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	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI
Peripherals	POR, WDT
Number of I/O	15
Program Memory Size	1KB (512 x 16)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	-
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s1200-4si

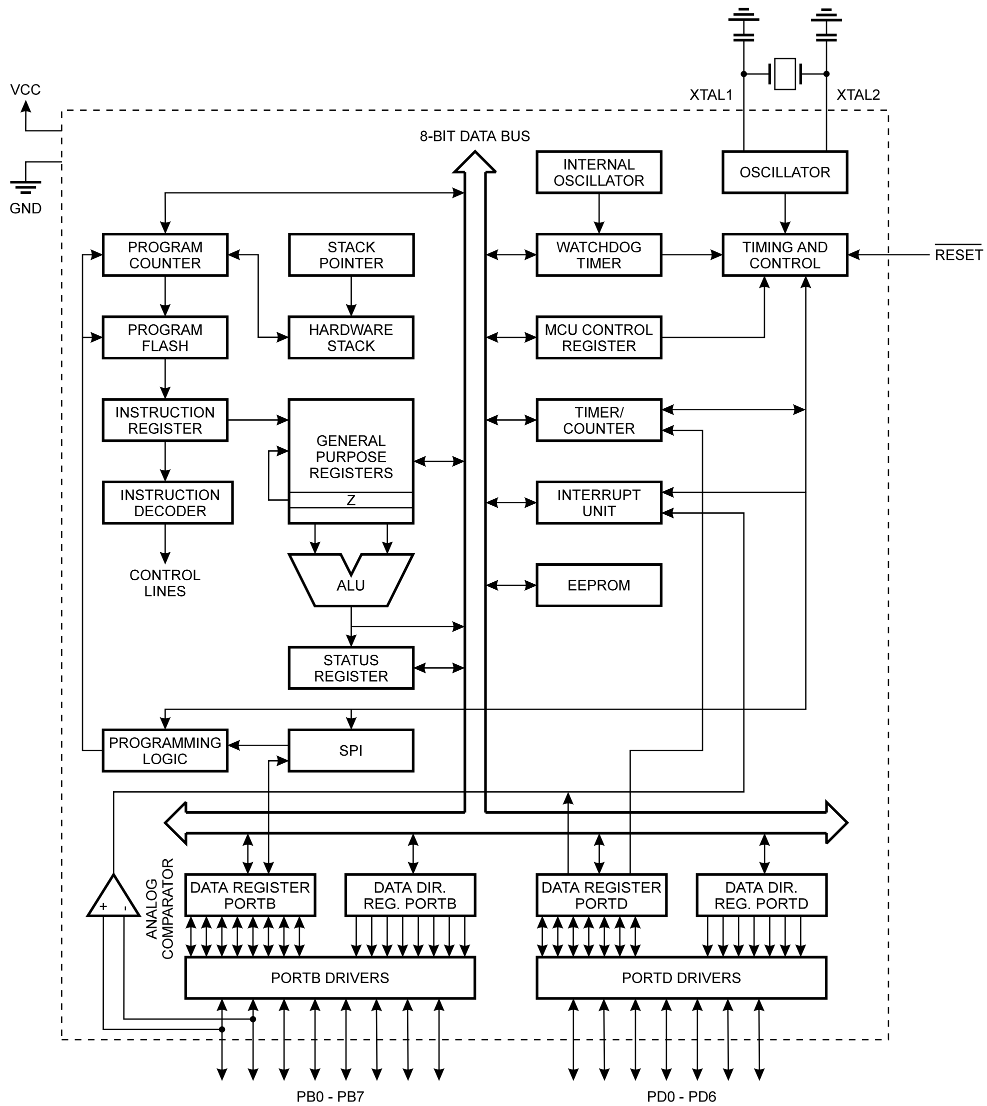
Description

The AT90S1200 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S1200 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

The AVR core combines a rich instruction set with the 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

Block Diagram

Figure 1. The AT90S1200 Block Diagram



The architecture supports high-level languages efficiently as well as extremely dense assembler code programs. The AT90S1200 provides the following features: 1K byte of In-System Programmable Flash, 64 bytes EEPROM, 15 general purpose I/O lines, 32 general purpose working registers, internal and external interrupts, programmable watchdog timer with internal oscillator, an SPI serial port for program downloading and two software selectable power-saving modes. The Idle Mode stops the CPU while allow-

EEPROM Data Memory

The AT90S1200 contains 64 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 25 specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register. For the SPI data downloading, see page 44 for a detailed description.

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 11 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 11. The Parallel Instruction Fetches and Instruction Executions

