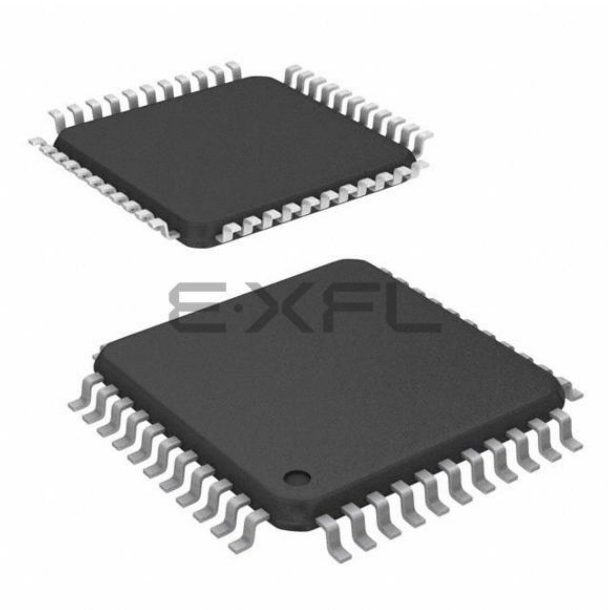


Welcome to E-XFL.COM

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| **Details** | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 80C51 |
| Core Size | 8-Bit |
| Speed | 60MHz |
| Connectivity | SPI, UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 32 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | 44-VQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at89c51rc2-rlrim |

The AT89C51RB2/RC2 retains all features of the 80C52 with 256 Bytes of internal RAM, a 9-source 4-level interrupt controller and three timer/counters.

In addition, the AT89C51RB2/RC2 has a Programmable Counter Array, an XRAM of 1024 Bytes, a Hardware Watchdog Timer, a Keyboard Interface, an SPI Interface, a more versatile serial channel that facilitates multiprocessor communication (EUART) and a speed improvement mechanism (X2 mode).

The Pinout is the standard 40/44 pins of the C52.

The fully static design reduces system power consumption of the AT89C51RB2/RC2 by allowing it to bring the clock frequency down to any value, even DC, without loss of data.

The AT89C51RB2/RC2 has 2 software-selectable modes of reduced activity and 8-bit clock prescaler for further reduction in power consumption. In Idle mode, the CPU is frozen while the peripherals and the interrupt system are still operating. In power-down mode, the RAM is saved and all other functions are inoperative.

The added features of the AT89C51RB2/RC2 make it more powerful for applications that need pulse width modulation, high speed I/O and counting capabilities such as alarms, motor control, corded phones, and smart card readers.

