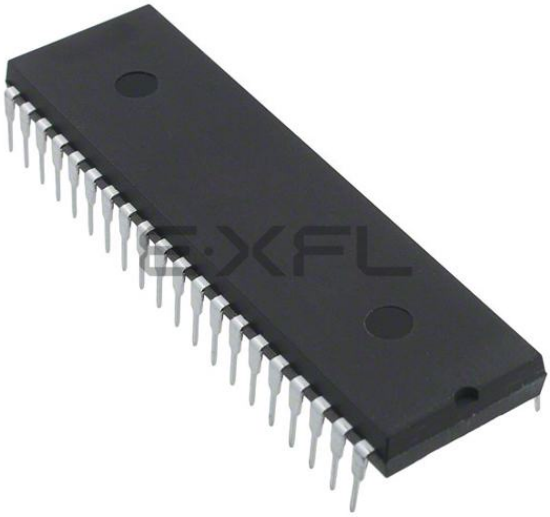


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

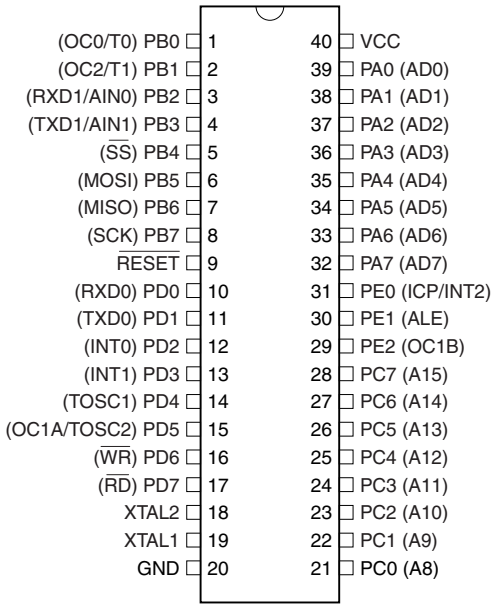
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

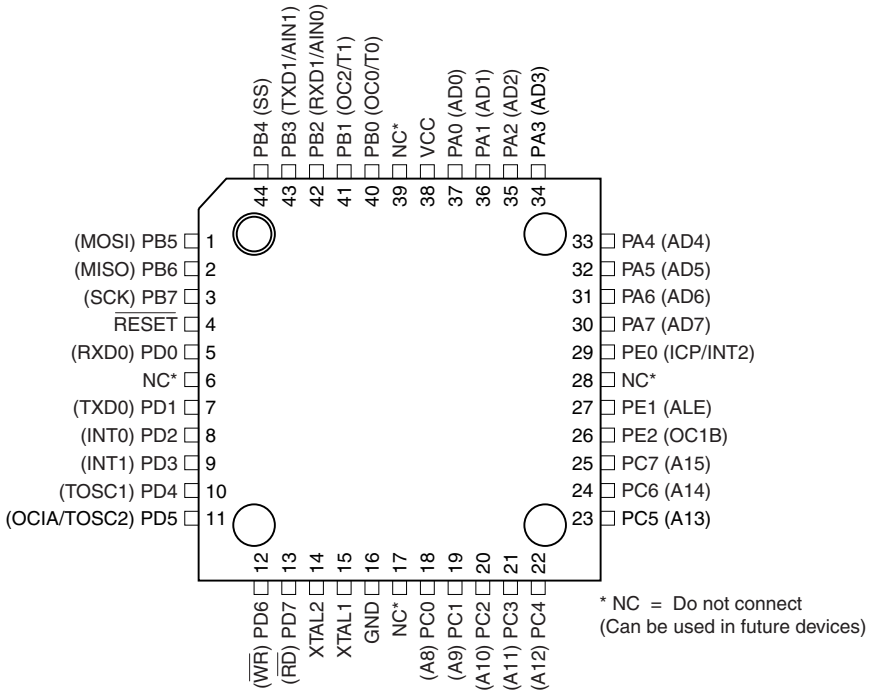
Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega161-8pi

Pin Configuration

PDIP



TQFP



Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip Oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

