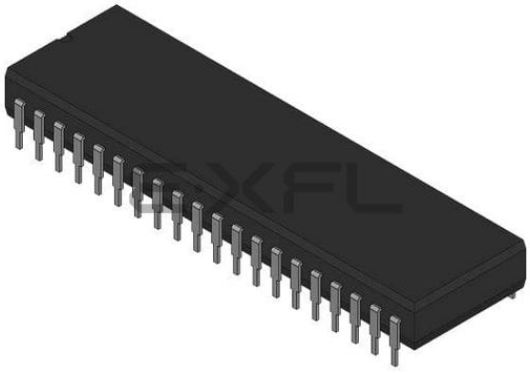


**Welcome to E-XFL.COM**

## 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 | 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 | -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/atmel/atmega161l-4pi |

In addition to the register operation, the conventional Memory Addressing modes can be used on the Register File. This is enabled by the fact that the Register File is assigned the 32 lowermost Data Space addresses ($00 - $1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions such as Control Registers, Timer/Counters, and other I/O functions. The I/O memory can be accessed directly or as the Data Space locations following those of the Register File, $20 - $5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The Program memory is executed with a two-stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the Program memory. This concept enables instructions to be executed in every clock cycle. The Program memory is Self-programmable Flash memory.

With the jump and call instructions, the whole 8K word address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every Program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM and, consequently, the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP (Stack Pointer) in the reset routine (before subroutines or interrupts are executed). The 16-bit Stack Pointer is read/write accessible in the I/O space.

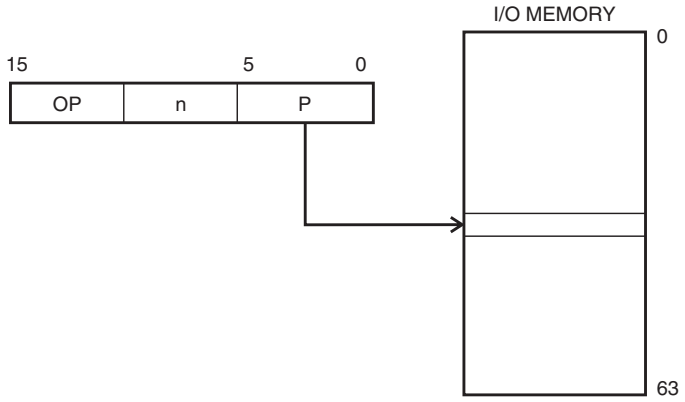
The 1K byte data SRAM can be easily accessed through the five different Addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

Operands are contained in registers r (Rr) and d (Rd). The result is stored in register d (Rd).

**I/O Direct**

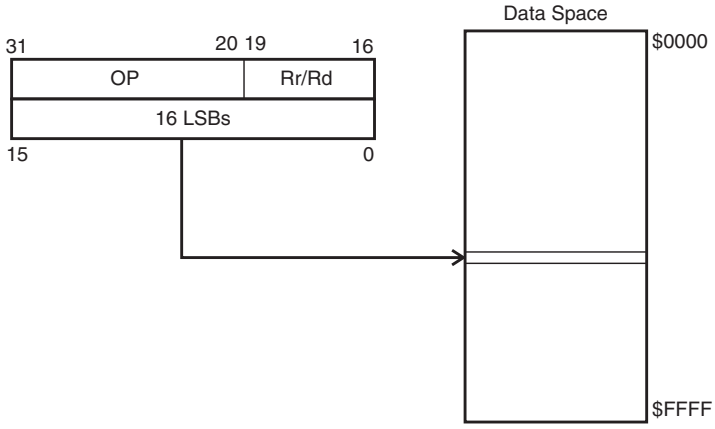
**Figure 11.** I/O Direct Addressing




Operand address is contained in six bits of the instruction word. n is the destination or Source Register Address.

**Data Direct**

**Figure 12.** Direct Data Addressing




A 16-bit data address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

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.”

