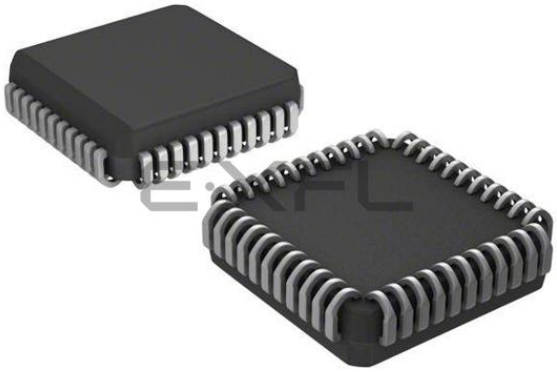


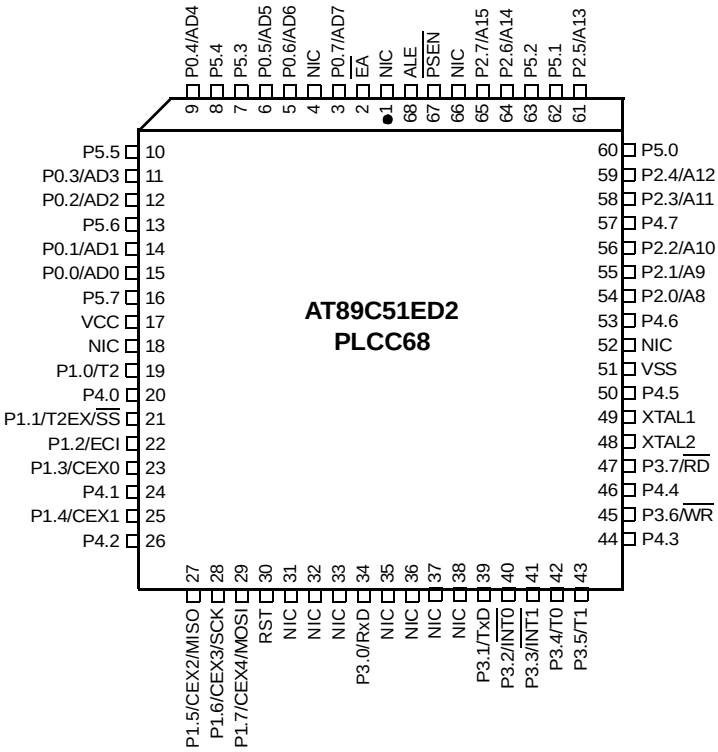
Welcome to E-XFL.COM

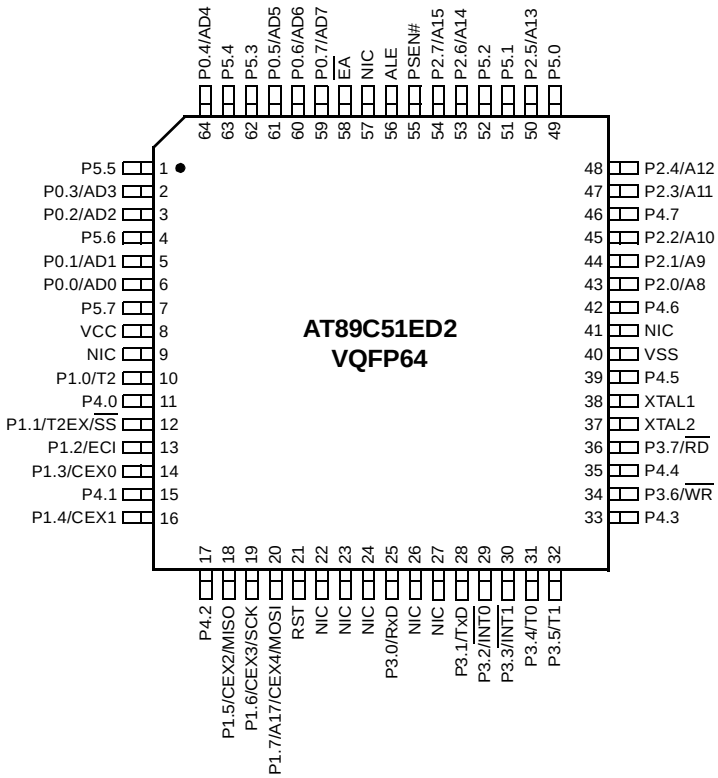
What is "Embedded - Microcontrollers"?