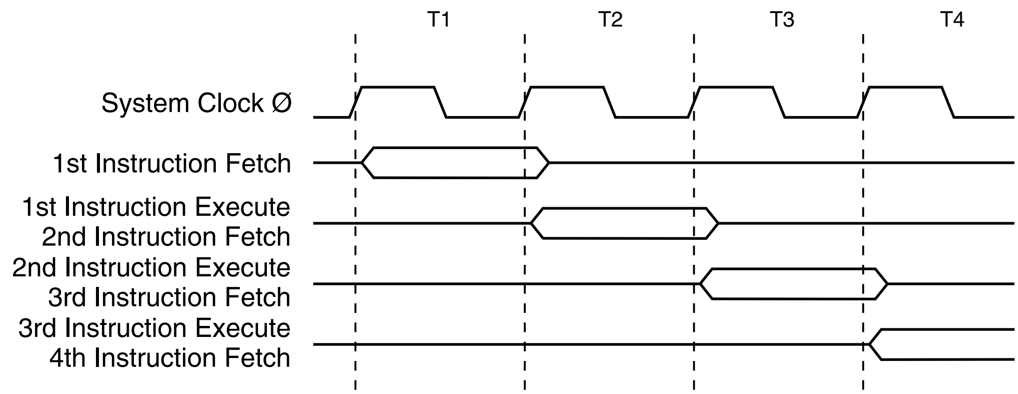
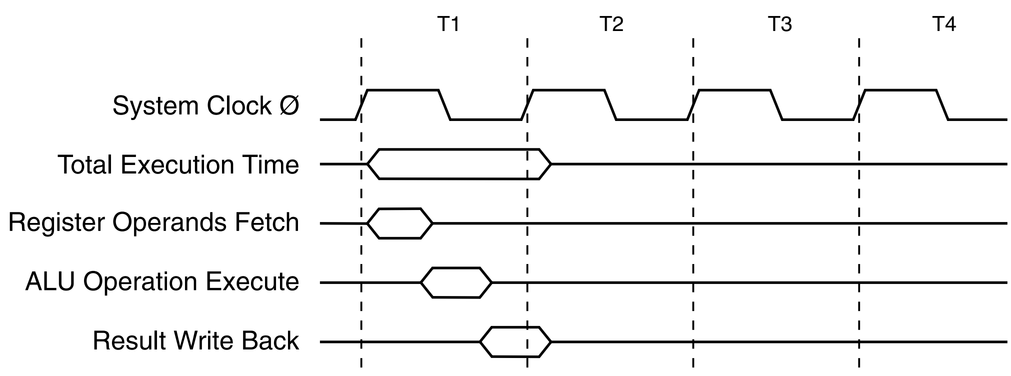


Figure 12 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 12. Single-cycle ALU Operation



I/O Memory

The I/O space definition of the AT90S1200 is shown in the following table.

Table 1. The AT90S1200 I/O Space

Address Hex	Name	Function
$3F	SREG	Status REGister
$3B	GIMSK	General Interrupt MaSK register
$39	TIMSK	Timer/Counter Interrupt MaSK register
$38	TIFR	Timer/Counter Interrupt Flag register
$35	MCUCR	MCU general Control Register
$33	TCCR0	Timer/Counter0 Control Register
$32	TCNT0	Timer/Counter0 (8-bit)
$21	WDTCR	Watchdog Timer Control Register
$1E	EEAR	EEPROM Address Register
$1D	EEDR	EEPROM Data Register
$1C	EECR	EEPROM Control Register
$18	PORTB	Data Register, Port B
$17	DDRB	Data Direction Register, Port B
$16	PINB	Input Pins, Port B
$12	PORTD	Data Register, Port D
$11	DDRD	Data Direction Register, Port D
$10	PIND	Input Pins, Port D
$08	ACSR	Analog Comparator Control and Status Register

Note: Reserved and unused locations are not shown in the table.

All AT90S1200 I/Os and peripherals are placed in the I/O space. The different I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

The different I/O and peripherals control registers are explained in the following sections.

Status Register – SREG

The AVR status register (SREG) at I/O space location $3F is defined as:

Bit	7	6	5	4	3	2	1	0	
$3F	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable bit is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 – T: Bit Copy Storage**

The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **Bit 5 – H: Half-carry Flag**

The half-carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set description for detailed information.

- **Bit 4 – S: Sign Bit, S = N⊕V**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set description for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set description for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set description for detailed information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

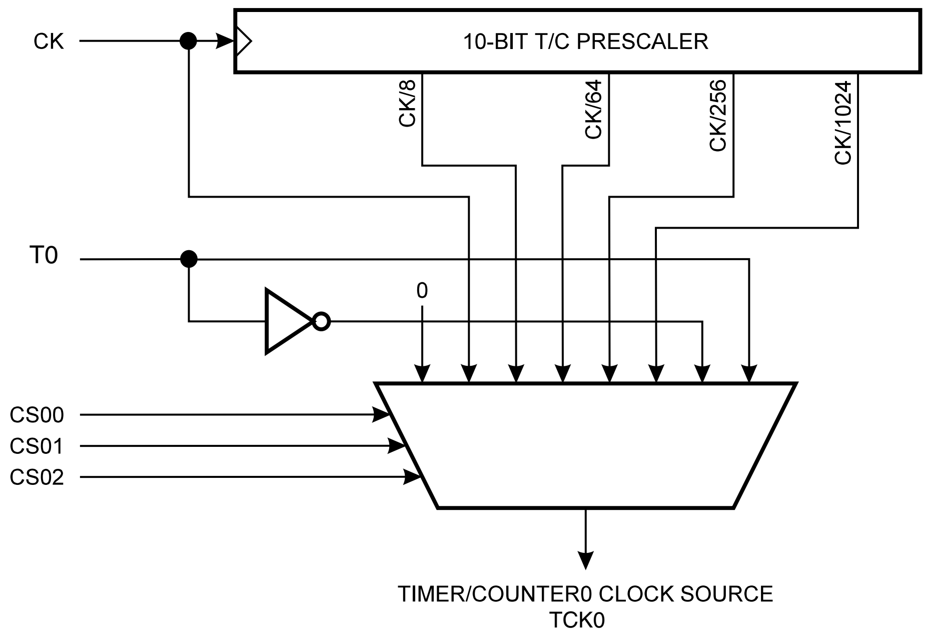
Timer/Counter0

The AT90S1200 provides one general purpose 8-bit Timer/Counter. The Timer/Counter0 gets the prescaled clock from the 10-bit prescaling timer. The Timer/Counter0 can either be used as a Timer with an internal clock time base or as a Counter with an external pin connection, which triggers the counting.

