**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 | C52X2 |
| Core Size | 8-Bit |
| Speed | 48MHz |
| Connectivity | I²C, SPI, UART/USART, USB |
| Peripherals | LED, POR, PWM, WDT |
| Number of I/O | 18 |
| 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 ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at89c5131a-tirul |

## Registers

**Table 14.** CKCON0 (S:8Fh)
Clock Control Register 0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TWIX2 | WDX2 | PCAX2 | SIX2 | T2X2 | T1X2 | T0X2 | X2 |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | TWIX2 | **TWI Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 6 | WDX2 | **Watchdog Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 5 | PCAX2 | **Programmable Counter Array Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 4 | SIX2 | **Enhanced UART Clock (Mode 0 and 2)**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 3 | T2X2 | **Timer2 Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 2 | T1X2 | **Timer1 Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 1 | T0X2 | **Timer0 Clock**<br>**This control bit is validated when the CPU clock X2 is set. When X2 is low, this bit has no effect.**<br>Clear to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |
| 0 | X2 | **System Clock Control bit**<br>Clear to select 12 clock periods per machine cycle (STD mode, $F_{CPU} = F_{PER} = F_{OSC}/2$).<br>Set to select 6 clock periods per machine cycle (X2 mode, $F_{CPU} = F_{PER} = F_{OSC}$). |

Reset Value = 0000 0000b

The table below shows all SFRs with their address and their reset value.

**Table 18.** SFR Descriptions

| | Bit Addressable | Non-Bit Addressable | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0/8 | 1/9 | 2/A | 3/B | 4/C | 5/D | 6/E | 7/F | |
| F8h | UEPINT 0000 0000 | CH 0000 0000 | CCAP0H XXXX XXXX | CCAP1H XXXX XXXX | CCAP2H XXXX XXXX | CCAP3H XXXX XXXX | CCAP4H XXXX XXXX | | FFh |
| F0h | B 0000 0000 | LEDCON 0000 0000 | | | | | | | F7h |
| E8h | | CL 0000 0000 | CCAP0L XXXX XXXX | CCAP1L XXXX XXXX | CCAP2L XXXX XXXX | CCAP3L XXXX XXXX | CCAP4L XXXX XXXX | | EFh |
| E0h | ACC 0000 0000 | | UBYCTLX 0000 0000 | UBYCTHX 0000 0000 | | | | | E7h |
| D8h | CCON 00X0 0000 | CMOD 00XX X000 | CCAPM0 X000 0000 | CCAPM1 X000 0000 | CCAPM2 X000 0000 | CCAPM3 X000 0000 | CCAPM4 X000 0000 | | DFh |
| D0h | PSW 0000 0000 | FCON (1) XXXX 0000 | EECON XXXX XX00 | | UEPCONX 1000 0000 | UEPRST 0000 0000 | | | D7h |
| C8h | T2CON 0000 0000 | T2MOD XXXX XX00 | RCAP2L 0000 0000 | RCAP2H 0000 0000 | TL2 0000 0000 | TH2 0000 0000 | UEPSTAX 0000 0000 | UEPDATX 0000 0000 | CFh |
| C0h | P4 XXXX 1111 | | UEPIEN 0000 0000 | SPCON 0001 0100 | SPSTA 0000 0000 | SPDAT XXXX XXXX | USBADDR 1000 0000 | UEPNUM 0000 0000 | C7h |
| B8h | IPL0 X000 000 | SADEN 0000 0000 | UFNUML 0000 0000 | UFNUMH 0000 0000 | USBCON 0000 0000 | USBINT 0000 0000 | USBIEN 0000 0000 | | BFh |
| B0h | P3 1111 1111 | IEN1 X0XX X000 | IPL1 X0XX X000 | IPH1 X0XX X000 | | | | IPH0 X000 0000 | B7h |
| A8h | IEN0 0000 0000 | SADDR 0000 0000 | | | | | | CKCON1 0000 0000 | AFh |
| A0h | P2 1111 1111 | | AUXR1 XXXX X0X0 | PLLCON XXXX XX00 | PLLDIV 0000 0000 | | WDTRST XXXX XXXX | WDTPRG XXXX X000 | A7h |
| 98h | SCON 0000 0000 | SBUF XXXX XXXX | BRL 0000 0000 | BDRCON XXX0 0000 | KBLS 0000 0000 | KBE 0000 0000 | KBF 0000 0000 | | 9Fh |
| 90h | P1 1111 1111 | | | SSCON 0000 0000 | SSCS 1111 1000 | SSDAT 1111 1111 | SSADR 1111 1110 | | 97h |
| 88h | TCON 0000 0000 | TMOD 0000 0000 | TL0 0000 0000 | TL1 0000 0000 | TH0 0000 0000 | TH1 0000 0000 | AUXR XX0X 0000 | CKCON0 0000 0000 | 8Fh |
| 80h | P0 1111 1111 | SP 0000 0111 | DPL 0000 0000 | DPH 0000 0000 | | | | PCON 00X1 0000 | 87h |
| | 0/8 | 1/9 | 2/A | 3/B | 4/C | 5/D | 6/E | 7/F | |

Note: 1. FCON access is reserved for the Flash API and ISP software.

