



Welcome to [E-XFL.COM](https://www.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	4MHz
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)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega161l-4ac

Pin Descriptions

VCC	Supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA and can 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 are tri-stated when a reset condition becomes active, even if the clock is not running. Port A serves as a Multiplexed Address/Data port when using external memory interface.
Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega161 as listed on page 92.
Port C (PC7..PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port C also serves as an address high output when using external memory interface.
Port D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. 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. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega161 as listed on page 101.
Port E (PE2..PE0)	Port E is a 3-bit bi-directional I/O port with internal pull-up resistors. The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port E also serves the functions of various special features of the ATmega161 as listed on page 107.
RESET	Reset input. A low level on this pin for more than 500 ns will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.

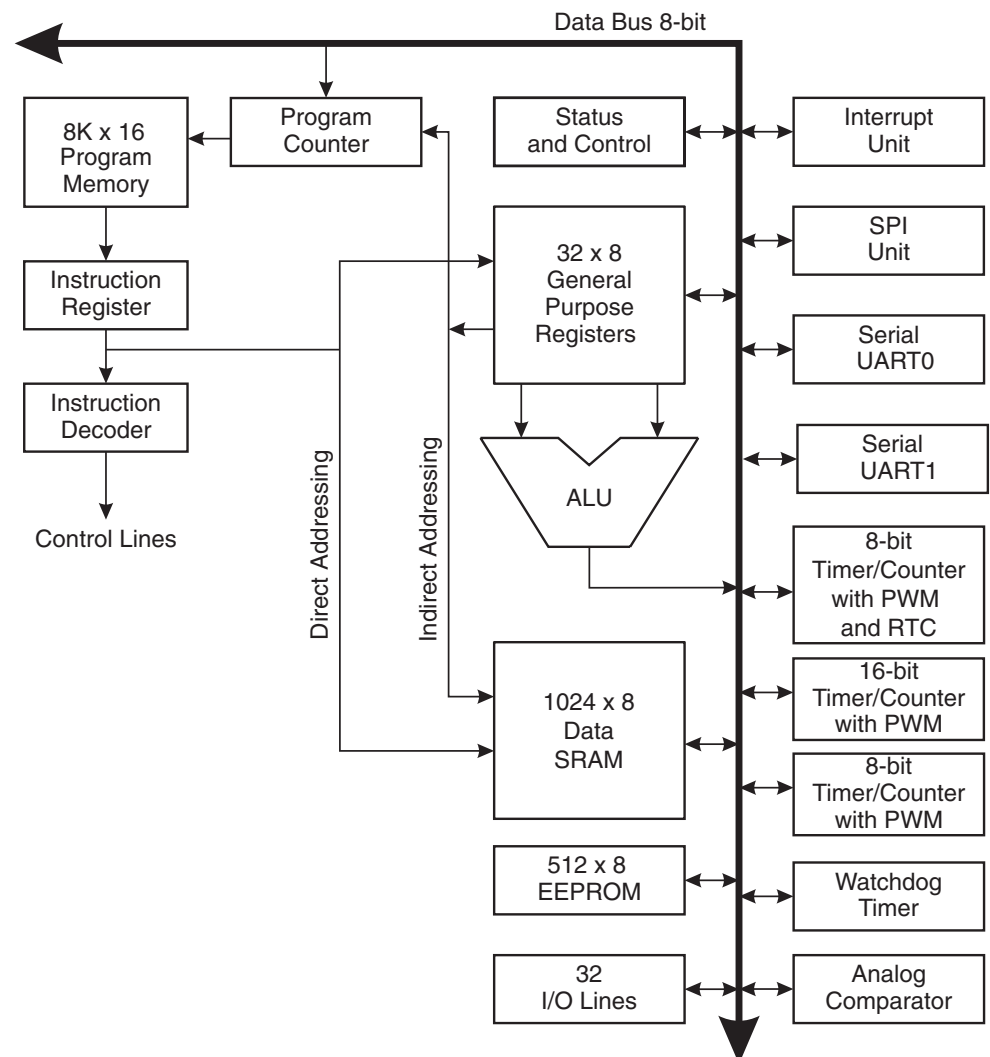
Architectural Overview

The fast-access Register File concept contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one Arithmetic Logic Unit (ALU) operation is executed. Two operands are output from the Register File, the operation is executed and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function. These added function registers are the 16-bit X-register, Y-register and Z-register.

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the ATmega161 AVR RISC microcontroller architecture.

Figure 4. The ATmega161 AVR RISC Architecture



The General Purpose Register File

Figure 6 shows the structure of the 32 general purpose working registers in the CPU.

