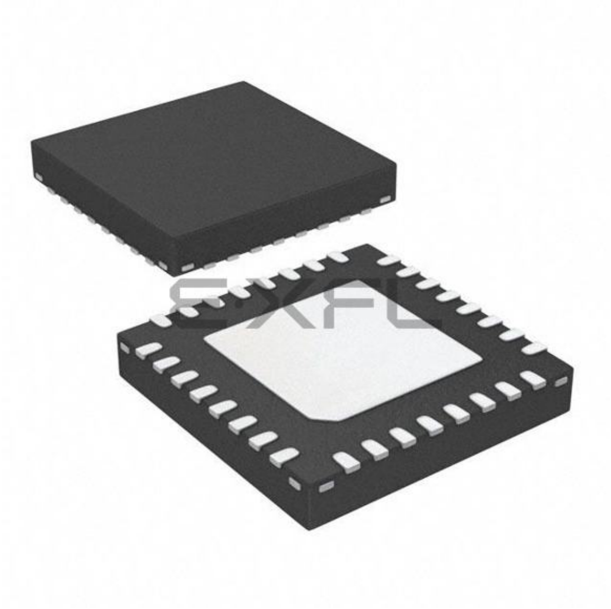


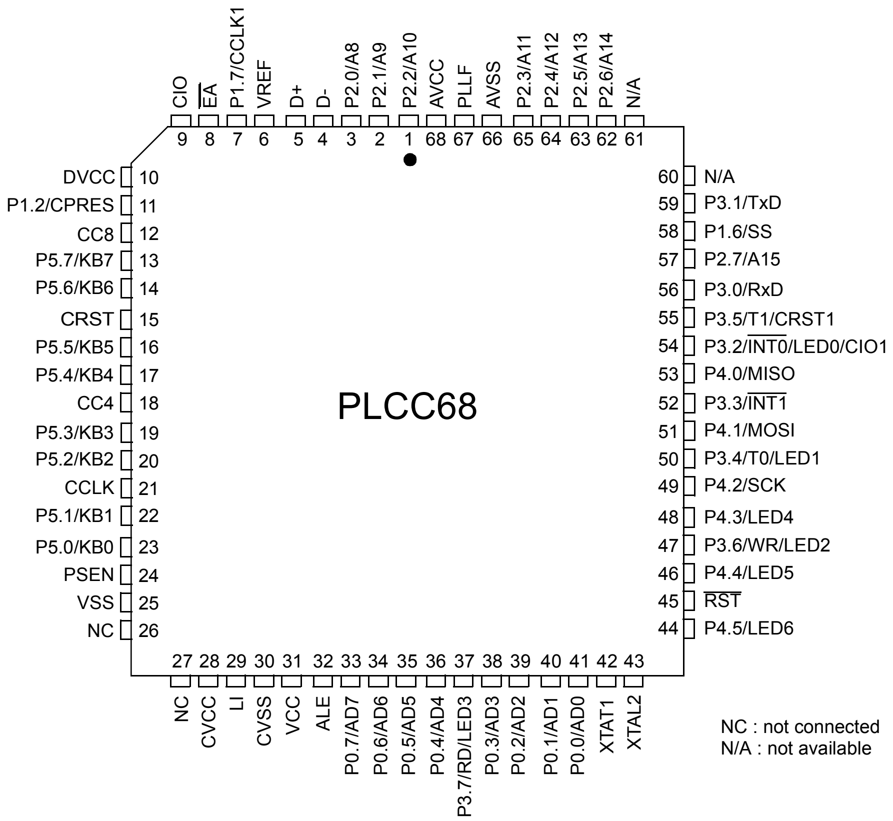
What is "Embedded - Microcontrollers"?

"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	CIP-51 8051
Core Size	8-Bit
Speed	72MHz
Connectivity	I²C, SMBus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	29
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 20x14b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-UFQFN Exposed Pad
Supplier Device Package	32-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/efm8lb11f32e-b-qfn32

Figure 2. PLCC68 Package Pinout (for engineering purpose only)