Compare match InterruptA Enable) and the OCF2 are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 1 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding Interrupt Handling Vector. Alternatively, TOV0 is cleared by writing a logical “1” to the Flag. When the SREG I-bit and TOIE0 (Timer/Counter0 Overflow Interrupt Enable) and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

- **Bit 2 – OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when compare match occurs between the Timer/Counter0 and the data in OCR0 (Output Compare Register 0). OCF0 is cleared by hardware when executing the corresponding Interrupt Handling Vector. Alternatively, OCF0 is cleared by writing a logical “1” to the Flag. When the I-bit in SREG and OCIE0 (Timer/Counter0 Compare match InterruptA Enable) and the OCF0 are set (one), the Timer/Counter0 Compare match Interrupt is executed.

### External Interrupts

The external interrupts are triggered by the INT0, INT1, and INT2 pins. Observe that, if enabled, the interrupts will trigger even if the INT0/INT1/INT2 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level (INT2 is only an edge triggered interrupt). This is set up as indicated in the specification for the MCU Control Register – MCUCR (INT0/INT1) and EMCUCR (INT2). When the external interrupt is enabled and is configured as level triggered (only INT0/INT1), the interrupt will trigger as long as the pin is held low.

## MCU Control Register – MCUCR

The MCU Control Register contains control bits for general MCU functions.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $35 ($55) | **SRE** | **SRW10** | **SE** | **SM1** | **ISC11** | **ISC10** | **ISC01** | **ISC00** | **MCUCR** |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – SRE: External SRAM Enable**

When the SRE bit is set (one), the external Data memory interface is enabled and the pin functions AD0 - 7 (Port A), A8 - 5 (Port C), ALE (Port E), $\overline{WR}$, and $\overline{RD}$ (Port D) are activated as the alternate pin functions. The SRE bit overrides any pin direction settings in the respective Data Direction Registers. See Figure 50 through Figure 53 for a description of the external memory pin functions. When the SRE bit is cleared (zero), the external Data memory interface is disabled and the normal pin and data direction settings are used.

- **Bit 6 – SRW10: External SRAM Wait State**

The SRW10 bit is used to set up extra wait states in the external memory interface. See “Double-speed Transmission” on page 78 for a detailed description.

- **Bit 5 – SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the Sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the Sleep mode unless it is the pro-

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 µs (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.

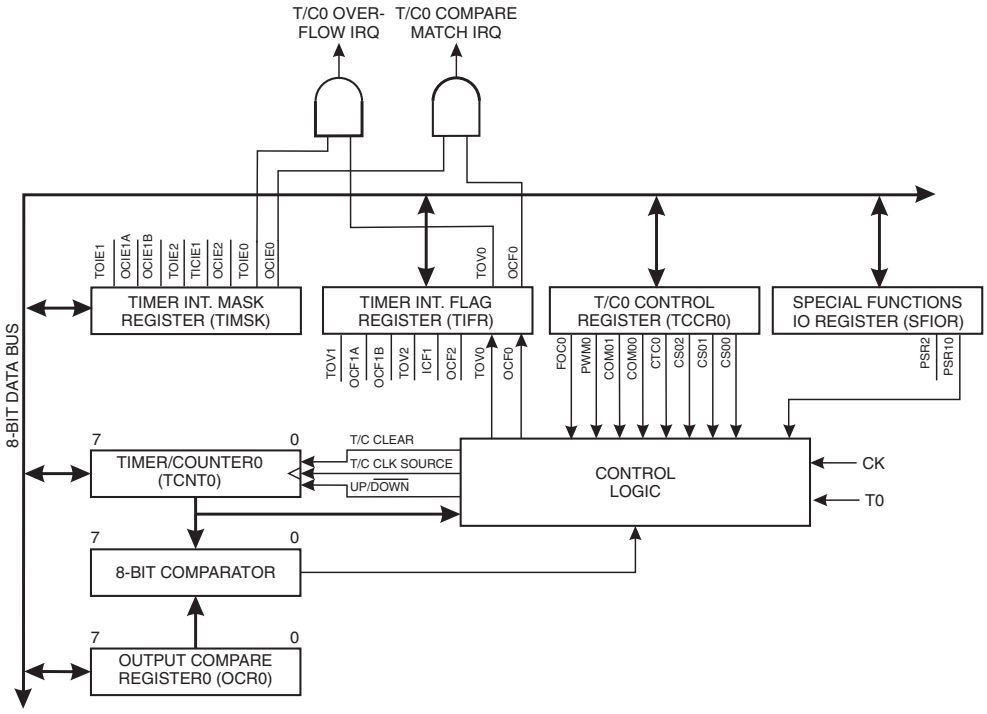
- **Bit 0 – PSR10: Prescaler Reset Timer/Counter1 and Timer/Counter0**

When this bit is set (one), the Timer/Counter1 and Timer/Counter0 prescaler will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers. This bit will always be read as zero.

