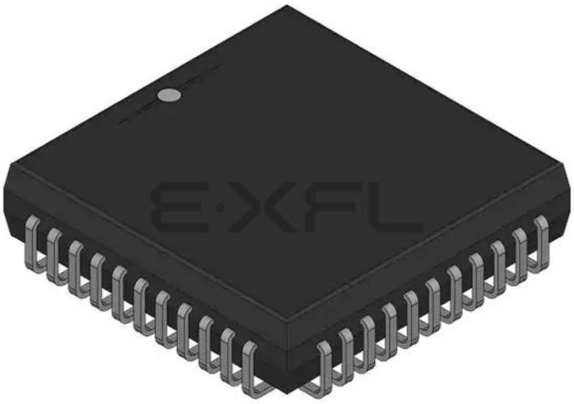


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

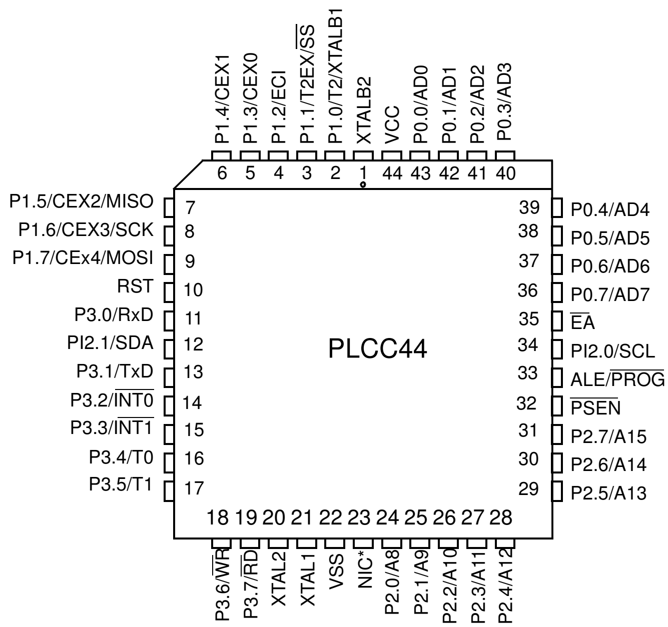
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	I²C, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	34
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 ~ 3.6V
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/atmel/at89c51ic2-slsul

Pin Configurations

Figure 2. Pin Configurations



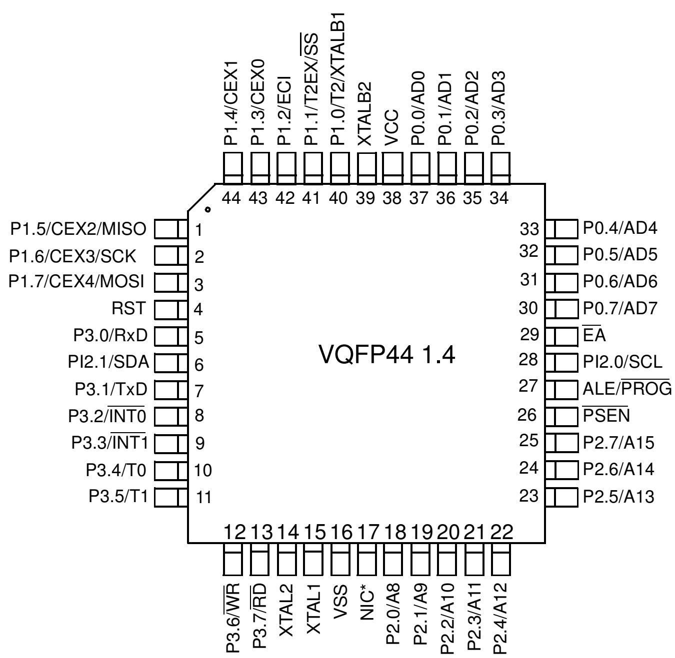


Table 28 shows the CCAPMn settings for the various PCA functions.

Table 28. CCAPMn Registers (n = 0-4)

CCAPM0 - PCA Module 0 Compare/Capture Control Register (0DAh)

CCAPM1 - PCA Module 1 Compare/Capture Control Register (0DBh)

CCAPM2 - PCA Module 2 Compare/Capture Control Register (0DCh)

CCAPM3 - PCA Module 3 Compare/Capture Control Register (0DDh)

CCAPM4 - PCA Module 4 Compare/Capture Control Register (0DEh)

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	ECOMn	**Enable Comparator** Cleared to disable the comparator function. Set to enable the comparator function.
5	CAPPn	**Capture Positive** Cleared to disable positive edge capture. Set to enable positive edge capture.
4	CAPNn	**Capture Negative** Cleared to disable negative edge capture. Set to enable negative edge capture.
3	MATn	**Match** When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.
2	TOGn	**Toggle** When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.
1	PWMn	**Pulse Width Modulation Mode** Cleared to disable the CEXn pin to be used as a pulse width modulated output. Set to enable the CEXn pin to be used as a pulse width modulated output.
0	CCF0	**Enable CCF interrupt** Cleared to disable compare/capture flag CCFn in the CCON register to generate an interrupt. Set to enable compare/capture flag CCFn in the CCON register to generate an interrupt.