**Table 1.** Memory Size

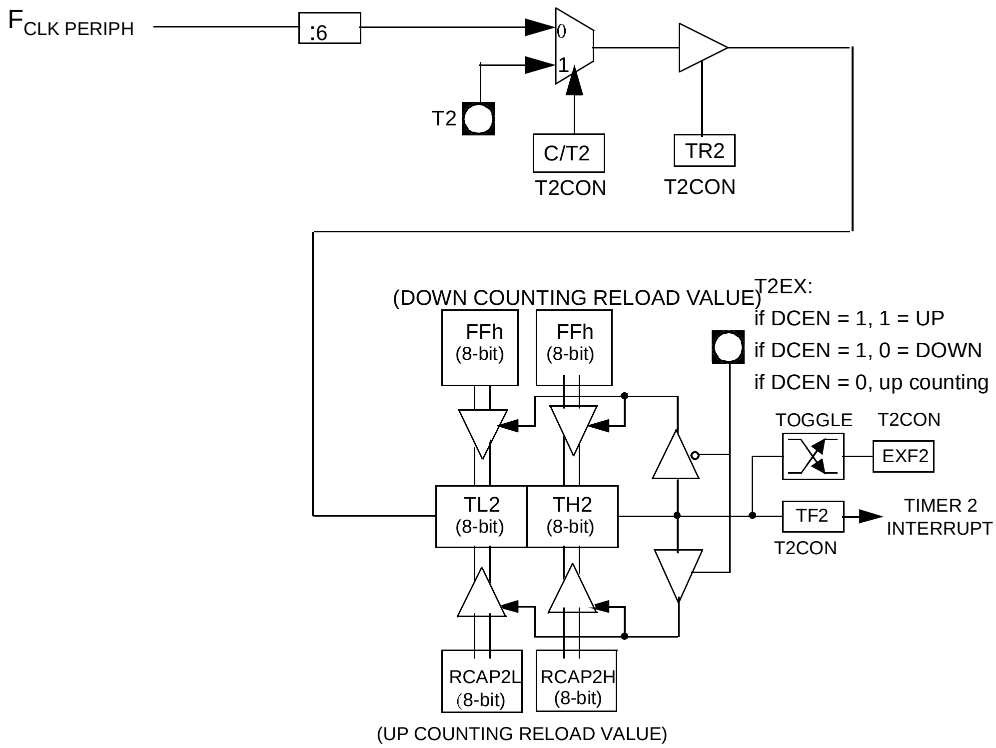
| Part Number | Flash (Bytes) | XRAM (Bytes) | TOTAL RAM (Bytes) | I/O |
|---|---|---|---|---|
| **AT89C51RB2** | 16K | 1024 | 1280 | 32 |
| **AT89C51RC2** | 32K | 1024 | 1280 | 32 |
| **AT89C51IC2** | 32K | 1024 | 1280 | 32 |

## SFR Mapping

The Special Function Registers (SFRs) of the AT89C51RB2/RC2 fall into the following categories:

- C51 core registers: ACC, B, DPH, DPL, PSW, SP
- I/O port registers: P0, P1, P2, P3
- Timer registers: T2CON, T2MOD, TCON, TH0, TH1, TH2, TMOD, TL0, TL1, TL2, RCAP2L, RCAP2H
- Serial I/O port registers: SADDR, SADEN, SBUF, SCON
- PCA (Programmable Counter Array) registers: CCON, CCAPMx, CL, CH, CCAPxH, CCAPxL (x: 0 to 4)
- Power and clock control registers: PCON
- Hardware Watchdog Timer registers: WDTRST, WDTPRG
- Interrupt system registers: IEN0, IPL0, IPH0, IEN1, IPL1, IPH1
- Keyboard Interface registers: KBE, KBF, KBLS
- SPI registers: SPCON, SPSTR, SPDAT
- BRG (Baud Rate Generator) registers: BRL, BDRCON
- Flash register: FCON
- Clock Prescaler register: CKRL
- Others: AUXR, AUXR1, CKCON0, CKCON1

**Figure 9.** Auto-Reload Mode Up/Down Counter (DCEN = 1)




## Programmable Clock-out Mode

In the clock-out mode, Timer 2 operates as a 50% duty-cycle, programmable clock generator (see Figure 10). The input clock increments TL2 at frequency $F_{CLK\ PERIPH}/2$. The timer repeatedly counts to overflow from a loaded value. At overflow, the contents of RCAP2H and RCAP2L registers are loaded into TH2 and TL2. In this mode, Timer 2 overflows do not generate interrupts. The formula gives the clock-out frequency as a function of the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$Clock\text{–}OutFrequency = \frac{F_{CLKPERIPH}}{4 \times (65536 - RCAP2H/RCAP2L)}$$

For a 16 MHz system clock, Timer 2 has a programmable frequency range of 61 Hz ($F_{CLK\ PERIPH}/2^{16}$) to 4 MHz ($F_{CLK\ PERIPH}/4$). The generated clock signal is brought out to T2 pin (P1.0).

Timer 2 is programmed for the clock-out mode as follows:

- Set T2OE bit in T2MOD register.
- Clear C/$\overline{T2}$ bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in RCAP2H/RCAP2L registers.
- Enter a 16-bit initial value in timer registers TH2/TL2. It can be the same as the reload value or a different one depending on the application.
- To start the timer, set TR2 run control bit in T2CON register.

