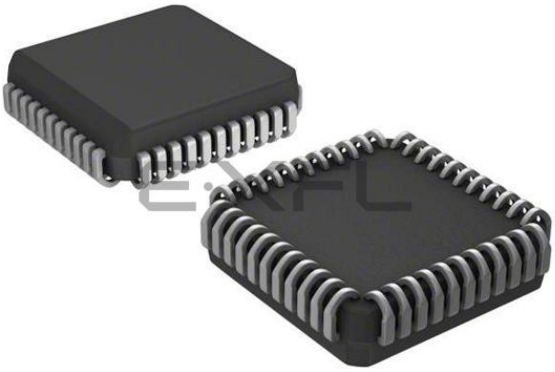


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	80C51
Core Size	8-Bit
Speed	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-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.6x16.6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51rc2-slsum

Table 12. Pin Description for 40 - 44 Pin Packages

	Pin Number				
Mnemonic	**DIL**	**LCC**	**VQFP44 1.4**	**Type**	**Name and Function**
V_{SS}	20	22	16	I	**Ground**: 0V reference
V_{CC}	40	44	38	I	**Power Supply**: This is the power supply voltage for normal, idle and power-down operation
P0.0 - P0.7	39 - 32	43 - 36	37 - 30	I/O	**Port 0**: Port 0 is an open-drain, bi-directional I/O port. Port 0 pins that have 1s written to them float and can be used as high impedance inputs. Port 0 must be polarized to V_{CC} or V_{SS} in order to prevent any parasitic current consumption. Port 0 is also the multiplexed low-order address and data bus during access to external program and data memory. In this application, it uses strong internal pull-up when emitting 1s. Port 0 also inputs the code Bytes during Flash programming. External pull-ups are required during program verification during which P0 outputs the code Bytes.
P1.0 - P1.7	1 - 8	2 - 9	40 - 44 1 - 3	I/O	**Port 1**: Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current because of the internal pull-ups. Port 1 also receives the low-order address Byte during memory programming and verification. Alternate functions for AT89C51RB2/RC2 Port 1 include:
	1	2	40	I/O	**P1.0**: Input/Output
				I/O	**T2 (P1.0):** Timer/Counter 2 external count input/Clockout
	2	3	41	I/O	**P1.1:** Input/Output
				I	**T2EX:** Timer/Counter 2 Reload/Capture/Direction Control
				I	**$\overline{\text{SS}}$:** SPI Slave Select
	3	4	42	I/O	**P1.2:** Input/Output
				I	**ECI:** External Clock for the PCA
	4	5	43	I/O	**P1.3:** Input/Output
				I/O	**CEX0:** Capture/Compare External I/O for PCA Module 0
	5	6	44	I/O	**P1.4:** Input/Output
				I/O	**CEX1:** Capture/Compare External I/O for PCA Module 1
	6	7	1	I/O	**P1.5:** Input/Output
				I/O	**CEX2:** Capture/Compare External I/O for PCA Module 2
				I/O	**MISO:** SPI Master Input Slave Output line
					When SPI is in master mode, MISO receives data from the slave peripheral. When SPI is in slave mode, MISO outputs data to the master controller.
	7	8	2	I/O	**P1.6**: Input/Output
				I/O	**CEX3:** Capture/Compare External I/O for PCA Module 3
				I/O	**SCK:** SPI Serial Clock
					SCK outputs clock to the slave peripheral
	8	9	3	I/O	**P1.7**: Input/Output:

Table 12. Pin Description for 40 - 44 Pin Packages (Continued)

