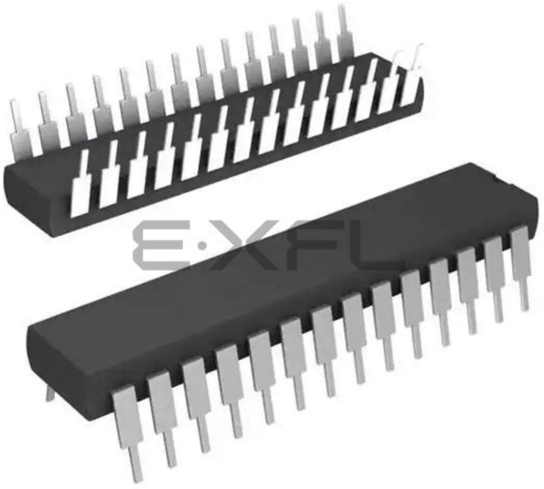
Microchip Technology - AT90S4433-8PC Datasheet



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	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	20
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	A/D 6x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s4433-8pc

Description

The AT90S4433 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S4433 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 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.

The AT90S4433 provides the following features: 4K bytes of In-System Programmable Flash, 256 bytes of EEPROM, 128 bytes of SRAM, 20 general purpose I/O lines, 32 general purpose working registers, two flexible Timer/Counters with compare modes, internal and external interrupts, a programmable serial UART, 6-channel, 10-bit ADC, programmable Watchdog Timer with internal Oscillator, an SPI serial port and two software-selectable Power-saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset.

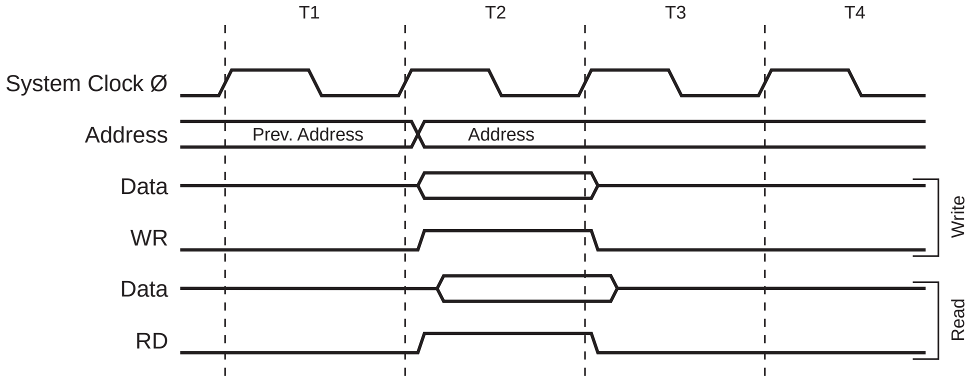
The device is manufactured using Atmel’s high-density non-volatile memory technology. The On-chip Flash Program memory can be re-programmed In-System through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a RISC 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S4433 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The AT90S4433 AVR is supported with a full suite of program and system development tools including: C Compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators and evaluation kits.

Table 1. Comparison Table

Device	Flash	EEPROM	SRAM	Voltage Range	Frequency
AT90S4433	4K	256B	128B	4.0V - 6.0V	0 - 8 MHz
AT90LS4433	4K	256B	128B	2.7V - 6.0V	0 - 4 MHz

Figure 23. On-chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the AT90S4433 is shown in Table 2.

Table 2. AT90S4433 I/O Space[1]

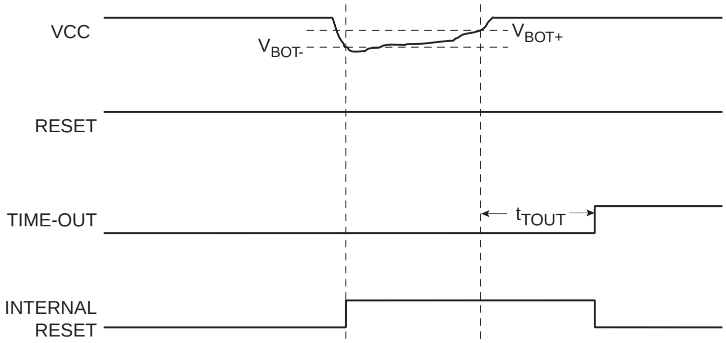
I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status Register
$3D ($5D)	SP	Stack Pointer
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU general Control Register
$34 ($54)	MCUSR	MCU general Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)
$2F ($4F)	TCCR1A	Timer/Counter1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter1 Low Byte
$2B ($4B)	OCR1H	Timer/Counter1 Output Compare Register High Byte
$2A ($4A)	OCR1L	Timer/Counter1 Output Compare Register Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1E ($3E)	EEAR	EEPROM Address Register
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register
$18 ($38)	PORTB	Data Register, Port B