Reserved

# Flash EEPROM Memory

## General Description

The Flash memory increases EPROM functionality with in-circuit electrical erasure and programming. It contains 32 Kbytes of program memory organized in 256 pages of 128 bytes, respectively. This memory is both parallel and serial In-System Programmable (ISP). ISP allows devices to alter their own program memory in the actual end product under software control. A default serial loader (bootloader) program allows ISP of the Flash.

The programming does not require 12V external programming voltage. The necessary high programming voltage is generated on-chip using the standard $V_{CC}$ pins of the microcontroller.

## Features

- Flash EEPROM internal program memory.
- Boot vector allows user-provided Flash loader code to reside anywhere in the Flash memory space. This configuration provides flexibility to the user.
- Default loader in Boot EEPROM allows programming via the serial port without the need of a user provided loader.
- Up to 64K bytes external program memory if the internal program memory is disabled (EA = 0).
- Programming and erase voltage with standard power supply.
- Read/Program/Erase:
- Byte-wise read (without wait state).
- Byte or page erase and programming (10 ms).
- Typical programming time (32 Kbytes) in 10 sec.
- Parallel programming with 87C51 compatible hardware interface to programmer.
- Programmable security for the code in the Flash.
- 100K write cycles
- 10 years data retention

## Flash Programming and Erasure

The 32 Kbytes Flash is programmed by bytes or by pages of 128 bytes. It is not necessary to erase a byte or a page before programming. The programming of a byte or a page includes a self erase before programming.

There are three methods of programming the Flash memory:

1. The on-chip ISP bootloader may be invoked which will use low level routines to program the pages. The interface used for serial downloading of Flash is the USB.
2. The Flash may be programmed or erased in the end-user application by calling low-level routines through a common entry point in the Boot Flash.
3. The Flash may be programmed using the parallel method .

The bootloader and the Application Programming Interface (API) routines are located in the Flash Bootloader.

**Table 45.** AUXR Register

AUXR - Auxiliary Register (8Eh)

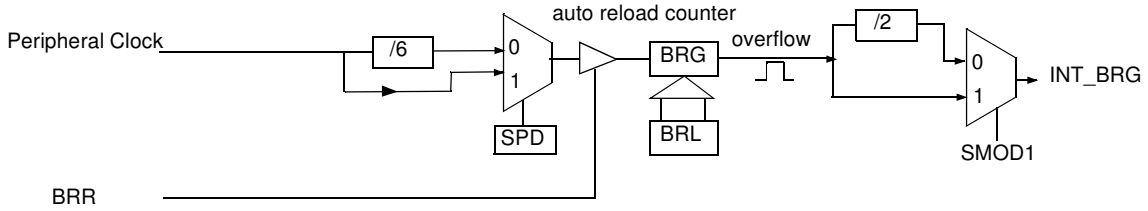
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **DPU** | **-** | **M0** | **-** | **XRS1** | **XRS0** | **EXTRAM** | **AO** |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | DPU | **Disable Weak Pull Up**<br>Cleared to enabled weak pull up on standard Ports.<br>Set to disable weak pull up on standard Ports. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit |
| 5 | M0 | **Pulse length**<br>Cleared to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 6 clock periods (default).<br>Set to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 30 clock periods. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit |
| 3 | XRS1 | **ERAM Size**<br>**XRS1 XRS0 — ERAM size**<br>0 0 — 256 bytes<br>0 1 — 512 bytes<br>1 0 — 768 bytes<br>1 1 — 1024 bytes (default) |
| 2 | XRS0 | |
| 1 | EXTRAM | **EXTRAM bit**<br>Cleared to access internal ERAM using MOVX at $\overline{Ri}$ at DPTR.<br>Set to access external memory. |
| 0 | AO | **ALE Output bit**<br>Cleared, ALE is emitted at a constant rate of 1/6 the oscillator frequency (or 1/3 if X2 mode is used) (default).<br>Set, ALE is active only when a MOVX or MOVC instruction is used. |

