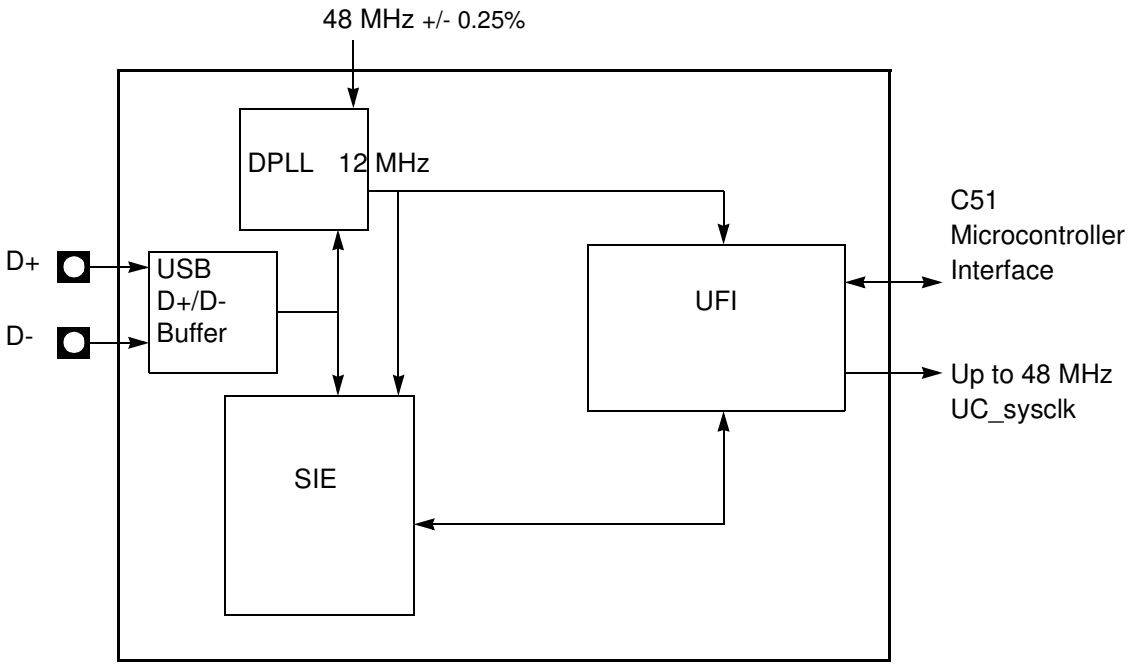
The USB device controller provides the hardware that the AT89C5131 needs to interface a USB link to a data flow stored in a double port memory (DPRAM).

The USB controller requires a 48 MHz ±0.25% reference clock, which is the output of the AT89C5131 PLL (see Section “PLL”, page 15) divided by a clock prescaler. This clock is used to generate a 12 MHz Full-speed bit clock from the received USB differential data and to transmit data according to full speed USB device tolerance. Clock recovery is done by a Digital Phase Locked Loop (DPLL) block, which is compliant with the jitter specification of the USB bus.

The Serial Interface Engine (SIE) block performs NRZI encoding and decoding, bit stuffing, CRC generation and checking, and the serial-parallel data conversion.

The Universal Function Interface (UFI) realizes the interface between the data flow and the Dual Port RAM.

Figure 21-1. USB Device Controller Block Diagram



21.1.1 Serial Interface Engine (SIE)

The SIE performs the following functions:

- NRZI data encoding and decoding.
- Bit stuffing and un-stuffing.
- CRC generation and checking.
- Handshakes.
- TOKEN type identifying.

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

1. Clear suspend interrupt bit in USBINT (required to allow the USB pads to enter power down mode).
2. Enable USB resume interrupt.
3. Disable of the 48 MHz clock input of the USB controller by setting to 1 the SUSPCLK bit in the USBCON register.
4. Disable the PLL by clearing the PLLEN bit in the PLLCON register.
5. Make the CPU core enter power down mode by setting PDOWN bit in PCON.