Figure 2. Oscillator Connections

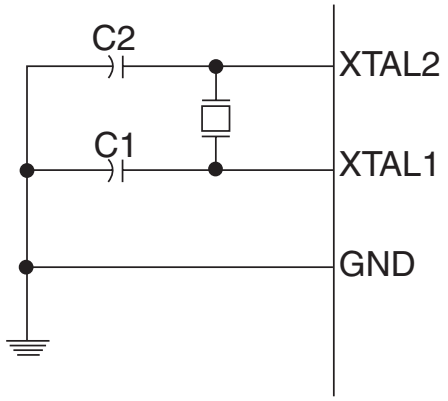
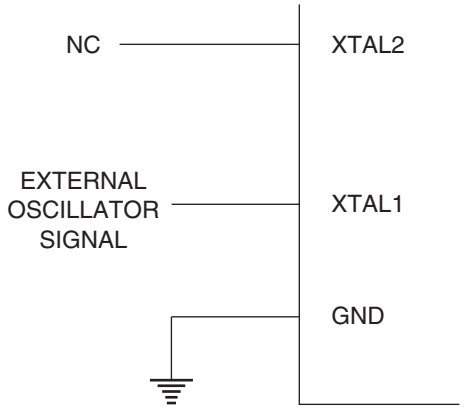


Figure 3. External Clock Drive Configuration



I/O Memory

The I/O space definition of the ATmega161 is shown in Table 1.

Table 1. ATmega161 I/O Space[1]

I/O Address (SRAM Address)	Name	Function
$3F($5F)	SREG	Status REGister
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$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
$37 ($57)	SPMCR	Store Program Memory Control Register
$36 ($56)	EMCUCR	Extended MCU general Control 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)
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register
$30 ($50)	SFIOR	Special Function IO Register
$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)	OCR1AH	Timer/Counter1 Output Compare RegisterA High Byte
$2A ($4A)	OCR1AL	Timer/Counter1 Output Compare RegisterA Low Byte
$29 ($49)	OCR1BH	Timer/Counter1 Output Compare RegisterB High Byte
$28 ($48)	OCR1BL	Timer/Counter1 Output Compare RegisterB Low Byte
$27 ($47)	TCCR2	Timer/Counter2 Control Register
$26 ($46)	ASSR	Asynchronous mode StatuS Register
$25 ($45)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$24 ($44)	ICR1L	Timer/Counter1 Input Capture Register Low Byte
$23 ($43)	TCNT2	Timer/Counter2 (8-bit)
$22 ($42)	OCR2	Timer/Counter2 Output Compare Register
$21 ($41)	WDTCR	Watchdog Timer Control Register
$20 ($40)	UBRRHI	UART Baud Register HIgh
$1F ($3F)	EEARH	EEPROM Address Register High
$1E ($3E)	EEARL	EEPROM Address Register Low
$1D ($3D)	EEDR	EEPROM Data Register

- **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.

Stack Pointer – SP

The ATmega161 Stack Pointer is implemented as two 8-bit registers in the I/O space locations $3E ($5E) and $3D ($5D). As the ATmega161 supports up to 64-Kbyte memory, all 16 bits are used.

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	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 Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above $60. The Stack Pointer is decremented by 1 when data is pushed onto the Stack with the PUSH instruction, and it is decremented by 2 when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by 1 when data is popped from the Stack with the POP instruction, and it is incremented by 2 when an address is popped from the Stack with return from subroutine RET or return from interrupt (RETI).

Reset and Interrupt Handling

The ATmega161 provides 20 different interrupt sources. These interrupts and the separate Reset Vector each have a separate Program Vector in the Program memory space. All interrupts are assigned individual enable bits that must be set (one) together with the I-bit in the Status Register in order to enable the interrupt.

The lowest addresses in the Program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of Vectors is shown in Table 2. The list also determines the priority levels of the different interrupts. The lower the address, the higher the priority level. RESET has the highest priority, and next is INT0 (the External Interrupt Request 0) and so on.