It is possible to use Timer 2 as a baud rate generator and a clock generator simultaneously. For this configuration, the baud rates and clock frequencies are not independent since both functions use the values in the RCAP2H and RCAP2L registers.

**Figure 10.** Clock-Out Mode C/$\overline{T2}$ = 0

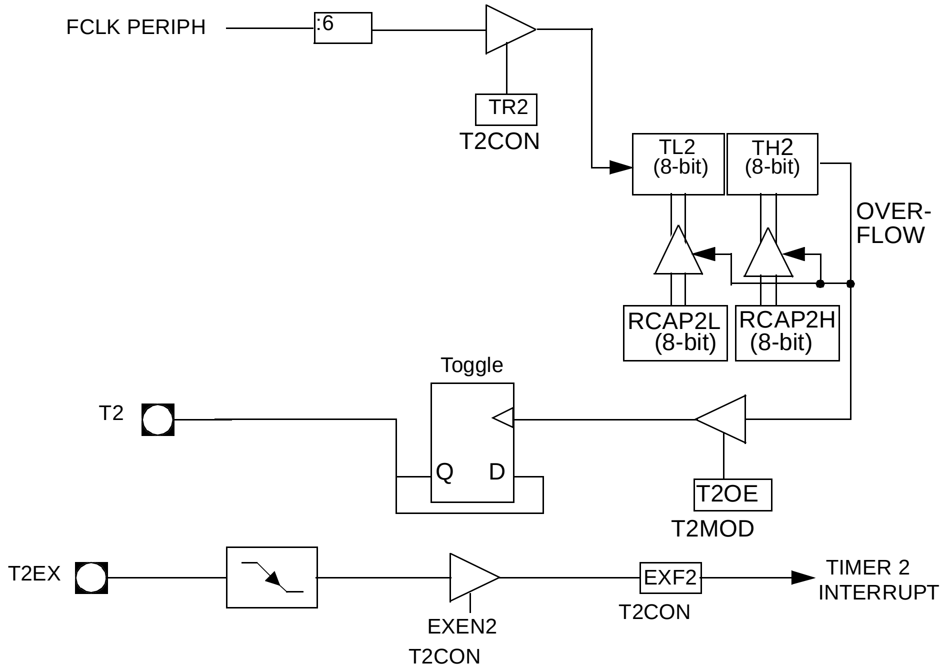


**Table 25.** PCA Module Modes (CCAPMn Registers)

| ECOMn | CAPPn | CAPNn | MATn | TOGn | PWMm | ECCFn | Module Function |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | No Operation |
| X | 1 | 0 | 0 | 0 | 0 | X | 16-bit capture by a positive-edge trigger on CEXn |
| X | 0 | 1 | 0 | 0 | 0 | X | 16-bit capture by a negative trigger on CEXn |
| X | 1 | 1 | 0 | 0 | 0 | X | 16-bit capture by a transition on CEXn |
| 1 | 0 | 0 | 1 | 0 | 0 | X | 16-bit Software Timer/Compare mode. |
| 1 | 0 | 0 | 1 | 1 | 0 | X | 16-bit High-speed Output |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 8-bit PWM |
| 1 | 0 | 0 | 1 | X | 0 | X | Watchdog Timer (Module 4 only) |

There are two additional registers associated with each of the PCA Modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a Module is used in the PWM mode these registers are used to control the duty cycle of the output (see Table 26 and Table 27).