Timer/Counter0 Prescaler

Figure 18 shows the general Timer/Counter0 prescaler.

Figure 18. Timer/Counter0 Prescaler



The four different prescaled selections are: CK/8, CK/64, CK/256, and CK/1024 where CK is the Oscillator Clock. For the Timer/Counter0, added selections as CK, external clock source and stop, can be selected as clock sources. Figure 19 shows the block diagram for Timer/Counter0.

- **Bits 2, 1, 0 – CS02, CS01, CS00: Clock Select0, Bits 2, 1 and 0**

The Clock Select0 bits 2, 1 and 0 define the prescaling source of Timer/Counter0.

Table 5. Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK Oscillator clock. If the external pin modes are used for Timer/Counter0, transitions on PD4/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter0 – TCNT0

Bit	7	6	5	4	3	2	1	0	
$32	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the timer clock cycle following the write operation.

Table 6. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K cycles	47 ms	15 ms
0	0	1	32K cycles	94 ms	30 ms
0	1	0	64K cycles	0.19 s	60 ms
0	1	1	128K cycles	0.38 s	0.12 s
1	0	0	256K cycles	0.75 s	0,24 s
1	0	1	512K cycles	1.5 s	0.49 s
1	1	0	1,024K cycles	3.0 s	0.97 s
1	1	1	2,048K cycles	6.0 s	1.9 s

Note: The frequency of the Watchdog Oscillator is voltage dependent as shown in "Typical Characteristics" on page 51.

The WDR (Watchdog Reset) instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without Reset, the Watchdog Timer may not start to count from zero.

To avoid unintentional MCU resets, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

Port D

Three I/O memory address locations are allocated for Port D, one each for the Data Register – PORTD ($12), Data Direction Register – DDRD ($11), and the Port D Input Pins – PIND ($10). The Port D Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

Port D has seven bi-directional I/O pins with internal pull-up resistors, PD6..PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Some Port D pins have alternate functions as shown in Table 10.

Table 10. Port D Pin Alternate Functions

Port Pin	Alternate Function
PD2	INT0 (External Interrupt 0 input)
PD4	T0 (Timer/Counter 0 external input)

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
$12	–	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
$11	–	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
$10	–	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port D Input Pins address (PIND) is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read; and when reading PIND, the logical values present on the pins are read.

Port D as General Digital I/O

PDn, general I/O pin: The DDDn bit in the DDRD Register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PORTDn is set (one) when DDDn is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTDn bit has to be cleared (zero) or the pin has to be configured as an output pin. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Table 11. DDDn Bits' Effect on Port D Pins

DDDn	PORTDn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (High-Z)
0	1	Input	Yes	PDn will source current if ext. pulled low.
1	0	Output	No	Push-pull Zero Output
1	1	Output	No	Push-pull One Output

Note: n: 6…0, pin number.

Alternate Functions for Port D

The alternate functions of Port D are:

- **T0 – Port D, Bit 4**

T0, Timer/Counter0 clock source. See the timer description for further details.

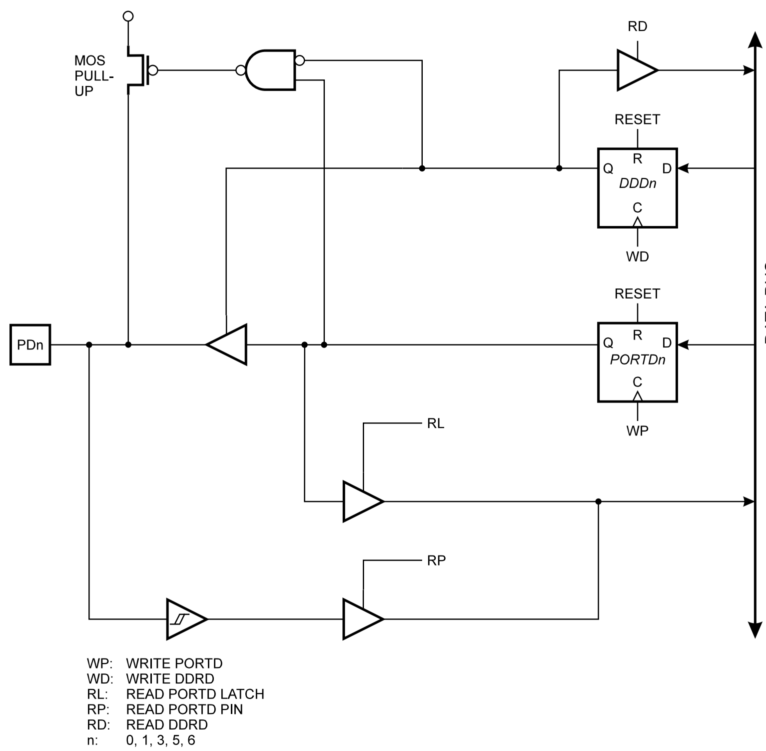
- **INT0 – Port D, Bit 2**

INT0, External Interrupt source 0. See the interrupt description for further details.