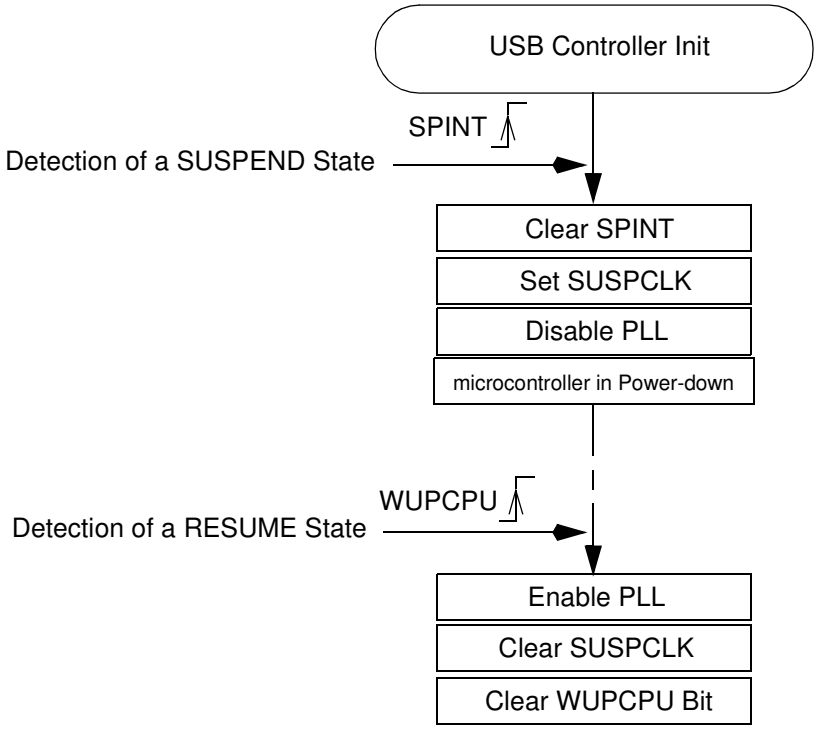
21.8.2 Resume

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.

Figure 21-11. Example of a Suspend/Resume Management



21.8.3 Upstream Resume

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

Figure 21-13. Example of V_{REF} Connection

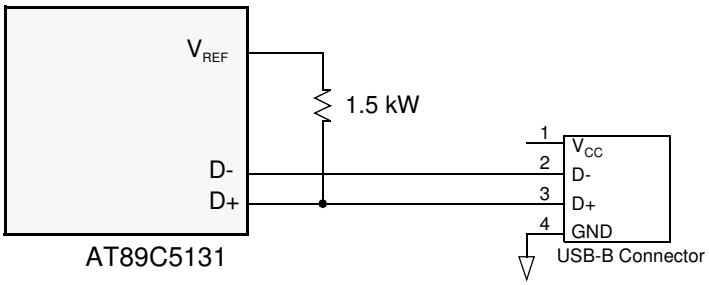
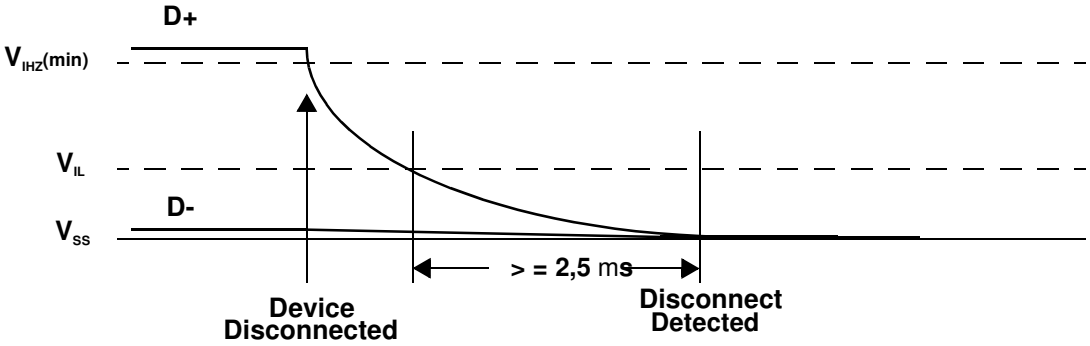


Figure 21-14. Disconnect Timing



21.10 USB Interrupt System

21.10.1 Interrupt System Priorities

Figure 21-15. USB Interrupt Control System

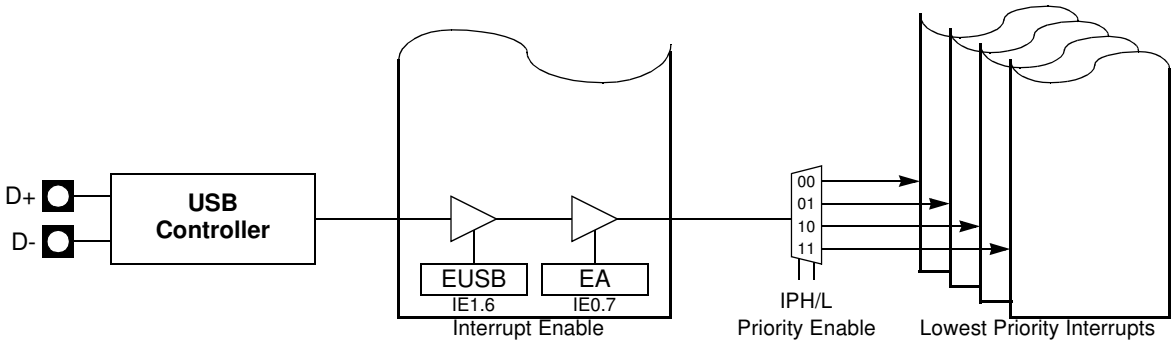


Table 21-2. Priority Levels

IPHUSB	IPLUSB	USB Priority Level
0	0	0 Lowest
0	1	1
1	0	2
1	1	3 Highest

Table 21-15. UEPIEN Register
UEPIEN (S:C2h)
USB Endpoint Interrupt Enable Register

