



Welcome to [E-XFL.COM](http://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](#)"

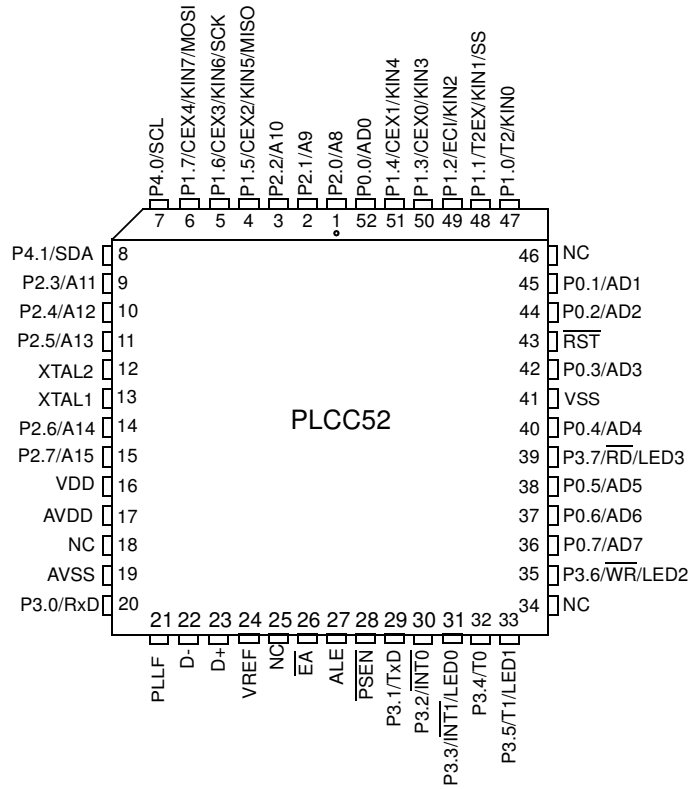
#### Details

Product Status	Active
Core Processor	80C51
Core Size	8-Bit
Speed	48MHz
Connectivity	I <sup>2</sup> C, 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)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at89c5131a-rdtul">https://www.e-xfl.com/product-detail/microchip-technology/at89c5131a-rdtul</a>

## Pinout Description

### Pinout

**Figure 1.** AT89C5131A-L 52-pin PLCC Pinout



**Table 4.** Timer 0, Timer 1 and Timer 2 Signal Description (Continued)

Signal Name	Type	Description	Alternate Function
T0	I	<b>Timer Counter 0 External Clock Input</b> When Timer 0 operates as a counter, a falling edge on the T0 pin increments the count.	P3.4
T1	I	<b>Timer/Counter 1 External Clock Input</b> When Timer 1 operates as a counter, a falling edge on the T1 pin increments the count.	P3.5
T2	I O	<b>Timer/Counter 2 External Clock Input</b> Timer/Counter 2 Clock Output	P1.0
T2EX	I	Timer/Counter 2 Reload/Capture/Direction Control Input	P1.1

**Table 5.** LED Signal Description

Signal Name	Type	Description	Alternate Function
LED[3:0]	O	<b>Direct Drive LED Output</b> These pins can be directly connected to the Cathode of standard LEDs without external current limiting resistors. The typical current of each output can be programmed by software to 2, 6 or 10 mA. Several outputs can be connected together to get higher drive capabilities.	P3.3 P3.5 P3.6 P3.7

**Table 6.** TWI Signal Description

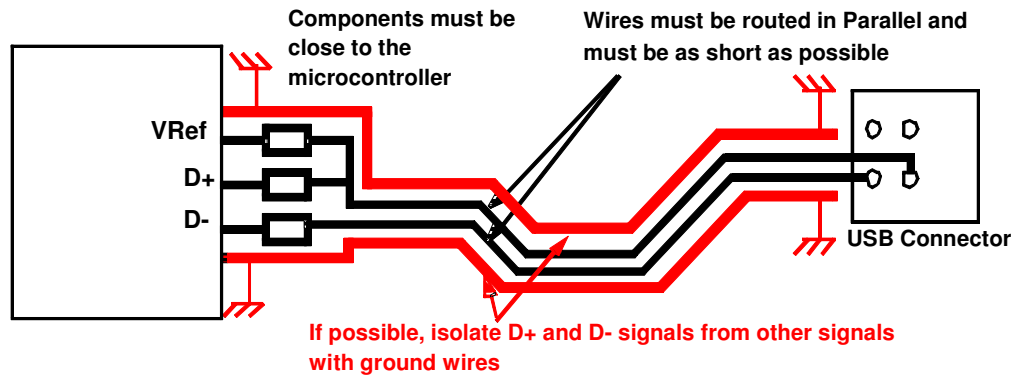
Signal Name	Type	Description	Alternate Function
SCL	I/O	<b>SCL: TWI Serial Clock</b> SCL output the serial clock to slave peripherals. SCL input the serial clock from master.	P4.0
SDA	I/O	<b>SDA: TWI Serial Data</b> SCL is the bidirectional TWI data line.	P4.1