Port D Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 27. Port D Schematic Diagram (Pins PD0, PD1, PD3, PD5, and PD6)



Memory Programming

Program and Data Memory Lock Bits

The AT90S1200 MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 12. The Lock bits can only be erased with the Chip Erase command.

Table 12. Lock Bit Protection Modes

Memory Lock Bits			
Mode	LB1	LB2	Protection Type
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, further programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The AT90S1200 has two Fuse bits: SPIEN and RCEN.

- When the SPIEN Fuse bit is programmed (“0”), Serial Program Downloading is enabled. Default value is programmed (“0”).
- When the RCEN Fuse bit is programmed (“0”), MCU clocking from the Internal RC Oscillator is selected. Default value is erased (“1”). Parts with this bit pre-programmed (“0”) can be delivered on demand.
- The Fuse bits are not accessible in Serial Programming mode. The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a 3-byte signature code that identifies the device. This code can be read in both Serial and Parallel modes. The three bytes reside in a separate address space.

For the AT90S1200 they are:

1. $00: $1E (indicates manufactured by Atmel)
2. $01: $90 (indicates 1 Kb Flash memory)
3. $02: $01 (indicates AT90S1200 device when $01 is $90)

Note: When both Lock bits are programmed (lock mode 3), the signature bytes cannot be read in Serial mode. Reading the signature bytes will return: $00, $01 and $02.

Programming the Flash and EEPROM

Atmel’s AT90S1200 offers 1K byte of in-System Reprogrammable Flash program memory and 64 bytes of EEPROM data memory.

The AT90S1200 is normally shipped with the On-chip Flash program memory and EEPROM data memory arrays in the erased state (i.e., contents = $FF) and ready to be programmed. This device supports a High-voltage (12V) Parallel Programming mode and a Low-voltage Serial Programming mode. The +12V is used for programming enable only, and no current of significance is drawn by this pin. The Serial Programming mode provides a convenient way to download program and data into the AT90S1200 inside the user’s system.

The program and data memory arrays on the AT90S1200 are programmed byte-by-byte in either programming mode. For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction in the Serial Programming mode. During programming, the supply voltage must be in accordance with Table 13.

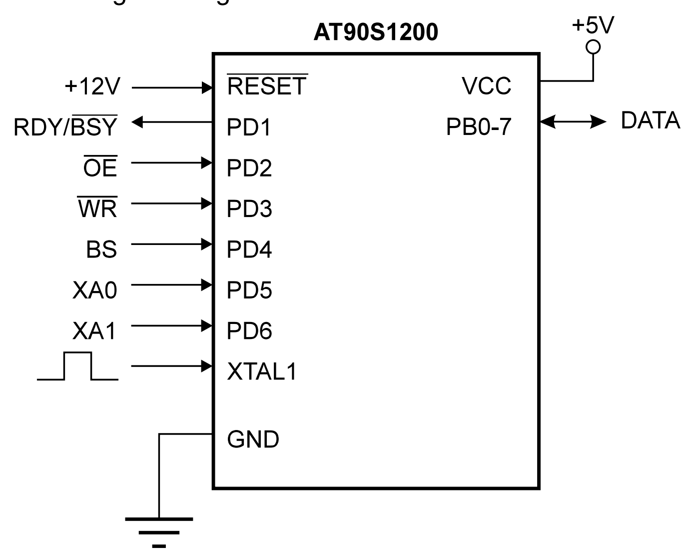
Table 13. Supply Voltage during Programming

Part	Serial Programming	Parallel Programming
AT90S1200	2.7 - 6.0V	4.5 - 5.5V

Parallel Programming

This section describes how to parallel program and verify Flash program memory, EEPROM data memory, Lock bits and Fuse bits in the AT90S1200.

Figure 30. Parallel Programming



Signal Names

In this section, some pins of the AT90S1200 are referenced by signal names describing their function during parallel programming rather than their pin names, see Figure 30 and Table 14. Pins not described in Table 14 are referenced by pin names.

The XA1/XA0 pins determines the action executed when the XTAL1 pin is given a positive pulse. The coding is shown in Table 15.

When pulsing $\overline{\text{WR}}$ or $\overline{\text{OE}}$, the command loaded determines the action executed. The command is a byte where the different bits are assigned functions as shown in Table 16.

Table 14. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{\text{BSY}}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command
$\overline{\text{OE}}$	PD2	I	Output Enable (Active low)
$\overline{\text{WR}}$	PD3	I	Write Pulse (Active low)
BS	PD4	I	Byte Select (“0” selects low byte, “1” selects high byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
DATA	PB0-7	I/O	Bi-directional Data Bus (Output when $\overline{\text{OE}}$ is low)

Table 15. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte for Flash determined by BS).
0	1	Load Data (High or low data byte for Flash determined by BS).
1	0	Load Command
1	1	No Action, Idle

Table 16. Command Byte Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse Bits
0010 0000	Write Lock Bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes
0000 0100	Read Fuse and Lock Bits
0000 0010	Read Flash
0000 0011	Read EEPROM

Enter Programming Mode

The following algorithm puts the device in Parallel Programming mode:

1. Apply supply voltage according to Table 13, between V_{CC} and GND.
2. Set the $\overline{\text{RESET}}$ and BS pin to "0" and wait at least 100 ns.
3. Apply 11.5 - 12.5V to $\overline{\text{RESET}}$. Any activity on BS within 100 ns after +12V has been applied to $\overline{\text{RESET}}$, will cause the device to fail entering Programming mode.