Reset Value = 0X0X 1100b

Not bit addressable

**Figure 38.** Internal Baud Rate




- The baud rate for UART is token by formula:

$$Baud\_Rate = \frac{2^{SMOD1} \times F_{CLK\ PERIPH}}{2 \times 6^{(1-SPD)} \times 16 \times [256 - (BRL)]}$$

$$(BRL) = 256 - \frac{2^{SMOD1} \times F_{CLK\ PERIPH}}{2 \times 6^{(1-SPD)} \times 16 \times Baud\_Rate}$$

Example of computed value when X2 = 1, SMOD1 = 1, SPD = 1

| Baud Rates | $F_{osc}$ = 16.384 MHz | | $F_{osc}$ = 24 MHz | |
|---|---|---|---|---|
| | BRL | Error (%) | BRL | Error (%) |
| 115200 | 247 | 1.23 | 243 | 0.16 |
| 57600 | 238 | 1.23 | 230 | 0.16 |
| 38400 | 229 | 1.23 | 217 | 0.16 |
| 28800 | 220 | 1.23 | 204 | 0.16 |
| 19200 | 203 | 0.63 | 178 | 0.16 |
| 9600 | 149 | 0.31 | 100 | 0.16 |
| 4800 | 43 | 1.23 | - | - |

Example of computed value when X2 = 0, SMOD1 = 0, SPD = 0

| Baud Rates | $F_{osc}$ = 16.384 MHz | | $F_{osc}$ = 24 MHz | |
|---|---|---|---|---|
| | BRL | Error (%) | BRL | Error (%) |
| 4800 | 247 | 1.23 | 243 | 0.16 |
| 2400 | 238 | 1.23 | 230 | 0.16 |
| 1200 | 220 | 1.23 | 202 | 3.55 |
| 600 | 185 | 0.16 | 152 | 0.16 |

The baud rate generator can be used for mode 1 or 3 (refer to Figure 37.), but also for mode 0 for UART, thanks to the bit SRC located in BDRCON register (Table 59.)

## UART Registers

SADEN - Slave Address Mask Register for UART (B9h)

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

Reset Value = 0000 0000b

SADDR - Slave Address Register for UART (A9h)

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

Reset Value = 0000 0000b

SBUF - Serial Buffer Register for UART (99h)

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

Reset Value = XXXX XXXXb

**Table 58.** 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**<br>Set to select double baud rate in mode 1, 2 or 3. |
| 6 | SMOD0 | **Serial port Mode bit 0 for UART**<br>Cleared to select SM0 bit in SCON register.<br>Set to select FE bit in SCON register. |
| 5 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 4 | POF | **Power-Off Flag**<br>Cleared to recognize next reset type.<br>Set by hardware when $V_{CC}$ rises from 0 to its nominal voltage. Can also be set by software. |
| 3 | GF1 | **General-purpose Flag**<br>Cleared by user for general-purpose usage.<br>Set by user for general-purpose usage. |
| 2 | GF0 | **General-purpose Flag**<br>Cleared by user for general-purpose usage.<br>Set by user for general-purpose usage. |
| 1 | PD | **Power-down Mode Bit**<br>Cleared by hardware when reset occurs.<br>Set to enter power-down mode. |
| 0 | IDL | **Idle Mode Bit**<br>Cleared by hardware when interrupt or reset occurs.<br>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 65.** IPL1 Register

IPL1 - Interrupt Priority Register (B2h)

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

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | PUSBL | **USB Interrupt Priority bit**<br>Refer to PUSBH for priority level. |
| 5 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 3 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 2 | PSPIL | **SPI Interrupt Priority bit**<br>Refer to PSPIH for priority level. |
| 1 | PTWIL | **TWI Interrupt Priority bit**<br>Refer to PTWIH for priority level. |
| 0 | PKBL | **Keyboard Interrupt Priority bit**<br>Refer to PKBH for priority level. |

Reset Value = X0XX X000b
Not bit addressable

## Interrupt Sources and Vector Addresses

**Table 67.** Vector Table

| Number | Polling Priority | Interrupt Source | Interrupt Request | Vector Address |
|---|---|---|---|---|
| 0 | 0 | Reset | | 0000h |
| 1 | 1 | INT0 | IE0 | 0003h |
| 2 | 2 | Timer 0 | TF0 | 000Bh |
| 3 | 3 | INT1 | IE1 | 0013h |
| 4 | 4 | Timer 1 | IF1 | 001Bh |
| 5 | 6 | UART | RI+TI | 0023h |
| 6 | 7 | Timer 2 | TF2+EXF2 | 002Bh |
| 7 | 5 | PCA | CF + CCFn (n = 0-4) | 0033h |
| 8 | 8 | Keyboard | KBDIT | 003Bh |
| 9 | 9 | TWI | TWIIT | 0043h |
| 10 | 10 | SPI | SPIIT | 004Bh |
| 11 | 11 | | | 0053h |
| 12 | 12 | | | 005Bh |
| 13 | 13 | | | 0063h |
| 14 | 14 | USB | UEPINT + USBINT | 006Bh |
| 15 | 15 | | | 0073h |