NC : not connected
N/A : not available

Low Pin Count Package Description

AT83C5123 version

Figure 6. VQFP32 Package Pinout

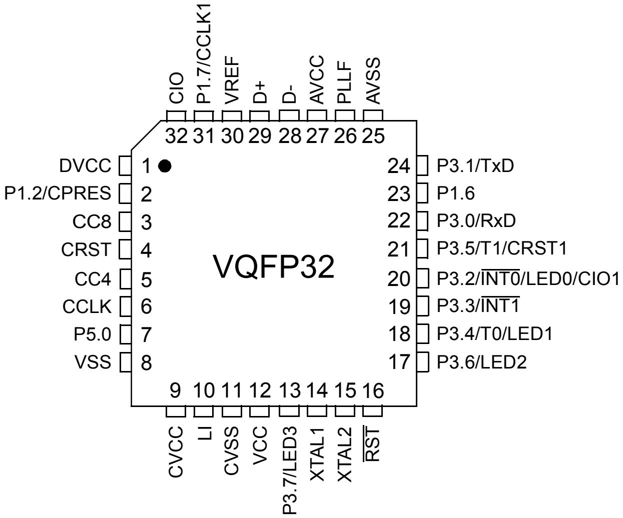
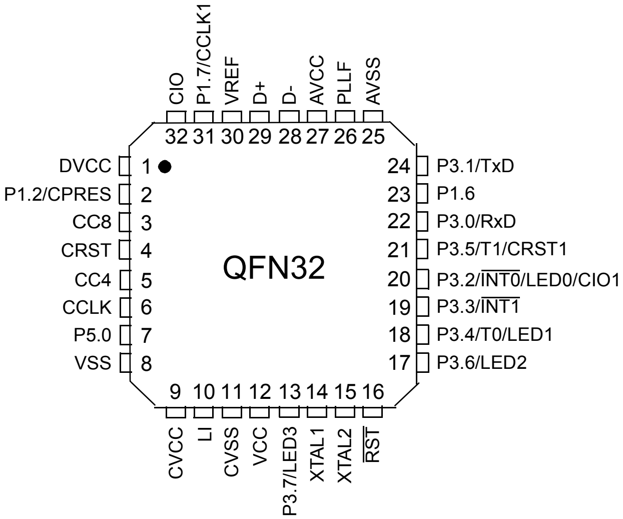
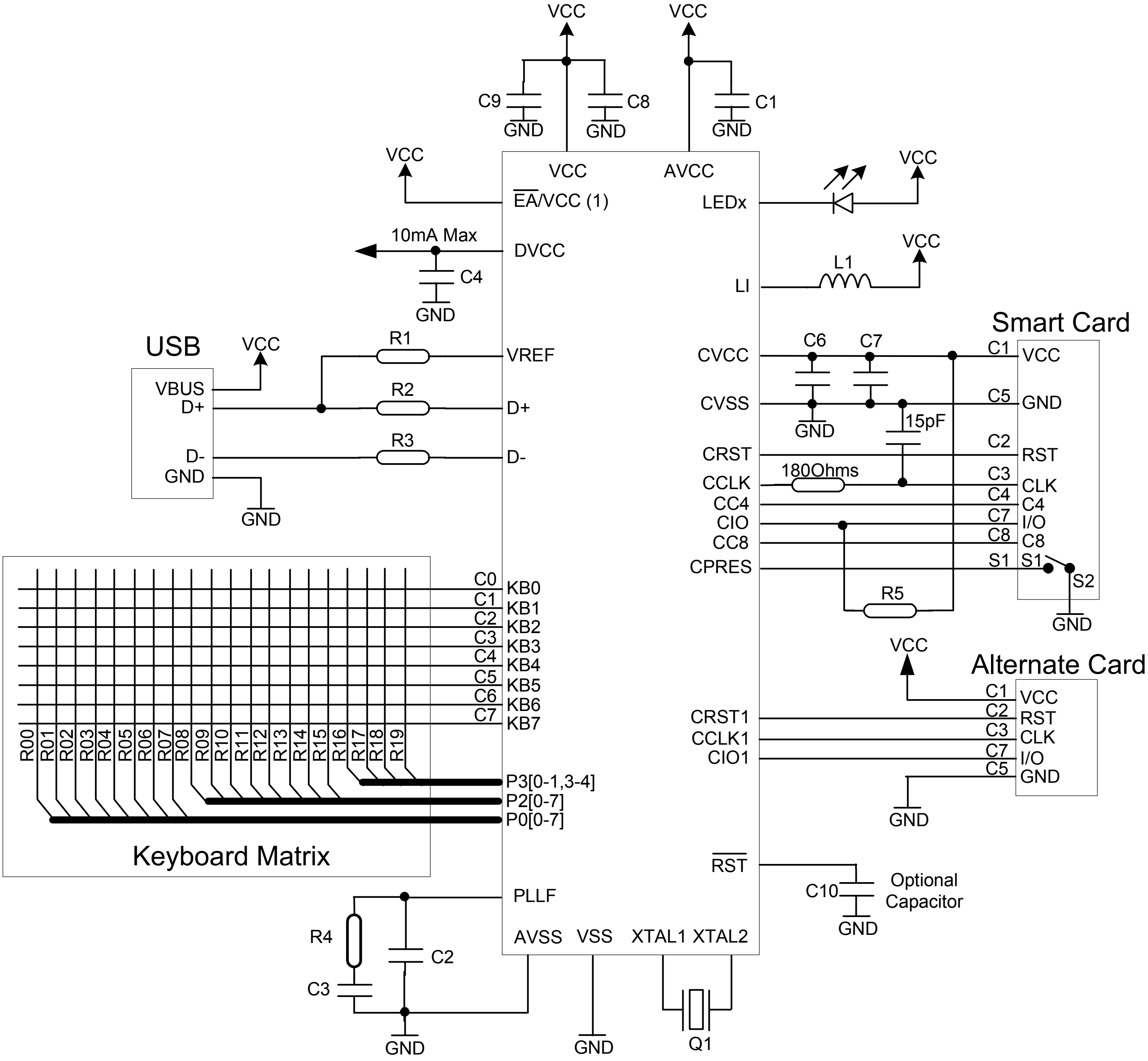


Figure 7. QFN32 Package Pinout



USB Keyboard with Smart Card Reader Using AT83R5122, AT8xC5122/AT89C5122DS



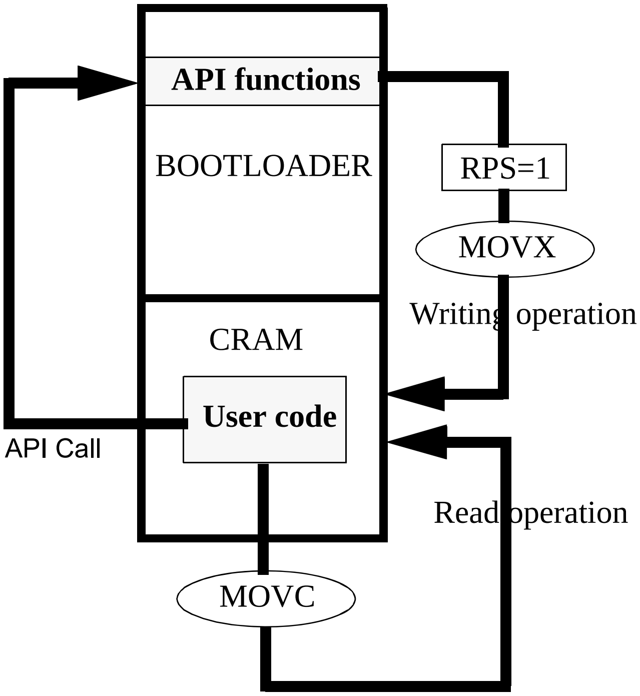
Notes :

1 - Pin configuration depends on product versions

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



Transmit / Receive Buffer

The contents of the SCIBUF Transmit / Receive Buffer is transferred or received into / from the Shift Register. The Shift Register is not accessible by microcontroller. Its role is to prepare the byte to be copied on the I/O pin for a transmission or in the SCIBUF buffer after a reception.

During a character transmission process, as soon as the contents of the SCIBUF buffer is transferred to the shift register, the SCTBE bit is set in SCISR register to indicate that the SCIBUF buffer is empty and ready to accept a new byte. This mechanism avoids to wait for the complete transmission of the previous byte before writing a new byte in the buffer and enables to speed up the transmission.