Chip Erase

The Chip Erase command will erase the Flash and EEPROM memories, and the Lock bits. The Lock bits are not Reset until the Flash and EEPROM have been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash or EEPROM is reprogrammed.

Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to "1000 0000". This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{\text{WR}}$ a t_{WLWH_CE} wide negative pulse to execute Chip Erase, t_{WLWH_CE} is found in Table 17. Chip Erase does not generate any activity on the RDY/$\overline{\text{BSY}}$ pin.

Programming the Flash

A: Load Command "Write Flash"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to "0001 0000". This is the command for Write Flash.

Figure 31. Programming the Flash Waveforms

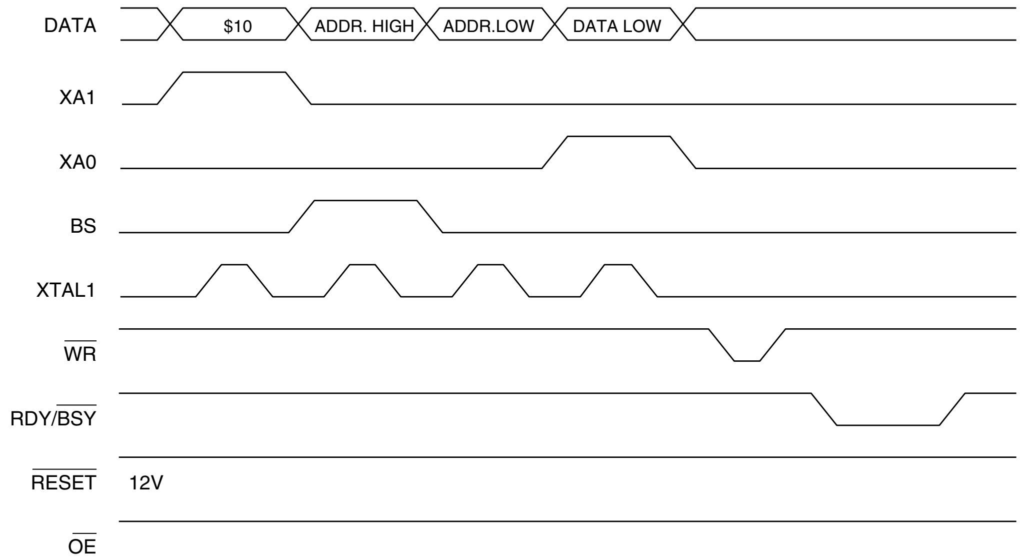
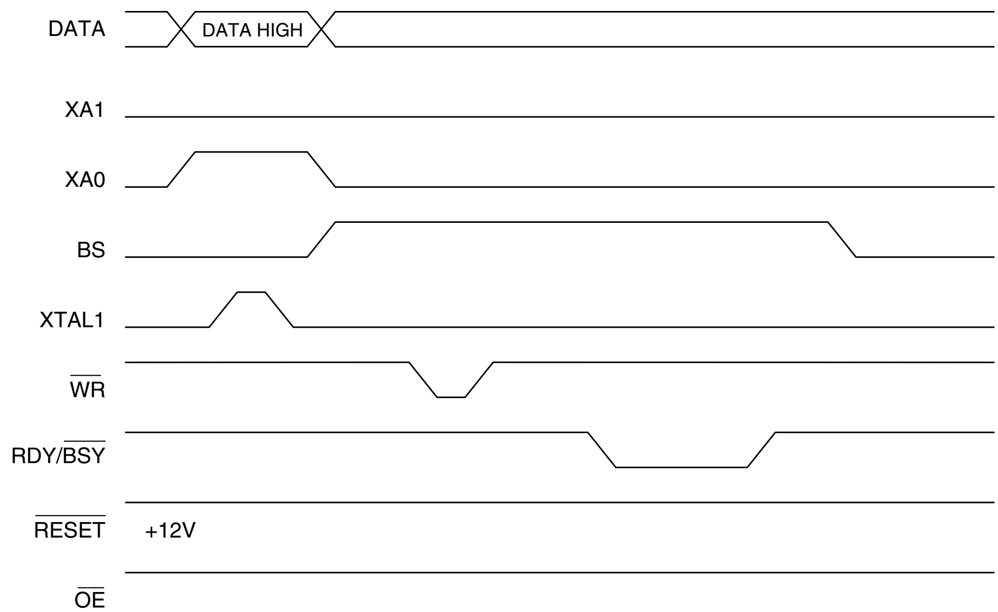


Figure 32. Programming the Flash Waveforms (Continued)



Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to "Programming the Flash" for details on command and address loading):

1. A: Load Command "0000 0010".
2. B: Load Address High Byte ($00 - $01).
3. C: Load Address Low Byte ($00 - $FF).
4. Set $\overline{OE}$ to "0", and BS to "0". The Flash word low byte can now be read at DATA.
5. Set BS to "1". The Flash word high byte can now be read from DATA.
6. Set $\overline{OE}$ to "1".

Programming the EEPROM

The programming algorithm for the EEPROM data memory is as follows (refer to "Programming the Flash" for details on command, address and data loading):

1. A: Load Command "0001 0001".
2. C: Load Address Low Byte ($00 - $3F).
3. D: Load Data Low Byte ($00 - $FF).
4. E: Write Data Low Byte.

Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to "Programming the Flash" for details on command and address loading):

1. A: Load Command "0000 0011".
2. C: Load Address Low Byte ($00 - $3F).
3. Set $\overline{OE}$ to "0", and BS to "0". The EEPROM data byte can now be read at DATA.
4. Set $\overline{OE}$ to "1".