Figure 6. AVR CPU General Purpose Working Registers

	7	0	Addr.	
	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
General	R16		\$10	
Purpose	R17		\$11	
Working	...			
Registers	R26		\$1A	X-register Low Byte
	R27		\$1B	X-register High Byte
	R28		\$1C	Y-register Low Byte
	R29		\$1D	Y-register High Byte
	R30		\$1E	Z-register Low Byte
	R31		\$1F	Z-register High Byte

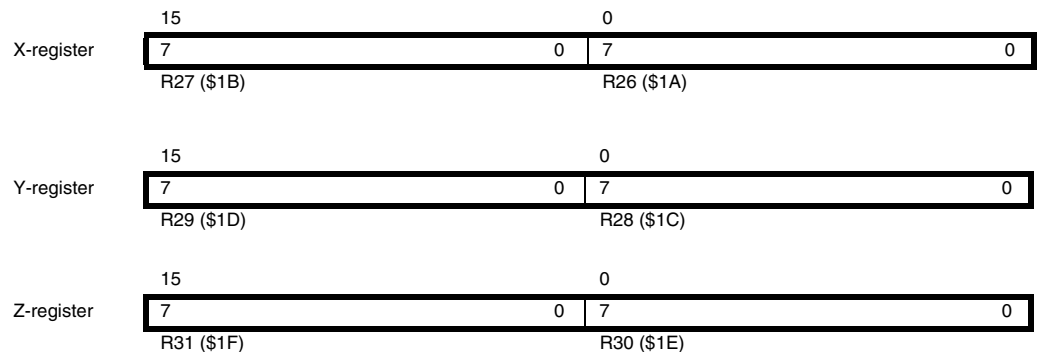
All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exceptions are the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register, and the LDI instruction for load immediate constant data. These instructions apply to the second half of the registers in the Register File – R16..R31. The general SBC, SUB, CP, AND, and OR, and all other operations between two registers or on a single register apply to the entire Register File.

As shown in Figure 6, each register is also assigned a Data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-registers can be set to index any register in the file.

The X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as:

Figure 7. X-, Y-, and Z-registers



Note that the Status Register is not automatically stored when entering an interrupt routine or restored when returning from an interrupt routine. This must be handled by software.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles, the Program Vector address for the actual interrupt handling routine is executed. During this four-clock-cycle period, the Program Counter (13 bits) is pushed onto the Stack. The Vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in Sleep mode, the interrupt execution response time is increased by four clock cycles.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (2 bytes) is popped back from the Stack, the Stack Pointer is incremented by 2, and the I-Flag in SREG is set. When AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
\$3B (\$5B)	INT1	INT0	INT2	–	–	–	–	–	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – INT1: External Interrupt Request 1 Enable

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or is level-sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from Program memory address \$004. See also “External Interrupts”.

• Bit 6 – INT0: External Interrupt Request 0 Enable

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or is level-sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from Program memory address \$002. See also “External Interrupts.”

• Bit 5 – INT2: External Interrupt Request 2 Enable

When the INT2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control2 bit (ISC02 in the Extended MCU Control Register [EMCUCR]) defines whether the external interrupt is activated on rising or falling edge of the INT2 pin. Activity on the pin will cause an interrupt request even if INT2 is configured as an output. The corresponding interrupt of External Interrupt Request 2 is executed from Program memory address \$006. See also “External Interrupts.”

Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register (ACSR). This will reduce power consumption in Idle mode.

Power-down Mode

When the SM1/SM0 bits are set to 10, the SLEEP instruction makes the MCU enter the Power-down mode. In this mode, the external Oscillator is stopped while the external interrupts and the Watchdog (if enabled) continue operating. Only an External Reset, a Watchdog Reset (if enabled), an external level interrupt on INT0 or INT1, or an external edge interrupt on INT2 can wake up the MCU.

If INT2 is used for wake-up from Power-down mode, the edge is remembered until the MCU wakes up.

If a level-triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of the Watchdog Oscillator is 1 μ s (nominal) at 5.0V and 25°C. The frequency of the Watchdog Oscillator is voltage-dependent as shown in the Electrical Characteristics section.

When waking up from Power-down mode, a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL fuses that define the Reset Time-out period. The wake-up period is equal to the clock counting part of the reset period, as shown in Table 4. If the wake-up condition disappears before the MCU wakes up and starts to execute, e.g., a low level on INT0 is not held long enough, the interrupt causing the wake-up will not be executed.

Power-save Mode

When the SM1/SM0 bits are 11, the SLEEP instruction makes the MCU enter the Power-save mode. This mode is identical to Power-down, with one exception.

If Timer/Counter2 is clocked asynchronously, i.e., the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. In addition to the Power-down wake-up sources, the device can also wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK and the global interrupt enable bit in SREG is set.

If the asynchronous timer is *not* clocked asynchronously, Power-down mode is recommended instead of Power-save mode because the contents of the register in the asynchronous timer should be considered undefined after wake-up in Power-save mode even if AS2 is 0.