- **Bit 5 – OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Match interrupt is enabled. The corresponding interrupt (at Vector $010) is executed if a Compare B Match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 4 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at Vector $00a) is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 3 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at Vector $00C) is executed if a capture-triggering event occurs on pin 31, ICP, i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 2 – OCIE2: Timer/Counter2 Output Compare Match Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match interrupt is enabled. The corresponding interrupt (at Vector $008) is executed if a Compare2 match in Timer/Counter2 occurs, i.e., when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at Vector $016) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 0 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable**

When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at Vector $014) is executed if a Compare 0 Match in Timer/Counter0 occurs, i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bits 5, 4 – COM1B1, COM1B0: Compare Output Mode1B, Bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1B (Output CompareB). This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

Table 14. Compare 1 Mode Select(1)

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Note: 1. X = A or B

In PWM mode, these bits have a different function. Refer to Table 18 for a detailed description.

- **Bit 3 – FOC1A: Force Output Compare1A**

Writing a logical “1” to this bit forces a change in the compare match output pin PD5 according to the values already set in COM1A1 and COM1A0. If the COM1A1 and COM1A0 bits are written in the same cycle as FOC1A, the new settings will not take effect until the next compare match or forced compare match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a compare match in the timer. The automatic action programmed in COM1A1 and COM1A0 happens as if a Compare Match had occurred, but no interrupt is generated and it will not clear the timer even if CTC1 in TCCR1B is set. The FOC1A bit will always be read as zero. The setting of the FOC1A bit has no effect in PWM mode.

- **Bit 2 – FOC1B: Force Output Compare1B**

Writing a logical “1” to this bit forces a change in the compare match output pin PE2 according to the values already set in COM1B1 and COM1B0. If the COM1B1 and COM1B0 bits are written in the same cycle as FOC1B, the new settings will not take effect until the next compare match or forced compare match occurs. The Force Output Compare bit can be used to change the output pin without waiting for a compare match in the timer. The automatic action programmed in COM1B1 and COM1B0 happens as if a Compare Match had occurred, but no interrupt is generated. The FOC1B bit will always be read as zero. The setting of the FOC1B bit has no effect in PWM mode.

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 Register (TCNT1) is transferred to the Input Capture Register (ICR1). In the same cycle, 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, OCR1A and OCR1B. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program and interrupt routine.

Timer/Counter1 in PWM Mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A (OCR1A) and the Output Compare Register1B (OCR1B) form a dual 8-, 9-, or 10-bit, free-running, glitch-free, and phase-correct PWM with outputs on the PD5(OC1A) and PE2(OC1B) pins. In this mode the Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Table 17), 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 (depends of the resolution) of OCR1A or OCR1B, the PD5(OC1A)/PE2(OC1B) pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register (TCCR1A). Refer to Table 18 for details.

Alternatively, the Timer/Counter1 can be configured to a PWM that operates at twice the speed as in the mode described above. Then the Timer/Counter1 and the Output Compare Register1A (OCR1A) and the Output Compare Register1B (OCR1B) form a dual 8-, 9- or 10-bit, free-running and glitch-free PWM with outputs on the PD5(OC1A) and PE2(OC1B) pins.

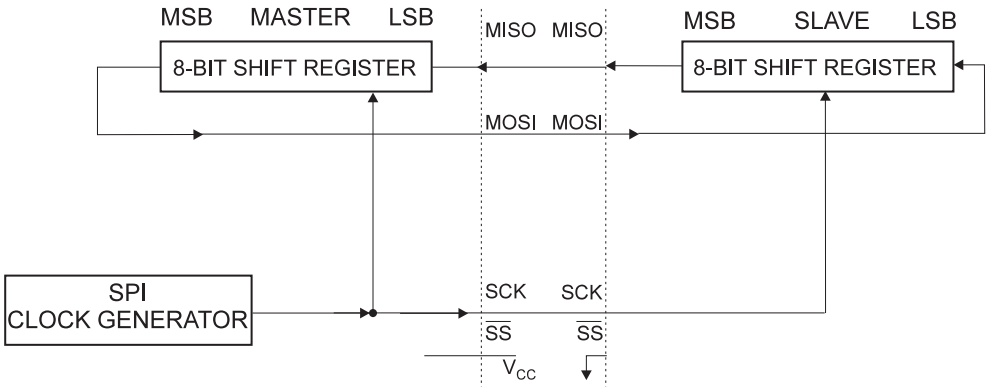
Table 17. Timer TOP Values and PWM Frequency