Brown-out Detection

AT90S4433 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during the operation. The power supply must be decoupled with a 47 nF to 100 nF capacitor if the BOD function is used. The BOD circuit can be enabled/disabled by the fuse BODEN. When BODEN is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the Brown-out Reset is deactivated after a delay. The delay is defined by the user in the same way as the delay of POR signal (see Table 5). The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis of 50 mV to ensure spike-free Brown-out Detection.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than 3 µs for trigger level 4.0V, 7 µs for trigger level 2.7V (typical values).

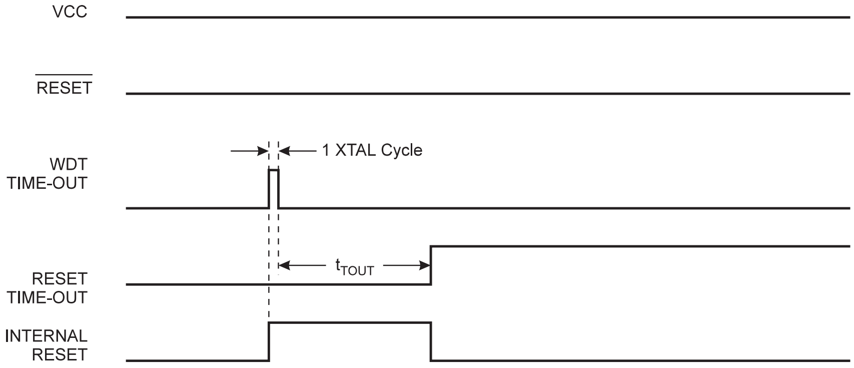
Figure 28. Brown-out Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period (t_{TOUT}). See page 43 for details on operation of the Watchdog.

Figure 29. Watchdog Reset during Operation



Interrupt Flag Register (TIFR). The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter1 is externally clocked, the external signal is synchronized with the Oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

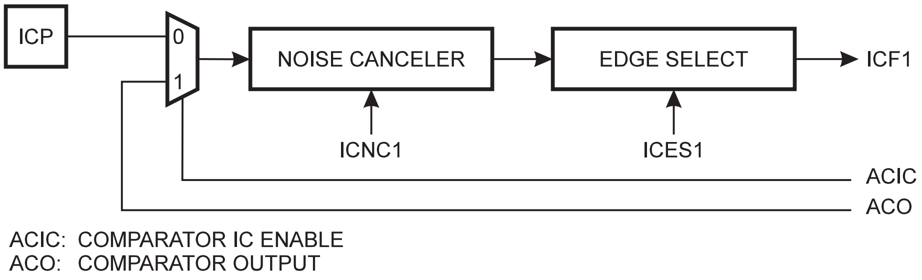
The 16-bit Timer/Counter1 features both a high resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports an Output Compare function using the Output Compare Register1 (OCR1) as the data source to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compare matches and actions on the Output Compare pin 1 on compare matches.

Timer/Counter1 can also be used as a 8-, 9-, or 10-bit Pulse Width Modulator. In this mode, the counter and the OCR1 Register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 41 for a detailed description of this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register (ICR1), triggered by an external event on the Input Capture Pin (ICP). The actual capture event settings are defined by the Timer/Counter1 Control Register (TCCR1). In addition, the Analog Comparator can be set to trigger the Input Capture. Refer to the section, "The Analog Comparator", for details of this. The ICP pin logic is shown in Figure 33.

Figure 33. ICP Pin Schematic Diagram



If the Noise Canceler function is enabled, the actual trigger condition for the capture event is monitored over four samples, and all four must be equal to activate the Capture Flag. The input pin signal is sampled at XTAL clock frequency.

the TEMP Register. Consequently, the Low Byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

Timer/Counter1 Output Compare Register – OCR1H and OCR1L

Bit	15	14	13	12	11	10	9	8	
$2B ($4B)	**MSB**								**OCR1H**
$2A ($4A)								**LSB**	**OCR1L**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

The Output Compare Register is a 16-bit read/write register.

The Timer/Counter1 Output Compare Register contains the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status Register.

Since the Output Compare Register (OCR1) is a 16-bit register, a temporary register TEMP is used when OCR1 is written to ensure that both bytes are updated simultaneously. When the CPU writes the High Byte, OCR1H, the data is temporarily stored in the TEMP Register. When the CPU writes the Low Byte, OCR1L, the TEMP Register is simultaneously written to OCR1H. Consequently, the High Byte OCR1H must be written first for a full 16-bit register write operation.