**Table 7.** SPI Signal Description

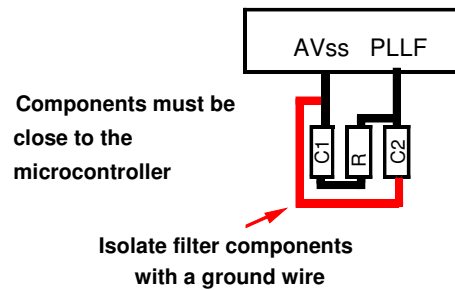
Signal Name	Type	Description	Alternate Function
SS	I/O	<b>SS: SPI Slave Select</b>	P1.1
MISO	I/O	<b>MISO: SPI Master Input Slave Output line</b> When SPI is in master mode, MISO receives data from the slave peripheral. When SPI is in slave mode, MISO outputs data to the master controller.	P1.5
SCK	I/O	<b>SCK: SPI Serial Clock</b> SCK outputs clock to the slave peripheral or receive clock from the master	P1.6
MOSI	I/O	<b>MOSI: SPI Master Output Slave Input line</b> When SPI is in master mode, MOSI outputs data to the slave peripheral. When SPI is in slave mode, MOSI receives data from the master controller	P1.7

## PCB Recommendations

**Figure 5. USB Pads**



**Figure 6. USB PLL**



**Table 27. Keyboard SFRs**

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
KBF	9Eh	Keyboard Flag Register	KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0
KBE	9Dh	Keyboard Input Enable Register	KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0
KBLS	9Ch	Keyboard Level Selector Register	KBLS7	KBLS6	KBLS5	KBLS4	KBLS3	KBLS2	KBLS1	KBLS0

**Table 28. TWI SFRs**

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SSCON	93h	Synchronous Serial Control	CR2	SSIE	STA	STO	SI	AA	CR1	CR0
SSCS	94h	Synchronous Serial Control-Status	SC4	SC3	SC2	SC1	SC0	-	-	-
SSDAT	95h	Synchronous Serial Data	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
SSADR	96h	Synchronous Serial Address	A7	A6	A5	A4	A3	A2	A1	A0

**Table 29. SPI SFRs**

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SPCON	C3h	Serial Peripheral Control	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
SPSTA	C4h	Serial Peripheral Status-Control	SPIF	WCOL	SSERR	MODF	-	-	-	-
SPDAT	C5h	Serial Peripheral Data	R7	R6	R5	R4	R3	R2	R1	R0

**Table 30. USB SFR's**

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
USBCON	BC h	USB Global Control	USBE	SUSPCLK	SDRMWUP	DETACH	UPRSM	RMWUPE	CONFIG	FADDEN
USBADDR	C6h	USB Address	FEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
USBINT	BDh	USB Global Interrupt	-	-	WUPCPU	EORINT	SOFINT	-	-	SPINT
USBIEN	BEh	USB Global Interrupt Enable	-	-	EWUPCPU	EEORINT	ESOFINT	-	-	ESPINT
UEPNUM	C7h	USB Endpoint Number	-	-	-	-	EPNUM3	EPNUM2	EPNUM1	EPNUM0
UEPCONX	D4h	USB Endpoint X Control	EPEN	-	-	-	DTGL	EPDIR	EPTYPE1	EPTYPE0
UEPSTAX	CEh	USB Endpoint X Status	DIR	RXOUTB1	STALLRQ	TXRDY	STLCRC	RXSETUP	RXOUTB0	TXCMP
UEPRST	D5h	USB Endpoint Reset	-	EP6RST	EP5RST	EP4RST	EP3RST	EP2RST	EP1RST	EP0RST
UEPINT	F8h	USB Endpoint Interrupt	-	EP6INT	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

**Table 38.** Program Lock bits

Program Lock Bits				Protection Description
Security level	LB0	LB1	LB2	
1	U	U	U	No program lock features enabled.
2	P	U	U	MOVC instruction executed from external program memory is disabled from fetching code bytes from any internal memory, $\overline{EA}$ is sampled and latched on reset, and further parallel programming of the Flash and of the EEPROM (boot and Xdata) is disabled. ISP and software programming with API are still allowed.
3	X	P	U	Same as 2, also verify through parallel programming interface is disabled and serial programming ISP is still allowed.
4	X	X	P	Same as 3, also external execution is disabled.

Notes: 1. U: unprogrammed or “one” level.  
 2. P: programmed or “zero” level.  
 3. X: don’t care  
 4. WARNING: Security level 2 and 3 should only be programmed after verification.

These security bits protect the code access through the parallel programming interface. They are set by default to level 4. The code access through the ISP is still possible and is controlled by the “software security bits” which are stored in the extra Flash memory accessed by the ISP firmware.

To load a new application with the parallel programmer, a chip erase must be done first. This will set the HSB in its inactive state and will erase the Flash memory. The part reference can always be read using Flash parallel programming modes.

#### Default Values

The default value of the HSB provides parts ready to be programmed with ISP:

- BLJB: Cleared to force ISP operation.
- X2: Set to force X1 mode (Standard Mode)
- OSCON1-0: Set to start with 32 MHz oscillator configuration value.
- LB2-0: Security level four to protect the code from a parallel access with maximum security.

#### Software Registers

Several registers are used, in factory and by parallel programmers, to make copies of hardware registers contents. These values are used by Atmel ISP (see Section “In-System Programming (ISP)”).

These registers are in the “Extra Flash Memory” part of the Flash memory. This block is also called “XAF” or eXtra Array Flash. They are accessed in the following ways:

- Commands issued by the parallel memory programmer.
- Commands issued by the ISP software.
- Calls of API issued by the application software.

Several software registers are described in Table 39.