	Pin Number				
Mnemonic	**DIL**	**LCC**	**VQFP44 1.4**	**Type**	**Name and Function**
				I/O	**CEX4:** Capture/Compare External I/O for PCA Module 4
P1.0 - P1.7				I/O	**MOSI**: SPI Master Output Slave Input line When SPI is in master mode, MOSI outputs data to the slave peripheral. When SPI is in slave mode, MOSI receives data from the master controller.
XTAL1	19	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier
P2.0 - P2.7	21 - 28	24 - 31	18 - 25	I/O	**Port 2**: Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current because of the internal pull-ups. Port 2 emits the high - order address Byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @Ri), port 2 emits the contents of the P2 SFR. Some Port 2 pins receive the high order address bits during EPROM programming and verification: P2.0 to P2.5 for 16 KB devices P2.0 to P2.6 for 32KB devices
P3.0 - P3.7	10 - 17	11, 13 - 19	5, 7 - 13	I/O	**Port 3:** Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally pulled low will source current because of the internal pull-ups. Port 3 also serves the special features of the 80C51 family, as listed below.
	10	11	5	I	**RXD (P3.0):** Serial input port
	11	13	7	O	**TXD (P3.1):** Serial output port
	12	14	8	I	$\overline{\textbf{INT0}}$ **(P3.2):** External interrupt 0
	13	15	9	I	$\overline{\textbf{INT1}}$ **(P3.3):** External interrupt 1
	14	16	10	I	**T0 (P3.4):** Timer 0 external input
	15	17	11	I	**T1 (P3.5):** Timer 1 external input
	16	18	12	O	$\overline{\textbf{WR}}$ **(P3.6):** External data memory write strobe
	17	19	13	O	$\overline{\textbf{RD}}$ **(P3.7):** External data memory read strobe
RST	9	10	4	I/O	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V_{SS} permits a power-on reset using only an external capacitor to V_{CC}. This pin is an output when the hardware watchdog forces a system reset.
ALE/$\overline{\text{PROG}}$	30	33	27	O (I)	**Address Latch Enable/Program Pulse:** Output pulse for latching the low Byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 (1/3 in X2 mode) the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming. ALE can be disabled by setting SFR's AUXR. 0 bit. With this bit set, ALE will be inactive during internal fetches.

Port Types

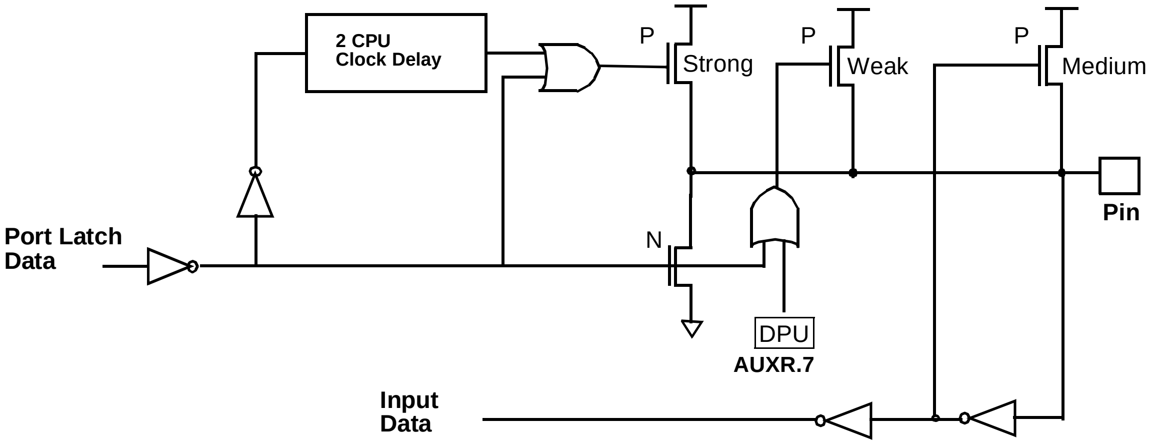
AT89C51RB2/RC2 I/O ports (P1, P2, P3) implement the quasi-bidirectional output that is common on the 80C51 and most of its derivatives. This output type can be used as both an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is pulled low, it is driven strongly and able to sink a fairly large current. These features are somewhat similar to an open drain output except that there are three pull-up transistors in the quasi-bidirectional output that serve different purposes. One of these pull-ups, called the "weak" pull-up, is turned on whenever the port latch for the pin contains a logic 1. The weak pull-up sources a very small current that will pull the pin high if it is left floating. A second pull-up, called the "medium" pull-up, is turned on when the port latch for the pin contains a logic 1 and the pin itself is also at a logic 1 level. This pull-up provides the primary source current for a quasi-bidirectional pin that is outputting a 1. If a pin that has a logic 1 on it is pulled low by an external device, the medium pull-up turns off, and only the weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to overpower the medium pull-up and take the voltage on the port pin below its input threshold.