The TEMP Register is also used when accessing TCNT1 and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program.

Timer/Counter1 Input Capture Register – ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($47)	**MSB**								**ICR1H**
$26 ($46)								**LSB**	**ICR1L**
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Input Capture Register is a 16-bit, read only register.

When the rising or falling edge (according to the input capture edge setting [ICES1]) of the signal at the Input Capture Pin (ICP) is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register (ICR1). At the same time, the Input Capture Flag (ICF1) is set (one).

Since the Input Capture Register (ICR1) is a 16-bit register, a temporary register (TEMP) is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the Low Byte, ICR1L, the data is sent to the CPU and the data of the High Byte, ICR1H, is placed in the TEMP Register. When the CPU reads the data in the High Byte, ICR1H, the CPU receives the data in the TEMP Register. Consequently, the Low Byte, ICR1L, must be accessed first for a full 16-bit register read operation.

The TEMP Register is also used when accessing TCNT1 and OCR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program.

Timer/Counter1 in PWM Mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1 (OCR1) form a 8-, 9-, or 10-bit, free-running, glitch-free, phase correct PWM with output on the PB1(OC1) pin. Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Table 13), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 8, 9, or 10 least significant bits of OCR1, the PB1(OC1) pin is set or cleared according to the settings of the COM11 and COM10 bits in the Timer/Counter1 Control Register (TCCR1). Refer to Table 14 for details.

Table 13. Timer TOP Values and PWM Frequency(1)

PWM Resolution	Timer TOP Value	Frequency
8-bit	$00FF (255)	f_{TCK1}/510
9-bit	$01FF (511)	f_{TCK1}/1022
10-bit	$03FF(1023)	f_{TCK1}/2046

Note: 1. If the Compare Register contains the TOP value and the prescaler is not in use (CS12..CS10 = 001), the PWM output will not produce any pulse at all, because the up-counting and down-counting values are reached simultaneously. When the prescaler is in use (CS12..CS10 $\neq$ 001 or 000), the PWM output goes active when the counter reaches the TOP value, but the down-counting compare match is not interpreted to be reached before the next time the counter reaches the TOP value, making a one-period PWM pulse.

Prevent EEPROM Corruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board-level systems using the EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Second, the CPU itself can execute instructions incorrectly if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating speed matches the detection level. If not, an external low V_{CC} Reset Protection circuit can be applied.
2. Keep the AVR core in Power-down sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM Registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory cannot be updated by the CPU and will not be subject to corruption.

Multi-processor Communication Mode

The Multi-processor Communication mode enables several slave MCUs to receive data from a Master MCU. This is done by first decoding an address byte to find out which MCU has been addressed. If a particular slave MCU has been addressed, it will receive the following data bytes as normal, while the other slave MCUs will ignore the data bytes until another address byte is received.

For an MCU to act as a Master MCU, it should enter 9-bit Transmission mode (CHR9 in UCSRB set). The ninth bit must be one to indicate that an address byte is being transmitted, and zero to indicate that a data byte is being transmitted.

For the Slave MCUs, the mechanism appears slightly differently for 8-bit and 9-bit Reception mode. In 8-bit Reception mode (CHR9 in UCSRB cleared), the stop bit is one for an address byte and zero for a data byte. In 9-bit Reception mode (CHR9 in UCSRB set), the ninth bit is one for an address byte and zero for a data byte, whereas the stop bit is always high.

The following procedure should be used to exchange data in Multi-processor Communication mode:

1. All slave MCUs are in Multi-processor Communication mode (MPCM in UCSRA is set).
2. The Master MCU sends an address byte, and all Slaves receive and read this byte. In the slave MCUs, the RXC Flag in UCSRA will be set as normal.
3. Each Slave MCU reads the UDR Register and determines if it has been selected. If so, it clears the MPCM bit in UCSRA, otherwise it waits for the next address byte.
4. For each received data byte, the receiving MCU will set the Receive Complete Flag (RXC in UCSRA). In 8-bit mode, the receiving MCU will also generate a Framing Error (FE in UCSRA set), since the stop bit is zero. The other Slave MCUs, which still have the MPCM bit set, will ignore the data byte. In this case, the UDR Register and the RXC or FE Flags will not be affected.
5. After the last byte has been transferred, the process repeats from step 2.