## Registers

**Table 42.** EECON (S:0D2h)  
EECON Register

7	6	5	4	3	2	1	0
EEPL3	EEPL2	EEPL1	EEPL0	-	-	EEE	EEBUSY
Bit Number	Bit Mnemonic	Description					
7-4	EEPL3-0	<b>Programming Launch command bits</b> Write 5Xh followed by AXh to EEPL to launch the programming.					
3	-	<b>Reserved</b> The value read from this bit is indeterminate. Do not set this bit.					
2	-	<b>Reserved</b> The value read from this bit is indeterminate. Do not set this bit.					
1	EEE	<b>Enable EEPROM Space bit</b> Set to map the EEPROM space during MOVX instructions (Write in the column latches) Clear to map the ERAM space during MOVX.					
0	EEBUSY	<b>Programming Busy flag</b> Set by hardware when programming is in progress. Cleared by hardware when programming is done. Cannot be set or cleared by software.					

Reset Value = XXXX XX00b

Not bit addressable

When an instruction accesses an internal location above address 7Fh, the CPU knows whether the access is to the upper 128 bytes of data RAM or to SFR space by the addressing mode used in the instruction.

- Instructions that use direct addressing access SFR space. For example: MOV 0A0H, # data, accesses the SFR at location 0A0h (which is P2).
- Instructions that use indirect addressing access the Upper 128 bytes of data RAM. For example: MOV @R0, # data where R0 contains 0A0h, accesses the data byte at address 0A0h, rather than P2 (whose address is 0A0h).
- The ERAM bytes can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory which is physically located on-chip, logically occupies the first bytes of external data memory. The bits XRS0 and XRS1 are used to hide a part of the available ERAM as explained in Table 44. This can be useful if external peripherals are mapped at addresses already used by the internal ERAM.
- With EXTRAM = 0, the ERAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to ERAM will not affect ports P0, P2, P3.6 (WR) and P3.7 (RD). For example, with EXTRAM = 0, MOVX @R0, # data where R0 contains 0A0H, accesses the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than the accessible size of the ERAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, with P0 and P2 as data/address busses, and P3.6 and P3.7 as write and read timing signals. Accesses to ERAM above 0FFH can only be done by the use of DPTR.
- With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 80C51. MOVX @Ri will provide an eight-bit address multiplexed with data on Port0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX @Ri and MOVX @DPTR will generate either read or write signals on P3.6 (WR) and P3.7 (RD).

The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the ERAM.

The M0 bit allows to stretch the ERAM timings; if M0 is set, the read and write pulses are extended from 6 to 30 clock periods. This is useful to access external slow peripherals.



The CMOD register includes three additional bits associated with the PCA (See Figure 28 and Table 48).

- The CIDL bit allows the PCA to stop during idle mode.
- The WDTE bit enables or disables the watchdog function on module 4.
- The ECF bit when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows.

The CCON register contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (see Table 49).

- Bit CR (CCON.6) must be set by software to run the PCA. The PCA is shut off by clearing this bit.
- Bit CF: The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software.
- Bits 0 through 4 are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags can only be cleared by software.

**Table 49.** CCON Register

CCON - PCA Counter Control Register (D8h)

7	6	5	4	3	2	1	0
CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0
Bit Number	Bit Mnemonic	Description					
7	CF	<b>PCA Counter Overflow flag</b> Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.					
6	CR	<b>PCA Counter Run control bit</b> Must be cleared by software to turn the PCA counter off. Set by software to turn the PCA counter on.					
5	–	<b>Reserved</b> The value read from this bit is indeterminate. Do not set this bit.					
4	CCF4	<b>PCA Module 4 interrupt flag</b> Must be cleared by software. Set by hardware when a match or capture occurs.					
3	CCF3	<b>PCA Module 3 interrupt flag</b> Must be cleared by software. Set by hardware when a match or capture occurs.					
2	CCF2	<b>PCA Module 2 interrupt flag</b> Must be cleared by software. Set by hardware when a match or capture occurs.					
1	CCF1	<b>PCA Module 1 Interrupt Flag</b> Must be cleared by software. Set by hardware when a match or capture occurs.					
0	CCF0	<b>PCA Module 0 Interrupt Flag</b> Must be cleared by software. Set by hardware when a match or capture occurs.					

Reset Value = 000X 0000b

Not bit addressable

The SADEN byte is selected so that each slave may be addressed separately.

For slave A, bit 0 (the LSB) is a don't care bit; for slaves B and C, bit 0 is a 1. To communicate with slave A only, the master must send an address where bit 0 is clear (e.g. 1111 0000b).

For slave A, bit 1 is a 1; for slaves B and C, bit 1 is a don't care bit. To communicate with slaves B and C, but not slave A, the master must send an address with bits 0 and 1 both set (e.g. 1111 0011b).

To communicate with slaves A, B and C, the master must send an address with bit 0 set, bit 1 clear, and bit 2 clear (e.g. 1111 0001b).

## Broadcast Address

A broadcast address is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't care bits, e.g.:

SADDR0101 0110b

SADEN1111 1100b

Broadcast = SADDR OR SADEN1111 111Xb

The use of don't care bits provides flexibility in defining the broadcast address, in most applications, a broadcast address is FFh. The following is an example of using broadcast addresses:

Slave A: SADDR1111 0001b

SADEN1111 1010b

Broadcast1111 1X11b,

Slave B: SADDR1111 0011b

SADEN1111 1001b

Broadcast1111 1X11B,

Slave C: SADDR = 1111 0011b

SADEN1111 1101b

Broadcast1111 1111b

For slaves A and B, bit 2 is a don't care bit; for slave C, bit 2 is set. To communicate with all of the slaves, the master must send an address FFh. To communicate with slaves A and B, but not slave C, the master can send an address FBh.