Timer Counter0 – TCNT0

Bit	7	6	5	4	3	2	1	0	
\$32 (\$52)	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	

Timer/Counter2 – TCNT2

Bit	7	6	5	4	3	2	1	0	
\$23 (\$43)	MSB							LSB	TCNT2
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	

These 8-bit registers contain the value of the Timer/Counters.

Both Timer/Counters are realized as up or up/down (in PWM mode) counters with read and write access. If the Timer/Counter is written to and a clock source is selected, it continues counting in the timer clock cycle following the write operation.

Timer/Counter0 Output Compare Register – OCR0

Bit	7	6	5	4	3	2	1	0	
\$31 (\$51)	MSB							LSB	OCR0
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	

Timer/Counter2 Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
\$22 (\$42)	MSB							LSB	OCR2
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 Output Compare Registers are 8-bit read/write registers. The Timer/Counter Output Compare Registers contain the data to be continuously compared with the Timer/Counter. Actions on compare matches are specified in TCCR0 and TCCR2. A software write to the Timer/Counter Register blocks compare matches in the next Timer/Counter clock cycle. This prevents immediate interrupts when initializing the Timer/Counter.

A compare match will set the Compare Interrupt Flag in the CPU clock cycle following the compare event.

Timer/Counters 0 and 2 in PWM Mode

When PWM mode is selected, the Timer/Counter either wraps (overflows) when it reaches \$FF or it acts as an up/down counter.

If the up/down mode is selected, the Timer/Counter and the Output Compare Registers (OCR0 or OCR2) form an 8-bit, free-running, glitch-free and phase-correct PWM with outputs on the PB0(OC0/PWM0) or PB1(OC2/PWM2) pin.

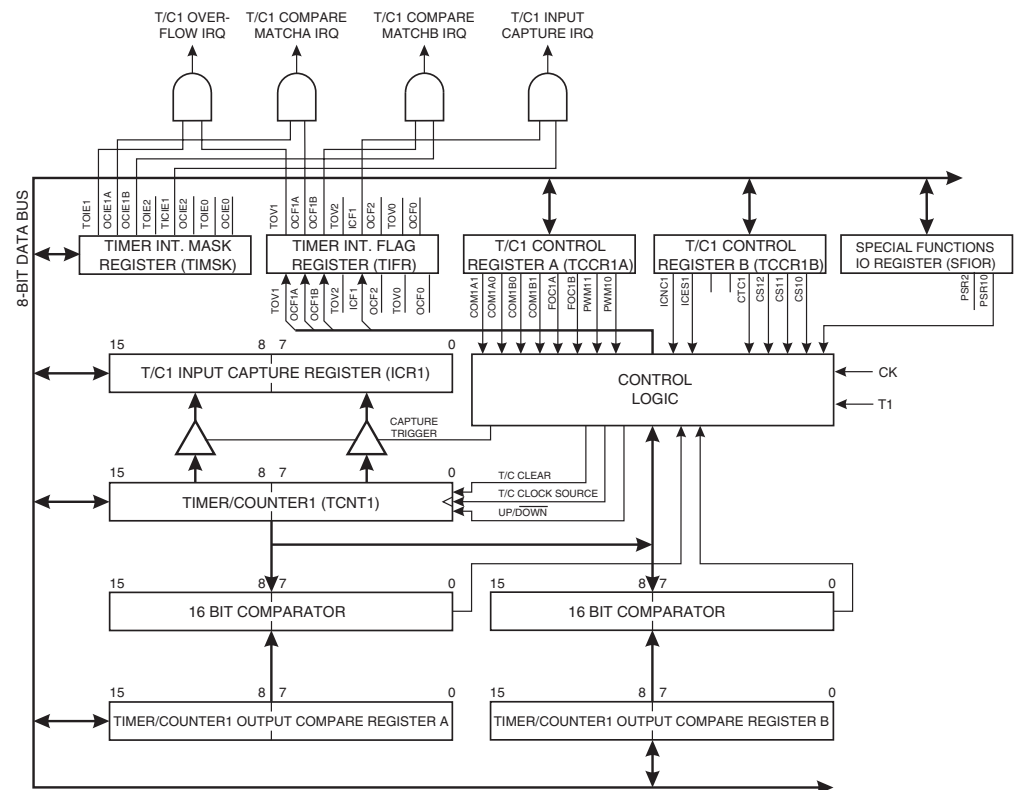
If the Overflow mode is selected, the Timer/Counter and the Output Compare Registers (OCR0 or OCR2) form an 8-bit, free-running and glitch-free PWM, operating with twice the speed of the up/down counting mode.

- Description of wake-up from Power-save mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake-up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least 1 before the processor can read the counter value. The Interrupt Flags are updated three processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable and the interrupt routine has not started yet.
- During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes three processor cycles plus one timer cycle. The timer is therefore advanced by at least 1 before the processor can read the timer value, causing the setting of the Interrupt Flag. The output compare pin is changed on the timer clock and is not synchronized to the processor clock.

Timer/Counter1

Figure 35 shows the block diagram for Timer/Counter1.