Reset Value = X000 0000b
Not bit addressable

Table 31. CCAPnL Registers (n = 0-4)

CCAP0L - PCA Module 0 Compare/Capture Control Register Low (0EAh)

CCAP1L - PCA Module 1 Compare/Capture Control Register Low (0EBh)

CCAP2L - PCA Module 2 Compare/Capture Control Register Low (0ECh)

CCAP3L - PCA Module 3 Compare/Capture Control Register Low (0EDh)

CCAP4L - PCA Module 4 Compare/Capture Control Register Low (0EEh)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA Module n Compare/Capture Control** CCAPnL Value

Reset Value = 0000 0000b
Not bit addressable

Table 32. CH Register

CH - PCA Counter Register High (0F9h)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA counter** CH Value

Reset Value = 0000 0000b
Not bit addressable

Table 33. CL Register

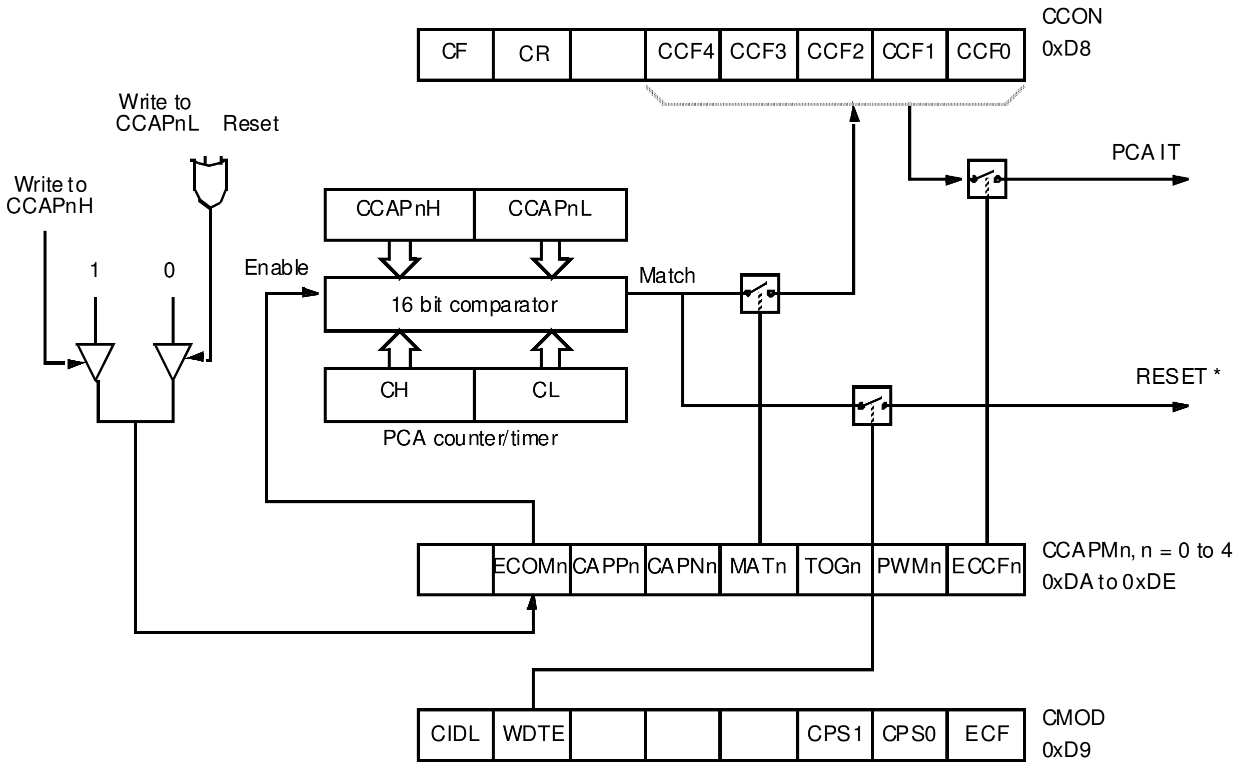
CL - PCA Counter Register Low (0E9h)

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

Bit Number	Bit Mnemonic	Description
7-0	-	**PCA Counter** CL Value

Reset Value = 0000 0000b
Not bit addressable

Figure 14. PCA Compare Mode and PCA Watchdog Timer



Before enabling ECOM bit, CCAPnL and CCAPnH should be set with a non zero value, otherwise an unwanted match could happen. Writing to CCAPnH will set the ECOM bit.

Once ECOM set, writing CCAPnL will clear ECOM so that an unwanted match doesn't occur while modifying the compare value. Writing to CCAPnH will set ECOM. For this reason, user software should write CCAPnL first, and then CCAPnH. Of course, the ECOM bit can still be controlled by accessing to CCAPMn register.

High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (See Figure 15).

A prior write must be done to CCAPnL and CCAPnH before writing the ECOMn bit.

Table 45. 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.