## Reset Addresses

On reset, the SADDR and SADEN registers are initialized to 00h, i.e. the given and broadcast addresses are XXXX XXXXb (all don't care bits). This ensures that the serial port will reply to any address, and so, that it is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition.

SADEN - Slave Address Mask Register (B9h)

7	6	5	4	3	2	1	0

Reset Value = 0000 0000b

Not bit addressable

SADDR - Slave Address Register (A9h)

7	6	5	4	3	2	1	0

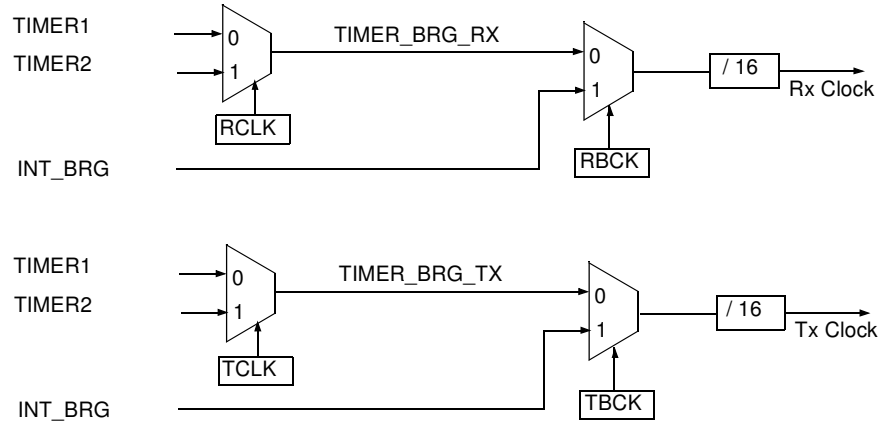
Reset Value = 0000 0000b

Not bit addressable

## Baud Rate Selection for UART for Mode 1 and 3

The Baud Rate Generator for transmit and receive clocks can be selected separately via the T2CON and BDRCON registers.

**Figure 37.** Baud Rate Selection



## Baud Rate Selection Table for UART

TCLK (T2CON)	RCLK (T2CON)	TBCK (BDRCON)	RBCK (BDRCON)	Clock Source UART Tx	Clock Source UART Rx
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG	Timer 1
X	1	1	0	INT_BRG	Timer 2
0	X	0	1	Timer 1	INT_BRG
1	X	0	1	Timer 2	INT_BRG
X	X	1	1	INT_BRG	INT_BRG

## Internal Baud Rate Generator (BRG)

When the internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow depending on the BRL reload value, the value of SPD bit (Speed Mode) in BDRCON register and the value of the SMOD1 bit in PCON register.

BRL - Baud Rate Reload Register for the internal baud rate generator, UART (9Ah)

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

Reset Value = 0000 0000b

**Table 57.** 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	<b>Timer 2 overflow Flag</b> Must be cleared by software. Set by hardware on Timer 2 overflow, if RCLK = 0 and TCLK = 0.
6	EXF2	<b>Timer 2 External Flag</b> 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	<b>Receive Clock bit for UART</b> 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	<b>Transmit Clock bit for UART</b> 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	<b>Timer 2 External Enable bit</b> 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	<b>Timer 2 Run control bit</b> Cleared to turn off Timer 2. Set to turn on Timer 2.
1	C/T2#	<b>Timer/Counter 2 select bit</b> 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#	<b>Timer 2 Capture/Reload bit</b> 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

**Table 64.** IEN1 Register

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	<b>SPI interrupt Enable bit</b> Cleared to disable SPI interrupt. Set to enable SPI interrupt.
1	ETWI	<b>TWI interrupt Enable bit</b> Cleared to disable TWI interrupt. Set to enable TWI interrupt.
0	EKB	<b>Keyboard interrupt Enable bit</b> Cleared to disable keyboard interrupt. Set to enable keyboard interrupt.

Reset Value = X0XX X000b

Not bit addressable

## USB Controller

### Description

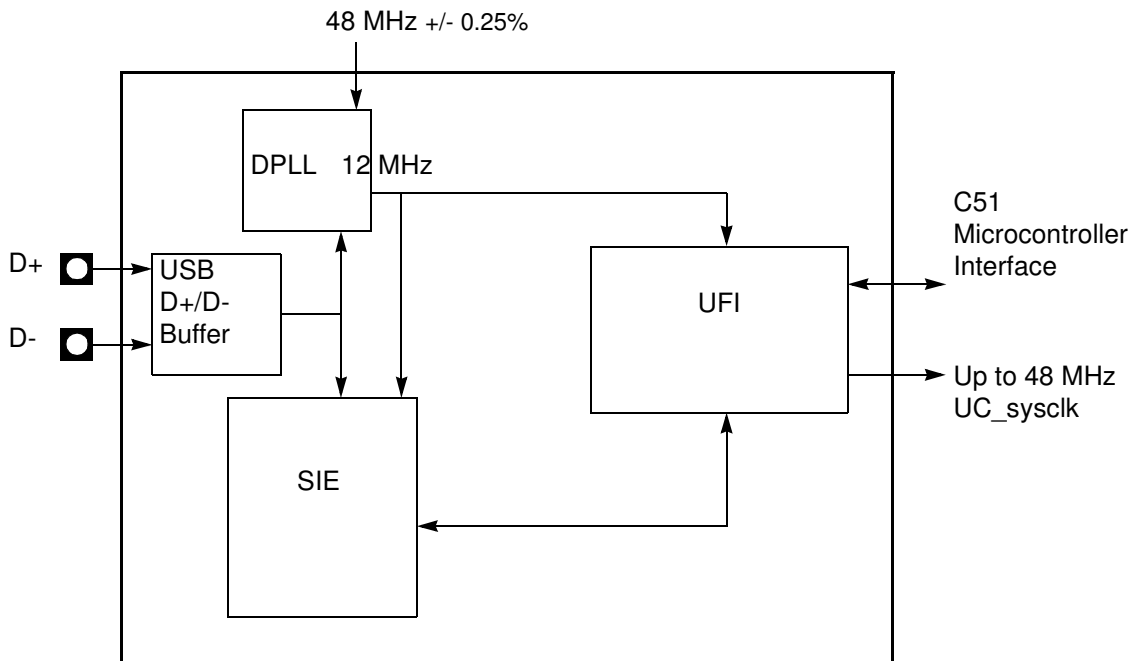
The USB device controller provides the hardware that the AT89C5131 needs to interface a USB link to a data flow stored in a double port memory (DPRAM).