**Table 26.** CCAPnH Registers (n = 0-4)

CCAP0H – PCA Module 0 Compare/Capture Control Register High (0FAh)

CCAP1H – PCA Module 1 Compare/Capture Control Register High (0FBh)

CCAP2H – PCA Module 2 Compare/Capture Control Register High (0FCh)

CCAP3H – PCA Module 3 Compare/Capture Control Register High (0FDh)

CCAP4H – PCA Module 4 Compare/Capture Control Register High (0FEh)

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

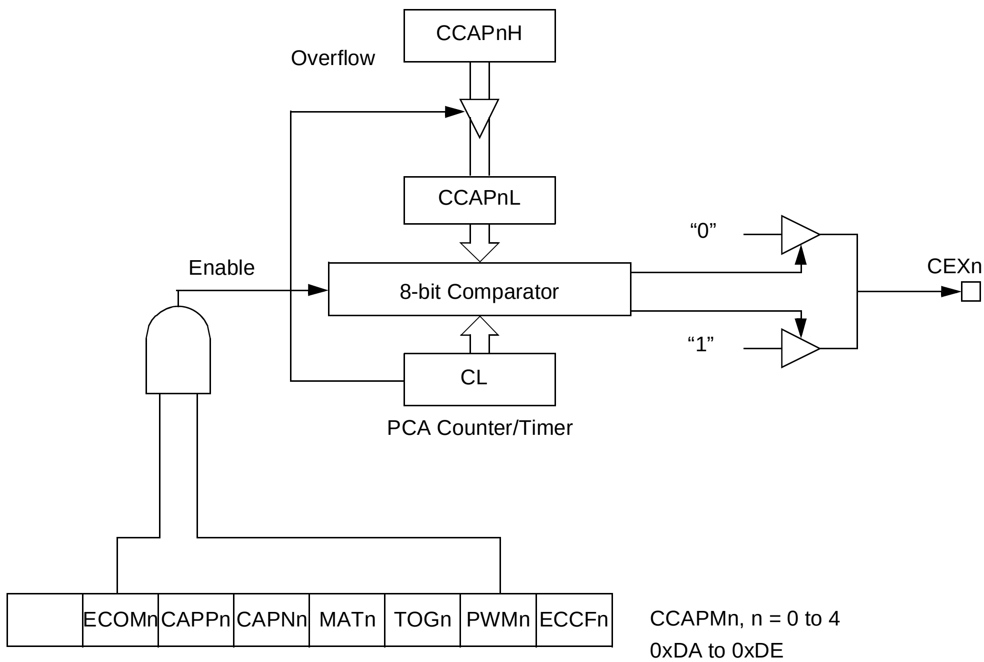
| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 - 0 | - | **PCA Module n Compare/Capture Control**<br>CCAPnH Value |

Reset Value = 0000 0000b
Not bit addressable

## Pulse Width Modulator Mode

All of the PCA Modules can be used as PWM outputs. Figure 16 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the Modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each Module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. This allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

**Figure 16.** PCA PWM Mode




## PCA Watchdog Timer

An on-board watchdog timer is available with the PCA to improve the reliability of the system without increasing chip count. Watchdog timers are useful for systems that are susceptible to noise, power glitches, or electrostatic discharge. Module 4 is the only PCA Module that can be programmed as a watchdog. However, this Module can still be used for other modes if the watchdog is not needed. Figure 14 shows a diagram of how the watchdog works. The user pre-loads a 16-bit value in the compare registers. Just like the other compare modes, this 16-bit value is compared to the PCA timer value. If a match is allowed to occur, an internal reset will be generated. This will not cause the RST pin to be driven high.

In order to hold off the reset, the user has the following three options:

1. Periodically change the compare value so it will never match the PCA timer.
2. Periodically change the PCA timer value so it will never match the compare values.
3. Disable the watchdog by clearing the WDTE bit before a match occurs and then re-enable it.

The first two options are more reliable because the watchdog timer is never disabled as in option #3. If the program counter ever goes astray, a match will eventually occur and cause an internal reset. The second option is also not recommended if other PCA Modules are being used. Remember, the PCA timer is the time base for all modules;

**Table 40.** T2CON Register

T2CON - Timer 2 Control Register (C8h)

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

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

Reset Value = 0000 0000b
Bit addressable

**Table 42.** BDRCON Register

BDRCON - Baud Rate Control Register (9Bh)