"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	34
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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/at89c51rd2-slrim





NIC: Not Internaly Connected

Reset Output

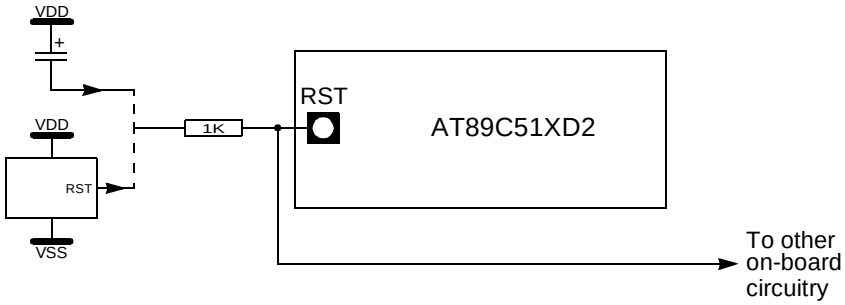
Reset output can be generated by two sources:

- Internal POR/PFD
- Hardware watchdog timer

As detailed in Section “Hardware Watchdog Timer”, page 86, the WDT generates a 96-clock period pulse on the RST pin.

In order to properly propagate this pulse to the rest of the application in case of external capacitor or power-supply supervisor circuit, a 1 kΩ resistor must be added as shown Figure 11.

Figure 11. Recommended Reset Output Schematic



Programmable Counter Array (PCA)

The PCA provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy. The PCA consists of a dedicated timer/counter which serves as the time base for an array of five compare/capture modules. Its clock input can be programmed to count any one of the following signals:

- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 6
- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 2
- Timer 0 overflow
- External input on ECI (P1.2)

Each compare/capture module can be programmed in any one of the following modes:

- Rising and/or falling edge capture
- Software timer
- High-speed output
- Pulse width modulator

Module 4 can also be programmed as a watchdog timer (see Section "PCA Watchdog Timer", page 47).

When the compare/capture modules are programmed in the capture mode, software timer, or high speed output mode, an interrupt can be generated when the module executes its function. All five modules plus the PCA timer overflow share one interrupt vector.

The PCA timer/counter and compare/capture modules share Port 1 for external I/O. These pins are listed below. If one or several bits in the port are not used for the PCA, they can still be used for standard I/O.

PCA Component	External I/O Pin
16-bit Counter	P1.2/ECI
16-bit Module 0	P1.3/CEX0
16-bit Module 1	P1.4/CEX1
16-bit Module 2	P1.5/CEX2
16-bit Module 3	P1.6/CEX3

The PCA timer is a common time base for all five modules (see Figure 16). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD register (Table 22) and can be programmed to run at:

- 1/6 the peripheral clock frequency ($F_{CLK\ PERIPH}$)
- 1/2 the peripheral clock frequency ($F_{CLK\ PERIPH}$)
- The Timer 0 overflow
- The input on the ECI pin (P1.2)

Table 23. CCON Register

CCON - PCA Counter Control Register (D8h)

7	6	5	4	3	2	1	0
CF	**CR**	**-**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

Bit Number	Bit Mnemonic	Description
7	CF	**PCA Counter Overflow flag** Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
6	CR	**PCA Counter Run control bit** Must be cleared by software to turn the PCA counter off. Set by software to turn the PCA counter on.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	CCF4	**PCA Module 4 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
3	CCF3	**PCA Module 3 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
2	CCF2	**PCA Module 2 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
1	CCF1	**PCA Module 1 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.
0	CCF0	**PCA Module 0 interrupt flag** Must be cleared by software. Set by hardware when a match or capture occurs.

Reset Value = 00X0 0000b
Bit addressable

The watchdog timer function is implemented in Module 4 (See Figure 19).

The PCA interrupt system is shown in Figure 17.

Table 27. 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 28. 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 29. CL Register