The USB controller requires a 48 MHz  $\pm 0.25\%$  reference clock, which is the output of the AT89C5131 PLL (see Section “PLL”, page 14) 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.

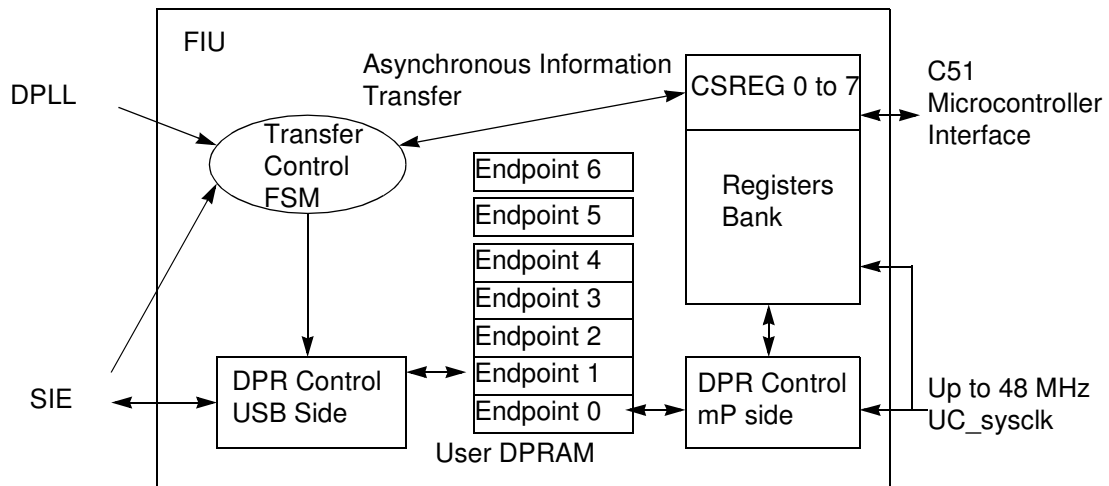
The Universal Function Interface (UFI) realizes the interface between the data flow and the Dual Port RAM.

**Figure 56.** USB Device Controller Block Diagram



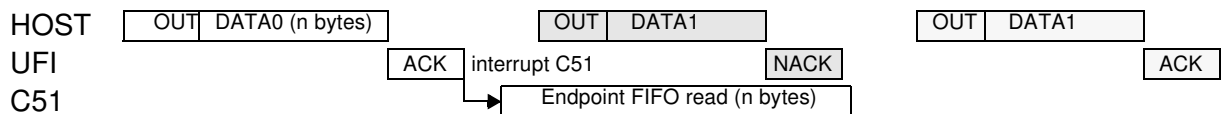
**Function Interface Unit (FIU)** The Function Interface Unit provides the interface between the AT89C5131 and the SIE. It manages transactions at the packet level with minimal intervention from the device firmware, which reads and writes the endpoint FIFOs.

**Figure 58.** UFI Block Diagram



**Figure 59.** Minimum Intervention from the USB Device Firmware

**OUT Transactions:**



**IN Transactions:**



- Endpoint FIFO reset

Before using an endpoint, its FIFO will be reset. This action resets the FIFO pointer to its original value, resets the byte counter of the endpoint (UBYCTLX and UBYCTHX registers), and resets the data toggle bit (DTGL bit in UEPCONX).

The reset of an endpoint FIFO is performed by setting to 1 and resetting to 0 the corresponding bit in the UEPRST register.

For example, in order to reset the Endpoint number 2 FIFO, write 0000 0100b then 0000 0000b in the UEPRST register.