Figure 35. Timer/Counter1 Block Diagram



The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK or an external pin. In addition, it can be stopped as described in “Timer/Counter1 Control Register B – TCCR1B”. The different Status Flags (Overflow, Compare Match, and Capture Event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter1 Control Registers (TCCR1A and TCCR1B). 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

- **Bits 1..0 – PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 15. This mode is described on page 55.

Table 15. PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled.
0	1	Timer/Counter1 is an 8-bit PWM.
1	0	Timer/Counter1 is a 9-bit PWM.
1	1	Timer/Counter1 is a 10-bit PWM.

Timer/Counter1 Control Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
\$2E (\$4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the Input Capture trigger noise canceler function is disabled. The Input Capture is triggered at the first rising/falling edge sampled on the ICP (Input Capture pin) as specified. When the ICNC1 bit is set (one), four successive samples are measured on the ICP (Input Capture pin), and all samples must be high/low according to the Input Capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 – ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the falling edge of the Input Capture pin (ICP). While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the rising edge of the Input Capture pin (ICP).

- **Bits 5, 4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega161 and always read as zero.

- **Bit 3 – CTC1: Clear Timer/Counter1 on Compare Match**

When the CTC1 control bit is set (one), the Timer/Counter1 is reset to \$0000 in the clock cycle after a compareA match. If the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match. When a prescaling of 1 is used and the Compare A Register is set to C, the timer will count as follows if CTC1 is set:

... | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | 0, 0, 0, 0, 0, 0, 0, 0 | ...

In PWM mode, this bit has a different function. If the CTC1 bit is cleared in PWM mode, the Timer/Counter1 acts as an up/down counter. If the CTC1 bit is set (one), the Timer/Counter wraps when it reaches the TOP value. Refer to page 55 for a detailed description.

• Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency (f_{cl}) is shown in Table 23:

Table 23. Relationship between SCK and the Oscillator Frequency⁽¹⁾

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{cl}/4$
0	0	1	$f_{cl}/16$
0	1	0	$f_{cl}/64$
0	1	1	$f_{cl}/128$
1	0	0	$f_{cl}/2$
1	0	1	$f_{cl}/8$
1	1	0	$f_{cl}/32$
1	1	1	$f_{cl}/64$

Note: 1. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{cl}/4$.

SPI Status Register – SPSR

Bit	7	6	5	4	3	2	1	0	
\$0E (\$2E)	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – SPIF: SPI Interrupt Flag

When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) and global interrupts are enabled. If $\overline{SS}$ is an input and is driven low when the SPI is in Master mode, this will also set the SPIF Flag. SPIF is cleared by hardware when executing the corresponding Interrupt Handling Vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set (one), then by accessing the SPI Data Register (SPDR).

• Bit 6 – WCOL: Write Collision Flag

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared (zero) by first reading the SPI Status Register with WCOL set (one), and then by accessing the SPI Data Register.

• Bits 5..1 – Res: Reserved Bits

These bits are reserved bits in the ATmega161 and will always read as zero.

• Bit 0 – SPI2X: Double SPI Speed Bit

When this bit is set (one), the SPI speed (SCK frequency) will be doubled when the SPI is in Master mode (see Table 23). This means that the maximum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{cl}/4$.

The SPI interface on the ATmega161 is also used for Program memory and EEPROM downloading or uploading. See page 125 for serial programming and verification.

UART0 and UART1 High Byte Baud Rate Register UBRRHI

Bit	7	6	5	4	3	2	1	0	
\$20 (\$40)	MSB1			LSB1	MSB0			LSB0	UBRRHI
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 UART Baud Register is a 12-bit register. The four most significant bits are located in a separate register, UBRRHI. Note that both UART0 and UART1 share this register. Bit 7 to bit 4 of UBRRHI contain the four most significant bits of the UART1 Baud Register. Bit 3 to bit 0 contain the four most significant bits of the UART0 Baud Register.

UART0 Baud Rate Register Low Byte – UBRR0

Bit	7	6	5	4	3	2	1	0	
\$09 (\$29)	MSB							LSB	UBRR0
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	

UART1 Baud Rate Register Low Byte – UBRR1

Bit	7	6	5	4	3	2	1	0	
\$00 (\$20)	MSB							LSB	UBRR1
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	

UBRRn stores the eight least significant bits of the UART Baud Rate Register.

Double-speed Transmission

The ATmega161 provides a separate UART mode that allows the user to double the communication speed. By setting the U2X bit in UART Control and Status Register UCSRnA, the UART speed will be doubled. The data reception will differ slightly from Normal mode. Since the speed is doubled, the Receiver front-end logic samples the signals on RXDn pin at a frequency 8 times the baud rate. While the line is idle, one single sample of logical “0” will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the Receiver samples the RXDn pin at samples 4, 5 and 6. If two or more of these three samples are found to be logical “1”s, the start bit is rejected as a noise spike and the Receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 4, 5 and 6. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift Register as they are sampled. Sampling of an incoming character is shown in Figure 47.