- If the Character repetition mode is not selected (bit CREP=0 in SCICR), as soon as the contents of the Shift Register is transferred to I/O pin, the SCTC bit is set in SCISR register to indicate that the byte has been transmitted.
- If the Character repetition mode is selected (bit CREP=1 in SCICR) The TERMINAL will be able to repeat characters as requested by the ICC (See the Parity Error in T=0 protocol description in the definition paragraph above). The SCTC bit in SCISR register will be set after a successful transmission (no retry or no further retry requested by the ICC). If the number of retries is exhausted (up to 4 retries depending on CREPSEL bit in SCSR) and the last retry is still unsuccessful, the SCTC bit in SCISR will not be set and the SCPE bit in SCISR register will be set instead.

During a character reception process, the contents of the Shift Register is transferred in the SCIBUF buffer.

- If the Character repetition mode is not selected (bit CREP=0 in SCICR), as soon as the contents of the Shift Register is transferred to the SCIBUF the SCRC bit is set in SCISR register to indicate that the byte has been received, and the SCIBUF contains a valid character ready to be red by the microcontroller.
- If the Character repetition mode is selected (bit CREP=1 in SCICR) The TERMINAL will be able to request repetition if the received character exhibit a parity error. Up to 4 retries can be requested depending on CREPSEL bit in SCSR. The SCRC bit will be set in SCISR register after a successful reception, first reception or after retry(ies). If the number of retries is exhausted (up to 4 retries depending on CREPSEL bit in SCSR) and the last retry is still unsuccessful, the SCRC bit and the SCPE bit in SCISR register will be set. It will be possible to read the erroneous character.

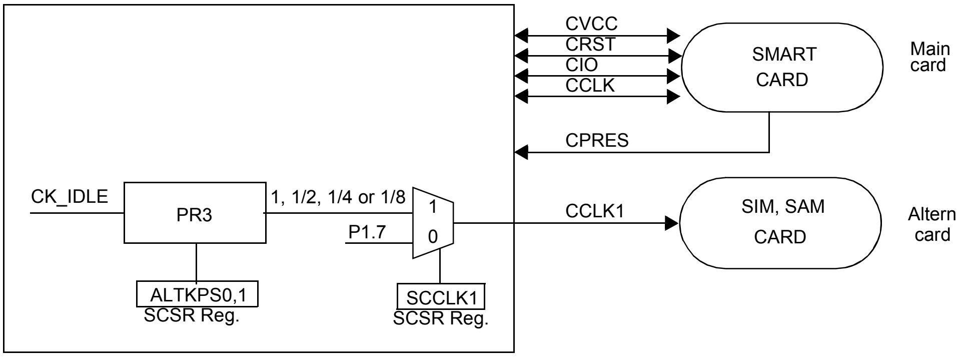
Warning : the SCTBI, SCTI SCRI and SCPI bits have the same function as SCTBE, SCTC, SCRC and SCPE bits. The first ones are able to generate interruptions if the interruptions are enabled in SCIER register while the second ones are only status bits to be used in pulling mode. If the interruption mode is not used, the status bits must be used. The SCTBI, SCTI and SCRI bits do not contain valid information while their respective interrupt enable bits ESCTBI, EXCTI, ESCRI are cleared.

Alternate Card

A second card named ‘Alternate Card’ can be controlled.

The Clock signal CCLK1 can be adapted to the XTAL frequency. Thanks to the clock prescaler which can divide the frequency by 1, 2, 4 or 8. The bits ALTKPS0 and ALTKPS1 in SCSR Register are used to set this factor.

Figure 44. Alternate Card



Registers

There are fifteen registers to control the SCIB macro-cell. They are described from Table 58 to Table 45.

Some of the register widths are greater than a byte. Despite the 8 bits access provided by the BIU, the address mapping of this kind of register respects the following rule :

The Low significant byte register is implemented at the higher address.

This implementation makes access to these registers easier when using high level programming languages (C,C++).

Table 45. Smart Card Contacts Register - SCCON (S:ACh, SCRS=0)

7	6	5	4	3	2	1	0
CLK	-	CARDC8	CARDC4	CARDIO	CARDCLK	CARDRST	CARDVCC

Bit Number	Bit Mnemonic	Description
7	CLK	**Card Clock Selection** Clear this bit to use the Card CLK bit (CARDCLK bit below) to drive Card CLK (P1.4) pin. Set this bit to use CK_XTAL1 or CK_PLL signals for CK_ISO to drive the Card CLK pin (CCLK = P1.4 pin) **Note:** internal synchronization avoids glitches on the CLK pin when switching this bit.
6	-	**Reserved** This bit can be changed by software but the read value is indeterminate.
5	CARDC8	**Card C8** Clear this bit to drive a low level on the Card C8 pin (CC8 = P1.1 pin). Set this bit to set a high level on the Card C8 pin (CC8 = P1.1 pin).. The CC8 pin can be used as a pseudo bi-directional I/O when this bit is set. **Warning** : VCARDOK=1 (SCISR.4 bit) condition must be true to change the state of CC8 pin
4	CARDC4	**Card C4** Clear this bit to drive a low level on the Card C4 pin (CC4 = P1.3 pin). Set this bit to set a high level on the Card C4 pin (CC4 = P1.3 pin). The CC4 pin can be used as a pseudo bi-directional I/O when this bit is set. **Warning** : VCARDOK=1 (SCISR.4 bit) condition must be true to change the state of CC4 pin
3	CARDIO	**Card I/O** If UART bit is cleared in SCICR register, this bit enables the use of the Card IO pin (CIO = P1.0) as a C51 pseudo bi-directional port : To read from CIO (P1.0) port pin : set CARDIO (P1.0) bit then read CARDIO (P1.0) bit to have the CIO port value To write in CIO (P1.0) port pin : set CARDIO (P1.0) bit to write a 1 in CIO (P1.0) port pin , clear CARDIO (P1.0) bit to write a 0 in CIO (P1.0) port pin. **Warning** : VCARDOK=1 (SCISR.4 bit) condition must be true to change the state of CIO pin
2	CARDCLK	**Card CLK** When the CLK bit is cleared in SCCON Register, the value of this bit is driven to the Card CLK pin. **Warning** : VCARDOK=1 (SCISR.4 bit) condition must be true to change the state of Card CLK pin
1	CARDRST	**Card RST** Clear this bit to drive a low level on the Card RST pin. Set this bit to set a high level on the Card RST pin. **Warning** : VCARDOK=1 (SCISR.4 bit) condition must be true to change the state of Card RST pin
0	CARDVCC	**Card VCC Control** Clear this bit to desactivate the Card interface and set its power-off. The other bits of SCCON register have no effect while this bit is cleared. Set this bit to power-on the Card interface. The activation sequence should be handled by software.