CTC1	PWM11	PWM10	PWM Resolution	Timer TOP Value	Frequency
0	0	1	8-bit	$00FF (255)	$f_{TCK1}/510$
0	1	0	9-bit	$01FF (511)	$f_{TCK1}/1022$
0	1	1	10-bit	$03FF (1023)	$f_{TCK1}/2046$
1	0	1	8-bit	$00FF (255)	$f_{TCK1}/256$
1	1	0	9-bit	$01FF (511)	$f_{TCK1}/512$
1	1	1	10-bit	$03FF (1023)	$f_{TCK1}/1024$

As shown in Table 17, the PWM operates at either 8-, 9-, or 10-bit resolution. Note the unused bits in OCR1A, OCR1B and TCNT1 will automatically be written to zero by hardware, i.e., bits 9 to 15 will be set to zero in OCR1A, OCR1B and TCNT1 if the 9-bit PWM resolution is selected. This makes it possible for the user to perform read-modify-write operations in any of the three resolution modes and the unused bits will be treated as don't care.

The interconnection between Master and Slave CPUs with SPI is shown in Figure 41. The PB7(SCK) pin is the Clock Output in the Master mode and is the clock input in the Slave mode. Writing to the SPI Data Register of the Master CPU starts the SPI clock generator, and the data written shifts out of the PB5(MOSI) pin and into the PB5(MOSI) pin of the Slave CPU. After shifting one byte, the SPI clock generator stops, setting the End-of-Transmission Flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Slave Select input, PB4($\overline{SS}$), is set low to select an individual Slave SPI device. The two Shift Registers in the Master and the Slave can be considered as one distributed 16-bit circular Shift Register. This is shown in Figure 41. When data is shifted from the Master to the Slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the Master and the Slave are interchanged.

Figure 41. SPI Master-Slave Interconnection



The system is single-buffered in the transmit direction and double-buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifted in. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and $\overline{SS}$ pins is overridden according to Table 22.

Table 22. SPI Pin Overrides[1]

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

Note: 1. See “Alternate Functions of Port B” on page 92 for a detailed description of how to define the direction of the user defined SPI pins.

$\overline{SS}$ Pin Functionality

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ is configured as an output, the pin is a general output pin, which does not affect the SPI system. If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as Master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another Master selecting the SPI as a Slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

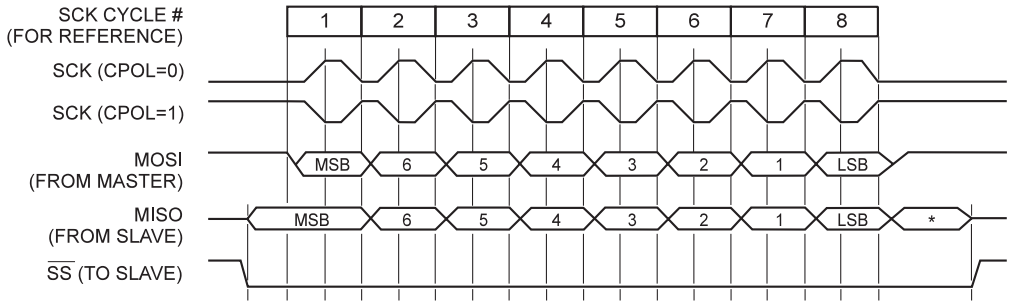
Thus, when interrupt-driven SPI transmission is used in Master mode, and there exists a possibility that $\overline{SS}$ is driven low. The interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a Slave select, it must be set by the user to re-enable SPI Master mode.

When the SPI is configured as a Slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately and both data received and data sent must be considered lost.

Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 42 and Figure 43.

Figure 42. SPI Transfer Format with CPHA = 0 and DORD = 0



* Not defined but normally MSB of character just received

SPI Data Register – SPDR

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	x	x	x	x	x	x	x	x	Undefined

The SPI Data Register is a read/write register used for data transfer between the Register File and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

- **Bit 2 – Res: Reserved Bit**

This bit is reserved bit in the ATmega161 and will always read as zero.