Figure 47. Sampling Received Data when the Transmission Speed is Doubled



Figure 49. External Memory with Page Select

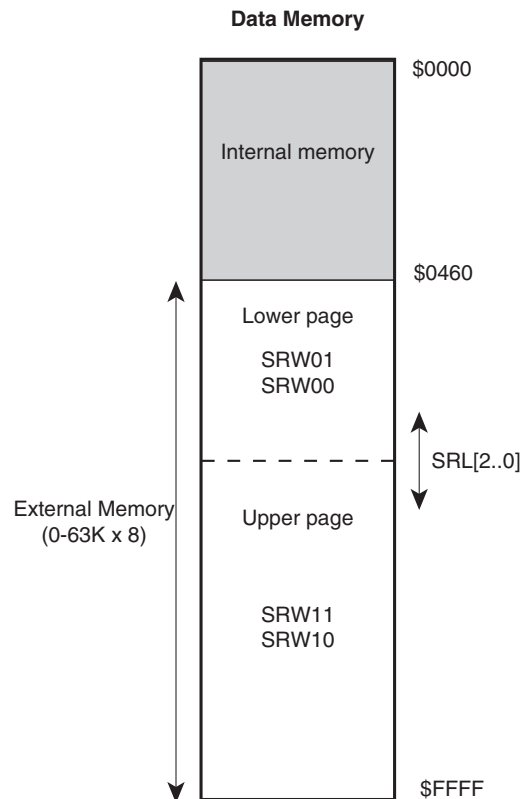
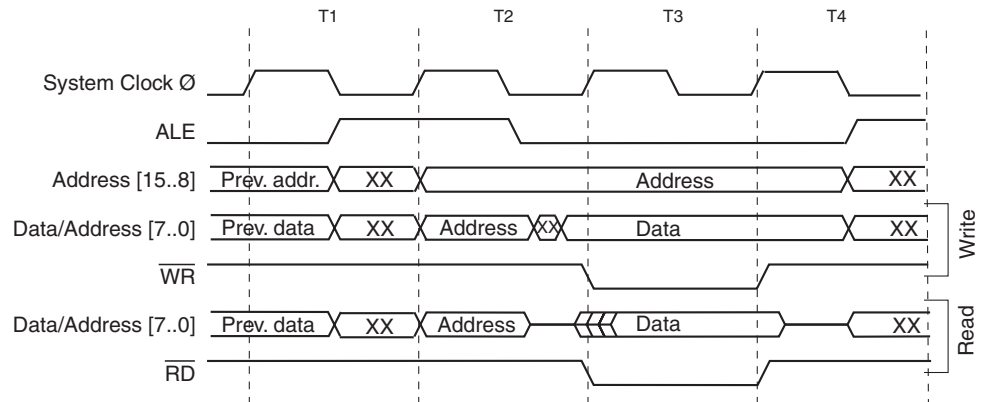


Figure 50. External Data Memory Cycles without Wait State (SRWn1 = 0 and SRWn0 = 0)⁽¹⁾



Note: 1. SRWn1 = SRW11 (upper page) or SRW01 (lower page), SRWn0 = SRW10 (upper page) or SRW00 (lower page).
 The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external). The Data and Address will only change in T4 if ALE is present (the next instruction accesses the RAM).

Figure 57. Port B Schematic Diagram (Pin PB2)