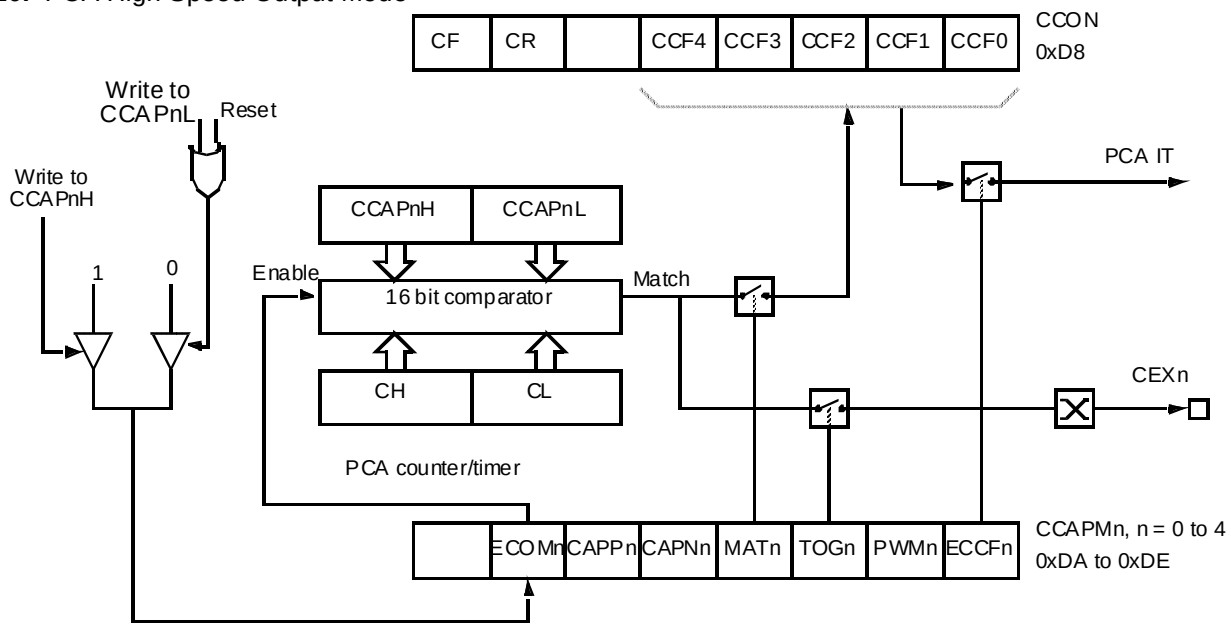
CL - PCA Counter Register Low (0E9h)

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 20. PCA High Speed Output Mode



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

Once ECOM is 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.

Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 21 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 modules capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the modules 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.

Registers

Table 43. KBF Register

KBF-Keyboard Flag Register (9Eh)

7	6	5	4	3	2	1	0
KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0

Bit Number	Bit Mnemonic	Description
7	KBF7	**Keyboard line 7 flag** Set by hardware when the Port line 7 detects a programmed level. It generates a Keyboard interrupt request if the KBKBIE.7 bit in KBIE register is set. Must be cleared by software.
6	KBF6	**Keyboard line 6 flag** Set by hardware when the Port line 6 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.6 bit in KBIE register is set. Must be cleared by software.
5	KBF5	**Keyboard line 5 flag** Set by hardware when the Port line 5 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.5 bit in KBIE register is set. Must be cleared by software.
4	KBF4	**Keyboard line 4 flag** Set by hardware when the Port line 4 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.4 bit in KBIE register is set. Must be cleared by software.
3	KBF3	**Keyboard line 3 flag** Set by hardware when the Port line 3 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.3 bit in KBIE register is set. Must be cleared by software.
2	KBF2	**Keyboard line 2 flag** Set by hardware when the Port line 2 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.2 bit in KBIE register is set. Must be cleared by software.
1	KBF1	**Keyboard line 1 flag** Set by hardware when the Port line 1 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.1 bit in KBIE register is set. Must be cleared by software.
0	KBF0	**Keyboard line 0 flag** Set by hardware when the Port line 0 detects a programmed level. It generates a Keyboard interrupt request if the KBIE.0 bit in KBIE register is set. Must be cleared by software.

Reset Value = 0000 0000b

This register is read only access, all flags are automatically cleared by reading the register.

Serial Port Interface (SPI)

The Serial Peripheral Interface Module (SPI) allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs.

Features

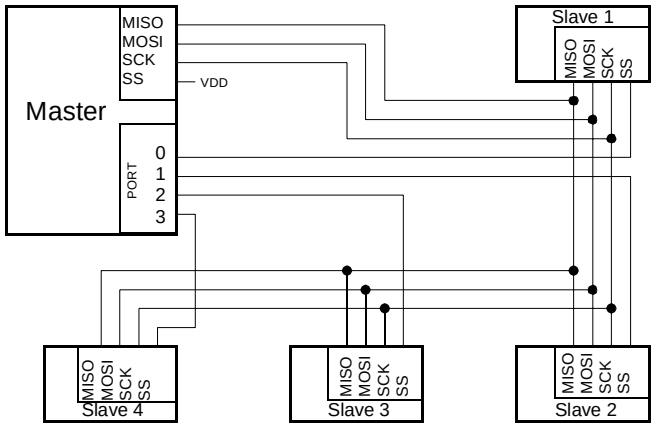
Features of the SPI Module include the following:

- Full-duplex, three-wire synchronous transfers
- Master or Slave operation
- Eight programmable Master clock rates
- Serial clock with programmable polarity and phase
- Master Mode fault error flag with MCU interrupt capability
- Write collision flag protection

Signal Description

Figure 29 shows a typical SPI bus configuration using one Master controller and many Slave peripherals. The bus is made of three wires connecting all the devices.

Figure 29. SPI Master/Slaves Interconnection



The Master device selects the individual Slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the Slave devices.