Reset Value = 0X00 0000b

Table 49. Smart Card Selection Register - SCSR (S:ABh)

7	6	5	4	3	2	1	0
-	BGTEN	-	CREPSEL	ALTKPS1	ALTKPS0	SCCLK1	SCRS

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
6	BGTEN	**Block Guard Time Enable** Set this bit to select the minimum interval between the leading edge of the start bits of the last character received from the ICC and the first character sent by the Terminal. The transfer of GT[8-0] value to the BGT counter is done on the rising edge of the BGTEN. Clear this bit to suppress the minimum time between reception and transmission.
5	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
4	CREPSEL	**Character repetition selection** Clear this bit to select 5 times transmission (1 original + 4 repetitions) before parity error indication (conform to EMV) Set this bit to select 4 times transmission (1 original + 3 repetitions) before parity error indication
3-2	ALTKPS1:0	**Alternate Card Clock prescaler factor** 00 ALTKPS = 0: prescaler factor equals 1 01 ALTKPS = 1: prescaler factor equals 2 10 ALTKPS = 2: prescaler factor equals 4 (reset value) 11 ALTKPS = 3: prescaler factor equals 8
1	SCCLK1	**Alternate card clock selection** Set to select the prescaled PR3 clock for CCLK1 (P1.7) pin Clear to select P1.7 port bit
0	SCRS	**Smart Card Register Selection** The SCRS bit selects which set of the SCIB registers is accessed.

Reset Value = X000 1000b

Table 50. Smart Card Transmit / Receive Buffer - SCIBUF (S:AA)

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

Bit Number	Bit Mnemonic	Description
-	-	**Smart Card Transmit / Receive Buffer** - A new byte can be written in the buffer to be transmitted on the I/O pin when SCTBE bit is set. The bits are sorted and copied on the I/O pin versus the active convention. - A new byte received from I/O pin is ready to be read when SCRI bit is set. The bits are sorted versus the active convention.

Reset Value = 0000 0000b

USB Controller

The AT8xC5122D implements a USB device controller supporting Full Speed data transfer. In addition to the default control endpoint 0, it provides 6 other endpoints, which can be configured in Control, Bulk, Interrupt or Isochronous modes:

- Endpoint 0: 32-byte FIFO, default control endpoint
- Endpoint 1,2,3: 8-byte FIFO
- Endpoint 4,5: 64-byte FIFO
- Endpoint 6: 2 x 64-byte Ping-pong FIFO

This allows the firmware to be developed conforming to most USB device classes, for example:

- USB Mass Storage Class Control/Bulk/Interrupt (CBI) Transport, Revision 1.0 - December 14, 1998.
- USB Mass Storage Class Bulk-Only Transport, Revision 1.0 - September 31, 1999.
- USB Human Interface Device Class, Version 1.1 - April 7, 1999.
- USB Device Firmware Upgrade Class, Revision 1.0 - May 13, 1999.

USB Mass Storage Classes

USB Mass Storage Class CBI Transport

Within the CBI framework, the Control endpoint is used to transport command blocks as well as to transport standard USB requests. One Bulk-Out endpoint is used to transport data from the host to the device. One Bulk-In endpoint is used to transport data from the device to the host. And one interrupt endpoint may also be used to signal command completion (protocol 0); it is optional and may not be used (protocol 1).

The following configuration adheres to these requirements:

- Endpoint 0: 8 bytes, Control In-Out
- Endpoint 4: 64 bytes, Bulk-Out
- Endpoint 5: 64 bytes, Bulk-In
- Endpoint 1: 8 bytes, Interrupt In

USB Mass Storage Class Bulk-Only Transport

Within the Bulk-Only framework, the Control endpoint is only used to transport class-specific and standard USB requests for device set-up and configuration. One Bulk-Out endpoint is used to transport commands and data from the host to the device. One Bulk-In endpoint is used to transport status and data from the device to the host. No interrupt endpoint is needed.

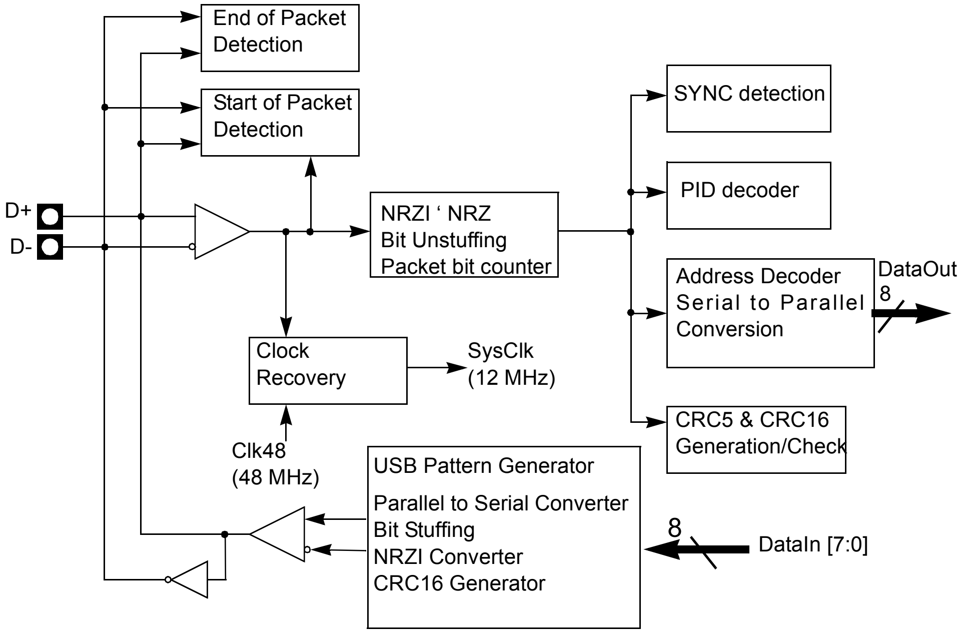
The following configuration adheres to these requirements:

- Endpoint 0: 8 bytes, Control In-Out
- Endpoint 4: 64 bytes, Bulk-Out
- Endpoint 5: 64 bytes, Bulk-In

USB Device Firmware Upgrade (DFU)

The USB Device Firmware Update (DFU) protocol can be used to upgrade the on-chip program memory of the AT8xC5122D. This allows the implementation of product enhancements and patches to devices that are already in the field. Two different configurations and description sets are used to support DFU functions. The Run-Time configuration co-exists with the usual functions of the device, which may be USB Mass Storage for the AT8xC5122D. It is used to initiate DFU from the normal operating mode. The DFU configuration is used to perform the firmware update after device re-configuration and USB reset. It excludes any other function. Only the default control pipe (endpoint 0) is used to support DFU services in both configurations.

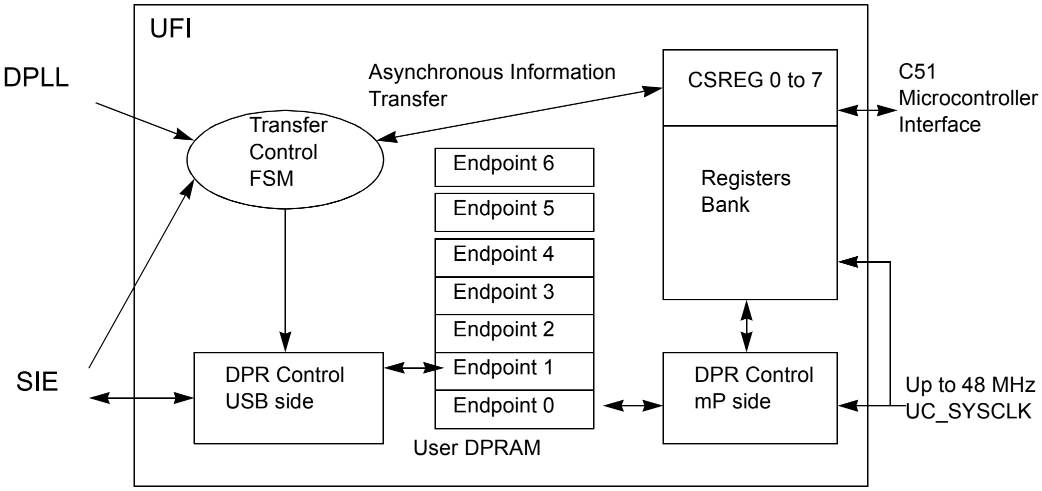
Figure 49. SIE Block Diagram



Function Interface Unit (UFI)

The Function Interface Unit provides the interface between the AT8xC5122D (or AT83C5123) 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 50. UFI Block Diagram



USB Interrupt System

Interrupt System Priorities

Figure 62. USB Interrupt Control System

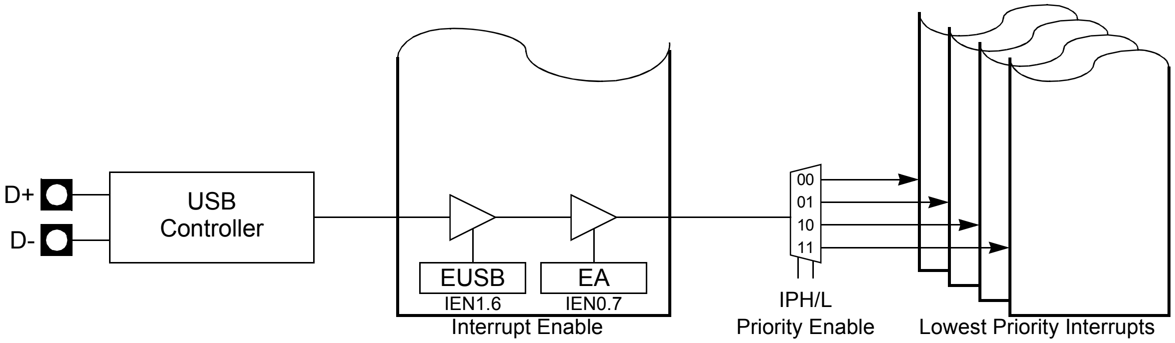


Table 63. Priority Levels

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

Interrupt Control System

As shown in Figure 63, many events can produce a USB interrupt:

- TXCMPL: Transmitted In Data (Table 70 on page 119). This bit is set by hardware when the Host accept a In packet.
- RXOUTB0: Received Out Data Bank 0 (Table 70 on page 119). This bit is set by hardware when an Out packet is accepted by the endpoint and stored in bank 0.
- RXOUTB1: Received Out Data Bank 1 (only for Ping-Pong endpoints) (Table 70 on page 119). This bit is set by hardware when an Out packet is accepted by the endpoint and stored in bank 1.
- RXSETUP: Received Setup (Table 70 on page 119). This bit is set by hardware when an SETUP packet is accepted by the endpoint.
- NAKIN and NAKOUT: These bits are set by hardware when a Nak Handshake has been received on the corresponding endpoint. These bits are cleared by software.
- STLCRC: STALLED (only for Control, Bulk and Interrupt endpoints) (Table on page 120). This bit is set by hardware when a STALL handshake has been sent as requested by STALLRQ, and is reset by hardware when a SETUP packet is received.
- SOFINT: Start Of Frame Interrupt (Table 65 on page 116). This bit is set by hardware when a USB start of frame packet has been received.
- WUPCPU: Wake-Up CPU Interrupt (Table 65 on page 116). This bit is set by hardware when a USB resume is detected on the USB bus, after a SUSPEND state.
- SPINT: Suspend Interrupt (Table 65 on page 116). This bit is set by hardware when a USB suspend is detected on the USB bus.