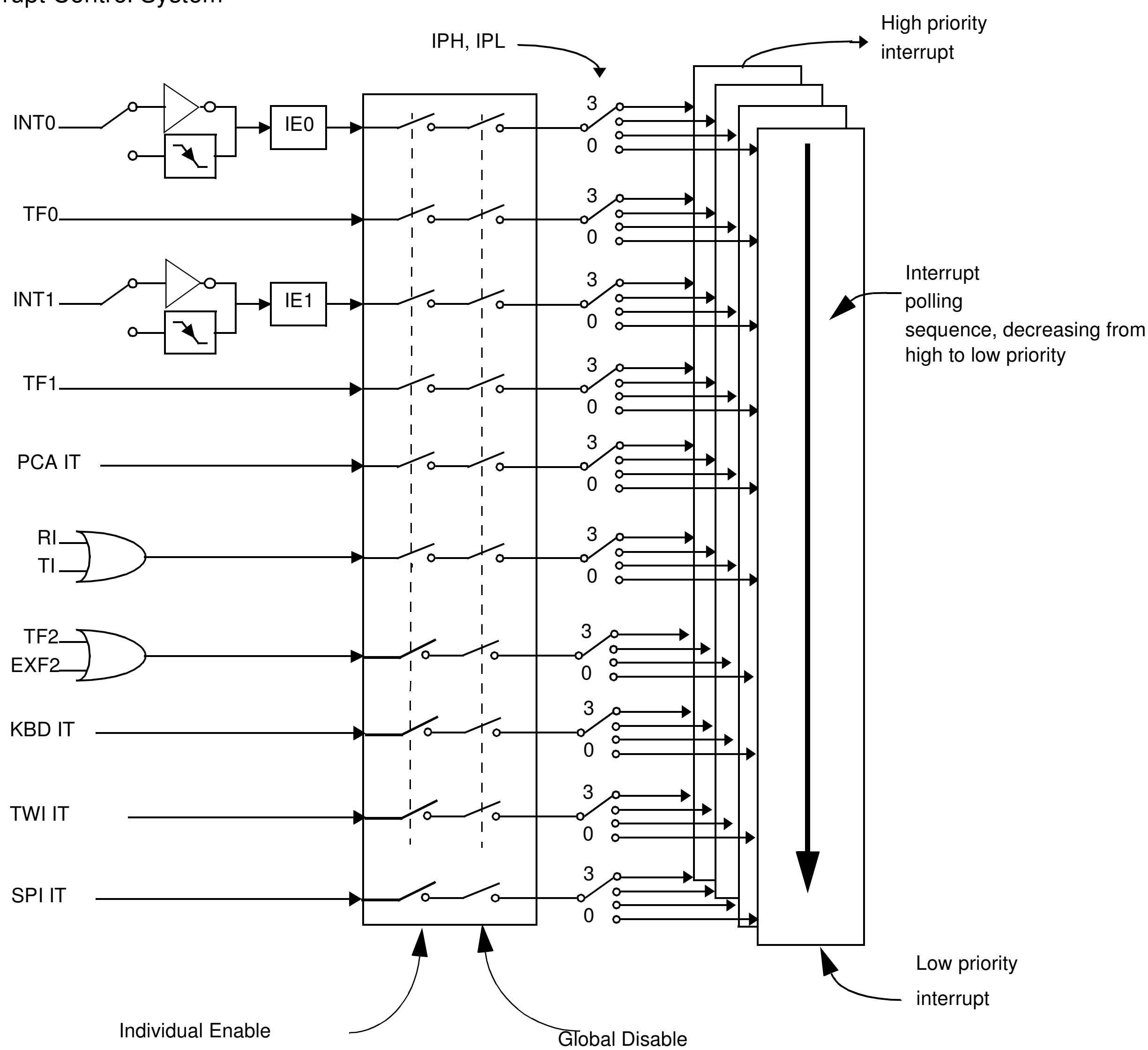
Reset Value = 00X1 0000b
Not bit addressable

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.

Interrupt System

The AT89C51IC2 has a total of 10 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (timers 0, 1 and 2), the serial port interrupt, SPI interrupt, Two Wire Interface (I2C) interrupt, Keyboard interrupt and the PCA global interrupt. These interrupts are shown in Figure 22.

Figure 22. Interrupt Control System



Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable register (Table 51 and Table 49). This register also contains a global disable bit, which must be cleared to disable all interrupts at once.

Each interrupt source can also be individually programmed to one out of four priority levels by setting or clearing a bit in the Interrupt Priority register (Table 52) and in the Interrupt Priority High register (Table 50 and Table 51) shows the bit values and priority levels associated with each combination.

Reset Recommendation to Prevent Flash Corruption

An example of bad initialization situation may occur in an instance where the bit ENBOOT in AUXR1 register is initialized from the hardware bit BLJB upon reset. Since this bit allows mapping of the bootloader in the code area, a reset failure can be critical.

If one wants the ENBOOT cleared in order to unmap the boot from the code area (yet due to a bad reset) the bit ENBOOT in SFRs may be set. If the value of Program Counter is accidently in the range of the boot memory addresses then a Flash access (write or erase) may corrupt the Flash on-chip memory.

It is recommended to use an external reset circuitry featuring power supply monitoring to prevent system malfunction during periods of insufficient power supply voltage (power supply failure, power supply switched off).