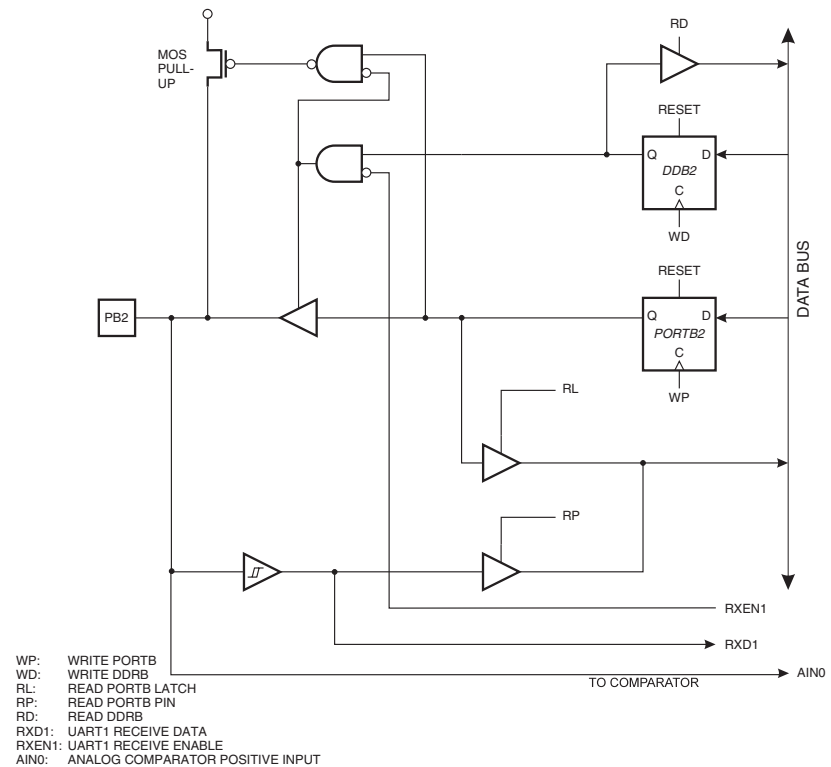
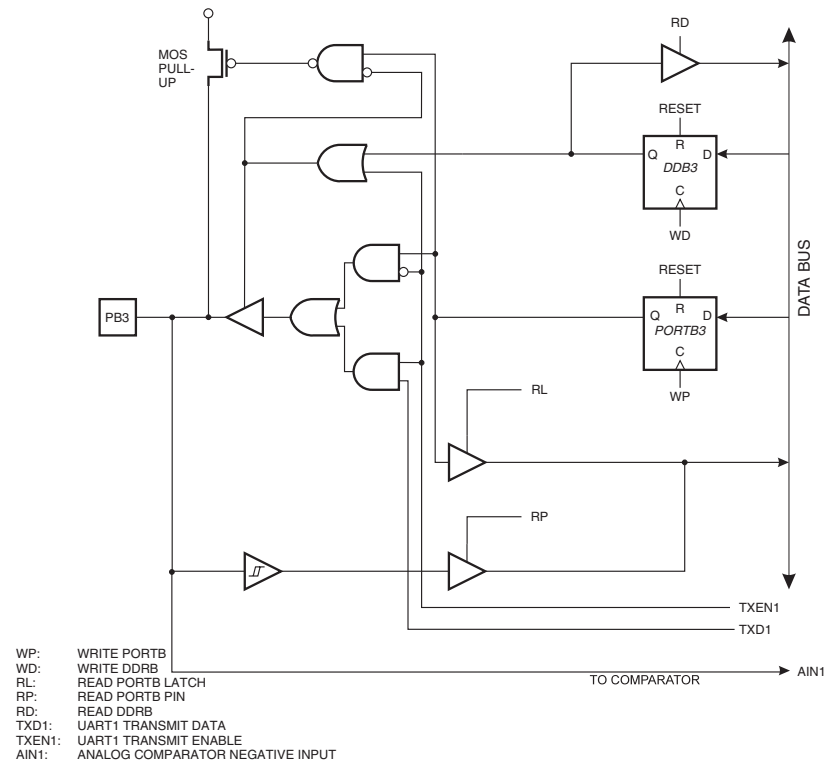


Figure 58. Port B Schematic Diagram (Pin PB3)



Port C

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

Three I/O memory address locations are allocated for the Port C, one each for the Data Register – PORTC, \$15(\$35), Data Direction Register – DDRC, \$14(\$34) and the Port C Input Pins – PINC, \$13(\$33). The Port C 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 C output buffers can sink 20 mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port C pins have alternate functions related to the optional external memory interface. Port C can be configured to be the high-order address byte during accesses to external Data memory.

When Port C 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 C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0									
\$15 (\$35)	<table><tr><td>PORTC7</td><td>PORTC6</td><td>PORTC5</td><td>PORTC4</td><td>PORTC3</td><td>PORTC2</td><td>PORTC1</td><td>PORTC0</td></tr></table>								PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0										
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 C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0									
\$14 (\$34)	<table><tr><td>DDC7</td><td>DDC6</td><td>DDC5</td><td>DDC4</td><td>DDC3</td><td>DDC2</td><td>DDC1</td><td>DDC0</td></tr></table>								DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0										
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 C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0									
\$13 (\$33)	<table><tr><td>PINC7</td><td>PINC6</td><td>PINC5</td><td>PINC4</td><td>PINC3</td><td>PINC2</td><td>PINC1</td><td>PINC0</td></tr></table>								PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0										
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 C Input Pins address (PINC) is not a register; this address enables access to the physical value on each Port C pin. When reading PORTC, the Port C Data Latch is read and when reading PINC, the logical values present on the pins are read.

Port C as General Digital I/O

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

PCn, general I/O pin: The DDcn bit in the DDRC Register selects the direction of this pin. If DDcn is set (one), PCn is configured as an output pin. If DDcn is cleared (zero), PCn is configured as an input pin. If PORTCn 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, PORTCn has to be cleared (zero) or the pin has to be configured as an output pin. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Table 42. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
PAGEL	PD7	I	Program Memory Page Load
BS2	PA0	I	Byte Select 2 (Always low)
DATA	PB7 - 0	I/O	Bi-directional Data Bus (Output when $\overline{OE}$ is low)

Table 43. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS1)
0	1	Load Data (High or low data byte for Flash determined by BS1)
1	0	Load Command
1	1	No Action, Idle

Table 44. Command Byte Bit 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 4.5 - 5.5V between V_{CC} and GND.
2. Set $\overline{RESET}$ and BS pins to “0” and wait at least 500 ns.
3. Apply 11.5 - 12.5V to $\overline{RESET}$, and wait for at least 500 ns.