Scanning Multiple Channels

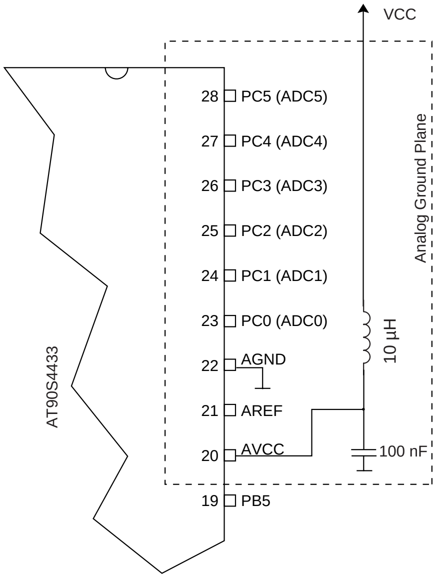
Since change of analog channel always is delayed until a conversion is finished, the Free Run mode can be used to scan multiple channels without interrupting the converter. Typically, the ADC Conversion Complete interrupt will be used to perform the channel shift. However, the user should take the following fact into consideration: The interrupt triggers once the result is ready to be read. In Free Run mode, the next conversion will start immediately when the interrupt triggers. If ADMUX is changed after the interrupt triggers, the next conversion has already started and the old setting is used.

ADC Noise Canceling Techniques

Digital circuitry inside and outside the AT90S4433 generates EMI, which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the AT90S4433 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane and keep them well away from high-speed switching digital tracks.
3. The AVCC pin on the AT90S4433 should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 49.
4. Use the ADC Noise Canceler function to reduce induced noise from the CPU.
5. If some Port C pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 49. ADC Power Connections



Note that since AVCC feeds the Port C output drivers, the RC network shown should not be employed if any Port C serve as outputs.

controlled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See the description of the SPI port for further details.

- **OC1 – Port B, Bit 1**

OC1, Output Compare Match output: PB1 pin can serve as an external output for the Timer/Counter1 Output Compare. The pin has to be configured as an output (DDB1 set [one]) to serve this function. See the timer description on how to enable this function. The OC1 pin is also the output pin for the PWM mode timer function.

- **ICP – Port B, Bit 0**

ICP, Input Capture Pin: PB0 pin can serve as an external input for the Timer/Counter1 input capture. The pin has to be configured as an input (DDB0 cleared [zero]) to serve this function. See the timer description on how to enable this function.

Figure 50. Port B Schematic Diagram (Pin PB0)

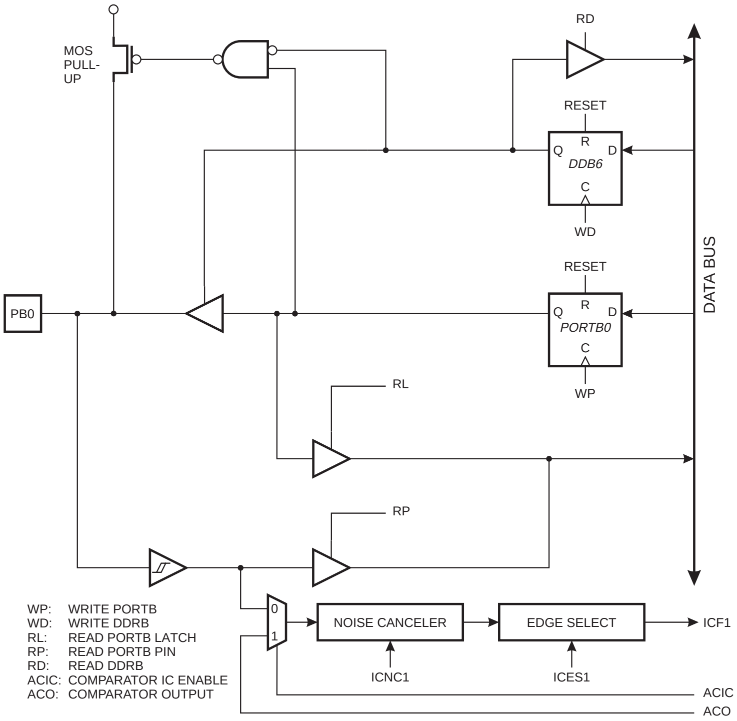
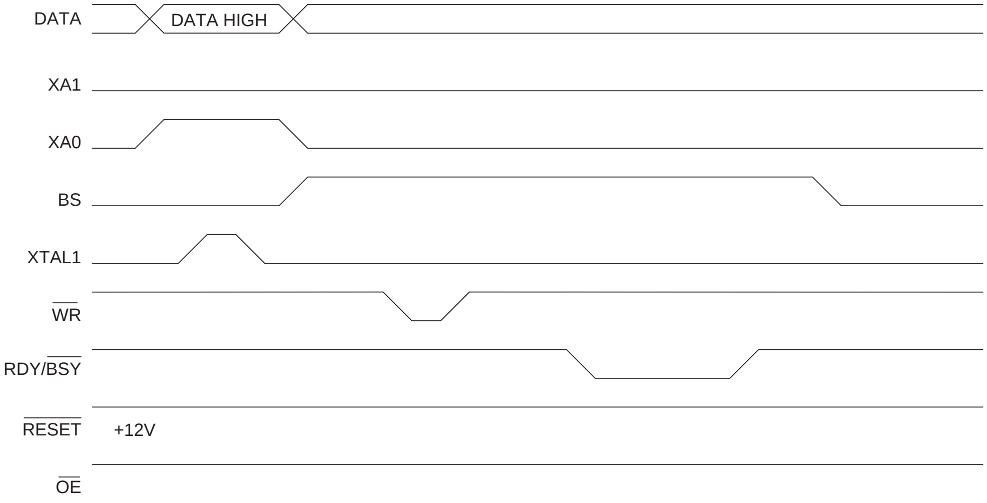