Master Output Slave Input (MOSI)

This 1-bit signal is directly connected between the Master Device and a Slave Device. The MOSI line is used to transfer data in series from the Master to the Slave. Therefore, it is an output signal from the Master, and an input signal to a Slave. A Byte (8-bit word) is transmitted most significant bit (MSB) first, least significant bit (LSB) last.

Master Input Slave Output (MISO)

This 1-bit signal is directly connected between the Slave Device and a Master Device. The MISO line is used to transfer data in series from the Slave to the Master. Therefore, it is an output signal from the Slave, and an input signal to the Master. A Byte (8-bit word) is transmitted most significant bit (MSB) first, least significant bit (LSB) last.

SPI Serial Clock (SCK)

This signal is used to synchronize the data movement both in and out of the devices through their MOSI and MISO lines. It is driven by the Master for eight clock cycles which allows to exchange one Byte on the serial lines.

Slave Select ($\overline{SS}$)

Each Slave peripheral is selected by one Slave Select pin ($\overline{SS}$). This signal must stay low for any message for a Slave. It is obvious that only one Master ($\overline{SS}$ high level) can

Error Conditions

The following flags in the SPSTA signal SPI error conditions:

Mode Fault (MODF)

Mode Fault error in Master mode SPI indicates that the level on the Slave Select ($\overline{SS}$) pin is inconsistent with the actual mode of the device. MODF is set to warn that there may be a multi-master conflict for system control. In this case, the SPI system is affected in the following ways:

- An SPI receiver/error CPU interrupt request is generated
- The SPEN bit in SPCON is cleared. This disables the SPI
- The MSTR bit in SPCON is cleared

When $\overline{SS}$ Disable (SSDIS) bit in the SPCON register is cleared, the MODF flag is set when the $\overline{SS}$ signal becomes '0'.

However, as stated before, for a system with one Master, if the $\overline{SS}$ pin of the Master device is pulled low, there is no way that another Master attempts to drive the network. In this case, to prevent the MODF flag from being set, software can set the SSDIS bit in the SPCON register and therefore making the $\overline{SS}$ pin as a general-purpose I/O pin.

Clearing the MODF bit is accomplished by a read of SPSTA register with MODF bit set, followed by a write to the SPCON register. SPEN Control bit may be restored to its original set state after the MODF bit has been cleared.

Write Collision (WCOL)

A Write Collision (WCOL) flag in the SPSTA is set when a write to the SPDAT register is done during a transmit sequence.

WCOL does not cause an interruption, and the transfer continues uninterrupted.

Clearing the WCOL bit is done through a software sequence of an access to SPSTA and an access to SPDAT.

Overrun Condition

An overrun condition occurs when the Master device tries to send several data Bytes and the Slave devise has not cleared the SPIF bit issuing from the previous data Byte transmitted. In this case, the receiver buffer contains the Byte sent after the SPIF bit was last cleared. A read of the SPDAT returns this Byte. All others Bytes are lost.

This condition is not detected by the SPI peripheral.

SS Error Flag (SSERR)

A Synchronous Serial Slave Error occurs when $\overline{SS}$ goes high before the end of a received data in slave mode. SSERR does not cause in interruption, this bit is cleared by writing 0 to SPEN bit (reset of the SPI state machine).

Interrupts

Two SPI status flags can generate a CPU interrupt requests:

Table 47. SPI Interrupts

Flag	Request
SPIF (SP data transfer)	SPI Transmitter Interrupt request
MODF (Mode Fault)	SPI Receiver/Error Interrupt Request (if SSDIS = '0')

Serial Peripheral data transfer flag, SPIF: This bit is set by hardware when a transfer has been completed. SPIF bit generates transmitter CPU interrupt requests.

Mode Fault flag, MODF: This bit becomes set to indicate that the level on the SS is inconsistent with the mode of the SPI. MODF with SSDIS reset, generates receiver/error CPU interrupt requests. When SSDIS is set, no MODF interrupt request is generated.

Figure 35 gives a logical view of the above statements.

Registers

The PCA interrupt vector is located at address 0033H, the SPI interrupt vector is located at address 004BH and Keyboard interrupt vector is located at address 003BH. All other vectors addresses are the same as standard C52 devices.

Table 51. Priority Level Bit Values

IPH.x	IPL.x	Interrupt Level Priority
0	0	0 (Lowest)
0	1	1
1	0	2
1	1	3 (Highest)

A low-priority interrupt can be interrupted by a high priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

the SFR and RAM contents are preserved. The status of the Port pins during Power-Down mode is detailed in Table 59.

Note: VCC may be reduced to as low as V_{RET} during Power-Down mode to further reduce power dissipation. Take care, however, that VDD is not reduced until Power-Down mode is invoked.