- **Bit 1 – U2X0/U2X1: Double the UART Transmission Speed**

When this bit is set (one), the UART speed will be doubled. This means that a bit will be transmitted/received in 8 CPU clock periods instead of 16 CPU clock periods. For a detailed description, see “Double-speed Transmission” on page 78.

- **Bit 0 – MPCM0/MPCM1: Multi-processor Communication Mode**

This bit is used to enter Multi-processor Communication mode. The bit is set when the Slave MCU waits for an address byte to be received. When the MCU has been addressed, the MCU switches off the MPCMn bit and starts data reception.

For a detailed description, see “Multi-processor Communication mode”.

UART0 Control and Status Registers – UCSR0B

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	CHR90	RXB80	TXB80	UCSR0B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	1	0	

UART1 Control and Status Registers – UCSR1B

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	CHR91	RXB81	TXB81	UCSR1B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	1	0	

- **Bit 7 – RXCIE0/RXCIE1: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXCn bit in UCSRnA will cause the Receive Complete interrupt routine to be executed, provided that global interrupts are enabled.

- **Bit 6 – TXCIE0/TXCIE1: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXCn bit in UCSRnA will cause the Transmit Complete interrupt routine to be executed, provided that global interrupts are enabled.

- **Bit 5 – UDRIE0/UDREI1: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDREn bit in UCSRnA will cause the UART Data Register Empty interrupt routine to be executed, provided that global interrupts are enabled.

- **Bit 4 – RXEN0/RXEN1: Receiver Enable**

This bit enables the UART Receiver when set (one). When the Receiver is disabled, the TXCn, ORn and FEn Status Flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

- **Bit 3 – TXEN0/TXEN1: Transmitter Enable**

This bit enables the UART Transmitter when set (one). When disabling the Transmitter while transmitting a character, the Transmitter is not disabled before the character in the Shift Register plus any following character in UDRn has been completely transmitted.

I/O Ports

All AVR ports have true read-modify-write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies for changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input).

Port A

Port A is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port A, one each for the Data Register – PORTA, $1B($3B), Data Direction Register – DDRA, $1A($3A) and the Port A Input Pins – PINA, $19($39). The Port A Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20 mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port A pins have alternate functions related to the optional external memory interface. Port A can be configured to be the multiplexed low-order address/data bus during accesses to the external Data memory. In this mode, Port A has internal pull-up resistors.

When Port A is set to the alternate function by the SRE (External SRAM Enable) bit in the MCUCR (MCU Control Register), the alternate settings override the Data Direction Register.

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

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

Port A as General Digital I/O

All eight pins in Port A have equal functionality when used as digital I/O pins.

PAn, general I/O pin: The DDAn bit in the DDRA Register selects the direction of this pin. If DDAn is set (one), PAn is configured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the pin is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTAn has to be cleared (zero) or the pin has to be configured as an output pin. The

Port B

Port B is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port B, one each for the Data Register – PORTB, $18($38), Data Direction Register – DDRB, $17($37) and the Port B Input Pins – PINB, $16($36). The Port B Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20 mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in Table 30.

Table 30. Port B Pin Alternate Functions(1)

Port Pin	Alternate Functions
PB0	OC0 (Timer/Counter0 Compare Match Output)/T0 (Timer/Counter0 External Counter Input)
PB1	OC2 (Timer/Counter2 Compare Match Output)/T1 (Timer/Counter1 External Counter Input)
PB2	RXD1 (UART1 Input Line)/AIN0 (Analog Comparator Positive Input)
PB3	TXD1 (UART1 Output Line)/AIN1 (Analog Comparator Negative Input)
PB4	$\overline{SS}$ (SPI Slave Select Input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

Note: 1. When the pins are used for the alternate function, the DDRB and PORTB Registers have to be set according to the alternate function description.

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
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	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
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	

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

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

EEPROM Write Prevents Writing to SPMCR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuse and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user check the status bit (EEWE) in the EECR Register and verify that the bit is cleared before writing to the SPMCR Register.