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

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

Reset Value = XXX0 0000b
Not bit addressablef

**Table 48.** IPL1 Register

IPL1 - Interrupt Priority Register (B2h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | SPIL | - | KBDL |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 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 | SPIL | **SPI Interrupt Priority Bit**<br>see SPIH for priority level. |
| 1 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 0 | KBDL | **Keyboard Interrupt Priority Bit**<br>see KBDH for priority level. |

Reset Value = XXXX X000b
Bit addressable

**Table 52.** 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 53.** KBLS Register

KBLS - Keyboard Level Selector Register (9Ch)

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

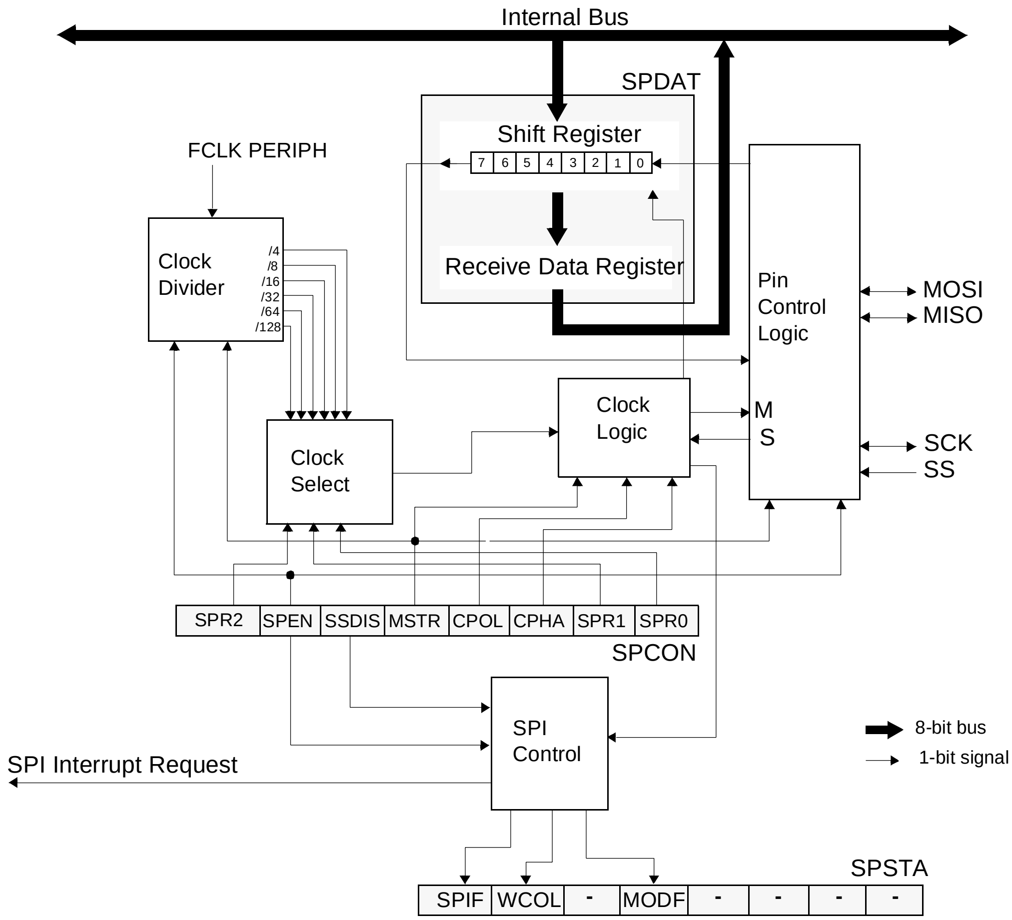
| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | KBLS7 | **Keyboard Line 7 Level Selection Bit**<br>Cleared to enable a low level detection on Port line 7.<br>Set to enable a high level detection on Port line 7. |
| 6 | KBLS6 | **Keyboard Line 6 Level Selection Bit**<br>Cleared to enable 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

## Functional Description

Figure 26 shows a detailed structure of the SPI Module.