Idle Mode

An instruction that sets PCON.0 indicates that it is the last instruction to be executed before going into Idle mode. In Idle mode, the internal clock signal is gated off to the CPU, but not to the interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator and all other registers maintain their data during idle. The port pins hold the logical states they had at the time Idle was activated. ALE and PSEN hold at logic high level.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into idle.

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during idle. For example, an instruction that activates idle can also set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

Power-down Mode

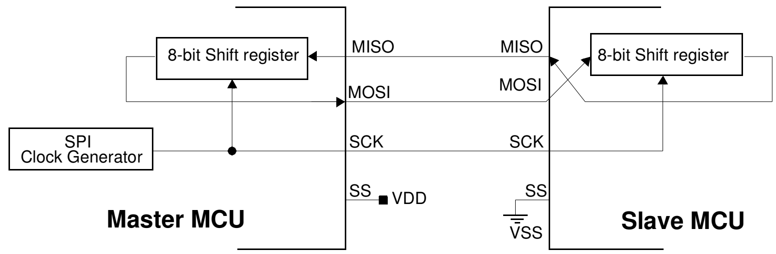
To save maximum power, a Power-down mode can be invoked by software (see PCON register).

In Power-down mode, the oscillator is stopped and the instruction that invoked Power-down mode is the last instruction executed. The internal RAM and SFRs retain their value until the Power-down mode is terminated. V_{CC} can be lowered to save further power. Either a hardware reset or an external interrupt can cause an exit from Power-down. To properly terminate Power-down, the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize.

Only external interrupts $\overline{\text{INT0}}$, $\overline{\text{INT1}}$ and Keyboard Interrupts are useful to exit from Power-down. For that, interrupt must be enabled and configured as level or edge sensitive interrupt input. When Keyboard Interrupt occurs after a power down mode, 1024 clocks are necessary to exit to power down mode and enter in operating mode.

Holding the pin low restarts the oscillator but bringing the pin high completes the exit as detailed in Figure 25. When both interrupts are enabled, the oscillator restarts as soon as one of the two inputs is held low and power down exit will be completed when the first input will be released. In this case, the higher priority interrupt service routine is executed. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that puts the AT89C51IC2 into Power-down mode.

Figure 28. Full-Duplex Master-Slave Interconnection



Master Mode

The SPI operates in Master mode when the Master bit, MSTR [1], in the SPCON register is set. Only one Master SPI device can initiate transmissions. Software begins the transmission from a Master SPI Module by writing to the Serial Peripheral Data Register (SPDAT). If the shift register is empty, the Byte is immediately transferred to the shift register. The Byte begins shifting out on MOSI pin under the control of the serial clock, SCK. Simultaneously, another Byte shifts in from the Slave on the Master's MISO pin. The transmission ends when the Serial Peripheral transfer data flag, SPIF, in SPSTA becomes set. At the same time that SPIF becomes set, the received Byte from the Slave is transferred to the receive data register in SPDAT. Software clears SPIF by reading the Serial Peripheral Status register (SPSTA) with the SPIF bit set, and then reading the SPDAT.

Slave Mode

The SPI operates in Slave mode when the Master bit, MSTR [2], in the SPCON register is cleared. Before a data transmission occurs, the Slave Select pin, $\overline{SS}$, of the Slave device must be set to '0'. $\overline{SS}$ must remain low until the transmission is complete.

In a Slave SPI Module, data enters the shift register under the control of the SCK from the Master SPI Module. After a Byte enters the shift register, it is immediately transferred to the receive data register in SPDAT, and the SPIF bit is set. To prevent an overflow condition, Slave software must then read the SPDAT before another Byte enters the shift register [3]. A Slave SPI must complete the write to the SPDAT (shift register) at least one bus cycle before the Master SPI starts a transmission. If the write to the data register is late, the SPI transmits the data already in the shift register from the previous transmission. The maximum SCK frequency allowed in slave mode is $F_{CLK\ PERIPH}$ /4.

Transmission Formats

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPCON: the Clock Polarity (CPOL [4]) and the Clock Phase (CPHA[4]). CPOL defines the default SCK line level in idle state. It has no significant effect on the transmission format. CPHA defines the edges on which the input data are sampled and the edges on which the output data are shifted (Figure 29 and Figure 30). The clock phase and polarity should be identical for the Master SPI device and the communicating Slave device.

1. The SPI Module should be configured as a Master before it is enabled (SPEN set). Also, the Master SPI should be configured before the Slave SPI.
2. The SPI Module should be configured as a Slave before it is enabled (SPEN set).
3. The maximum frequency of the SCK for an SPI configured as a Slave is the bus clock speed.
4. Before writing to the CPOL and CPHA bits, the SPI should be disabled (SPEN = '0').