Reading the Fuse and Lock bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with $0001 and set the BLBSET and SPMEN bits in SPMCR. If an LPM instruction is executed within three CPU cycles after the BLBSET and SPMEN bits are set in SPMCR, the Lock bits will be written to the destination register. The BLBSET and SPMEN bits will auto-clear upon completion of reading the Lock bits or if no LPM/SPM instruction is executed within three/four CPU cycles. When BLBSET and SPMEN are cleared, LPM will work as described in “Constant Addressing Using the LPM Instruction” on page 16 and in the Instruction Set manual.

Bit	7	6	5	4	3	2	1	0	
	–	–	BLB12	BLB11	BLB02	BLB01	LB2	LB1	R0/Rd

The algorithm for reading the Fuse bits is similar to the one described above for reading the Lock bits. But when reading the Fuse bits, load $0000 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SPMEN bits are set in the SPMCR, the Fuse bits can be read in the destination register as shown below.

Bit	7	6	5	4	3	2	1	0	
	–	BOOTRST	SPIEN	SUT	–	CKSEL[2]	CKSEL[1]	CKSEL[0]	R0/Rd

Fuse and Lock bits that are programmed will be read as zero.

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. H: Load Address High Byte ($00 - $01)
3. B: Load Address Low Byte ($00 - $FF)
4. E: Load Data Low Byte ($00 - $FF)

L: Write Data Low Byte

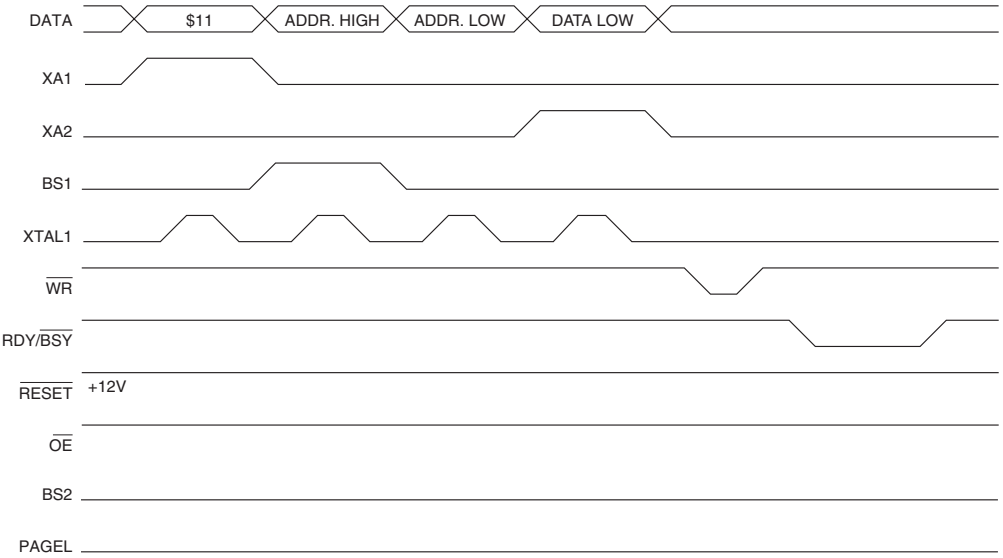
1. Set BS to “0”. This selects low data.
2. Give $\overline{WR}$ a negative pulse. This starts programming of the data byte. RDY/$\overline{BSY}$ goes low.
3. Wait until to RDY/$\overline{BSY}$ goes high before programming the next byte. (See Figure 78 for signal waveforms.)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered:

- The command needs to be loaded only once when writing or reading multiple memory locations.
- Address High byte only needs to be loaded before programming a new 256-word page in the EEPROM.
- Skip writing the data value $FF, that is, the contents of the entire EEPROM after a Chip Erase.

These considerations also apply to Flash, EEPROM and Signature bytes reading.