## 8-bit Timer/Counters T/C0 and T/C2

Figure 31 shows the block diagram for Timer/Counter0. Figure 32 shows the block diagram for Timer/Counter2.

**Figure 31.** Timer/Counter0 Block Diagram



### Timer/Counter0 Control Register – TCCR0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $33 ($53) | FOC0 | PWM0 | COM01 | COM00 | CTC0 | CS02 | CS01 | CS00 | TCCR0 |
| 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 Control Register – TCCR2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $27 ($47) | FOC2 | PWM2 | COM21 | COM20 | CTC2 | CS22 | CS21 | CS20 | TCCR2 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – FOC0/FOC2: Force Output Compare**

Writing a logical “1” to this bit forces a change in the compare match output pin PB0 (Timer/Counter0) and PB1 (Timer/Counter2) according to the values already set in COMn1 and COMn0. If the COMn1 and COMn0 bits are written in the same cycle as FOC0/FOC2, the new settings will not take effect until the next compare match or Forced Output 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 COMn1 and COMn0 happens as if a Compare Match had occurred, but no interrupt is generated and the Timer/Counters will not be cleared even if CTC0/CTC2 is set. The FOC0/FOC2 bits will always be read as zero. The setting of the FOC0/FOC2 bits has no effect in PWM mode.

- **Bit 6 – PWM0/PWM2: Pulse Width Modulator Enable**

When set (one), this bit enables PWM mode for Timer/Counter0 or Timer/Counter2. This mode is described on page 44.

- **Bits 5, 4 – COM01, COM00/COM21, COM20: Compare Output Mode, Bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter0 or Timer/Counter2. Output pin actions affect pins PB0(OC0) or PB1(OC2). 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 control configuration is shown in Table 9.

**Table 9.** Compare Mode Select[1][2]

| COMn1 | COMn0 | Description |
|---|---|---|
| 0 | 0 | Timer/Counter disconnected from output pin OCn. |
| 0 | 1 | Toggle the OCn output line. |
| 1 | 0 | Clear the OCn output line (to zero). |
| 1 | 1 | Set the OCn output line (to one). |

Notes: 1. In PWM mode, these bits have a different function. Refer to Table 12 for a detailed description.
2. n = 0 or 2

- **Bit 3 – CTC0/CTC2: Clear Timer/Counter on Compare Match**

When the CTC0 or CTC2 control bit is set (one), Timer/Counter0 or Timer/Counter2 is reset to $00 in the CPU clock cycle after a compare match. If the control bit is cleared, the Timer/Counter continues counting and is unaffected by a compare match. When a