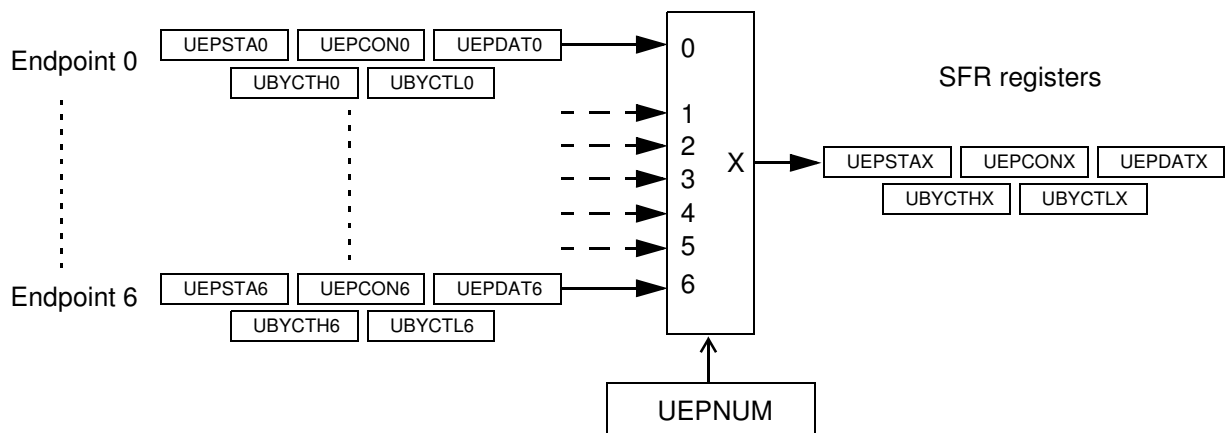
Note that the endpoint reset doesn't reset the bank number for ping-pong endpoints.

## Read/Write Data FIFO

### FIFO Mapping

Depending on the selected endpoint through the UEPNUM register, the UEPDATX register allows to access the corresponding endpoint data fifo.

**Figure 61.** Endpoint FIFO Configuration



### Read Data FIFO

The read access for each OUT endpoint is performed using the UEPDATX register.

After a new valid packet has been received on an Endpoint, the data are stored into the FIFO and the byte counter of the endpoint is updated (UBYCTLX and UBYCTHX registers). The firmware has to store the endpoint byte counter before any access to the endpoint FIFO. The byte counter is not updated when reading the FIFO.

To read data from an endpoint, select the correct endpoint number in UEPNUM and read the UEPDATX register. This action automatically decreases the corresponding address vector, and the next data is then available in the UEPDATX register.

### Write Data FIFO

The write access for each IN endpoint is performed using the UEPDATX register.

To write a byte into an IN endpoint FIFO, select the correct endpoint number in UEPNUM and write into the UEPDATX register. The corresponding address vector is automatically increased, and another write can be carried out.

Warning 1: The byte counter is not updated.

Warning 2: Do not write more bytes than supported by the corresponding endpoint.



## **Isochronous Transactions**

### **Isochronous OUT Transactions in Standard Mode**

An endpoint will be first enabled and configured before being able to receive Isochronous packets.

When a 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 UBYCTLX and UBYCTHX registers. If the received packet is a ZLP (Zero Length Packet), the UBYCTLX and UBYCTHX register values are equal to 0 and no data has to be read.

The STLCRC bit in the UEPSTAX register is set by the USB controller if the packet stored in FIFO has a corrupted CRC. This bit is updated after each new packet receipt.

When all the endpoint FIFO bytes have been read, the firmware will clear the RXOUTB0 bit to allow the USB controller to store the next OUT packet data into the endpoint FIFO. Until the RXOUTB0 bit has been cleared by the firmware, the data sent by the Host at each OUT transaction will be lost.

If the RXOUTB0 bit is cleared while the Host is sending data, the USB controller will store only the remaining bytes into the FIFO.

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.

### **Isochronous OUT Transactions in Ping-pong Mode**

An endpoint will be first enabled and configured before being able to receive Isochronous packets.

When a OUT packet is received on the endpoint bank 0, 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 UBYCTLX and UBYCTHX registers. If the received packet is a ZLP (Zero Length Packet), the UBYCTLX and UBYCTHX register values are equal to 0 and no data has to be read.

The STLCRC bit in the UEPSTAX register is set by the USB controller if the packet stored in FIFO has a corrupted CRC. This bit is updated after each new packet receipt.

When all the endpoint FIFO bytes have been read, the firmware will clear the RXOUB0 bit to allow the USB controller to store the next OUT packet data into the endpoint FIFO bank 0. This action switches the endpoint bank 0 and 1. Until the RXOUTB0 bit has been cleared by the firmware, the data sent by the Host on the bank 0 endpoint FIFO will be lost.

If the RXOUTB0 bit is cleared while the Host is sending data on the endpoint bank 0, the USB controller will store only the remaining bytes into the FIFO.

When a new OUT packet is received on the endpoint bank 1, the RXOUTB1 bit is set by the USB controller. This triggers an interrupt if enabled. The firmware empties the bank 1 endpoint FIFO before clearing the RXOUTB1 bit. Until the RXOUTB1 bit has been cleared by the firmware, the data sent by the Host on the bank 1 endpoint FIFO will be lost.

The RXOUTB0 and RXOUTB1 bits are alternatively set by the USB controller at each new packet receipt.

The firmware has to clear one of these two bits after having read all the data FIFO to allow a new packet to be stored in the corresponding bank.

## Miscellaneous

### USB Reset

The EORINT bit in the USBINT register is set by hardware when a End Of Reset has been detected on the USB bus. This triggers a USB interrupt if enabled. The USB controller is still enabled, but all the USB registers are reset by hardware. The firmware will clear the EORINT bit to allow the next USB reset detection.

### STALL Handshake

This function is only available for Control, Bulk, and Interrupt endpoints.