Figure 78. Programming the EEPROM Waveforms



Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin except $\overline{\text{RESET}}$ with Respect to Ground	-1.0V to V_{CC} + 0.5V
Voltage on $\overline{\text{RESET}}$ with Respect to Ground	-1.0V to +13.0V
Maximum Operating Voltage	6.6V
DC Current per I/O Pin	40.0 mA
DC Current VCC and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 53. External Data Memory Characteristics, 4.0 - 5.5 Volts, SRWn1 = 1, SRWn0 = 0

			8 MHz Oscillator		Variable Oscillator		
	Symbol	Parameter	Min	Max	Min	Max	Unit
0	$1/t_{CLCL}$	Oscillator Frequency			0.0	8.0	MHz
10	t_{RLDV}	Read Low to Data Valid		310		$3.0t_{CLCL}$-65	ns
12	t_{RLRH}	RD Pulse Width	355		$3.0t_{CLCL}$-20		ns
15	t_{DVWH}	Data Valid to WR High	345		$3.0t_{CLCL}$-30		ns
16	t_{WLWH}	WR Pulse Width	35		$3.0t_{CLCL}$-20		ns

Table 54. External Data Memory Characteristics, 4.0 - 5.5 Volts, SRWn1 = 1, SRWn0 = 1

			8 MHz Oscillator		Variable Oscillator		
	Symbol	Parameter	Min	Max	Min	Max	Unit
0	$1/t_{CLCL}$	Oscillator Frequency			0.0	8.0	MHz
10	t_{RLDV}	Read Low to Data Valid		310		$3.0t_{CLCL}$-65	ns
12	t_{RLRH}	RD Pulse Width	355		$3.0t_{CLCL}$-20		ns
14	t_{WHDX}	Data Hold After WR High	152.5		$1.5t_{CLCL}$-35		ns
15	t_{DVWH}	Data Valid to WR High	345		$3.0t_{CLCL}$-30		ns
16	t_{WLWH}	WR Pulse Width	355		$3.0t_{CLCL}$-20		ns

Table 55. External Data Memory Characteristics, 2.7 - 5.5 Volts, No Wait State

			4 MHz Oscillator		Variable Oscillator		
	Symbol	Parameter	Min	Max	Min	Max	Unit
0	$1/t_{CLCL}$	Oscillator Frequency			0.0	4.0	MHz
1	t_{LHLL}	ALE Pulse Width	195		t_{CLCL}-55		ns
2	t_{AVLL}	Address Valid A to ALE Low	60		$0.5t_{CLCL}$-65		ns
3a	t_{LLAX_ST}	Address Hold After ALE Low, write access	10		10		ns
3b	t_{LLAX_LD}	Address Hold after ALE Low, read access	15		15		ns
4	t_{AVLLC}	Address Valid C to ALE Low	60		$0.5t_{CLCL}$-65		ns
5	t_{AVRL}	Address Valid to RD Low	200		$1.0t_{CLCL}$-50		ns
6	t_{AVWL}	Address Valid to WR Low	200		$1.0t_{CLCL}$-50		ns
7	t_{LLWL}	ALE Low to WR Low	105	145	$0.5t_{CLCL}$-20	$0.5t_{CLCL}$+20	ns
8	t_{LLRL}	ALE Low to RD Low	105	145	$0.5t_{CLCL}$-20	$0.5t_{CLCL}$+20	ns
9	t_{DVRH}	Data Setup to RD High	95		95		ns
10	t_{RLDV}	Read Low to Data Valid		165		165	ns
11	t_{RHDX}	Data Hold After RD High	0		0		ns