**Figure 33.** Effects of Unsynchronized OCR Latching in Up/Down Mode

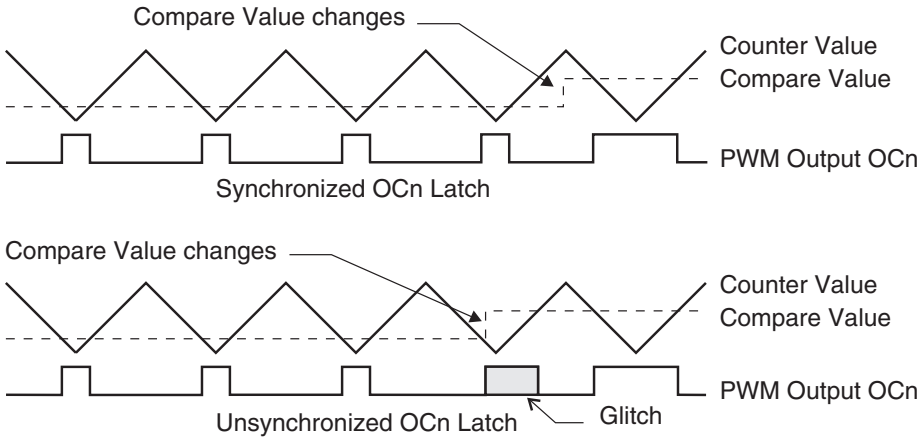
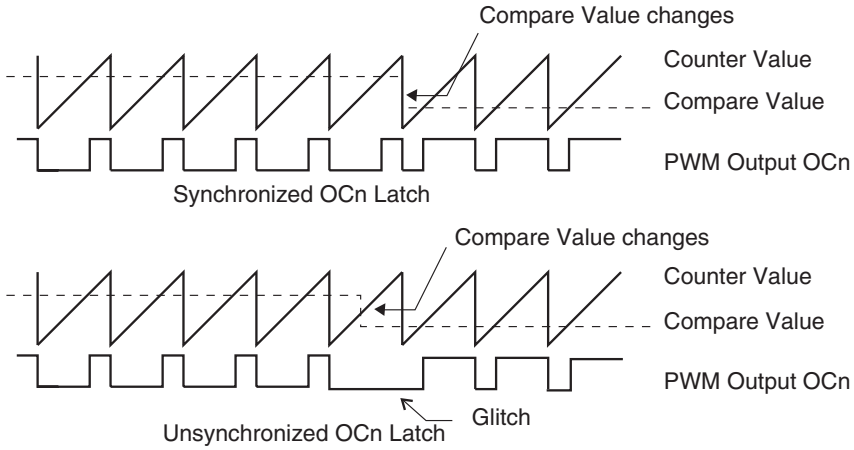



**Figure 34.** Effects of Unsynchronized OCR Latching in Overflow Mode.




Note: n = 0 or 2 (Figure 33 and Figure 34)

During the time between the write and the latch operation, a read from the Output Compare Registers will read the contents of the temporary location. This means that the most recently written value always will read out of OCR0 and OCR2.

When the Output Compare Register contains $00 or $FF, and the up/down PWM mode is selected, the output PB0(OC0/PWM0)/PB1(OC2/PWM2) is updated to low or high on the next compare match according to the settings of COMn1/COMn0. This is shown in Table 13. In overflow PWM mode, the output PB0(OC0/PWM0)/PB1(OC2/PWM2) is held low or high only when the Output Compare Register contains $FF.

**Table 13.** PWM Outputs OCRn = $00 or $FF[1][2]

| COMn1 | COMn0 | OCRn | Output PWMn |
|---|---|---|---|
| 1 | 0 | $00 | L |
| 1 | 0 | $FF | H |
| 1 | 1 | $00 | H |
| 1 | 1 | $FF | L |

Note: 1. n = 0 or 2

2. In overflow PWM mode, the table above is only valid for OCRn = $FF.

In up/down PWM mode, the Timer Overflow Flag (TOV0 or TOV2) is set when the counter advances from $00. In overflow PWM mode, the Timer Overflow Flag is set as in normal Timer/Counter mode. Timer Overflow Interrupt0 and 2 operate exactly as in normal Timer/Counter mode, i.e., they are executed when TOV0 or TOV2 are set, provided that Timer Overflow Interrupt and global interrupts are enabled. This also applies to the Timer Output Compare flag and interrupt.

**Asynchronous Status Register – ASSR**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $26 ($46) | – | – | – | – | AS2 | TCN2UB | OCR2UB | TCR2UB | ASSR |
| Read/Write | R | R | R | R | R/W | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..4 – Res: Reserved Bits**

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

- **Bit 3 – AS2: Asynchronous Timer/Counter2 Mode**

When this bit is cleared (zero), Timer/Counter2 is clocked from the internal system clock, CK. If AS2 is set, the Timer/Counter2 is clocked from the TOSC1 pin. Pins PD4 and PD5 become connected to a crystal Oscillator and cannot be used as general I/O pins. When the value of this bit is changed, the contents of TCNT2, OCR2 and TCCR2 might get corrupted.

- **Bit 2 – TCN2UB: Timer/Counter2 Update Busy**

When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set (one). When TCNT2 has been updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical “0” in this bit indicates that TCNT2 is ready to be updated with a new value.