The firmware has to set the STALLRQ bit in the UEPSTAX register to send a STALL handshake at the next request of the Host on the endpoint selected with the UEPNUM register. The RXSETUP, TXRDY, TXCMPL, RXOUTB0 and RXOUTB1 bits must be first reset to 0. The bit STLCRC is set at 1 by the USB controller when a STALL has been sent. This triggers an interrupt if enabled.

The firmware will clear the STALLRQ and STLCRC bits after each STALL sent.

The STALLRQ bit is cleared automatically by hardware when a valid SETUP PID is received on a CONTROL type endpoint.

Important note: when a Clear Halt Feature occurs for an endpoint, the firmware will reset this endpoint using the UEPRST register in order to reset the data toggle management.

### Start of Frame Detection

The SOFINT bit in the USBINT register is set when the USB controller detects a Start of Frame PID. This triggers an interrupt if enabled. The firmware will clear the SOFINT bit to allow the next Start of Frame detection.

### Frame Number

When receiving a Start of Frame, the frame number is automatically stored in the UFNUML and UFNUMH registers. The CRCOK and CRCERR bits indicate if the CRC of the last Start of Frame is valid (CRCOK set at 1) or corrupted (CRCERR set at 1). The UFNUML and UFNUMH registers are automatically updated when receiving a new Start of Frame.

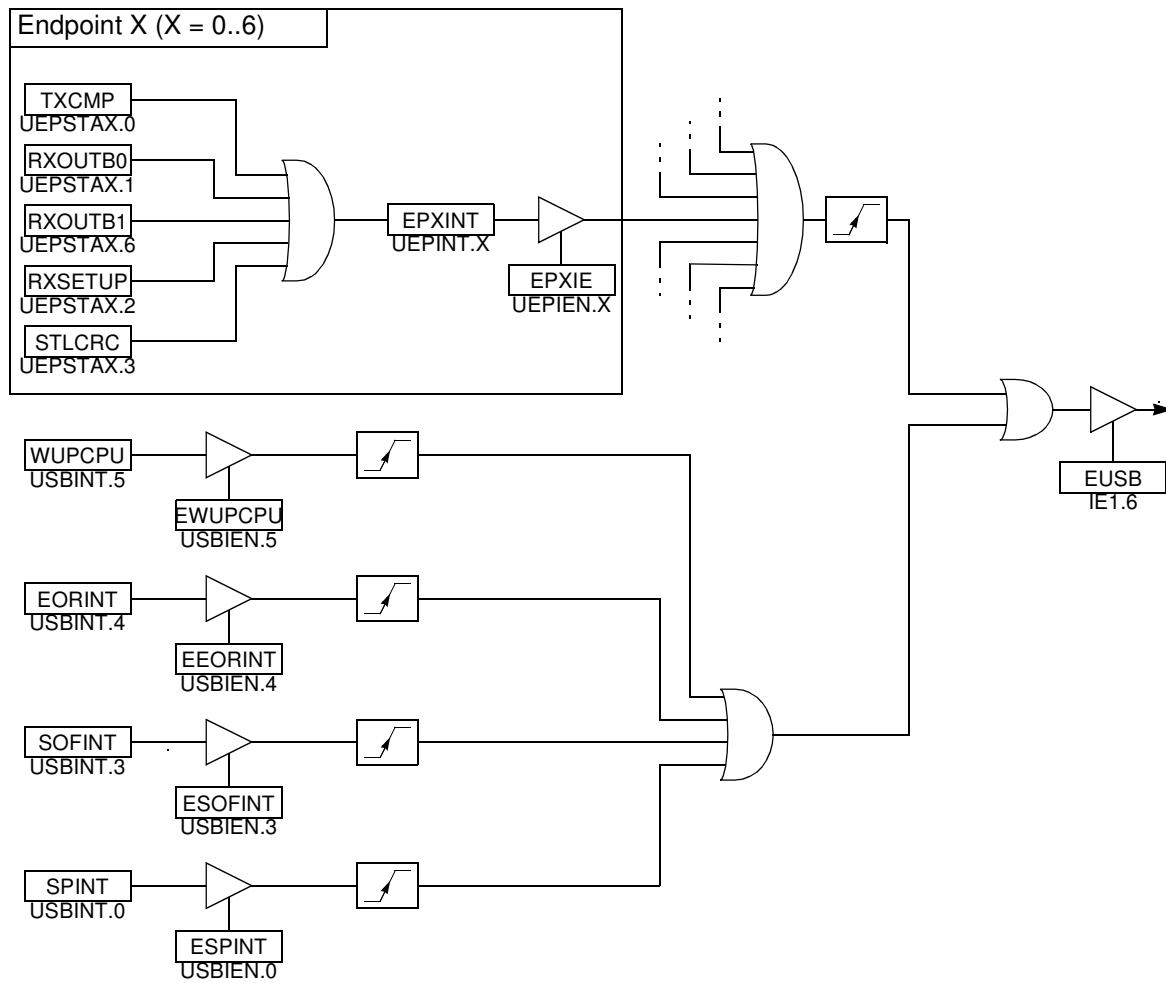
### Data Toggle Bit

The Data Toggle bit is set by hardware when a DATA0 packet is received and accepted by the USB controller and cleared by hardware when a DATA1 packet is received and accepted by the USB controller. This bit is reset when the firmware resets the endpoint FIFO using the UEPRST register.

For Control endpoints, each SETUP transaction starts with a DATA0 and data toggling is then used as for Bulk endpoints until the end of the Data stage (for a control write transfer). The Status stage completes the data transfer with a DATA1 (for a control read transfer).