**Table 69.** KBE Register

KBE **-** Keyboard Input Enable Register (9Dh)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| KBE7 | KBE6 | KBE5 | KBE4 | KBE3 | KBE2 | KBE1 | KBE0 |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | KBE7 | **Keyboard line 7 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.7 bit in KBF register to generate an interrupt request. |
| 6 | KBE6 | **Keyboard line 6 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.6 bit in KBF register to generate an interrupt request. |
| 5 | KBE5 | **Keyboard line 5 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.5 bit in KBF register to generate an interrupt request. |
| 4 | KBE4 | **Keyboard line 4 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.4 bit in KBF register to generate an interrupt request. |
| 3 | KBE3 | **Keyboard line 3 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.3 bit in KBF register to generate an interrupt request. |
| 2 | KBE2 | **Keyboard line 2 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.2 bit in KBF register to generate an interrupt request. |
| 1 | KBE1 | **Keyboard line 1 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.1 bit in KBF register to generate an interrupt request. |
| 0 | KBE0 | **Keyboard line 0 Enable bit**<br>Cleared to enable standard I/O pin.<br>Set to enable KBF.0 bit in KBF register to generate an interrupt request. |

Reset Value = 0000 0000b

**Table 70.** KBLS Register

KBLS-Keyboard Level Selector Register (9Ch)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| KBLS7 | KBLS6 | KBLS5 | KBLS4 | KBLS3 | KBLS2 | KBLS1 | KBLS0 |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | KBLS7 | **Keyboard line 7 Level Selection bit**<br>Cleared to enable a low level detection on Port line 7.<br>Set to enable a high level detection on Port line 7. |
| 6 | KBLS6 | **Keyboard line 6 Level Selection bit**<br>Cleared to enable a low level detection on Port line 6.<br>Set to enable a high level detection on Port line 6. |
| 5 | KBLS5 | **Keyboard line 5 Level Selection bit**<br>Cleared to enable a low level detection on Port line 5.<br>Set to enable a high level detection on Port line 5. |
| 4 | KBLS4 | **Keyboard line 4 Level Selection bit**<br>Cleared to enable a low level detection on Port line 4.<br>Set to enable a high level detection on Port line 4. |
| 3 | KBLS3 | **Keyboard line 3 Level Selection bit**<br>Cleared to enable a low level detection on Port line 3.<br>Set to enable a high level detection on Port line 3. |
| 2 | KBLS2 | **Keyboard line 2 Level Selection bit**<br>Cleared to enable a low level detection on Port line 2.<br>Set to enable a high level detection on Port line 2. |
| 1 | KBLS1 | **Keyboard line 1 Level Selection bit**<br>Cleared to enable a low level detection on Port line 1.<br>Set to enable a high level detection on Port line 1. |
| 0 | KBLS0 | **Keyboard line 0 Level Selection bit**<br>Cleared to enable a low level detection on Port line 0.<br>Set to enable a high level detection on Port line 0. |

Reset Value = 0000 0000b

**Table 84.** Status in Slave Transmitter Mode (Continued)

| Status Code (SSCS) | Status of the 2-wire bus and 2-wire hardware | Application Software Response: To/from SSDAT | Application Software Response: To SSCON STA | STO | SI | AA | Next Action Taken By 2-wire Software |
|---|---|---|---|---|---|---|---|
| C0h | Data byte in SSDAT has been transmitted; NOT ACK has been received | No SSDAT action or | 0 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA |
| | | No SSDAT action or | 0 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1 |
| | | No SSDAT action or | 1 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free |
| | | No SSDAT action | 1 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free |
| C8h | Last data byte in SSDAT has been transmitted (AA=0); ACK has been received | No SSDAT action or | 0 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA |
| | | No SSDAT action or | 0 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1 |
| | | No SSDAT action or | 1 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free |
| | | No SSDAT action | 1 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free |

**Table 85.** Miscellaneous Status

| Status Code (SSCS) | Status of the 2-wire bus and 2-wire hardware | Application Software Response: To/from SSDAT | Application Software Response: To SSCON STA | STO | SI | AA | Next Action Taken By 2-wire Software |
|---|---|---|---|---|---|---|---|
| F8h | No relevant state information available; SI= 0 | No SSDAT action | No SSCON action | | | | Wait or proceed current transfer |
| 00h | Bus error due to an illegal START or STOP condition | No SSDAT action | 0 | 1 | 0 | X | Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and STO is reset. |