Programming the Fuse Bits

The algorithm for programming the Fuse bits is as follows (refer to "Programming the Flash" for details on command and data loading):

1. A: Load Command "0100 0000".
2. D: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.

   Bit 5 = SPIEN Fuse
   Bit 0 = RCEN Fuse

   Bit 7 - 6, 4 - 1 = "1". These bits are reserved and should be left unprogrammed ("1").
3. Give $\overline{WR}$ a t_{WLWH_PFB} wide negative pulse to execute the programming; t_{WLWH_PFB} is found in Table 17. Programming the Fuse bits does not generate any activity on the RDY/$\overline{BSY}$ pin.

Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" for details on command and data loading):

1. A: Load Command "0010 0000".
2. D: Load Data Low Byte. Bit n = "0" programs the Lock bit.

   Bit 2 = Lock Bit2
   Bit 1 = Lock Bit1

   Bit 7 - 3, 0 = "1". These bits are reserved and should be left unprogrammed ("1").
3. E: Write Data Low Byte.

The Lock bits can only be cleared by executing Chip Erase.

Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 39 for details on command loading):

1. A: Load Command "0000 0100".
2. Set $\overline{OE}$ to "0", and BS to "1". The status of Fuse and Lock bits can now be read at DATA ("0" means programmed).

   Bit 7 = Lock Bit1
   Bit 6 = Lock Bit2
   Bit 5 = SPIEN Fuse
   Bit 0 = RCEN Fuse
3. Set $\overline{OE}$ to "1".

Observe especially that BS needs to be set to "1".

Reading the Signature Bytes

The algorithm for reading the signature bytes is as follows (refer to "Programming the Flash" on page 39 for details on command and address loading):

1. A: Load Command "0000 1000".
2. C: Load Address Low Byte ($00 - $02).
   Set $\overline{OE}$ to "0", and BS to "0". The selected signature byte can now be read at DATA.
   Set $\overline{OE}$ to "1".

Parallel Programming Characteristics

Figure 33. Parallel Programming Timing

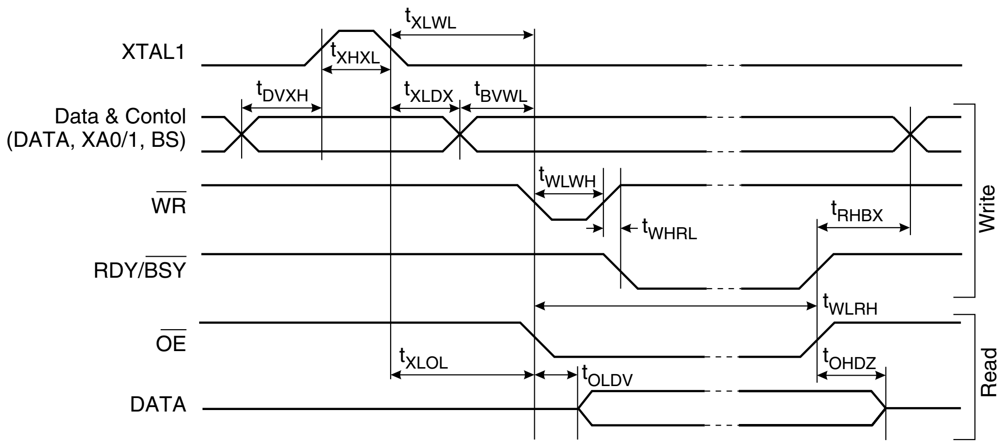


Table 17. Parallel Programming Characteristics, T_A = 25°C ± 10%, V_{CC} = 5V ± 10%

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250.0	µA
t_{DVXH}	Data and Control Setup before XTAL1 High	67.0			ns
t_{XHXL}	XTAL1 Pulse Width High	67.0			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67.0			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	67.0			ns
t_{BVWL}	BS Valid to $\overline{WR}$ Low	67.0			ns
t_{RHBX}	BS Hold after RDY/$\overline{BSY}$ High	67.0			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low$^{(1)}$	67.0			ns
t_{WHRL}	$\overline{WR}$ High to RDY/$\overline{BSY}$ Low$^{(2)}$		20.0		ns
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High$^{(2)}$	0.5	0.7	0.9	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	67.0			ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid		20.0		ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			20.0	ns
t_{WLWH_CE}	$\overline{WR}$ Pulse Width Low for Chip Erase	5.0	10.0	15.0	ms
t_{WLWH_PFB}	$\overline{WR}$ Pulse Width Low for Programming the Fuse Bits	1.0	1.5	1.8	ms

Notes: 1. Use t_{WLWH_CE} for chip erase and t_{WLWH_PFB} for programming the Fuse bits.
2. If t_{WLWH} is held longer than t_{WLRH}, no RDY/$\overline{BSY}$ pulse will be seen.

3. If a Chip Erase is performed (must be done to erase the Flash), wait t_{WD_ERASE} after the instruction, give $\overline{RESET}$ a positive pulse, and start over from step 2. See Table 21 on page 47 for t_{WD_ERASE} value.
4. The Flash or EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. Wait t_{WD_PROG} after transmitting the instruction. In an erased device, no $FFs in the data file(s) needs to be programmed. See Table 22 on page 47 for t_{WD_PROG} value.
5. Any memory location can be verified by using the Read instruction which returns the content at the selected address at the serial output MISO (PB6) pin.