Entering Power-Down Mode

To enter Power-Down mode, set PD bit in PCON register. The AT89C51RD2/ED2 enters the Power-Down mode upon execution of the instruction that sets PD bit. The instruction that sets PD bit is the last instruction executed.

Exiting Power-Down Mode

Note: If VCC was reduced during the Power-Down mode, do not exit Power-Down mode until VCC is restored to the normal operating level.

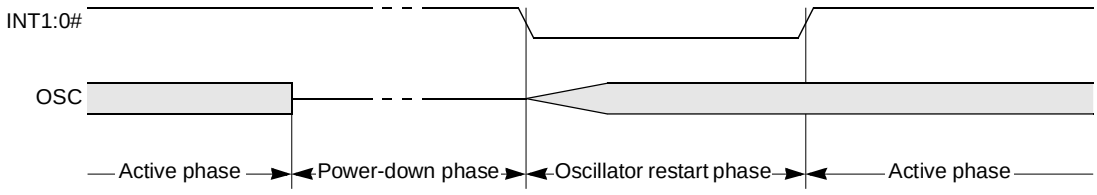
There are three ways to exit the Power-Down mode:

1. Generate an enabled external interrupt.
   - The AT89C51RD2/ED2 provides capability to exit from Power-Down using INT0#, INT1#.
     Hardware clears PD bit in PCON register which starts the oscillator and restores the clocks to the CPU and peripherals. Using INTx# input, execution resumes when the input is released (see Figure 37). Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.

Note: The external interrupt used to exit Power-Down mode must be configured as level sensitive (INT0# and INT1#) and must be assigned the highest priority. In addition, the duration of the interrupt must be long enough to allow the oscillator to stabilize. The execution will only resume when the interrupt is deasserted.

Note: Exit from power-down by external interrupt does not affect the SFRs nor the internal RAM content.

Figure 37. Power-Down Exit Waveform Using INT1:0#



2. Generate a reset.
   - A logic high on the RST pin clears PD bit in PCON register directly and asynchronously. This starts the oscillator and restores the clock to the CPU and peripherals. Program execution momentarily resumes with the instruction immediately following the instruction that activated Power-Down mode and may continue for a number of clock cycles before the internal reset algorithm takes control. Reset initializes the AT89C51RD2/ED2 and vectors the CPU to address 0000h.

Registers

Table 60. PCON Register

PCON (S87:h) Power configuration Register

7	6	5	4	3	2	1	0
-	-	-	-	GF1	GF0	PD	IDL

Bit Number	Bit Mnemonic	Description
7-4	-	**Reserved** The value read from these bits is indeterminate. Do not set these bits.
3	GF1	**General Purpose flag 1** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
2	GF0	**General Purpose flag 0** One use is to indicate whether an interrupt occurred during normal operation or during Idle mode.
1	PD	**Power-Down Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power-Down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence.

Reset Value= XXXX 0000b

Table 62. WDTPRG Register

WDTPRG - Watchdog Timer Out Register (0A7h)

7	6	5	4	3	2	1	0
-	-	-	-	-	S2	S1	S0

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is undetermined. Do not try to set this bit.
6	-	
5	-	
4	-	
3	-	
2	S2	**WDT Time-out select bit 2**
1	S1	**WDT Time-out select bit 1**
0	S0	**WDT Time-out select bit 0**
		S2 S1 S0 Selected Time-out 0 0 0 (2^{14} - 1) machine cycles, 16. 3 ms @ F_{OSCA} =12 MHz 0 0 1 (2^{15} - 1) machine cycles, 32.7 ms @ F_{OSCA}=12 MHz 0 1 0 (2^{16} - 1) machine cycles, 65. 5 ms @ F_{OSCA}=12 MHz 0 1 1 (2^{17} - 1) machine cycles, 131 ms @ F_{OSCA}=12 MHz 1 0 0 (2^{18} - 1) machine cycles, 262 ms @ F_{OSCA}=12 MHz 1 0 1 (2^{19} - 1) machine cycles, 542 ms @ F_{OSCA}=12 MHz 1 1 0 (2^{20} - 1) machine cycles, 1.05 ms @ F_{OSCA}=12 MHz 1 1 1 (2^{21} - 1) machine cycles, 2.09 ms @ F_{OSCA}=12 MHz

Reset Value = XXXX X000

WDT during Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode the user does not need to service the WDT. There are 2 methods of exiting Power-down mode: by a hardware reset or via a level activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally should whenever the AT89C51RD2/ED2 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service routine.

To ensure that the WDT does not overflow within a few states of exiting of powerdown, it is better to reset the WDT just before entering powerdown.