- **Bit 1 – OCR2UB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2 is written, this bit becomes set (one). When OCR2 has been updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical “0” in this bit indicates that OCR2 is ready to be updated with a new value.

- **Bit 0 – TCR2UB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2 is written, this bit becomes set (one). When TCCR2 has been updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical “0” in this bit indicates that TCCR2 is ready to be updated with a new value.

If a write is performed to any of the three Timer/Counter2 Registers while its update Busy Flag is set (one), the updated value might get corrupted and cause an unintentional interrupt to occur.

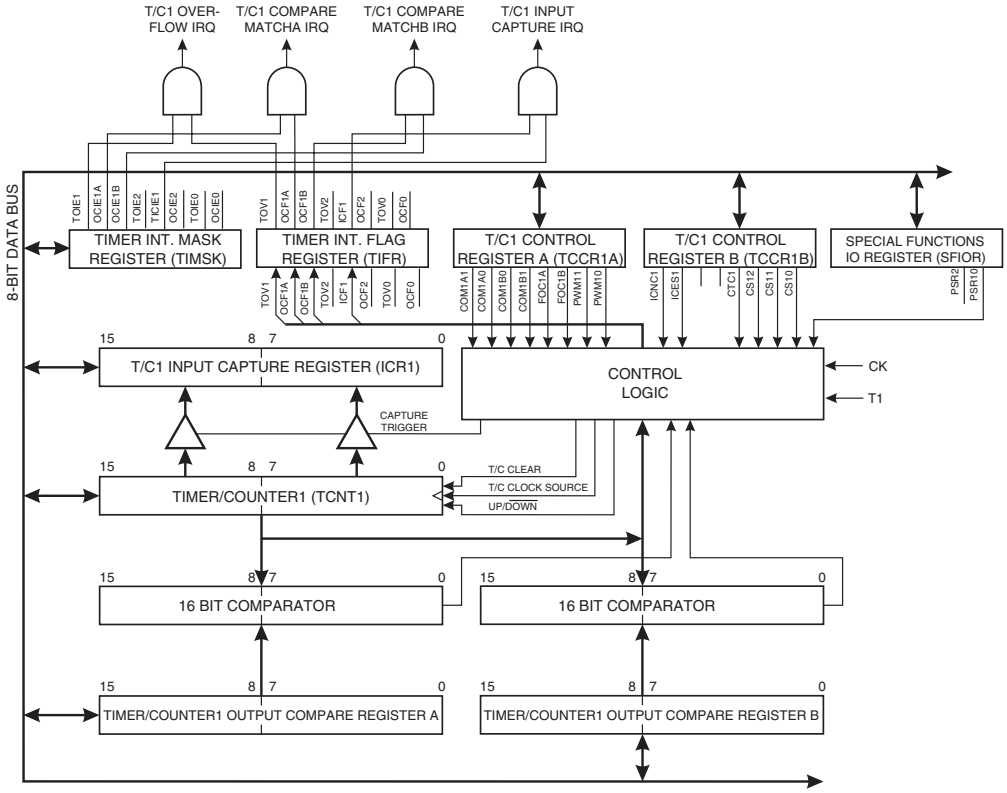
The mechanisms for reading TCNT2, OCR2 and TCCR2 are different. When reading TCNT2, the actual timer value is read. When reading OCR2 or TCCR2, the value in the temporary storage register is read.

- 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.