**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

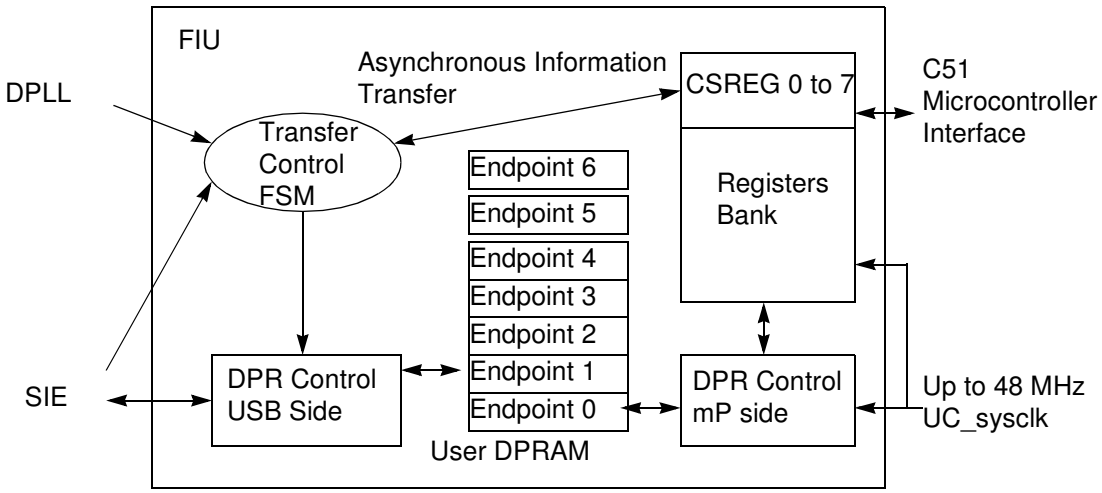
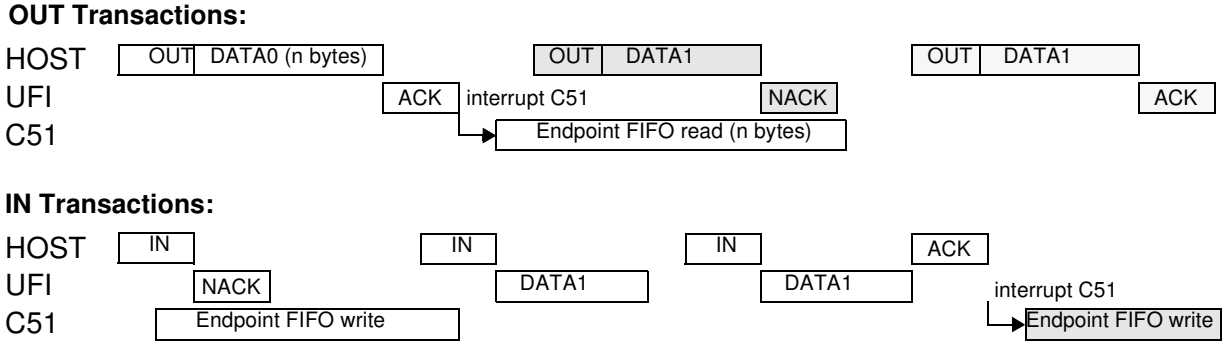



**Figure 59.** Minimum Intervention from the USB Device Firmware



- 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.

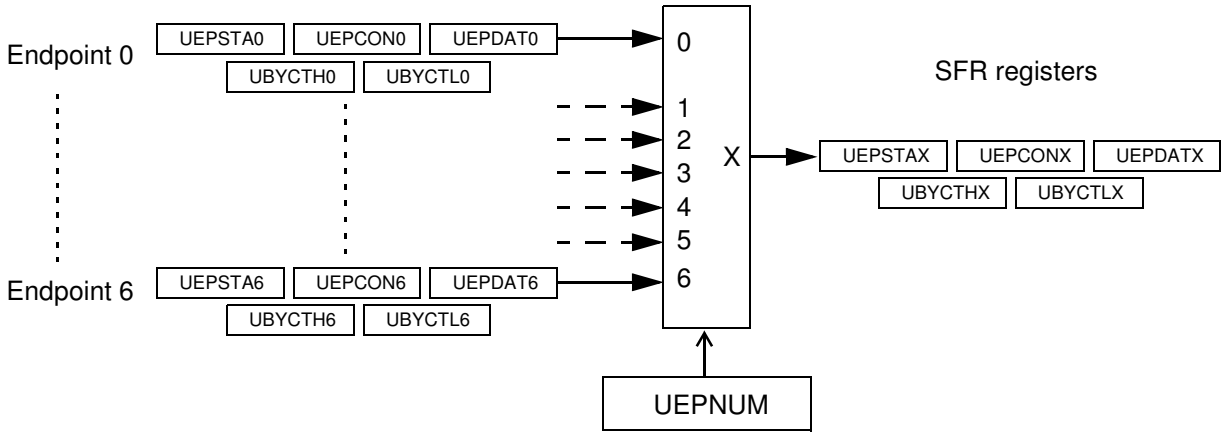
  Note that the endpoint reset doesn't reset the bank number for ping-pong endpoints.