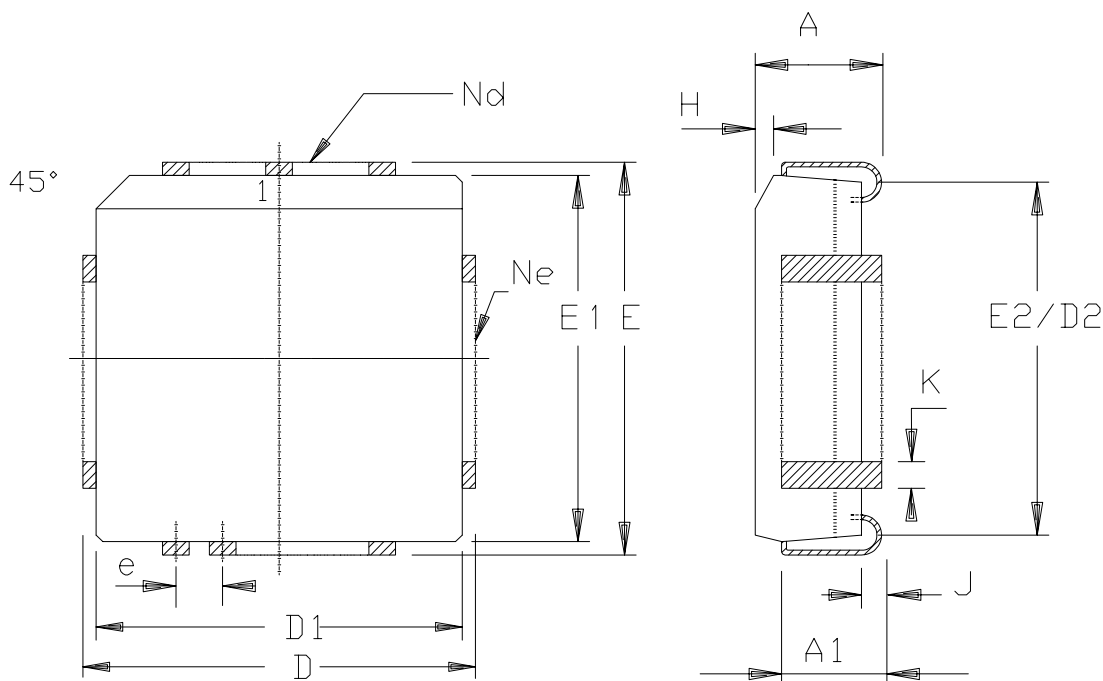
For Isochronous endpoints, the device firmware will ignore the data-toggle.

Figure 71. USB Interrupt Control Block Diagram



# 52-lead PLCC

52 PINS PLCC



	MM		INCH	
A	4. 20	4. 57	, 165	, 180
A1	2. 29	3. 30	, 090	, 130
D	19. 94	20. 19	, 785	, 795
D1	19. 05	19. 25	, 750	, 758
D2	17. 53	18. 54	, 690	, 730
E	19. 94	20. 19	, 785	, 795
E1	19. 05	19. 25	, 750	, 758
E2	17. 53	18. 54	, 690	, 730
e	1. 27	BSC	, 050	BSC
H	1. 07	1. 42	, 042	, 056
J	0. 51	—	, 020	—
K	0. 33	0. 53	, 013	, 021
Nd	13		13	
Ne	13		13	
PKG STD		00		

## STANDARD NOTES FOR PLCC:

1/ CONTROLLING DIMENSIONS : INCHES

2/ DIMENSIONING AND TOLERANCING PER ANSI Y 14.5M - 1982.

3/ "D" AND "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTUSIONS. MOLD FLASH OR PROTUSIONS SHALL NOT EXCEED 0.20 mm (.008 INCH) PER SIDE.

Flash Programming and Erasure .....	43
Boot Process .....	44
Application-Programming-Interface .....	45
XROW Bytes.....	45
Hardware Conditions .....	45
<b><i>On-chip Expanded RAM (ERAM).....</i></b>	<b><i>47</i></b>
<b><i>Timer 2 .....</i></b>	<b><i>50</i></b>
Auto-reload Mode .....	50
Programmable Clock Output .....	51
<b><i>Programmable Counter Array (PCA) .....</i></b>	<b><i>55</i></b>
PCA Capture Mode.....	62
16-bit Software Timer/Compare Mode.....	62
High Speed Output Mode .....	63
Pulse Width Modulator Mode.....	64
PCA Watchdog Timer .....	65
<b><i>Serial I/O Port .....</i></b>	<b><i>66</i></b>
Framing Error Detection .....	66
Automatic Address Recognition.....	67
Baud Rate Selection for UART for Mode 1 and 3.....	69
UART Registers.....	72
<b><i>Interrupt System .....</i></b>	<b><i>76</i></b>
Overview .....	76
Registers.....	77
Interrupt Sources and Vector Addresses.....	84
<b><i>Keyboard Interface .....</i></b>	<b><i>85</i></b>
Introduction .....	85
Description.....	85
Registers.....	86
<b><i>Programmable LED .....</i></b>	<b><i>89</i></b>
<b><i>Serial Peripheral Interface (SPI) .....</i></b>	<b><i>90</i></b>
Features.....	90
Signal Description.....	90
Functional Description .....	92
<b><i>Two Wire Interface (TWI).....</i></b>	<b><i>99</i></b>
Description.....	101
Notes .....	104
Registers.....	114