**Figure 26.** SPI Module Block Diagram




## Operating Modes

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

- The Serial Peripheral Control register (SPCON)

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

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

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

When the Master device transmits data to the Slave device via the MOSI line, the Slave device responds by sending data to the Master device via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock (Figure 27).

## Flash EEPROM Memory

The Flash memory increases EPROM and ROM functionality with in-circuit electrical erasure and programming. It contains 16K or 32K Bytes of program memory organized in 128 or 256 pages of 128 Bytes. 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 external dedicated 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 ROM allows programming via the serial port without the need of a user-provided loader.
- Up to 64K Byte external program memory if the internal program memory is disabled (EA = 0).
- Programming and erase voltage with standard 5V or 3V $V_{CC}$ supply.
- Read/Programming/Erase:
  – Byte-wise read without wait state
  – Byte or page erase and programming (10 ms)
- Typical programming time (32K Bytes) in 10 s
- 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 16K or 32K Bytes 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:

- First, 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 UART.
- Second, 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 ROM.
- Third, the Flash may be programmed using the parallel method by using a conventional EPROM programmer. The parallel programming method used by these devices is similar to that used by EPROM 87C51 but it is not identical and the commercially available programmers need to have support for the AT89C51RB2/RC2. The bootloader and the Application Programming Interface (API) routines are located in the BOOT ROM.

**Table 66.** Program Lock Bits

| Program Lock Bits | | | | |
|---|---|---|---|---|
| **Security Level** | **LB0** | **LB1** | **LB2** | **Protection Description** |
| 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 internal memory, $\overline{EA}$ is sampled and latched on reset, and further parallel programming of the Flash 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. |
| 4 | X | X | P | Same as 3, also external execution is disabled. (Default) |

Note: U: unprogrammed or "one" level.
P: programmed or "zero" level.
X: don't care
WARNING: Security level '2' and '3' should only be programmed after Flash and code 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 first be done. 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: Programmed force ISP operation.
- X2: Unprogrammed to force X1 mode (Standard Mode).
- XRAM: Unprogrammed to valid XRAM
- 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.

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 67.

*Example*

Programming Data (write 55h at address 0010h in the Flash)

```
HOST          : 01 0010 00 55 9A
BOOTLOADER    : 01 0010 00 55 9A . CR LF
```

Programming Atmel function (write SSB to level 2)

```
HOST          : 02 0000 03 05 01 F5
BOOTLOADER    : 02 0000 03 05 01 F5. CR LF
```

Writing Frame (write BSB to 55h)

```
HOST          : 03 0000 03 06 00 55 9F
BOOTLOADER    : 03 0000 03 06 00 55 9F . CR LF
```

## ISP Commands Summary

**Table 73.** ISP Commands Summary

| Command | Command Name | Data[0] | Data[1] | Command Effect |
|---|---|---|---|---|
| 00h | Program Data | | | Program Nb Data Byte. Bootloader will accept up to 128 (80h) data Bytes. The data Bytes should be 128 Byte page Flash boundary. |
| 03h | Write Function | 01h | 00h | Erase block0 (0000h-1FFFh) |
| | | | 20h | Erase block1 (2000h-3FFFh) |
| | | | 40h | Erase block2 (4000h-7FFFh) |
| | | | 80h | Erase block3 (8000h- BFFFh) |
| | | | C0h | Erase block4 (C000h- FFFFh) |
| | | 03h | 00h | Hardware Reset |
| | | 04h | 00h | Erase SBV & BSB |
| | | 05h | 00h | Program SSB level 1 |
| | | | 01h | Program SSB level 2 |
| | | 06h | 00h | Program BSB (value to write in data[2]) |
| | | | 01h | Program SBV (value to write in data[2]) |
| | | 07h | - | Full Chip Erase (This command needs about 6 sec to be executed) |
| | | 0Ah | 02h | Program Osc fuse (value to write in data[2]) |
| | | | 04h | Program BLJB fuse (value to write in data[2]) |
| | | | 08h | Program X2 fuse (value to write in data[2]) |
| 04h | Display Function | Data[0:1] = start address<br>Data [2:3] = end address<br>Data[4] = 00h -> Display data<br>Data[4] = 01h -> Blank check | | Display Data<br>Note: The maximum number of data that can be read with a single command frame (difference between start and end address) is 1kbyte. |
| | | | | Blank Check |
| 05h | Read Function | 00h | 00h | Manufacturer ID |
| | | | 01h | Device ID #1 |
| | | | 02h | Device ID #2 |
| | | | 03h | Device ID #3 |
| | | 07h | 00h | Read SSB |
| | | | 01h | Read BSB |
| | | | 02h | Read SBV |
| | | | 06h | Read Extra Byte |
| | | 0Bh | 00h | Read Hardware Byte |
| | | 0Eh | 00h | Read Device Boot ID1 |
| | | | 01h | Read Device Boot ID2 |
| | | 0Fh | 00h | Read Bootloader Version |