## 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**

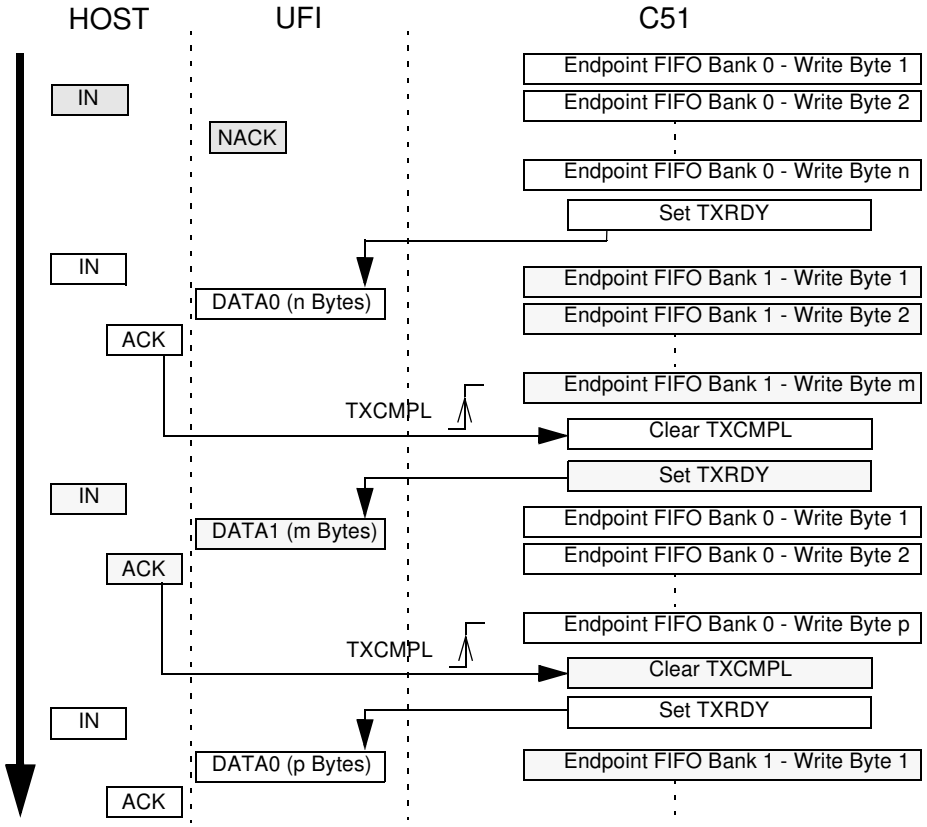
The write access for each IN endpoint is performed using the UEPDATX register.

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.

### Bulk/Interrupt IN Transactions in Ping-pong Mode

**Figure 65.** Bulk/Interrupt IN Transactions in Ping-pong Mode




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

The firmware will fill the FIFO bank 0 with the data to be sent and set the TXRDY bit in the UEPSTAX register to allow the USB controller to send the data stored in FIFO at the next IN request concerning the endpoint. The FIFO banks are automatically switched, and the firmware can immediately write into the endpoint FIFO bank 1.

When the IN packet concerning the bank 0 has been sent and acknowledged by the Host, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware will clear the TXCMPL bit before filling the endpoint FIFO bank 0 with new data. The FIFO banks are then automatically switched.

When the IN packet concerning the bank 1 has been sent and acknowledged by the Host, the TXCMPL bit is set by the USB controller. This triggers a USB interrupt if enabled. The firmware will clear the TXCMPL bit before filling the endpoint FIFO bank 1 with new data.

The bank switch is performed by the USB controller each time the TXRDY bit is set by the firmware. Until the TXRDY bit has been set by the firmware for an endpoint bank, the USB controller will answer a NAK handshake for each IN requests concerning this bank.

Note that in the example above, the firmware clears the Transmit Complete bit (TXCMPL) before setting the Transmit Ready bit (TXRDY). This is done in order to avoid the firmware to clear at the same time the TXCMPL bit for bank 0 and the bank 1.