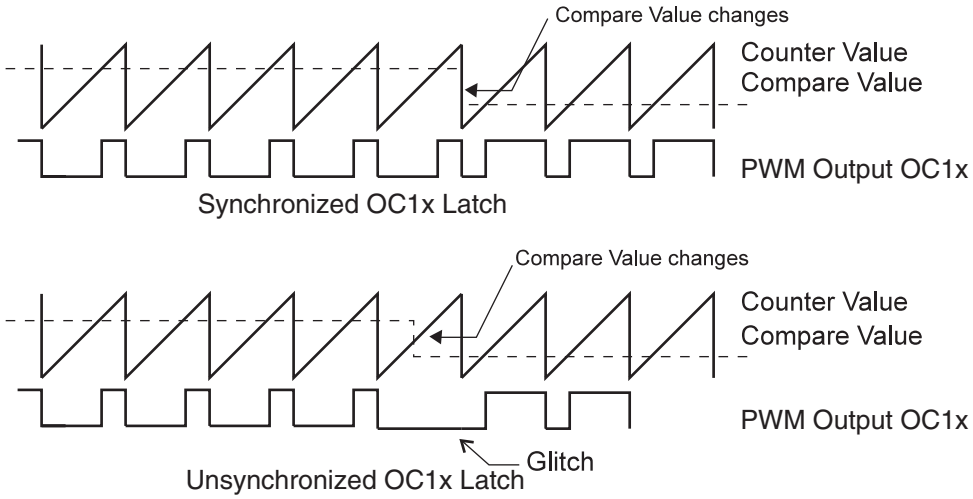
**Figure 35.** Timer/Counter1 Block Diagram




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

**Figure 38.** Effects of Unsynchronized OCR1 Latching in Overflow Mode[1]




Note: 1. Note: X = A or B

During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A/B.

When the OCR1X contains $0000 or TOP, and the up/down PWM mode is selected, the output OC1A/OC1B is updated to low or high on the next compare match according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 19. In overflow PWM mode, the output OC1A/OC1B is held low or high only when the Output Compare Register contains TOP.

**Table 19.** PWM Outputs OCR1X = $0000 or TOP(1)

| COM1X1 | COM1X0 | OCR1X | Output OC1X |
|---|---|---|---|
| 1 | 0 | $0000 | L |
| 1 | 0 | TOP | H |
| 1 | 1 | $0000 | H |
| 1 | 1 | TOP | L |

Note: 1. X = A or B
In overflow PWM mode, the table above is only valid for OCR1X = TOP.

In up/down PWM mode, the Timer Overflow Flag1 (TOV1) is set when the counter advances from $0000. In overflow PWM mode, the Timer Overflow Flag is set as in normal Timer/Counter mode. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e., it is executed when TOV1 is set, provided that Timer Overflow Interrupt1 and global interrupts are enabled. This also applies to the Timer Output Compare1 Flags and interrupts.

- **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.

**Figure 49.** External Memory with Page Select

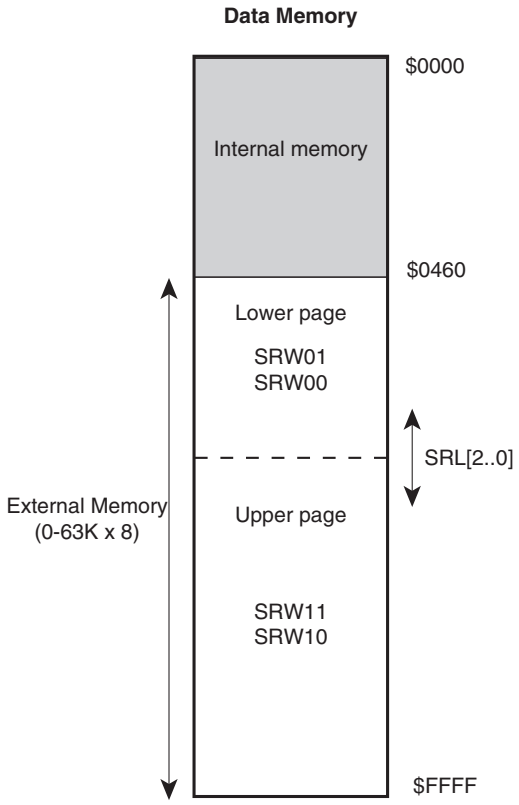
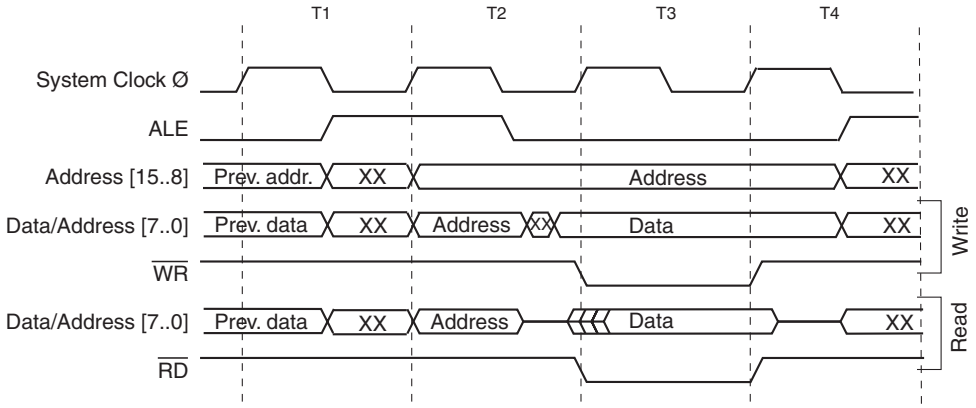