Figure 64. Programming the Flash Waveforms (Continued)



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):

A: Load Command "0000 0010".

B: Load Address High Byte ($00 - $07).

C: Load Address Low Byte ($00 - $FF).

1. Set $\overline{OE}$ to "0", and BS to "0". The Flash word Low Byte can now be read at DATA.
2. Set BS to "1". The Flash word High Byte can now be read from DATA.
3. 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):

A: Load Command "0001 0001".

B: Load Address Low Byte ($00 - $FF).

C: Load Data Low Byte ($00 - $FF).

D: 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):

A: Load Command "0000 0011".

B: Load Address Low Byte ($00 - $FF).

1. Set $\overline{OE}$ to "0", and BS to "0". The EEPROM data byte can now be read at DATA.
2. Set $\overline{OE}$ to "1".

Reading the Signature Bytes

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

A: Load Command “0000 1000”.

B: Load Address Low Byte ($00 - $02).

1. Set $\overline{OE}$ to “0”, and BS to “0”. The selected signature byte can now be read at DATA.
2. Set $\overline{OE}$ to “1”.

Parallel Programming Characteristics

Figure 65. Parallel Programming Timing

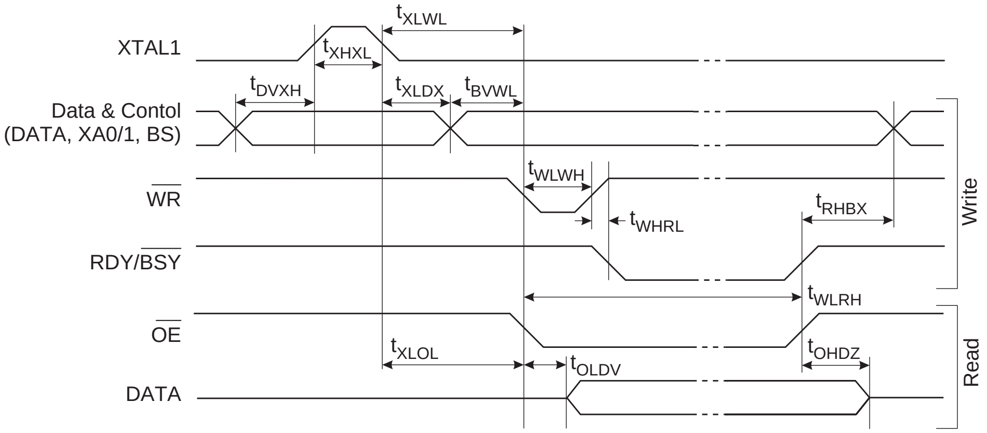


Table 33. Parallel Programming Characteristics T_A = 25°C ± 10%, V_{CC} = 5V ± 10%

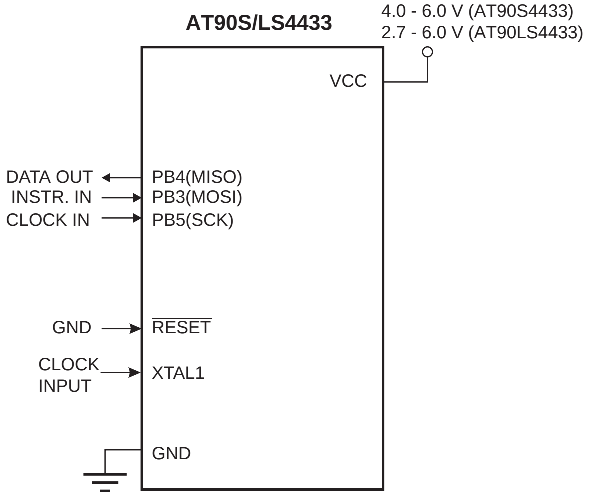
Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250.0	µA
t_{DVXH}	Data and Control Setup before XTAL1 High	67.0			ns
t_{XHXL}	XTAL1 Pulse Width High	67.0			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67.0			ns
t_{XLWL}	XTAL1 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.