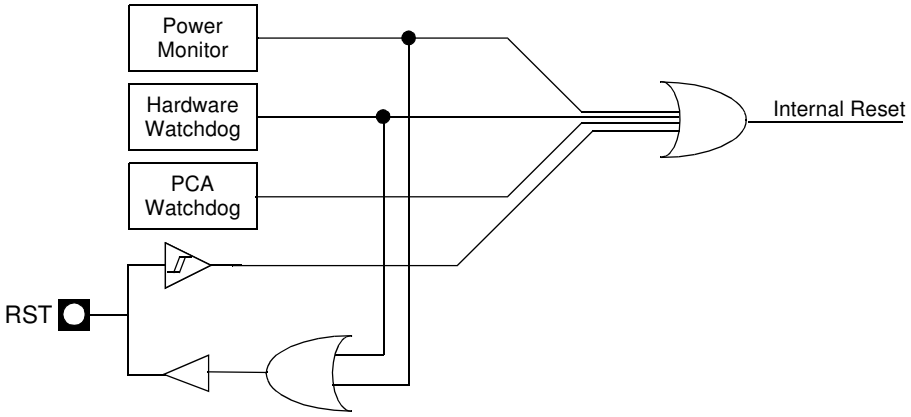
The firmware will never write more bytes than supported by the endpoint FIFO.

# Reset

## Introduction

The reset sources are: Power Management, Hardware Watchdog, PCA Watchdog and Reset input.

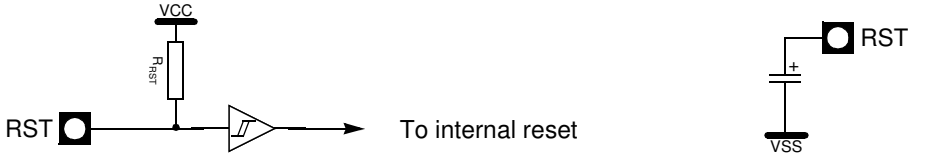
**Figure 72.** Reset schematic




## Reset Input

The Reset input can be used to force a reset pulse longer than the internal reset controlled by the Power Monitor. RST input has a pull-up resistor allowing power-on reset by simply connecting an external capacitor to $V_{SS}$ as shown in Figure 73. Resistor value and input characteristics are discussed in the Section “DC Characteristics” of the AT89C5131A-L datasheet.

**Figure 73.** Reset Circuitry and Power-On Reset




a. RST input circuitry

b. Power-on Reset

# Electrical Characteristics

## Absolute Maximum Ratings

Ambient Temperature Under Bias:

I = industrial .....-40°C to 85°C

Storage Temperature ..... -65°C to + 150°C

Voltage on $V_{CC}$ from $V_{SS}$ .....-0.5V to + 6V

Voltage on Any Pin from $V_{SS}$ .....-0.5V to $V_{CC}$ + 0.2V

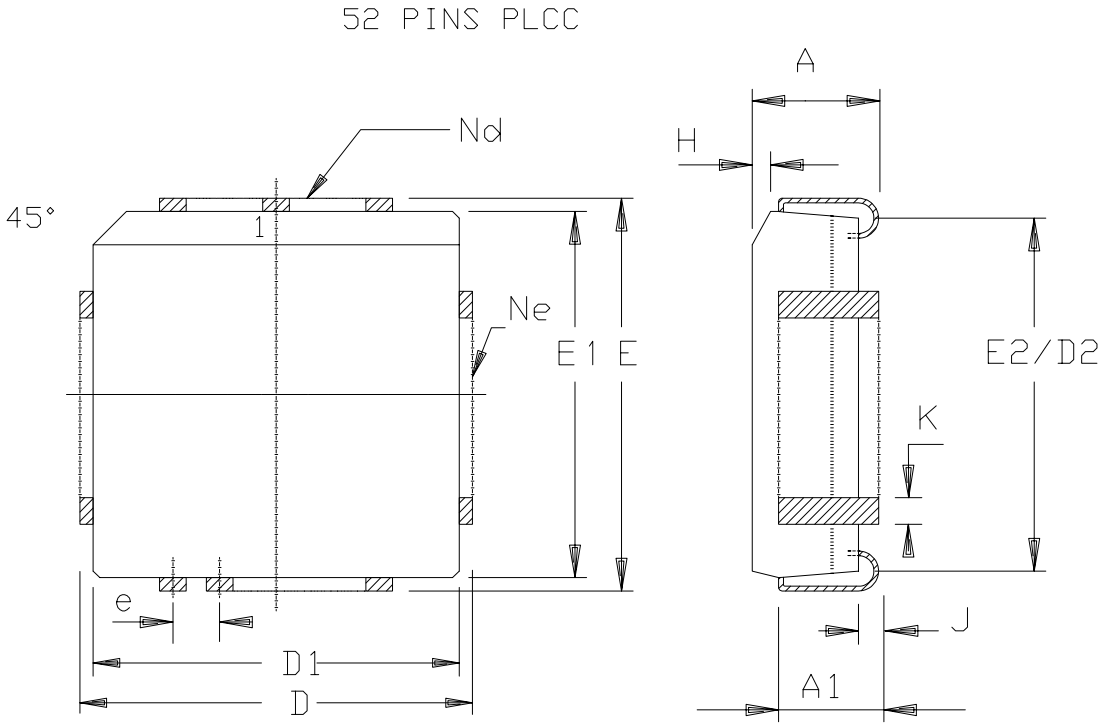
Note: Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