**Table 74.** API Call Summary (Continued)

| Command | R1 | A | DPTR0 | DPTR1 | Returned Value | Command Effect |
|---|---|---|---|---|---|---|
| PROGRAM X2 FUSE | 0Ah | Fuse value 00h or 01h | 0008h | XXh | none | Program X2 fuse bit with ACC |
| PROGRAM BLJB FUSE | 0Ah | Fuse value 00h or 01h | 0004h | XXh | none | Program BLJB fuse bit with ACC |
| READ HSB | 0Bh | XXh | XXXXh | XXh | ACC = HSB | Read Hardware Byte |
| READ BOOT ID1 | 0Eh | XXh | DPL = 00h | XXh | ACC = ID1 | Read boot ID1 |
| READ BOOT ID2 | 0Eh | XXh | DPL = 01h | XXh | ACC = ID2 | Read boot ID2 |
| READ BOOT VERSION | 0Fh | XXh | XXXXh | XXh | ACC = Boot_Version | Read bootloader version |

**Table 76.** AC Parameters for a Fix Clock

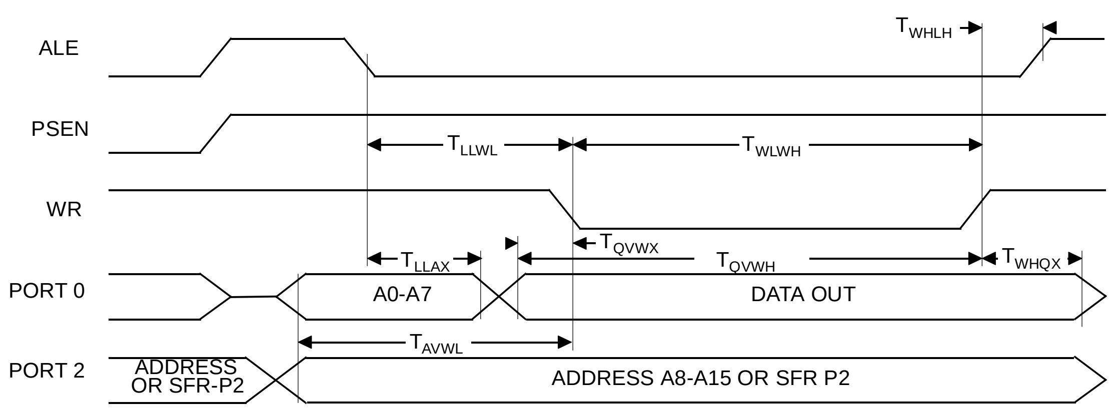
| Symbol | -M | | -L | | Units |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| T | 25 | | 25 | | ns |
| $T_{LHLL}$ | 35 | | 35 | | ns |
| $T_{AVLL}$ | 5 | | 5 | | ns |
| $T_{LLAX}$ | 5 | | 5 | | ns |
| $T_{LLIV}$ | | n 65 | | 65 | ns |
| $T_{LLPL}$ | 5 | | 5 | | ns |
| $T_{PLPH}$ | 50 | | 50 | | ns |
| $T_{PLIV}$ | | 30 | | 30 | ns |
| $T_{PXIX}$ | 0 | | 0 | | ns |
| $T_{PXIZ}$ | | 10 | | 10 | ns |
| $T_{AVIV}$ | | 80 | | 80 | ns |
| $T_{PLAZ}$ | | 10 | | 10 | ns |