Serial Downloading

Both the Program and Data memory arrays can be programmed using the SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output) (see Figure 66). After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 66. Serial Programming and Verify



For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content of every memory location in both the program and EEPROM arrays into $FF.

The Program and EEPROM memory arrays have separate address spaces: 0000 to $07FF for Program memory and $0000 to $00FF for EEPROM memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

Serial Programming Algorithm

When writing serial data to the AT90S4433, data is clocked on the rising edge of CLK.

When reading data from the AT90S4433, data is clocked on the falling edge of CLK. See Figure 67, Figure 68 and Table 36 for details.

To program and verify the AT90S4433 in the Serial Programming mode, the following sequence is recommended (see 4-byte instruction formats in Table 35):

1. Power-up sequence:
   Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ and SCK are set to "0". If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. In some systems, the programmer cannot guarantee that SCK is held low during Power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse of at least two XTAL1 cycles' duration after SCK has been set to "0".
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB3.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte ($53) will echo back when issu-

Figure 71. Active Supply Current vs. V_{CC}

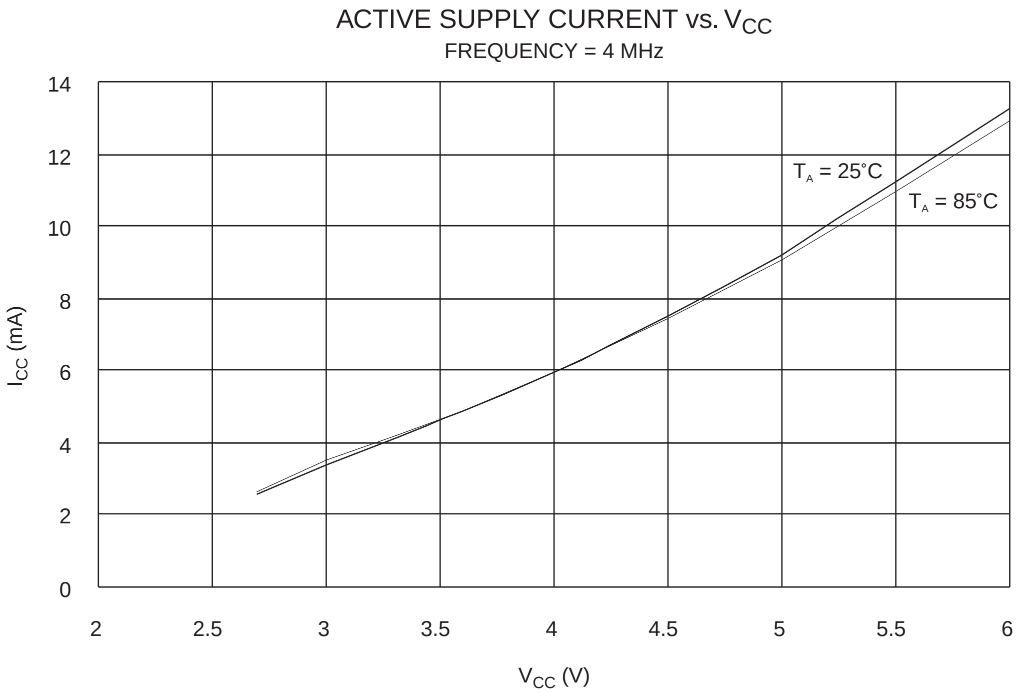


Figure 72. Idle Supply Current vs. Frequency

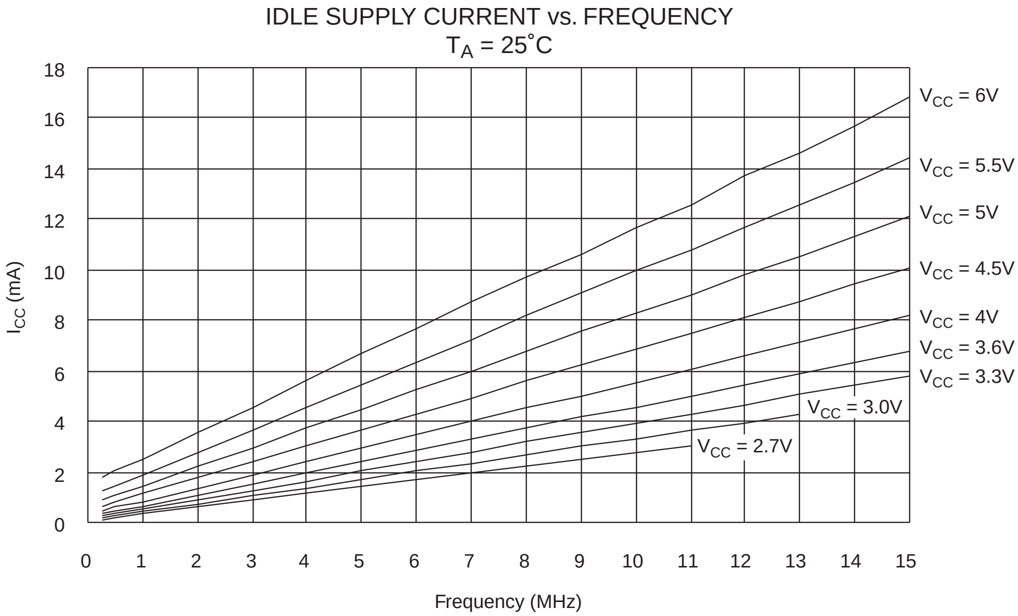


Figure 75. Power-down Supply Current vs. V_{CC}

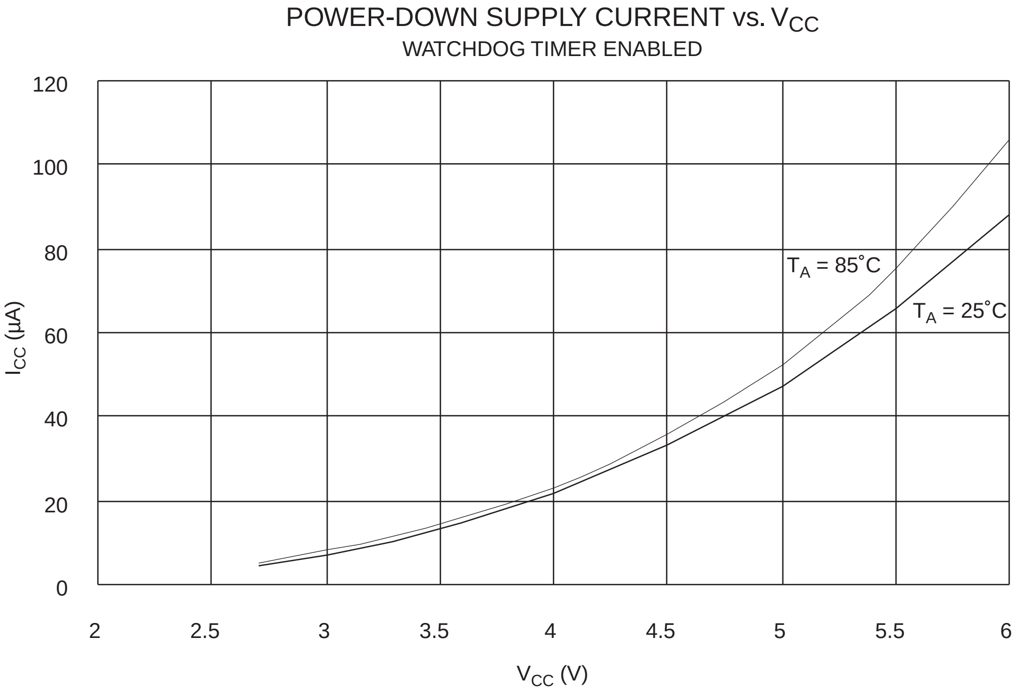
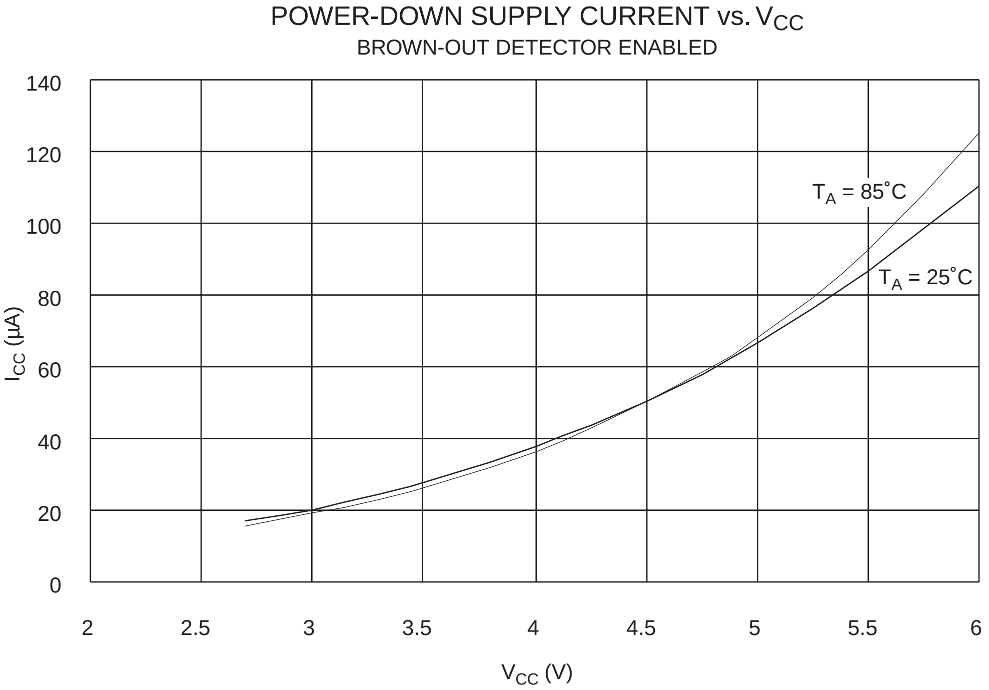