Serial I/O Port

The serial I/O port in the AT83R5122, AT8xC5122/23 is compatible with the serial I/O port in the 80C52.

The I/O port 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

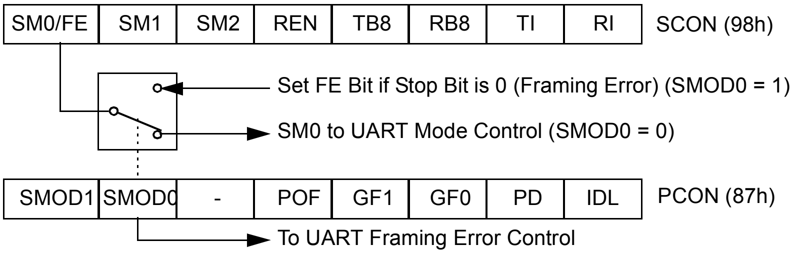
Serial I/O port includes the following enhancements:

- Framing error detection
- Automatic address recognition

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 64).

Figure 64. Framing Error Block Diagram


SM0/FE
SM1
SM2
REN
TB8
RB8
TI
RI
SCON (98h)
Set FE Bit if Stop Bit is 0 (Framing Error) (SMOD0 = 1)
SM0 to UART Mode Control (SMOD0 = 0)
SMOD1
SMOD0
-
POF
GF1
GF0
PD
IDL
PCON (87h)
To UART Framing Error Control

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 Figure 69 on page 128) 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 65 and Figure 66).

Figure 65. UART Timings in Mode 1

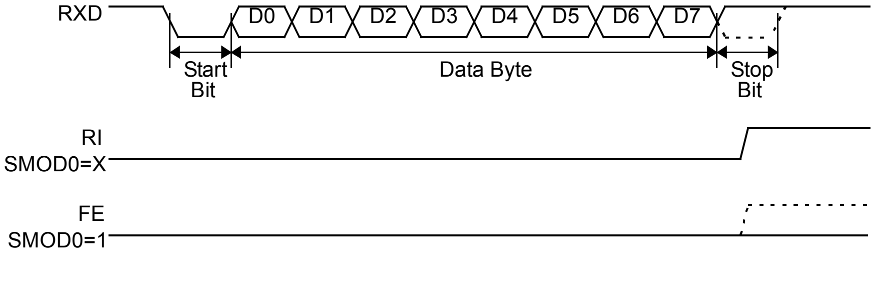
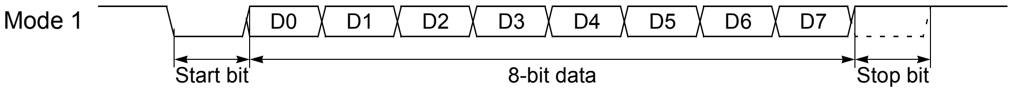


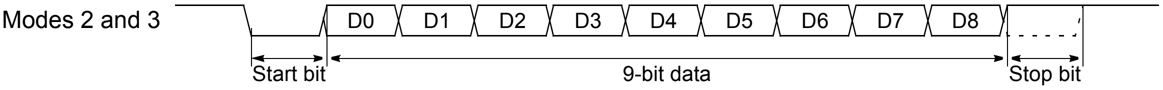
Figure 75. Data Frame Format (Mode 1)



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

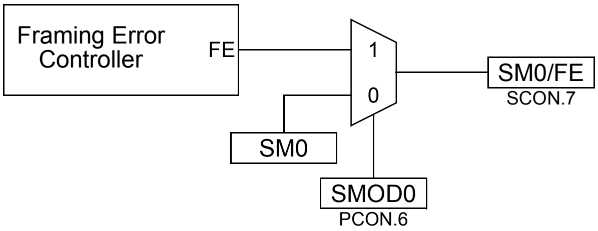
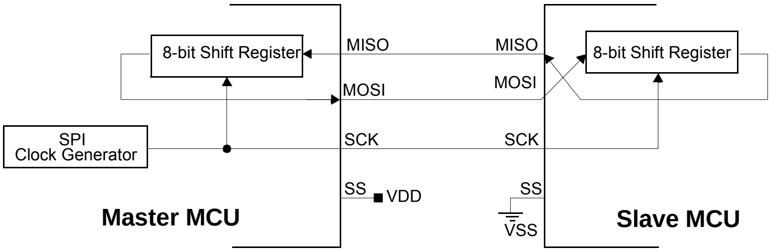


Figure 84. Full-duplex Master-Slave Interconnection



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 103. Interrupt Priority Low Register 1 - IPL1 (B2h) for AT8xC5122

7	6	5	4	3	2	1	0
-	PUSBL	-	-	PSCIL	PSPIL	-	PKBDL

Bit Number	Bit Mnemonic	Description
7	-	Reserved The value read from this bit is indeterminate. Do not change this bit.
6	PUSBL	**USB Interrupt Priority bit** Refer to PUSBH for priority level.
5 - 4	-	**Reserved** The value read from this bit is indeterminate. Do not change these bits.
3	PSCIL	**SCI Interrupt Priority bit** Refer to PSPIH for priority level.
2	PSPIL	**SPI Interrupt Priority bit** Refer to PSPIH for priority level.
1	-	**Reserved** The value read from this bit is indeterminate. Do not change this bit.
0	PKBL	**Keyboard Interrupt Priority bit** Refer to PKBDH for priority level.

Reset Value = X00X 00X0b (Bit addressable)

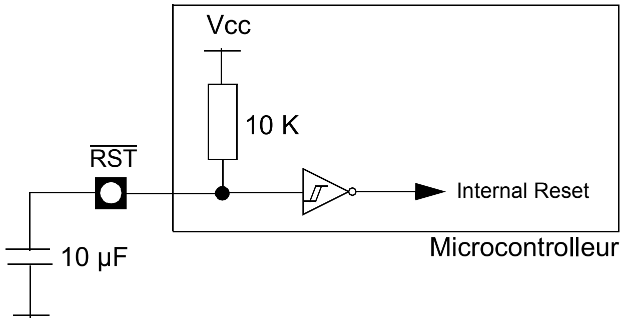
Reset pin

As explained in the POR section there is no need to use the reset pin as the internal reset function at power up is ensured by the POR. Anyway, if some applications requires a long reset, a reset controlled by the user or a reset controlled by external superviser device, the use of the reset pin is necessary.