The third pull-up is referred to as the "strong" pull-up. This pull-up is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port latch changes from a logic 0 to a logic 1. When this occurs, the strong pull-up turns on for a brief time, two CPU clocks, in order to pull the port pin high quickly. Then it turns off again.

The DPU bit (bit 7 in AUXR register) allows to disable the permanent weak pull up of all ports when latch data is logical 0.

The quasi-bidirectional port configuration is shown in Figure 3.

Figure 3. Quasi-Bidirectional Output

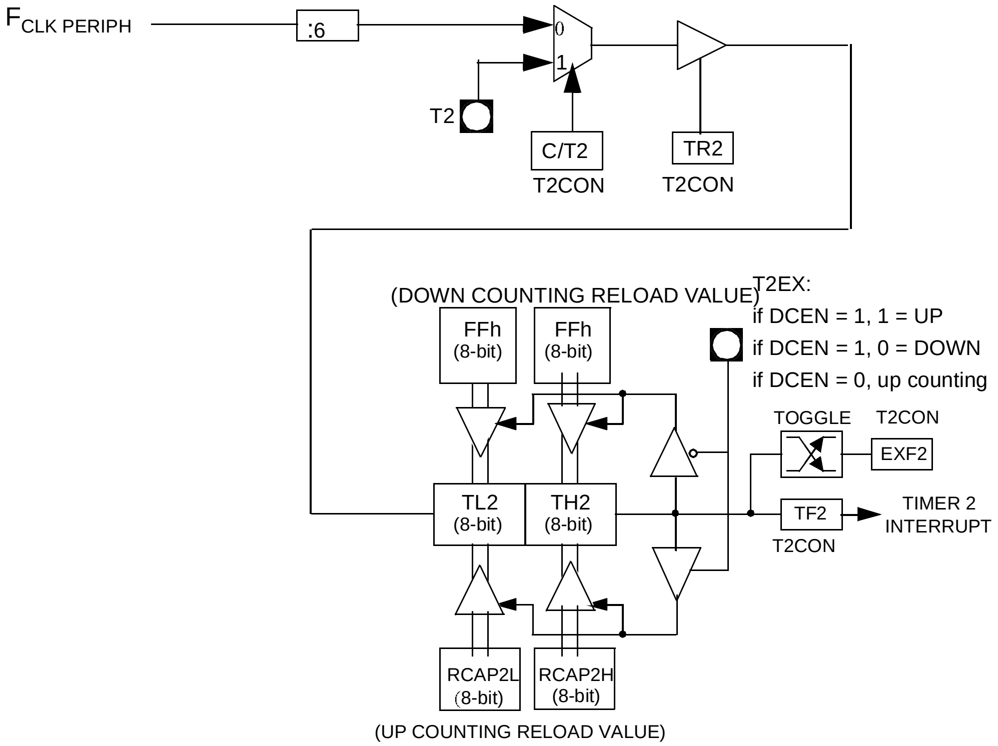


- Instructions that use indirect addressing access the Upper 128 Bytes of data RAM. For example: MOV @R0, # data where R0 contains 0A0h, accesses the data Byte at address 0A0h, rather than P2 (whose address is 0A0h).
- The XRAM Bytes can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory that is physically located on-chip, logically occupies the first Bytes of external data memory. The bits XRS0 and XRS1 are used to hide a part of the available XRAM as explained in Table 18. This can be useful if external peripherals are mapped at addresses already used by the internal XRAM.
- With EXTRAM = 0, the XRAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to XRAM will not affect ports P0, P2, P3.6 (WR) and P3.7 (RD). For example, with EXTRAM = 0, MOVX @R0, # data where R0 contains 0A0H, accesses the XRAM at address 0A0H rather than external memory. An access to external data memory locations higher than the accessible size of the XRAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, with P0 and P2 as data/address busses, and P3.6 and P3.7 as write and read timing signals. Accesses to XRAM above 0FFH can only be done by the use of DPTR.
- With EXTRAM = 1, MOVX @RI and MOVX @DPTR will be similar to the standard 80C51. MOVX @ Ri will provide an eight-bit address multiplexed with data on Port0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX @ RI and MOVX @DPTR will generate either read or write signals on P3.6 (WR) and P3.7 (RD).