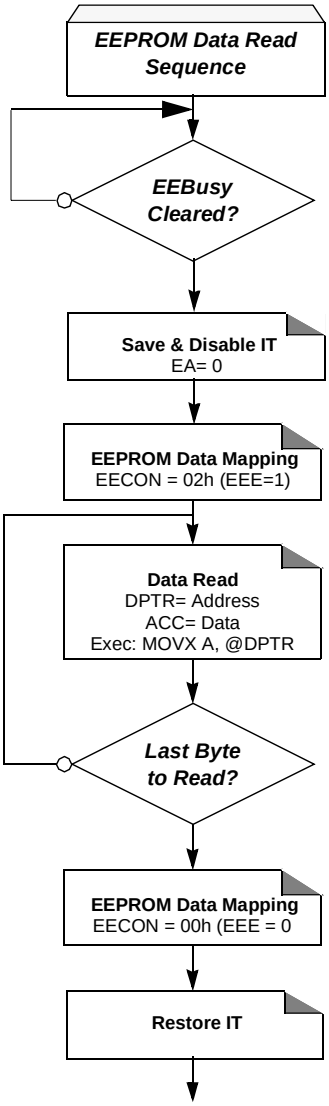
In the Idle mode, the oscillator continues to run. To prevent the WDT from resetting the AT89C51RD2/ED2 while in Idle mode, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle mode.

Read Data

The following procedure is used to read the data stored in the EEPROM memory:

- Check EEBUSY flag
- If the user application interrupts routines use XRAM memory space: Save and disable interrupts.
- Load DPTR with the address to read
- Set bit EEE of EECON register
- Execute a MOVX A, @DPTR
- Clear bit EEE of EECON register
- Restore interrupts.

Figure 39. Recommended EEPROM Data Read Sequence

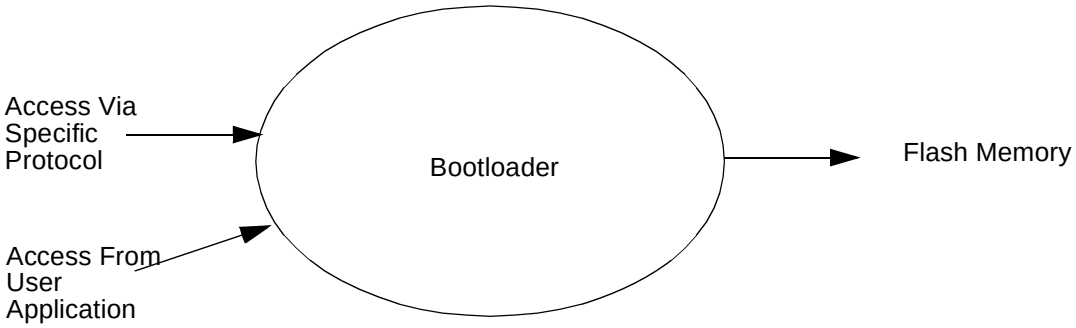


Bootloader Architecture

Introduction

The bootloader manages communication according to a specifically defined protocol to provide the whole access and service on Flash memory. Furthermore, all accesses and routines can be called from the user application.

Figure 41. Diagram Context Description



Acronyms

ISP: In-System Programming

SBV: Software Boot Vector

BSB: Boot Status Byte

SSB: Software Security Byte

HW: Hardware Byte

ISP Protocol Description

Physical Layer

The UART used to transmit information has the following configuration:

- Character: 8-bit data
- Parity: none
- Stop: 2 bits
- Flow control: none
- Baudrate: autobaud is performed by the bootloader to compute the baudrate chosen by the host.

Frame Description

The Serial Protocol is based on the Intel Hex-type records.

Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below.

Figure 45. Intel Hex Type Frame

Record Mark ':'	Reclen	Load Offset	Record Type	Data or Info	Checksum
1-byte	1-byte	2-bytes	1-byte	n-bytes	1-byte

- Record Mark:

Record Mark is the start of frame. This field must contain ':'.

- Reclen:

Reclen specifies the number of bytes of information or data which follows the Record Type field of the record.

- Load Offset:

Load Offset specifies the 16-bit starting load offset of the data bytes, therefore this field is used only for Data Program Record (see Section "ISP Commands Summary").

- Record Type:

Record Type specifies the command type. This field is used to interpret the remaining information within the frame. The encoding for all the current record types is described in Section "ISP Commands Summary".