Figure 29. Data Transmission Format (CPHA = 0)

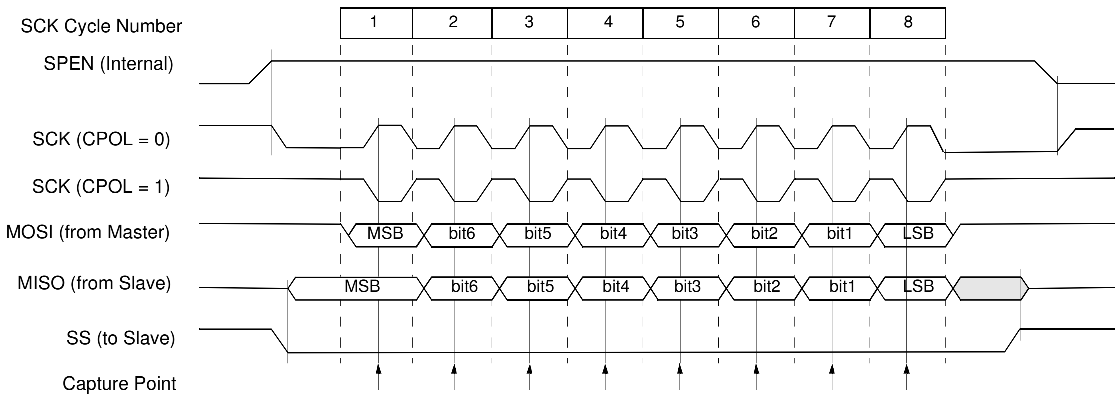


Figure 30. Data Transmission Format (CPHA = 1)

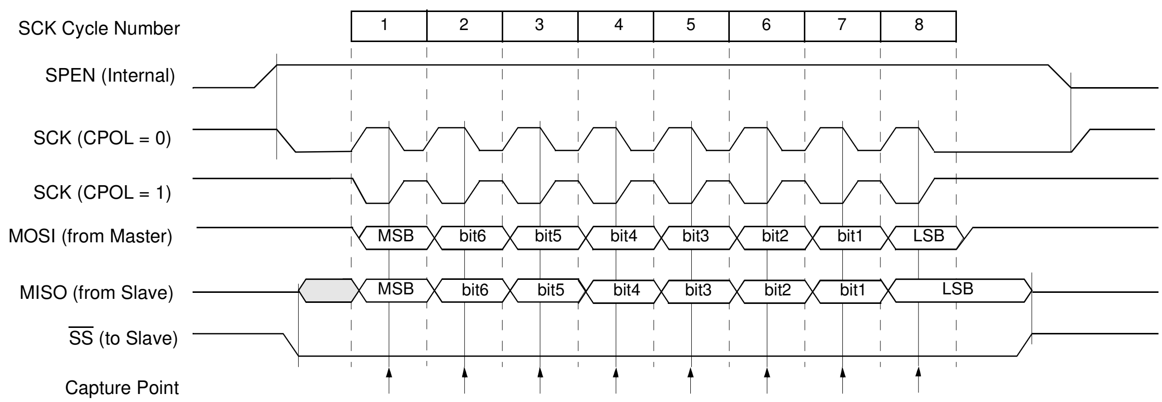
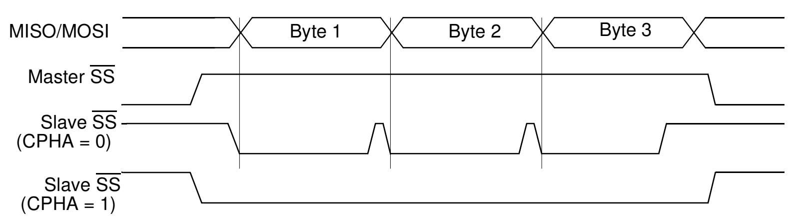


Figure 31. CPHA/$\overline{SS}$ Timing



As shown in Figure 29, 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 31).

Figure 30 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 31). This format may be preffered in systems having only one Master and only one Slave driving the MISO data line.

When the slave address and the direction bit have been transmitted and an acknowledgement bit has been received, the serial interrupt flag is set again and a number of status code in SSCS are possible. There are 40h, 48h or 38h for the master mode and also 68h, 78h or B0h if the slave mode was enabled (AA=logic 1). The appropriate action to be taken for each of these status code is detailed in Table 69. This scheme is repeated until a STOP condition is transmitted.

SSIE, CR2, CR1 and CR0 are not affected by the serial transfer and are referred to Table 7 to Table 11. After a repeated START condition (state 10h) SSLC may switch to the master transmitter mode by loading SSDAT with SLA+W.

Slave Receiver Mode

In the slave receiver mode, a number of data bytes are received from a master transmitter (Figure 40). To initiate the slave receiver mode, SSADR and SSCON must be loaded as follows:

Table 65. SSADR: slave receiver mode initialization

A6	A5	A4	A3	A2	A1	A0	GC
own slave address							

The upper 7 bits are the address to which SSLC will respond when addressed by a master. If the LSB (GC) is set SSLC will respond to the general call address (00h); otherwise it ignores the general call address.

Table 66. SSCON: slave receiver mode initialization

CR2	SSIE	STA	STO	SI	AA	CR1	CR0
bit rate	1	0	0	0	1	bit rate	bit rate

CR0, CR1 and CR2 have no effect in the slave mode. SSIE must be set to enable SSLC. The AA bit must be set to enable the own slave address or the general call address acknowledgement. STA, STO and SI must be cleared.

When SSADR and SSCON have been initialised, SSLC waits until it is addressed by its own slave address followed by the data direction bit which must be at logic 0 (W) for SSLC to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag is set and a valid status code can be read from SSCS. This status code is used to vector to an interrupt service routine.The appropriate action to be taken for each of these status code is detailed in Table 70. The slave receiver mode may also be entered if arbitration is lost while SSLC is in the master mode (states 68h and 78h).