7	6	5	4	3	2	1	0
-	EP6INTE	EP5INTE	EP4INTE	EP3INTE	EP2INTE	EP1INTE	EP0INTE

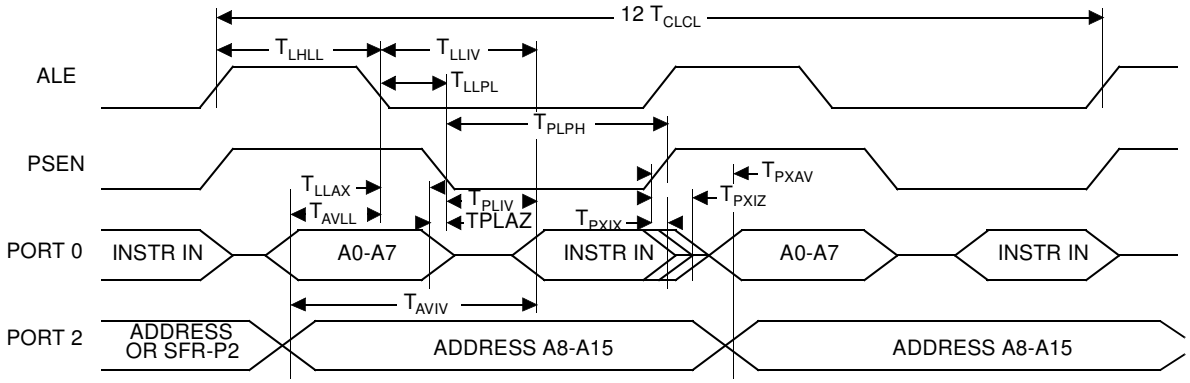
Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is always 0. Do not set this bit.
6	EP6INTE	**Endpoint 6 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
5	EP5INTE	**Endpoint 5 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
4	EP4INTE	**Endpoint 4 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
3	EP3INTE	**Endpoint 3 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
2	EP2INTE	**Endpoint 2 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
1	EP1INTE	**Endpoint 1 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.
0	EP0INTE	**Endpoint 0 Interrupt Enable** Set this bit to enable the interrupts for this endpoint. Clear this bit to disable the interrupts for this endpoint.

Reset Value = 00h

Table 27-4. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X Parameter	Units
T_{LHLL}	Min	2 T - x	T - x	10	ns
T_{AVLL}	Min	T - x	0.5 T - x	15	ns
T_{LLAX}	Min	T - x	0.5 T - x	15	ns
T_{LLIV}	Max	4 T - x	2 T - x	30	ns
T_{LLPL}	Min	T - x	0.5 T - x	10	ns
T_{PLPH}	Min	3 T - x	1.5 T - x	20	ns
T_{PLIV}	Max	3 T - x	1.5 T - x	40	ns
T_{PXIX}	Min	x	x	0	ns
T_{PXIZ}	Max	T - x	0.5 T - x	7	ns
T_{AVIV}	Max	5 T - x	2.5 T - x	40	ns
T_{PLAZ}	Max	x	x	10	ns

27.4.3 External Program Memory Read Cycle



27.4.4 External Data Memory Characteristics

Table 27-5. Symbol Description

Symbol	Parameter
T_{RLRH}	$\overline{RD}$ Pulse Width
T_{WLWH}	$\overline{WR}$ Pulse Width
T_{RLDV}	$\overline{RD}$ to Valid Data In
T_{RHDX}	Data Hold After $\overline{RD}$
T_{RHDZ}	Data Float After $\overline{RD}$
T_{LLDV}	ALE to Valid Data In
T_{AVDV}	Address to Valid Data In
T_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$
T_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$
T_{QVWX}	Data Valid to $\overline{WR}$ Transition
T_{QVWH}	Data set-up to $\overline{WR}$ High
T_{WHQX}	Data Hold After $\overline{WR}$
T_{RLAZ}	$\overline{RD}$ Low to Address Float
T_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE high

Table 27-6. AC Parameters for a Variable Clock (F = 40 MHz)

Symbol	Min	Max	Units
T_{RLRH}	130		ns
T_{WLWH}	130		ns
T_{RLDV}		100	ns
T_{RHDX}	0		ns
T_{RHDZ}		30	ns
T_{LLDV}		160	ns
T_{AVDV}		165	ns
T_{LLWL}	50	100	ns
T_{AVWL}	75		ns
T_{QVWX}	10		ns
T_{QVWH}	160		ns
T_{WHQX}	15		ns
T_{RLAZ}		0	ns
T_{WHLH}	10	40	ns