**Table 77.** AC Parameters for a Variable Clock

| Symbol | Type | Standard Clock | X2 Clock | X Parameter for -M Range | X Parameter for -L Range | Units |
|---|---|---|---|---|---|---|
| $T_{LHLL}$ | Min | 2 T - x | T - x | 15 | 15 | ns |
| $T_{AVLL}$ | Min | T - x | 0.5 T - x | 20 | 20 | ns |
| $T_{LLAX}$ | Min | T - x | 0.5 T - x | 20 | 20 | ns |
| $T_{LLIV}$ | Max | 4 T - x | 2 T - x | 35 | 35 | ns |
| $T_{LLPL}$ | Min | T - x | 0.5 T - x | 15 | 15 | ns |
| $T_{PLPH}$ | Min | 3 T - x | 1.5 T - x | 25 | 25 | ns |
| $T_{PLIV}$ | Max | 3 T - x | 1.5 T - x | 45 | 45 | ns |
| $T_{PXIX}$ | Min | x | x | 0 | 0 | ns |
| $T_{PXIZ}$ | Max | T - x | 0.5 T - x | 15 | 15 | ns |
| $T_{AVIV}$ | Max | 5 T - x | 2.5 T - x | 45 | 45 | ns |
| $T_{PLAZ}$ | Max | x | x | 10 | 10 | ns |

| Symbol | Type | Standard Clock | X2 Clock | X Parameter for - M Range | X Parameter for - L Range | Units |
|---|---|---|---|---|---|---|
| $T_{RLRH}$ | Min | 6 T - x | 3 T - x | 25 | 25 | ns |
| $T_{WLWH}$ | Min | 6 T - x | 3 T - x | 25 | 25 | ns |
| $T_{RLDV}$ | Max | 5 T - x | 2.5 T - x | 30 | 30 | ns |
| $T_{RHDX}$ | Min | x | x | 0 | 0 | ns |
| $T_{RHDZ}$ | Max | 2 T - x | T - x | 25 | 25 | ns |
| $T_{LLDV}$ | Max | 8 T - x | 4T -x | 45 | 45 | ns |
| $T_{AVDV}$ | Max | 9 T - x | 4.5 T - x | 65 | 65 | ns |
| $T_{LLWL}$ | Min | 3 T - x | 1.5 T - x | 30 | 30 | ns |
| $T_{LLWL}$ | Max | 3 T + x | 1.5 T + x | 30 | 30 | ns |
| $T_{AVWL}$ | Min | 4 T - x | 2 T - x | 30 | 30 | ns |
| $T_{QVWX}$ | Min | T - x | 0.5 T - x | 20 | 20 | ns |
| $T_{QVWH}$ | Min | 7 T - x | 3.5 T - x | 20 | 20 | ns |
| $T_{WHQX}$ | Min | T - x | 0.5 T - x | 15 | 15 | ns |
| $T_{RLAZ}$ | Max | x | x | 0 | 0 | ns |
| $T_{WHLH}$ | Min | T - x | 0.5 T - x | 20 | 20 | ns |
| $T_{WHLH}$ | Max | T + x | 0.5 T + x | 20 | 20 | ns |

## External Data Memory Write Cycle



## Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

## Regional Headquarters

*Europe*
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

*Asia*
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

*Japan*
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

## Atmel Operations

*Memory*
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

*Microcontrollers*
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

*ASIC/ASSP/Smart Cards*
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

*RF/Automotive*
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

*Biometrics/Imaging/Hi-Rel MPU/*
*High Speed Converters/RF Datacom*
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80


*Literature Requests*
www.atmel.com/literature




xxxxA–8051–10/06