Long Reset

As the pad integrates an internal pull-up of 10K, only an external capacitor of at least 10 µF is required to have an impact on the reset duration.

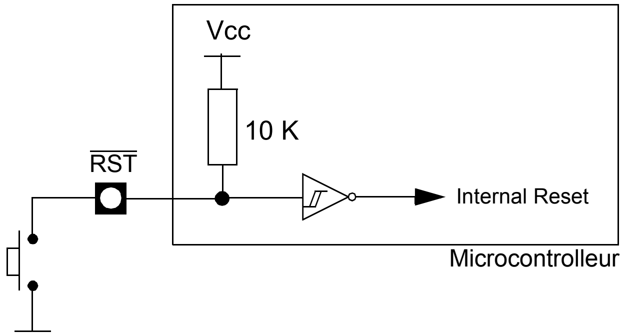
Figure 103. Long Reset



Reset Controlled by the User

The external capacitor is not needed if no long reset is required.

Figure 104. Reset Controlled by the User



Reset Controlled by an External Superviser Device

As the reset pin can be forced in output by the Watch-Dog timer (WDT) or the POR/PFD features, there can be a conflict between the external superviser device and the microcontroller's reset pin when in one side the external superviser is pulling the reset pin to VCC and in another side the WDT or POR/PFD features tries to force the reset pin to ground. Therefore, it recommended to insert a series resistor of 1.8K +/-10% or a diode (1N4148 for instance) between the external superviser device and the reset pin as detailed in the following figures.

Figure 105. Use of an External Serial Resistor

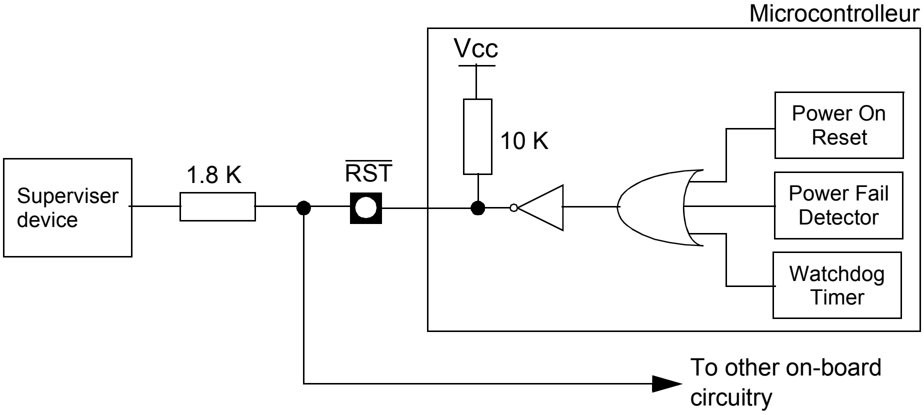
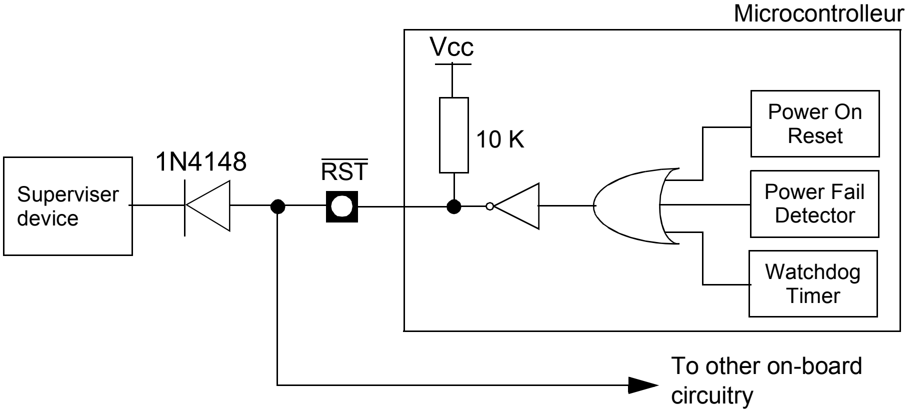


Figure 106. Use of an External Diode

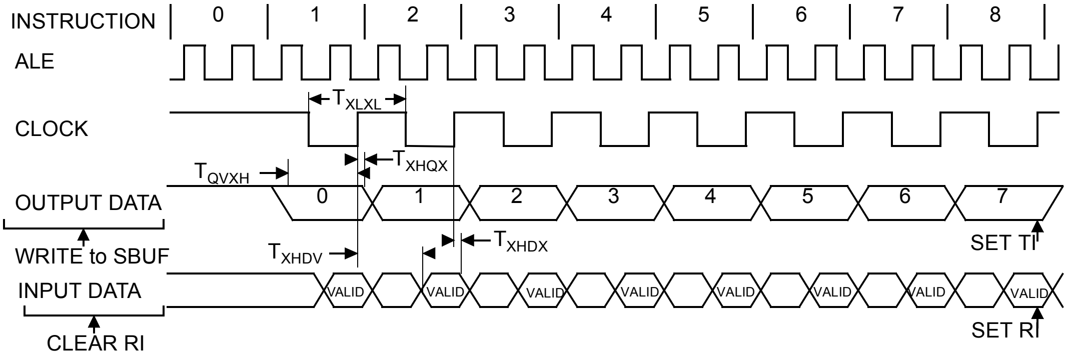


USB Interface

Figure 112. USB Interface

Symbol	Parameter	Min	Typ[5]	Max	Unit
V_{REF}	USB Reference Voltage	3.0		3.6	V
V_{IH}	Input High Voltage for D+ and D- (driven)	2.0		4.0	V
V_{IHZ}	Input High Voltage for D+ and D- (floating)	2.7		3.6	V
V_{IL}	Input Low Voltage for D+ and D-			0.8	V
V_{OH}	Output High Voltage for D+ and D-	2.8		3.6	V
V_{OL}	Output Low Voltage for D+ and D-	0.0		0.3	V