Figure 91. Pull-up Resistor Current vs. Input Voltage

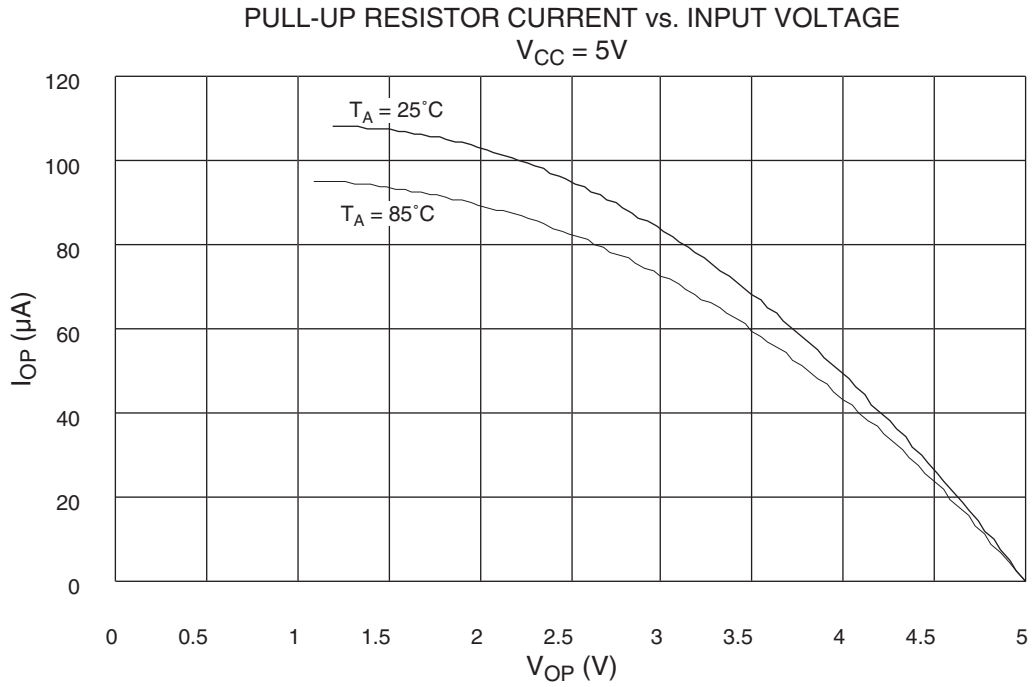
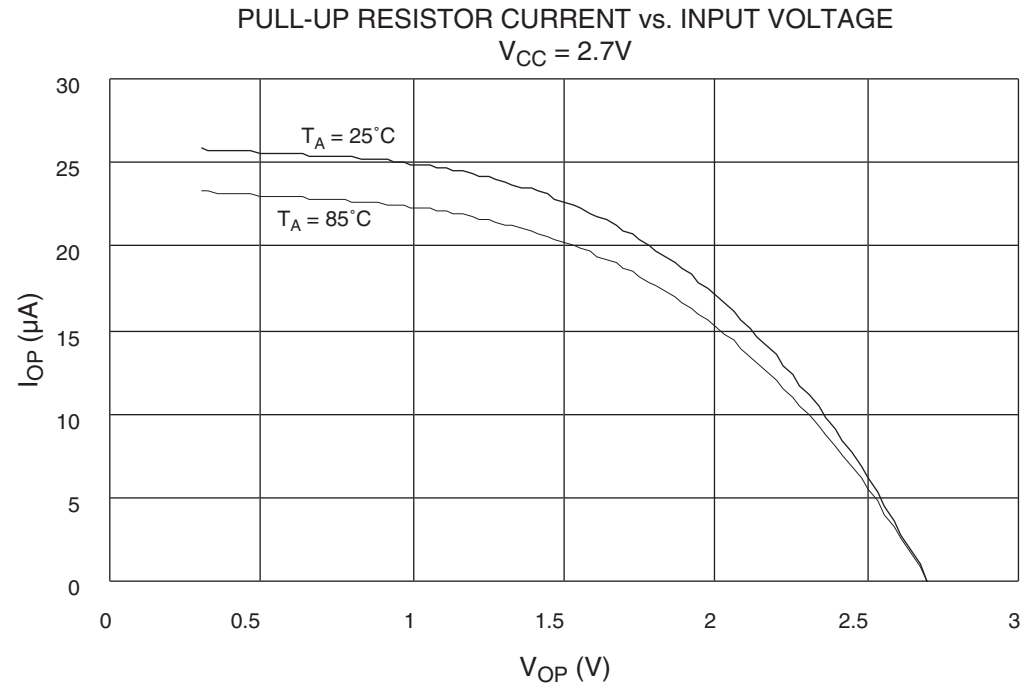


Figure 92. Pull-up Resistor Current vs. Input Voltage



Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	# Clocks
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half-carry Flag in SREG	H ← 1	H	1
CLH		Clear Half-carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1

Table of Contents