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

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

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/$\overline{\text{T2}}$ 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.

Registers

Table 20. T2CON Register

T2CON – Timer 2 Control Register (C8h)

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

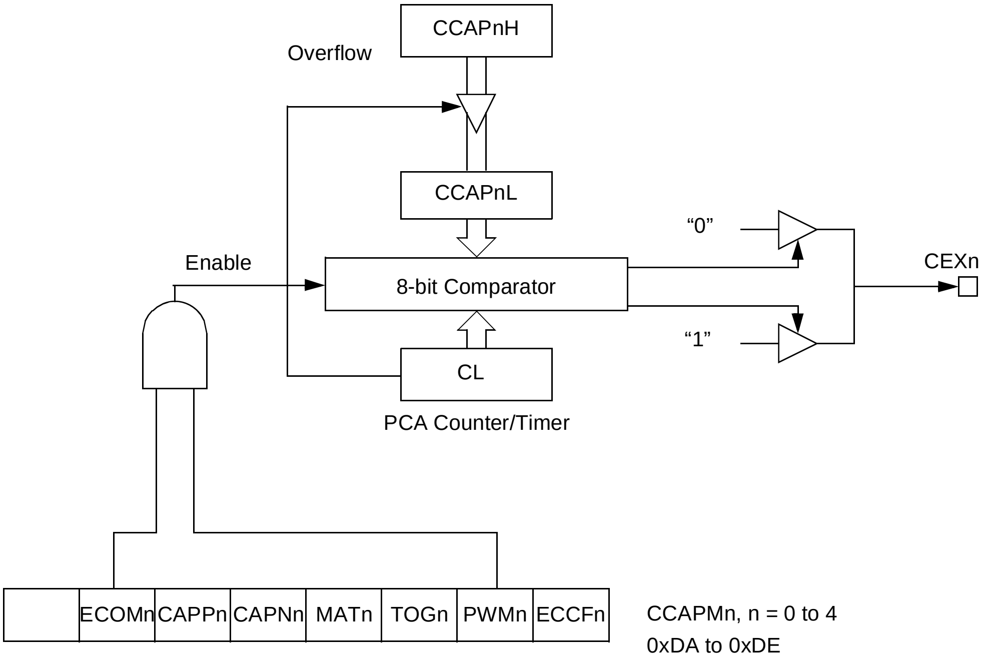
Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 Overflow Flag** Must be cleared by software. Set by hardware on Timer 2 overflow, if RCLK = 0 and TCLK = 0.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2 = 1. When set, causes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

Reset Value = 0000 0000b
Bit addressable

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 41. PCON Register

PCON - Power Control Register (87h)

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

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

Reset Value = 00X1 0000b
Not bit addressable

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

Table 47. IEN1 Register

IEN1 - Interrupt Enable Register (B1h)

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

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

Reset Value = XXXX X000b
Bit addressable

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** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	SPIL	**SPI Interrupt Priority Bit** see SPIH for priority level.
1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	KBDL	**Keyboard Interrupt Priority Bit** see KBDH for priority level.

Reset Value = XXXX X000b
Bit addressable

Interrupt Sources and Vector Addresses

Table 50. Interrupt Sources and Vector Addresses

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	SPI	SPIIT	004Bh

Figure 28. Data Transmission Format (CPHA = 0)