Shift Register Timing Waveform

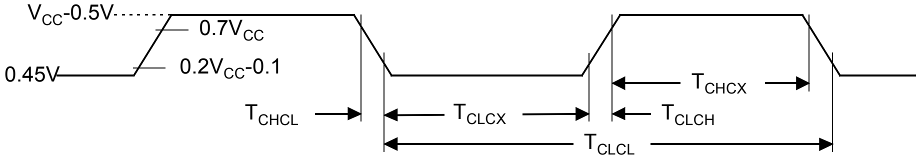


External Clock Drive Characteristics (XTAL1)

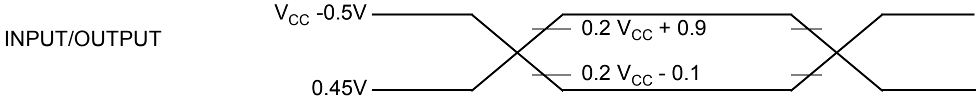
Table 120. AC Parameters

Symbol	Parameter	Min	Max	Units
T_{CLCL}	Oscillator Period	125		ns
T_{CHCX}	High Time	5		ns
T_{CLCX}	Low Time	5		ns
T_{CLCH}	Rise Time		5	ns
T_{CHCL}	Fall Time		5	ns
T_{CHCX}/T_{CLCX}	Cyclic ratio in X2 mode	40	60	%

External Clock Drive Waveforms

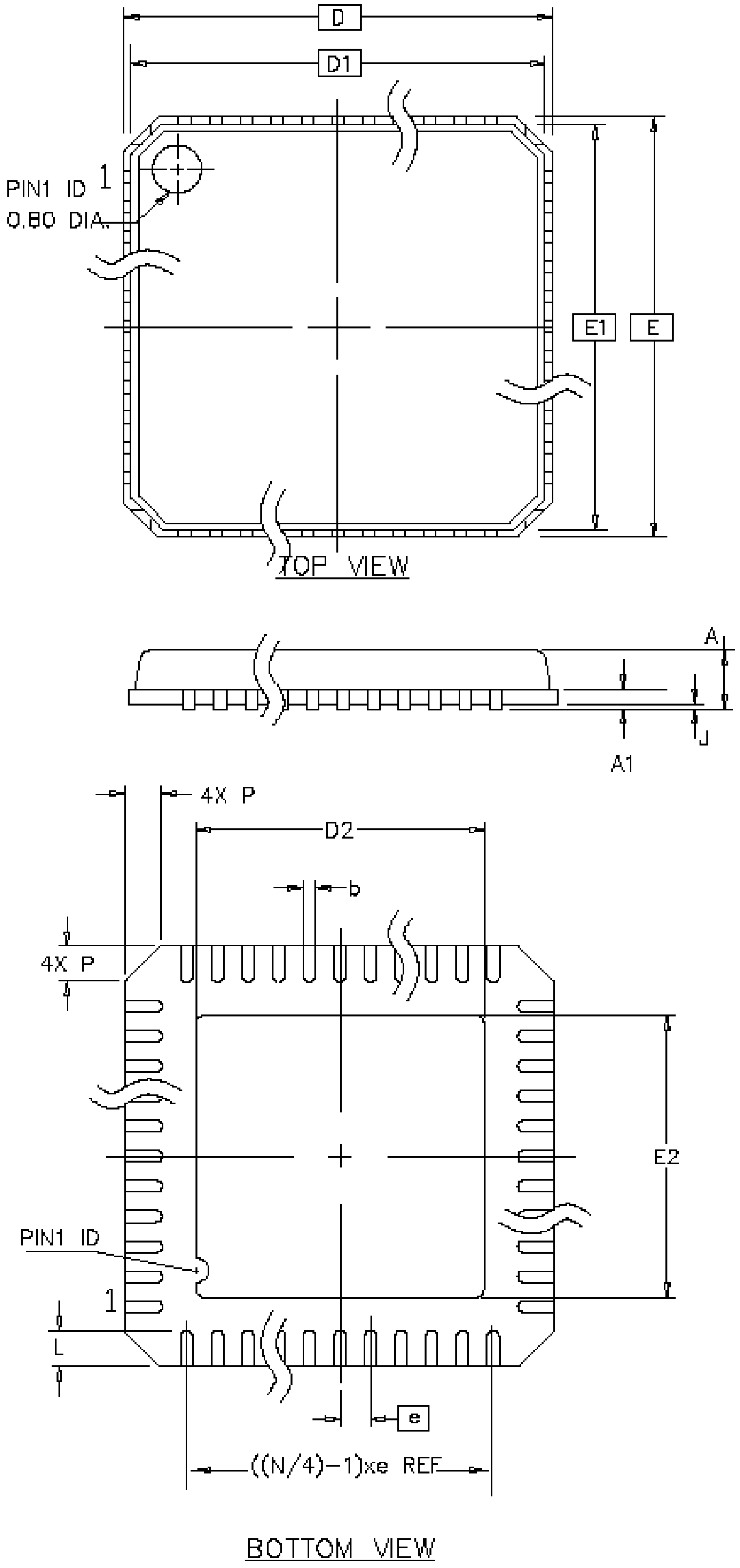


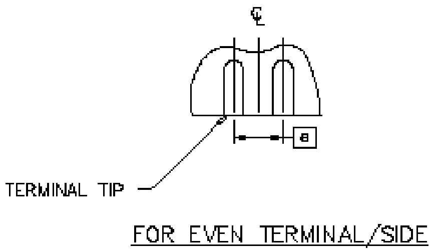
AC Testing Input/Output Waveforms



AC inputs during testing are driven at V_{CC} - 0.5 for a logic “1” and 0.45V for a logic “0”. Timing measurement are made at V_{IH} min for a logic “1” and V_{IL} max for a logic “0”.

QFN32 Package





	MM			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	0.85	0.90	-	.033	.035
J	0.00	0.01	0.05	.000	.000	.002
A1	0.20 ref			.008 ref		
D/E	7.00 BSC			.276 BSC		
D1/E1	6.75 BSC			.266 BSC		
D2/E2	4.95	5.10	5.25	.195	.201	.207
N	32					
P	0.24	0.42	0.60	.009	.016	.024
e	0.65 BSC			.026 BSC		
L	0.50	0.60	0.75	.020	.024	.030
b	0.23	0.28	0.35	.009	.011	.014