- Data/Info:

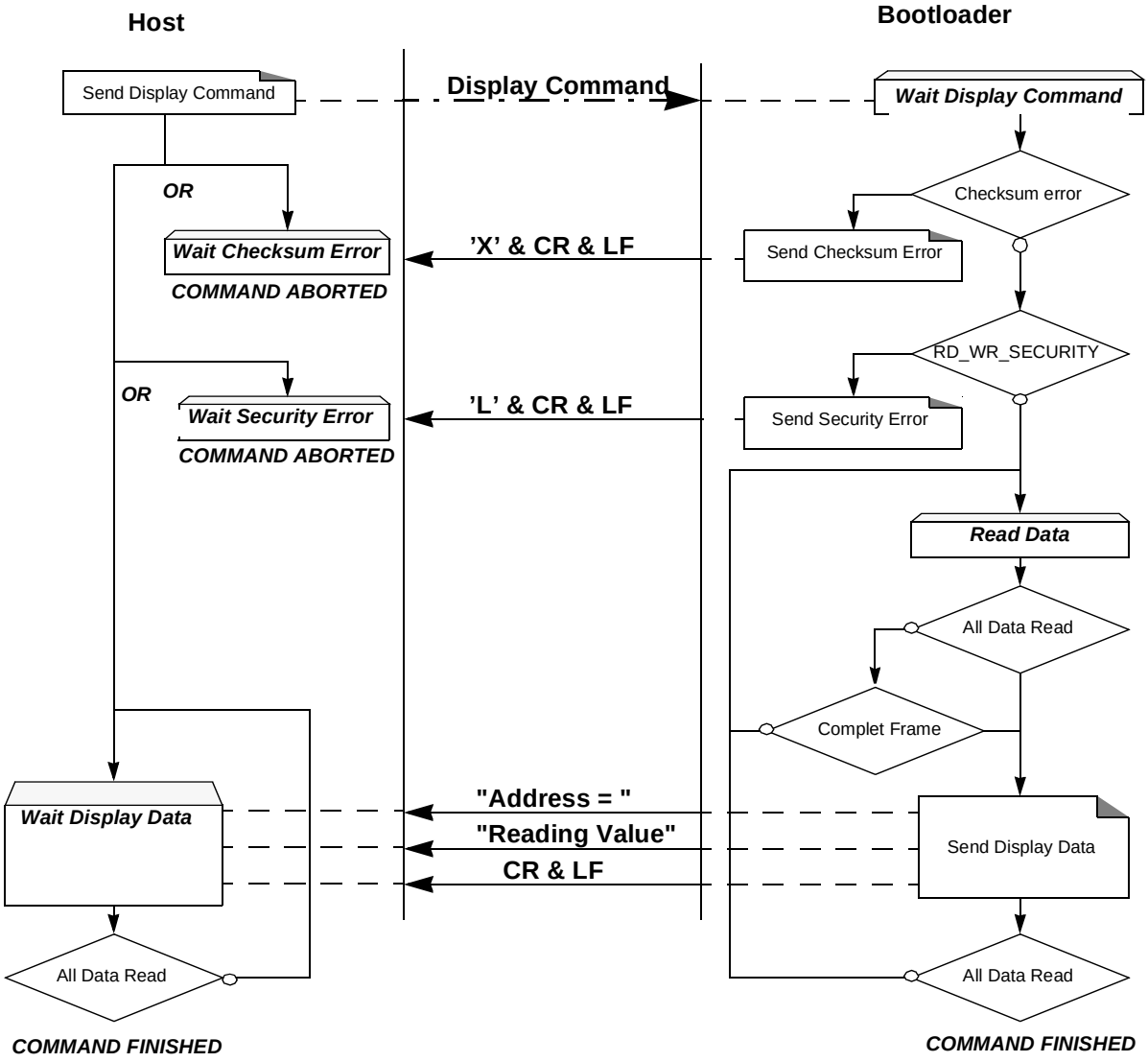
Data/Info is a variable length field. It consists of zero or more bytes encoded as pairs of hexadecimal digits. The meaning of data depends on the **Record Type**.

- Checksum:

The two's complement of the 8-bit bytes that result from converting each pair of ASCII hexadecimal digits to one byte of binary, and including the **Reclen** field to and including the last byte of the **Data/Info** field. Therefore, the sum of all the ASCII pairs in a record after converting to binary, from the **Reclen** field to and including the **Checksum** field, is zero.