If the AA bit is reset during a transfer, SSLC will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SSLC does not respond to its own slave address. However, the 2-wire bus is still monitored and address recognition may be resume at any time by setting AA. This means that the AA bit may be used to temporarily isolate SSLC from the 2-wire bus.

Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (Figure 41). Data transfer is initialized as in the slave receiver mode. When SSADR and SSCON have been initialized, SSLC waits until it is addressed by its own

Table 68. Status in master transmitter mode

Status Code SSSTA	Status of the Two-wire Bus and Two-wire Hardware	Application software response					Next Action Taken by Two-wire Hardware
			To SSCON				
		To/From SSDAT	SSSTA	SSSTO	SSI	SSAA	
08h	A START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted.
10h	A repeated START condition has been transmitted	Write SLA+W	X	0	0	X	SLA+W will be transmitted.
		Write SLA+R	X	0	0	X	SLA+R will be transmitted. Logic will switch to master receiver mode
18h	SLA+W has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
20h	SLA+W has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
28h	Data byte has been transmitted; ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
30h	Data byte has been transmitted; NOT ACK has been received	Write data byte	0	0	0	X	Data byte will be transmitted.
		No SSDAT action	1	0	0	X	Repeated START will be transmitted.
		No SSDAT action	0	1	0	X	STOP condition will be transmitted and SSSTO flag will be reset.
		No SSDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted and SSSTO flag will be reset.
38h	Arbitration lost in SLA+W or data bytes	No SSDAT action	0	0	0	X	Two-wire bus will be released and not addressed slave mode will be entered.
		No SSDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.

Table 77. SSADR (096h) - Synchronus Serial Address Register (read/write)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Table 78. SSADR Register - Reset value = FEh

Bit Number	Bit Mnemonic	Description
7	A7	Slave Address bit 7
6	A6	Slave Address bit 6
5	A5	Slave Address bit 5
4	A4	Slave Address bit 4
3	A3	Slave Address bit 3
2	A2	Slave Address bit 2
1	A1	Slave Address bit 1
0	GC	General Call bit Clear to disable the general call address recognition. Set to enable the general call address recognition.

Flash EEPROM Memory

The Flash memory increases EPROM and ROM functionality with in-circuit electrical erasure and programming. It contains 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
- 10K write cycles
- 10 years data retention

Flash Programming and Erasure

The 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 AT89C51IC2. The bootloader and the Application Programming Interface (API) routines are located in the BOOT ROM.

Flash Registers and Memory Map

The AT89C51IC2 Flash memory uses several registers for its management:

- Hardware registers can only be accessed through the parallel programming modes which are handled by the parallel programmer.
- Software registers are in a special page of the Flash memory which can be accessed through the API or with the parallel programming modes. This page, called "Extra Flash Memory", is not in the internal Flash program memory addressing space.

Hardware Register

The only hardware register of the AT89C51IC2 is called Hardware Security Byte (HSB).

Table 84. Hardware Security Byte (HSB)

7	6	5	4	3	2	1	0
X2	BLJB	OSC	-	XRAM	LB2	LB1	LB0

Bit Number	Bit Mnemonic	Description
7	X2	**X2 Mode** Programmed (‘0’ value) to force X2 mode (6 clocks per instruction) after reset. Unprogrammed (‘1’ Value) to force X1 mode, Standard Mode, after reset (Default).
6	BLJB	**Boot Loader Jump Bit** Unprogrammed (‘1’ value) to start the user’s application on next reset at address 0000h. Programmed (‘0’ value) to start the boot loader at address F800h on next reset (Default).
5	OSC	**Oscillator Bit** Programmed to allow oscillator B at startup Unprogrammed this bit to allow oscillator A at startup (Default).
4	-	**Reserved**
3	XRAM	**XRAM config bit (only programmable by programmer tools)** Programmed to inhibit XRAM Unprogrammed, this bit to valid XRAM (Default)
2-0	LB2-0	**User Memory Lock Bits (only programmable by programmer tools)** See Table 85

Boot Loader Jump Bit (BLJB)

One bit of the HSB, the BLJB bit, is used to force the boot address:

- When this bit is programmed (‘1’ value) the boot address is 0000h.
- When this bit is unprogrammed (‘1’ value) the boot address is F800h. By default, this bit is unprogrammed and the ISP is enabled.

Flash Memory Lock Bits

The three lock bits provide different levels of protection for the on-chip code and data, when programmed as shown in Table 85.