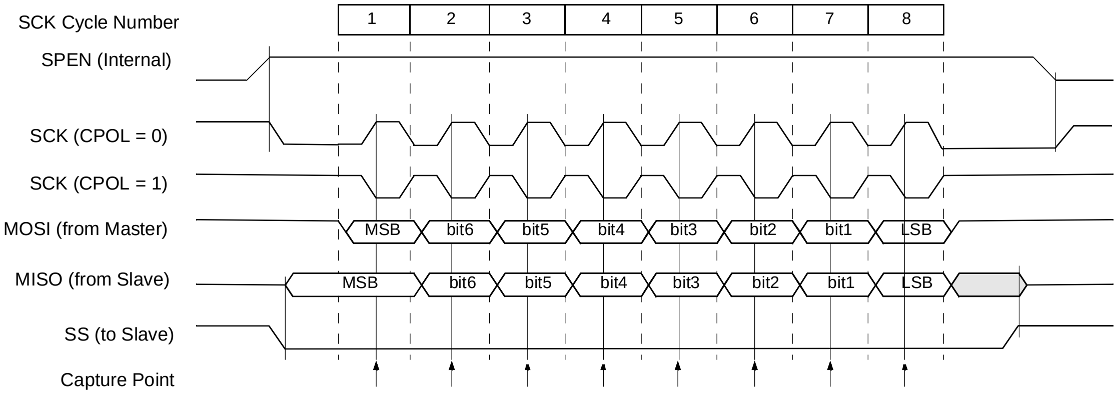


Figure 29. Data Transmission Format (CPHA = 1)

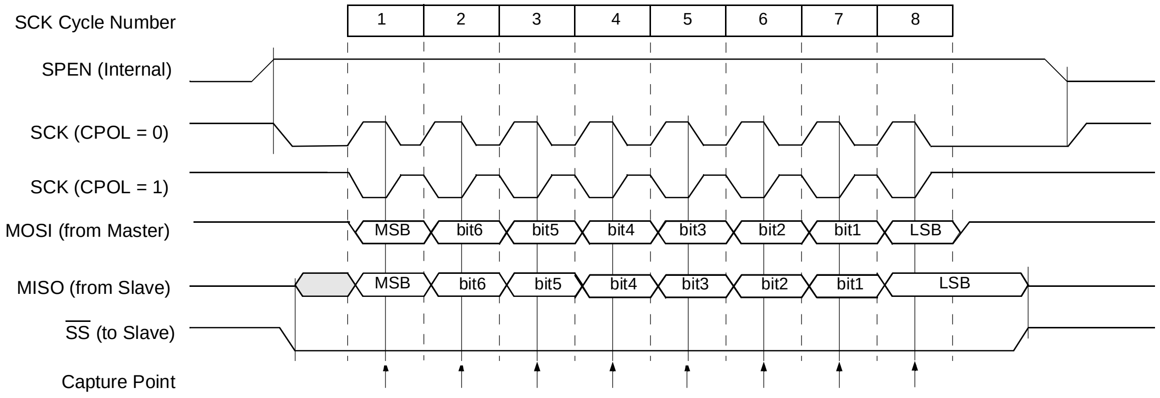
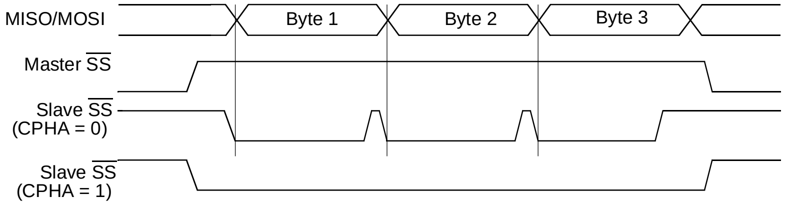


Figure 30. CPHA/$\overline{SS}$ Timing



As shown in Figure 28, the first SCK edge is the MSB capture strobe. Therefore, the Slave must begin driving its data before the first SCK edge, and a falling edge on the $\overline{SS}$ pin is used to start the transmission. The $\overline{SS}$ pin must be toggled high and then low between each Byte transmitted (Figure 30).

Figure 29 shows an SPI transmission in which CPHA is '1'. In this case, the Master begins driving its MOSI pin on the first SCK edge. Therefore, the Slave uses the first SCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions (Figure 30). This format may be preffered in systems having only one Master and only one Slave driving the MISO data line.

Bit Number	Bit Mnemonic	Description
1	SPR1	**SPR2 SPR1 SPR0 Serial Peripheral Rate** 0 0 0 $F_{CLK\ PERIPH}$ /2 0 0 1 $F_{CLK\ PERIPH}$ /4 0 1 0 $F_{CLK\ PERIPH}$ /8 0 1 1 $F_{CLK\ PERIPH}$ /16 1 0 0 $F_{CLK\ PERIPH}$ /32 1 0 1 $F_{CLK\ PERIPH}$ /64 1 1 0 $F_{CLK\ PERIPH}$ /128 1 1 1 Invalid
0	SPR0	

Reset Value = 0001 0100b

Not bit addressable

Serial Peripheral Status Register (SPSTA)

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

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

Table 57 describes the SPSTA register and explains the use of every bit in the register.