Display Data Description

Figure 50. Display Flow



Example

Display data from address 0000h to 0020h

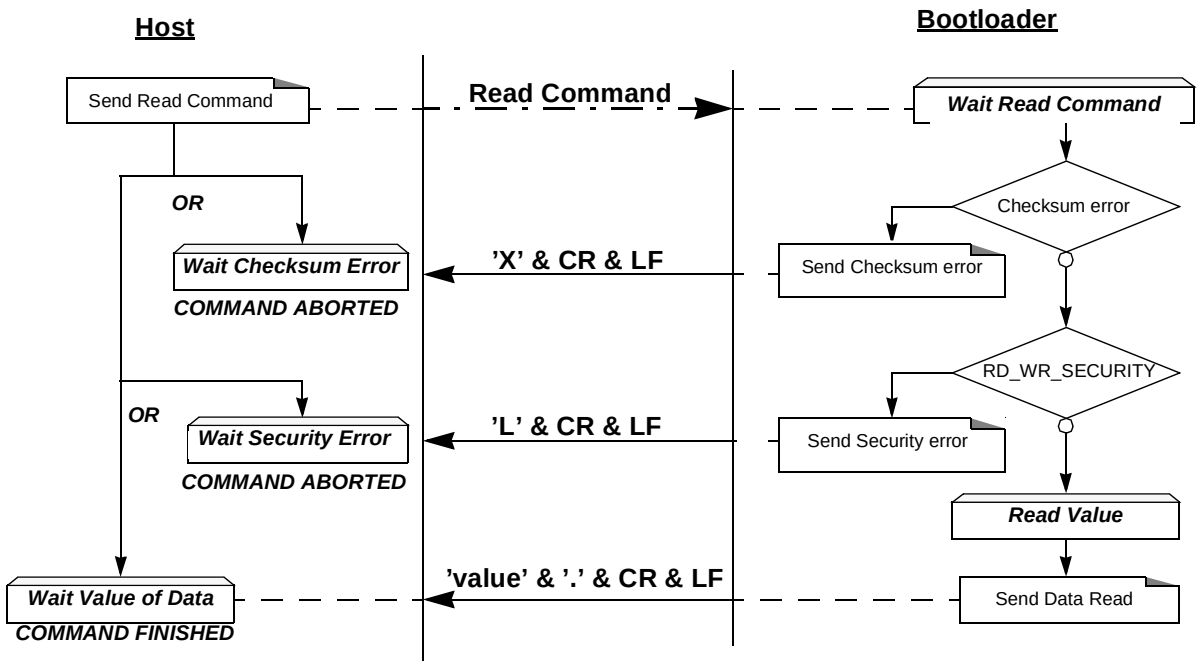
HOST	: 05 0000 04 0000 0020 00 D7
BOOTLOADER	: 05 0000 04 0000 0020 00 D7
BOOTLOADER	0000=-----data------ CR LF (16 data)
BOOTLOADER	0010=-----data------ CR LF (16 data)
BOOTLOADER	0020=data CR LF (1 data)

Read Function Description

This flow is similar for the following frames:

- Reading Frame
- EOF Frame/ Atmel Frame (only reading Atmel Frame)

Figure 51. Read Flow



Example

Read function (read SBV)

HOST : 02 0000 05 07 02 F0

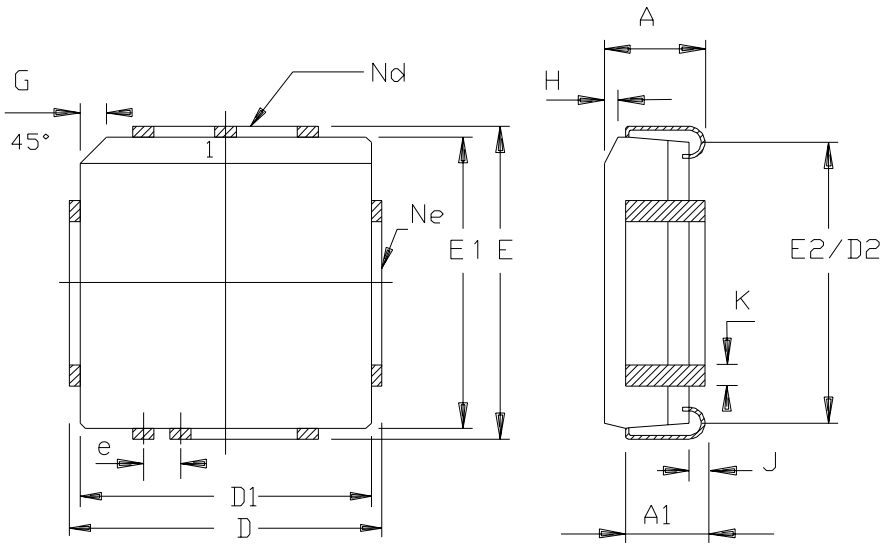
BOOTLOADER : 02 0000 05 07 02 F0 Value . CR LF

Atmel Read function (read Bootloader version)

HOST : 02 0000 01 02 00 FB

BOOTLOADER : 02 0000 01 02 00 FB Value . CR LF

PLCC68



	MM		INCH	
A	4.20	5.08	.165	.200
A1	2.29	3.30	.090	.130
D	25.02	25.27	.985	.995
D1	24.13	24.33	.950	.958
D2	22.61	23.62	.890	.930
E	25.02	25.27	.985	.995
E1	24.13	24.33	.950	.958
E2	22.61	23.62	.890	.930
e	1.27	BSC	.050	BSC
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	-	.020	-
K	0.33	0.53	.013	.021
Nd	17		17	
Ne	17		17	
PKG STD	00			