Table 88. Program Lock Bits of the SSB

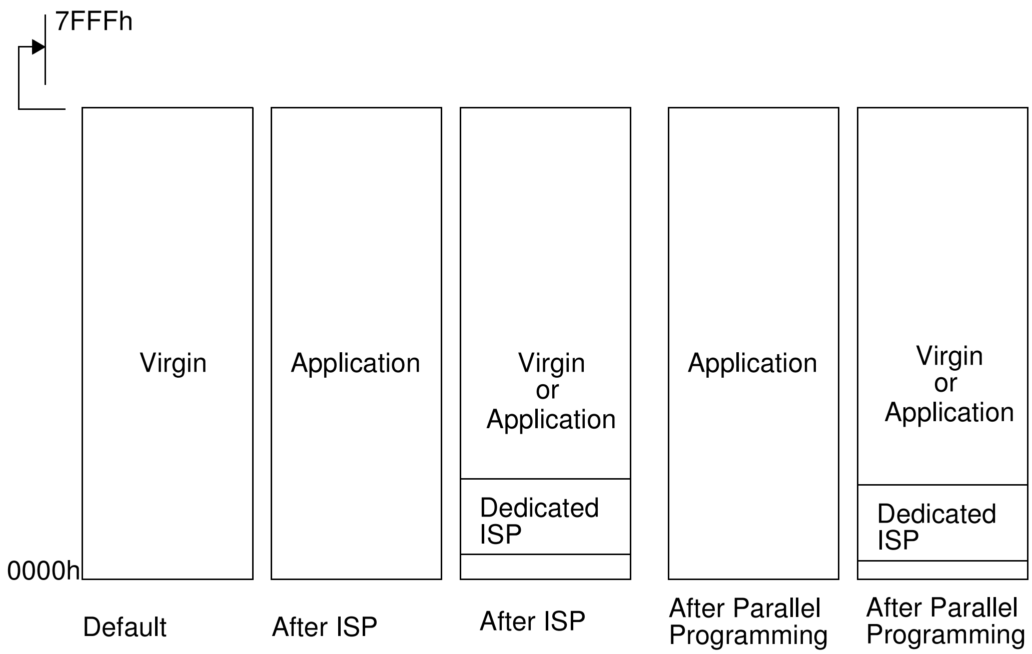
Program Lock Bits			
Security level	LB0	LB1	Protection Description
1	U	U	No program lock features enabled.
2	P	U	ISP programming of the Flash is disabled.
3	X	P	Same as 2, also verify through ISP programming interface is disabled.

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.

Flash Memory Status

AT89C51IC2 parts are delivered in standard with the ISP boot in the Flash memory. After ISP or parallel programming, the possible contents of the Flash memory are summarized on Figure 42.

Figure 42. Flash Memory Possible Contents



Memory Organization

In the AT89C51IC2, the lowest 32K of the 64 KB program memory address space is filled by internal Flash.

When the $\overline{EA}$ pin is high, the processor fetches instructions from internal program Flash. Bus expansion for accessing program memory from 32K upward automatic since external instruction fetches occur automatically when the program counter exceeds 7FFFh (32K). If the $\overline{EA}$ pin is tied low, all program memory fetches are from external memory.

Bootloader Functionality

Introduction

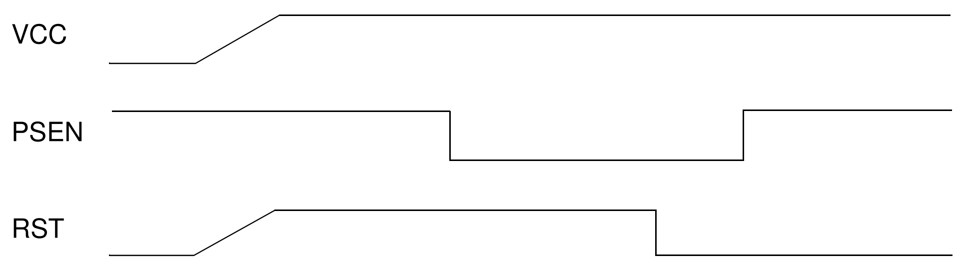
The bootloader can be activated by two means: Hardware conditions or regular boot process.

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 45).

Figure 45. Hardware conditions typical sequence during power-on.



The on-chip bootloader boot process is shown in Figure 46.

	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’).

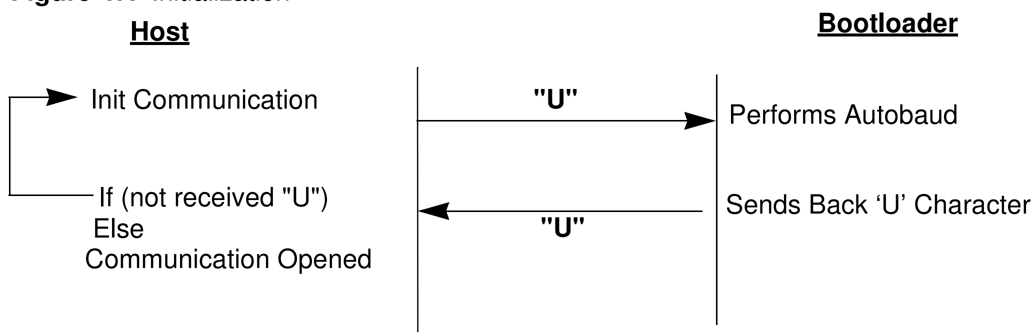
When the communication is initialized the protocol depends on the record type requested by the host.

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 47. Initialization



Autobaud Performances

The ISP feature allows a wide range of baud rates in the user application. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency. The ISP feature requires that an initial character (an uppercase U) be sent to the AT89C51IC2 to establish the baud rate. Table 91 shows the autobaud capability.