Table 57. SPSTA Register

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

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

Bit Number	Bit Mnemonic	Description
7	SPIF	**Serial Peripheral Data Transfer Flag** Cleared by hardware to indicate data transfer is in progress or has been approved by a clearing sequence. Set by hardware to indicate that the data transfer has been completed.
6	WCOL	**Write Collision Flag** Cleared by hardware to indicate that no collision has occurred or has been approved by a clearing sequence. Set by hardware to indicate that a collision has been detected.
5	SSERR	**Synchronous Serial Slave Error Flag** Set by hardware when $\overline{SS}$ is deasserted before the end of a received data. Cleared by disabling the SPI (clearing SPEN bit in SPCON).
4	MODF	**Mode Fault** Cleared by hardware to indicate that the $\overline{SS}$ pin is at appropriate logic level, or has been approved by a clearing sequence. Set by hardware to indicate that the $\overline{SS}$ pin is at inappropriate logic level.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Bootloader Functionality

Introduction

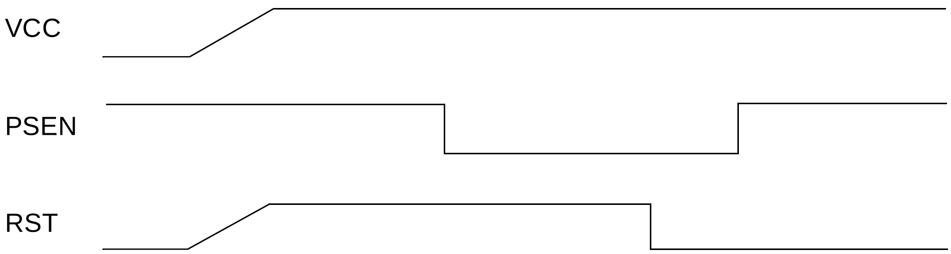
The bootloader can be activated by two means: Hardware conditions or regular boot process.

The Hardware conditions (EA = 1, PSEN = 0) during the Reset# falling edge force the on-chip bootloader execution. This allows an application to be built that will normally execute the end user’s code but can be manually forced into default ISP operation.

As PSEN is an output port in normal operating mode (running user application or boor-loader code) after reset, it is recommended to release PSEN after falling edge of reset signal. The hardware conditions are sampled at reset signal falling edge, thus they can be released at any time when reset input is low.

To ensure correct microcontroller startup, the PSEN pin should not be tied to ground during power-on (See Figure 38).

Figure 38. Hardware conditions typical sequence during power-on.



The on-chip bootloader boot process is shown in Figure 39.

	Purpose
Hardware Conditions	The Hardware Conditions force the bootloader execution whatever BLJB, BSB and SBV values.
BLJB	The Boot Loader Jump Bit forces the application execution. BLJB = 0 => Boot loader execution. BLJB = 1 => Application execution. The BLJB is a fuse bit in the Hardware Byte. That can be modified by hardware (programmer) or by software (API). Note: The BLJB test is perform by hardware to prevent any program execution.
SBV	The Software Boot Vector contains the high address of custumer bootloader stored in the application. SBV = FCh (default value) if no custumer bootloader in user Flash. Note: The costumer bootloader is called by JMP [SBV]00h instruction.

Full Chip Erase

The ISP command "Full Chip Erase" erases all User Flash memory (fills with FFh) and sets some Bytes used by the bootloader at their default values:

- BSB = FFh
- SBV = FCh
- SSB = FFh and finally erase the Software Security Bits

The Full Chip Erase does not affect the bootloader.

Checksum Error

When a checksum error is detected send ‘X’ followed with CR&LF.

Flow Description

Overview

An initialization step must be performed after each Reset. After microcontroller reset, the bootloader waits for an autobaud sequence (see section ‘autobaud performance’).