## DC Parameters

$T_A$ = -40°C to +85°C; $V_{SS}$ = 0V; $V_{CC}$ = 3.3V ± 10%; F = 0 to 40 MHz

| Symbol | Parameter | Min | Typ[5] | Max | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | -0.5 | | 0.2Vcc - 0.1 | V | |
| $V_{IH}$ | Input High Voltage except XTAL1, $\overline{RST}$ | 0.2 $V_{CC}$ + 0.9 | | $V_{CC}$ + 0.5 | V | |
| $V_{IH1}$ | Input High Voltage, XTAL1, $\overline{RST}$ | 0.7 $V_{CC}$ | | $V_{CC}$ + 0.5 | V | |
| $V_{OL}$ | Output Low Voltage, ports 1, 2, 3 and 4[6] | | | 0.3<br>0.45<br>1.0 | V<br>V<br>V | $I_{OL}$ = 100 µA[4]<br>$I_{OL}$ = 0.8 mA[4]<br>$I_{OL}$ = 1.6mA[4] |
| $V_{OL1}$ | Output Low Voltage, port 0, ALE, $\overline{PSEN}$ [6] | | | 0.3<br>0.45<br>1.0 | V<br>V<br>V | $I_{OL}$ = 200 µA[4]<br>$I_{OL}$ = 1.6 mA[4]<br>$I_{OL}$ = 3.5 mA[4] |
| $V_{OH}$ | Output High Voltage, ports 1, 2, 3, 4 and 5 | $V_{CC}$ - 0.3<br>$V_{CC}$ - 0.7<br>$V_{CC}$ - 1.5 | | | V<br>V<br>V | $I_{OH}$ = -10 µA<br>$I_{OH}$ = -30 µA<br>$I_{OH}$ = -60 µA<br>$V_{CC}$ = 3.3V ± 10% |
| $V_{OH1}$ | Output High Voltage, port 0, ALE, $\overline{PSEN}$ | $V_{CC}$ - 0.3<br>$V_{CC}$ - 0.7<br>$V_{CC}$ - 1.5 | | | V<br>V<br>V | $I_{OH}$ = -200 µA<br>$I_{OH}$ = -1.6 mA<br>$I_{OH}$ = -3.5 mA<br>$V_{CC}$ = 3.3V ± 10% |
| $R_{RST}$ | $\overline{RST}$ Pullup Resistor | 50 | 100 | 200 | kΩ | |
| $I_{IL}$ | Logical 0 Input Current ports 1, 2, 3 and 4 | | | -50 | µA | Vin = 0.45V |
| $I_{LI}$ | Input Leakage Current | | | ±10 | µA | 0.45V < Vin < $V_{CC}$ |
| $I_{TL}$ | Logical 1 to 0 Transition Current, ports 1, 2, 3 and 4 | | | -650 | µA | Vin = 2.0V |
| $C_{IO}$ | Capacitance of I/O Buffer | | | 10 | pF | Fc = 1 MHz<br>$T_A$ = 25°C |
| $I_{PD}$ | Power-down Current | | | 100 | µA | 3.0V < $V_{CC}$ < 3.6V[3] |
| $I_{CC}$ | Power Supply Current | $I_{CCOP}$ = 0.4xF(MHz)+5<br>$I_{CCIDLE}$ = 0.3xF(MHz)+5<br>$I_{CCwrite}$ = 0.8xF(MHz)+15 | | | | $V_{CC}$ = 3.3V [1][2] |
| $V_{PFDP}$ | Power Fail High Level Threshold | | | 3.0 | V | |

## 52-lead 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.

## Document Revision History

**Changes from 4338D - 09/05 to 4338E - 06/06**

1. Correction to Figure 4 on page 11.

**Changes from 4338E - 06/06 to 4338F - 08/07**

1. Hardware Conditions section Page 45 changed to recommend the use of 1K pull-up between PSEN and GND in ISP mode.
2. Updated 52-lead PLCC package.