Table 91. Autobaud Performances

Frequency (MHz) **Baudrate (bit/s)**	**1.8432**	**2**	**2.4576**	**3**	**3.6864**	**4**	**5**	**6**	**7.3728**	**8**
2400	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
4800	OK	-	OK	OK	OK	OK	OK	OK	OK	OK
9600	OK	-	OK	OK	OK	OK	OK	OK	OK	OK
19200	OK	-	OK	OK	OK	-	-	OK	OK	OK
38400	-	-	OK		OK	-	OK	OK	OK	
57600	-	-	-	-	OK	-	-	-	OK	
115200	-	-	-	-	-	-	-	-	OK	
Frequency (MHz) **Baudrate (bit/s)**	**10**	**11.0592**	**12**	**14.318**	**14.746**	**16**	**20**	**24**	**26.6**	
2400	OK	OK	OK	OK	OK	OK	OK	OK	OK	
4800	OK	OK	OK	OK	OK	OK	OK	OK	OK	
9600	OK	OK	OK	OK	OK	OK	OK	OK	OK	
19200	OK	OK	OK	OK	OK	OK	OK	OK	OK	
38400	-	OK	OK	OK	OK	OK	OK	OK	OK	
57600	-	OK	-	OK	OK	OK	OK	OK	OK	
115200	-	OK	-	OK	OK	-	-	-	-	

Command Data Stream Protocol

All commands are sent using the same flow. Each frame sent by the host is echoed by the bootloader.

Example

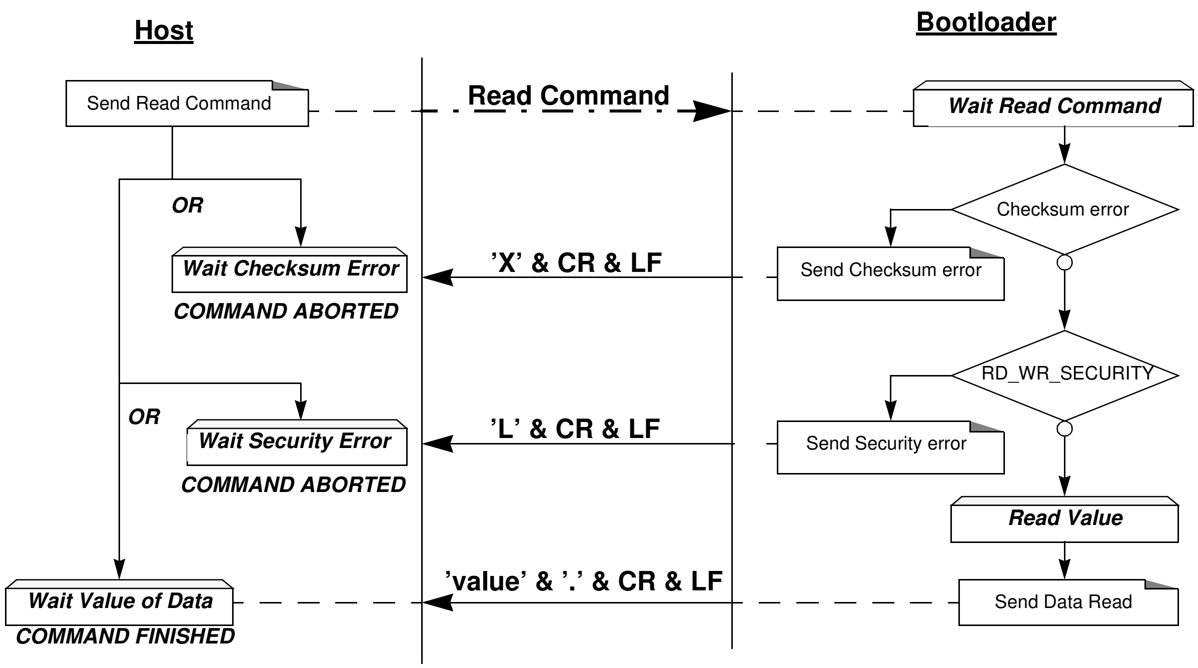
```
Display data from address 0000h to 0020h
HOST          : 05 0000 04 0000 0020 00 D7
BOOTLOADER    : 05 0000 04 0000 0020 00 D7
BOOTLOADER    0000=-----data------ CR LF   (16 data)
BOOTLOADER    0010=-----data------ CR LF   (16 data)
BOOTLOADER    0020=data CR LF              ( 1 data)
```

Read Function

This flow is similar for the following frames:
- Reading Frame
- EOF Frame/Atmel Frame (only reading Atmel Frame)

Description

Figure 52. Read Flow



Example

```
Read function (read SBV)
HOST          : 02 0000 05 07 02 F0
BOOTLOADER    : 02 0000 05 07 02 F0 Value . CR LF
Atmel Read function (read Bootloader version)
HOST          : 02 0000 01 02 00 FB
BOOTLOADER    : 02 0000 01 02 00 FB Value . CR LF
```

Datasheet Revision History

Changes from Rev. A 01/04 - Rev. B 01/06

1. Added green product ordering information.

Changes from Rev. B 01/06 - Rev. C 06/06

1. Correction to ordering information concerning product marking on green products.

Changes from Rev. C 06/06 - Rev. D 02/08

1. Removed non green part numbers from ordering information.