Chip Erase

The Chip Erase will erase the Flash and EEPROM memories plus Lock bits. The Lock bits are not reset until the Program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash is reprogrammed.

Load Command “Chip Erase”

1. Set XA1, XA0 to “10”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “1000 0000”. This is the command for Chip Erase.
4. Give $\overline{WR}$ a negative pulse. This starts the Chip Erase. RDY/BSY goes low.
5. Wait until RDY/BSY goes high before loading a new command.

t_{WD_FLASH} before programming the next page. As a chip-erased device contains \$FF in all locations, programming of addresses that are meant to contain \$FF can be skipped. See Table 46 for t_{WD_FLASH} value.

Data Polling EEPROM

When a new byte has been written and is being programmed into EEPROM, 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, but the user should keep the following in mind: 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 chip-erasing the device. In this case, data polling cannot be used for the value \$FF, and the user will have to wait at least t_{WD_EEPROM} before programming the next byte. See Table 46 for t_{WD_EEPROM} value.

Table 46. Minimum Wait Delay before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	14 ms
t_{WD_EEPROM}	3.4 ms

Table 47. Minimum Wait Delay after a Chip Erase Command

Symbol	Minimum Wait Delay
t_{WD_ERASE}	28 ms

Figure 80. Serial Programming Waveforms

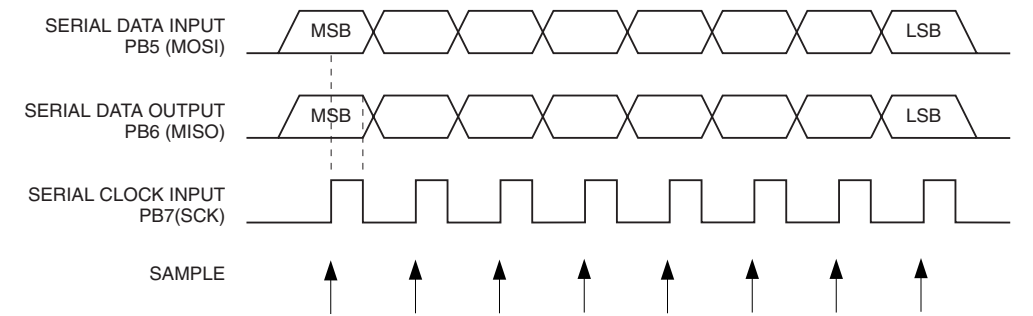


Figure 89. Analog Comparator Input Leakage Current

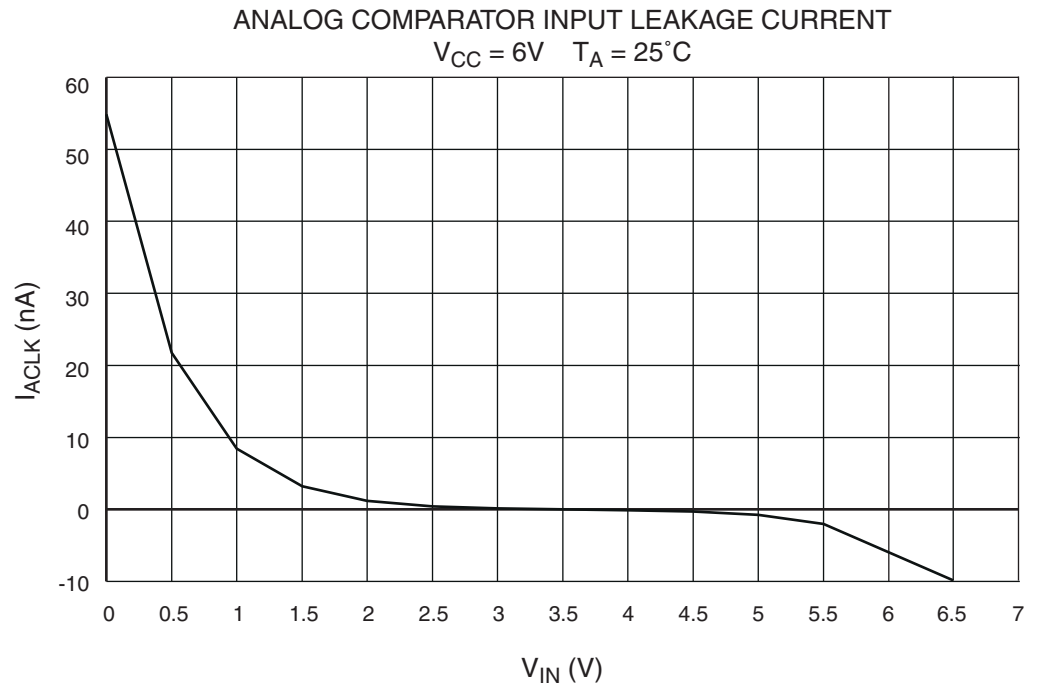
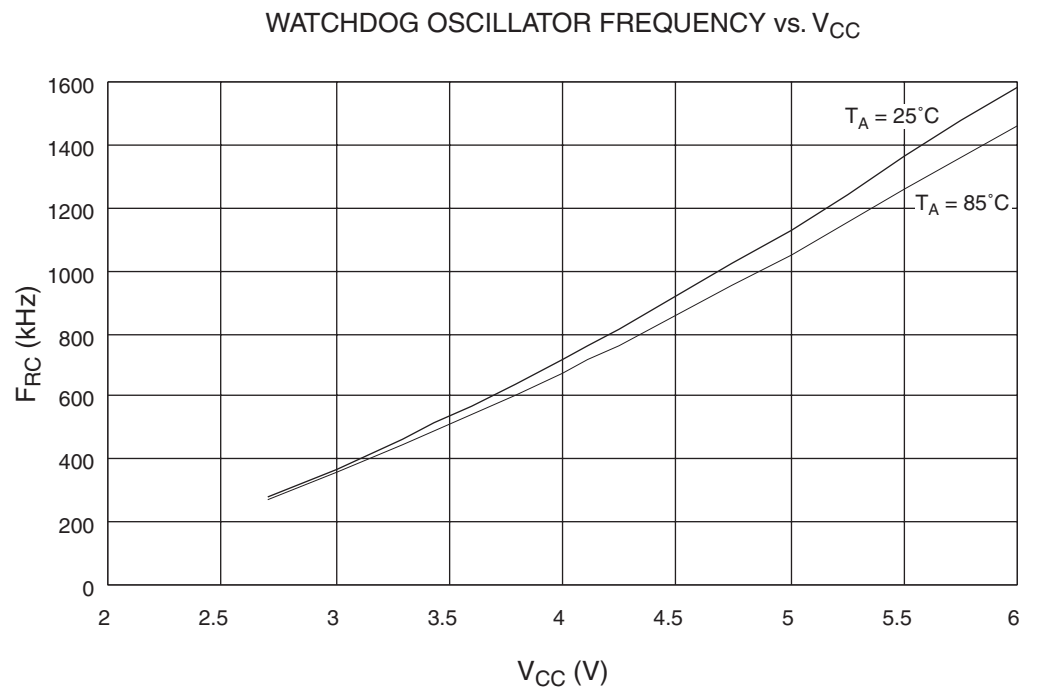


Figure 90. Watchdog Oscillator Frequency vs. V_{CC}



Sink and source capabilities of I/O ports are measured on one pin at a time.

Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	# Clocks
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					