   At the end of the programming session, $\overline{RESET}$ can be set high to commence normal operation.
6. Power-off sequence (if needed):

   Set XTAL1 to “0” (if a crystal is not used or the device is running from the Internal RC Oscillator).

   Set $\overline{RESET}$ to “1”.

   Turn V_{CC} power off.

Data Polling EEPROM

When a byte is being programmed into the EEPROM, reading the address location being programmed will give the value P1 until the auto-erase is finished, and then the value P2. See Table 18 for P1 and P2 values.

At the time the device is ready for a new EEPROM byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the values P1 and P2, so when programming these values, the user will have to wait for at least the prescribed time t_{WD_PROG} before programming the next byte. See Table 22 for t_{WD_PROG} value. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF can be skipped. This does not apply if the EEPROM is reprogrammed without first chip-erasing the device.

Table 18. Read Back Value during EEPROM Polling

Part	P1	P2
AT90S1200	$00	$FF

Data Polling Flash

When a byte is being programmed into the Flash, reading the address location being programmed will give the value $FF. At the time the device is ready for a new byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the value $FF, so when programming this value, the user will have to wait for at least t_{WD_PROG} before programming the next byte. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF, can be skipped.

Figure 53. I/O Pin Sink Current vs. Output Voltage

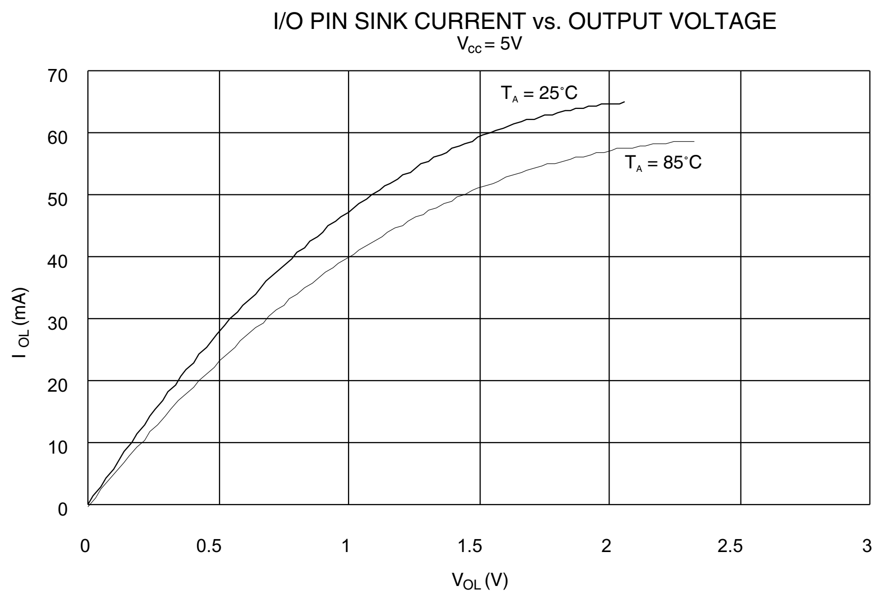
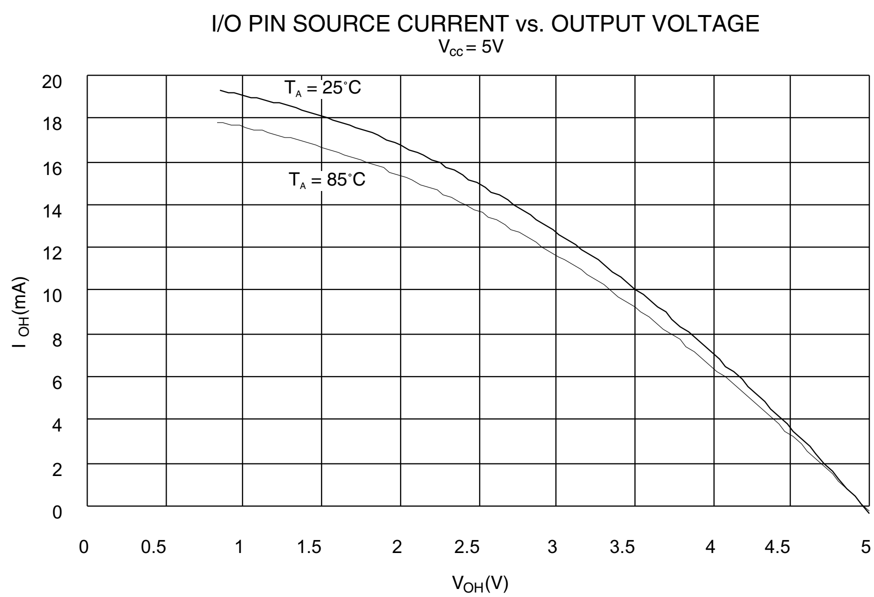