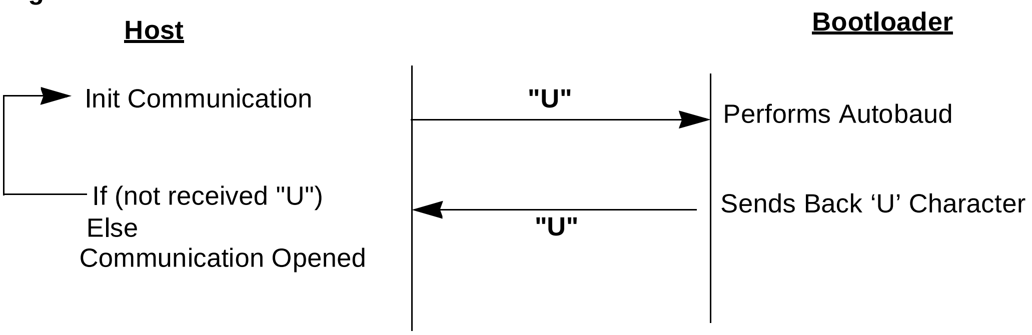
When the communication is initialized the protocol depends on the record type requested by the host.

FLIP, a software utility to implement ISP programming with a PC, is available from the Atmel the web site.

Communication Initialization

The host initializes the communication by sending a ’U’ character to help the bootloader to compute the baudrate (autobaud).

Figure 40. Initialization



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
		Data [2:3] = end address
		Data[4] = 00h -> Display data
		Data[4] = 01h -> Blank check
				Display Data
				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 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

Table 79. AC Parameters for a Fix Clock

Symbol	-M		-L		Units
	Min	Max	Min	Max	
T_{RLRH}	125		125		ns
T_{WLWH}	125		125		ns
T_{RLDV}		95		95	ns
T_{RHDX}	0		0		ns
T_{RHDZ}		25		25	ns
T_{LLDV}		155		155	ns
T_{AVDV}		160		160	ns
T_{LLWL}	45	105	45	105	ns
T_{AVWL}	70		70		ns
T_{QVWX}	5		5		ns
T_{QVWH}	155		155		ns
T_{WHQX}	10		10		ns
T_{RLAZ}	0		0		ns
T_{WHLH}	5	45	5	45	ns

Table of Contents

Features ***1***

Description ***1***

Block Diagram ***3***

SFR Mapping ***4***

Pin Configurations ***9***

Port Types ***13***

Oscillator ***14***

Registers 14
Functional Block Diagram 15

Enhanced Features ***16***

X2 Feature... 16

Dual Data Pointer Register (DPTR) ***20***

Expanded RAM (XRAM) ***23***

Registers 25

Timer 2 .... ***26***

Auto-reload Mode... 26
Programmable Clock-out Mode.. 27
Registers 29

Programmable Counter Array (PCA) ***31***

Registers 33
PCA Capture Mode 39
16-bit Software Timer/ Compare Mode 40
High-speed Output Mode 41
Pulse Width Modulator Mode 42
PCA Watchdog Timer.. 42

Serial I/O Port ***44***

Framing Error Detection 44
Automatic Address Recognition 45

Registers.. 47
Baud Rate Selection for UART for Mode 1 and 3.. 47
UART Registers.. 50

Interrupt System *55*
Registers.. 56
Interrupt Sources and Vector Addresses.. 63

Keyboard Interface *64*
Registers.. 65

Serial Port Interface (SPI) *68*
Features.. 68
Signal Description... 68
Functional Description 70

Hardware Watchdog Timer *77*
Using the WDT 77
WDT During Power-down and Idle 78

ONCE™ Mode (ON Chip Emulation) *79*

Power Management *80*
Reset 80
Reset Recommendation to Prevent Flash Corruption 82
Idle Mode 82
Power-down Mode.. 82

Power-off Flag *84*

Reduced EMI Mode *85*

Flash EEPROM Memory *86*
Features... 86
Flash Programming and Erasure... 86
Flash Registers and Memory Map... 87
Flash Memory Status.. 90
Memory Organization 90
Bootloader Architecture 91
ISP Protocol Description... 95
Functional Description 96
Flow Description 97
API Call Description... 105

Electrical Characteristics *107*
Absolute Maximum Ratings 107