MOV	Rd, Rr	Move between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-inc.	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-inc.	Rd ← (Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P, b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P, b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n = 0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit Load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1



Data Sheet Change Log for ATmega161

This document contains a log on the changes made to the data sheet for ATmega161.

**Changes from Rev.
1228C-08/02 to Rev.
1228D-02/07**

- 1 Package Drawing updated from rev. A to rev. B.**
- 2 Written "Not recommend in new designs" on features page.**

**Changes from Rev.
1228B-09/01 to Rev.
1228C-08/02**

All page numbers refers to this document.

- 1 Description of Brown-out Detector (BOD) removed from data sheet.**

Table of Contents

Features.....	1
Disclaimer.....	1
Pin Configuration.....	2
Description	3
Block Diagram	4
Pin Descriptions	5
Crystal Oscillator.....	6
Architectural Overview.....	7
The General Purpose Register File	10
ALU – Arithmetic Logic Unit.....	11
Self-programmable Flash Program Memory.....	11
EEPROM Data Memory.....	11
SRAM Data Memory	12
Program and Data Addressing Modes.....	13
Memory Access Times and Instruction Execution Timing	17
I/O Memory	19
Reset and Interrupt Handling.....	22
MCU Control Register – MCUCR	34
Sleep Modes.....	36
Timer/Counters	38
Timer/Counter Prescalers.....	38
8-bit Timer/Counters T/C0 and T/C2	40
Timer/Counter1	49
Watchdog Timer.....	58
EEPROM Read/Write Access.....	60
Prevent EEPROM Corruption	62
Serial Peripheral Interface – SPI.....	63
$\overline{SS}$ Pin Functionality.....	65
Data Modes	65
UARTs.....	69
Data Transmission.....	69
Data Reception	71
UART Control	73
Baud Rate Generator.....	76
Double-speed Transmission	78
Analog Comparator	81