Figure 54. I/O Pin Source Current vs. Output Voltage



Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	# Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add Two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry Two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract Two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry Two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd, K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd, K	Clear Bit(s) in Register	Rd ← Rd • (FFh - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd, Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1/2
CP	Rd, Rr	Compare	Rd - Rr	Z,N,V,C,H	1
CPC	Rd, Rr	Compare with Carry	Rd - Rr - C	Z,N,V,C,H	1
CPI	Rd, K	Compare Register with Immediate	Rd - K	Z,N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) PC ← PC + 2 or 3	None	1/2
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b) = 1) PC ← PC + 2 or 3	None	1/2
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)= 0) PC ← PC + 2 or 3	None	1/2
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b) = 1) PC ← PC + 2 or 3	None	1/2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC ← PC + k + 1	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC ← PC + k + 1	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V = 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Branch if Less than Zero, Signed	if (N ⊕ V = 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Branch if Half-carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Branch if Half-carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	Branch if T-Flag Set	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	Branch if T-Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
DATA TRANSFER INSTRUCTIONS					
LD	Rd, Z	Load Register Indirect	Rd ← (Z)	None	2
ST	Z, Rr	Store Register Indirect	(Z) ← Rr	None	2
MOV	Rd, Rr	Move between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1

Ordering Information[1]

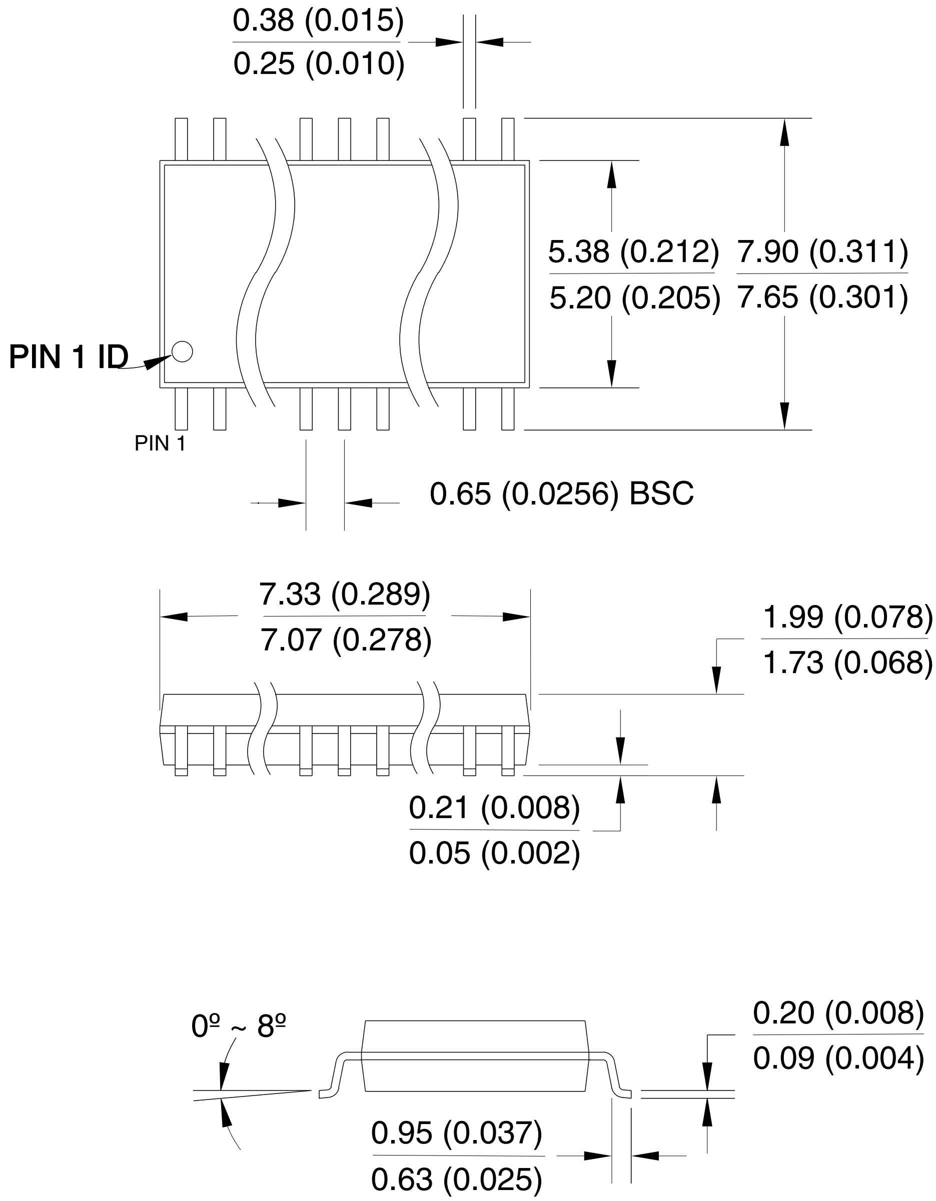
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
4	2.7 - 6.0V	AT90S1200-4PC AT90S1200-4SC AT90S1200-4YC	20P3 20S 20Y	Commercial (0°C to 70°C)
		AT90S1200-4PI AT90S1200-4SI AT90S1200-4YI	20P3 20S 20Y	Industrial (-40°C to 85°C)
12	4.0 - 6.0V	AT90S1200-12PC AT90S1200-12SC AT90S1200-12YC	20P3 20S 20Y	Commercial (0°C to 70°C)
		AT90S1200-12PI AT90S1200-12SI AT90S1200-12YI	20P3 20S 20Y	Industrial (-40°C to 85°C)

Note: 1. Order AT90S1200A-XXX for devices with the RCEN Fuse programmed.

Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
20Y	20-lead, 5.3 mm Wide, Plastic Shrink Small Outline Package (SSOP)

20Y

20Y, 20-lead Plastic Shrink Small Outline (SSOP), 5.3mm body Width. Dimensions in Millimeters and (inches)*



*Controlling dimension: millimeters

REV. A 04/11/2001