Figure 76. Power-down Supply Current vs. V_{CC}



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	page 19
$3E ($5E)	Reserved	–	–	–	–	–	–	–	–	page 20
$3D ($5D)	SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 20
$3C ($5C)	Reserved									
$3B ($5B)	GIMSK	INT1	INT0	–	–	–	–	–	–	page 27
$3A ($5A)	GIFR	INTF1	INTF0							page 27
$39 ($59)	TIMSK	TOIE1	OCIE1	–	–	TICIE1	–	TOIE0	–	page 28
$38 ($58)	TIFR	TOV1	OCF1	–	–	ICF1	–	TOV0	–	page 29
$37 ($57)	Reserved									
$36 ($56)	Reserved									
$35 ($55)	MCUCR	–		SE	SM	ISC11	ISC10	ISC01	ISC00	page 30
$34 ($54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	page 26
$33 ($53)	TCCR0	–	–	–	–	–	CS02	CS01	CS00	page 34
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								page 35
$31 ($51)	Reserved									
$30 ($50)	Reserved									
$2F ($4F)	TCCR1A	COM11	COM10	–	–	–	–	PWM11	PWM10	page 37
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	page 38
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								page 39
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								page 39
$2B ($4B)	OCR1H	Timer/Counter1 – Output Compare Register High Byte								page 40
$2A ($4A)	OCR1L	Timer/Counter1 – Output Compare Register Low Byte								page 40
$29 ($49)	Reserved									
$28 ($48)	Reserved									
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								page 41
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								page 41
$25 ($45)	Reserved									
$24 ($44)	Reserved									
$23 ($43)	Reserved									
$22 ($42)	Reserved									
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	page 43
$20 ($40)	Reserved									
$1F ($3F)	Reserved									
$1E ($3E)	EEAR	EEPROM Address Register								page 45
$1D ($3D)	EEDR	EEPROM Data Register								page 45
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	page 45
$1B ($3B)	Reserved									
$1A ($3A)	Reserved									
$19 ($39)	Reserved									
$18 ($38)	PORTB	–	–	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 72
$17 ($37)	DDRB	–	–	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	page 72
$16 ($36)	PINB	–	–	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 72
$15 ($35)	PORTC	–	–	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	page 78
$14 ($34)	DDRC	–	–	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	page 78
$13 ($33)	PINC	–	–	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	page 78
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 80
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 80
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 80
$0F ($2F)	SPDR	SPI Data Register								page 52
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	–	page 52
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 51
$0C ($2C)	UDR	UART I/O Data Register								page 57
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	OR	–	–	–	page 57
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 58
$09 ($29)	UBRR	UART Baud Rate Register								page 61
$08 ($28)	ACSR	ACD	AINBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 62
$07 ($27)	ADMUX	–	ADCBG	–	–	–	MUX2	MUX1	MUX0	page 68
$06 ($26)	ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 68
$05 ($25)	ADCH	–	–	–	–	–	–	ADC9	ADC8	page 69
$04 ($24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	page 69
$03 ($23)	UBRRHI					UART Baud Rate Register High				page 61
$02 ($22)	Reserved									
$01 ($21)	Reserved									
$00 ($20)	Reserved									

Notes: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

2. Some of the Status Flags are cleared by writing a logical "1" 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.

Table of Contents