**Figure 50.** External Data Memory Cycles without Wait State (SRWn1 = 0 and SRWn0 =0)[1]




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).

## Port D

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O address locations are allocated for the Port D, one each for the Data Register – PORTD, $12($32), Data Direction Register – DDRD, $11($31) and the Port D Input Pins – PIND, $10($30). The Port D Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Some Port D pins have alternate functions as shown in Table 33.

**Table 33.** Port D Pin Alternate Functions

| Port Pin | Alternate Function |
|---|---|
| PD0 | RXD0 (UART0 Input Line) |
| PD1 | TXD0 (UART0 Output Line) |
| PD2 | INT0 (External Interrupt0 Input) |
| PD3 | INT1 (External Interrupt1 Input) |
| PD3 | TOSC1 (RTC Oscillator Timer/Counter2) |
| PD5 | TOSC2 (RTC Oscillator Timer/Counter2)/OC1A (Timer/Counter1 Output CompareA Match Output) |
| PD6 | $\overline{WR}$ (Write Strobe to External Memory) |
| PD7 | $\overline{RD}$ (Read Strobe to External Memory) |

When the PD5 pin is used for the alternate function (OC1A), the DDRD and PORTD Registers have to be set according to the alternate function description.

### Port D Data Register – PORTD

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $12 ($32) | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | PORTD |
| 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 D Data Direction Register – DDRD

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $11 ($31) | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | DDRD |
| 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 D Input Pins Address – PIND

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $10 ($30) | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | PIND |
| 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 D Input Pins address (PIND) is not a register; this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read and when reading PIND, the logical values present on the pins are read.

### Port D as General Digital I/O

PDn, general I/O pin: The DDDn bit in the DDRD Register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PORTDn is set (one) when configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTDn

## Port E

Port E is a 3-bit bi-directional I/O port with internal pull-up resistors.

Three I/O address locations are allocated for the Port E, one each for the Data Register – PORTE, $07($27), Data Direction Register – DDRE, $06($26) and the Port E Input Pins – PINE, $05($25). The Port E Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

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.

Port E pins have alternate functions as shown in Table 35.

**Table 35.** Port E Pin Alternate Functions(1)

| Port Pin | Alternate Function |
|---|---|
| PE0 | ICP (Input Capture Pin Timer/Counter1)/INT2 (External Interrupt 2 Input) |
| PE1 | OC1B (Timer/Counter1 Output CompareB Match Output) |
| PE2 | ALE (Address Latch Enable, External Memory) |

Note: 1. When the PE1 pin is used for the alternate function, the DDRE and PORTE Registers have to be set according to the alternate function description.

### Port E Data Register – PORTE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $07 ($27) | – | – | – | – | – | PORTE2 | PORTE1 | PORTE0 | PORTE |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

### Port E Data Direction Register – DDRE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $06 ($26) | – | – | – | – | – | DDE2 | DDE1 | DDE0 | DDRE |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

### Port E Input Pins Address – PINE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $05 ($25) | – | – | – | – | – | PINE2 | PINE1 | PINE0 | PINE |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | N/A | N/A | N/A | |

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

### Port E as General Digital I/O

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

**Table 36.** DDEn Bits on Port E Pins[1]

| DDEn | PORTEn | I/O | Pull-up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PEn will source current if ext. pulled low. |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: 1. n: 2,1,0, pin number.

**Alternate Functions of Port E**

The alternate pin configuration is as follows:

- **OC1B – Port E, Bit 2**

OC1B, Output compare match output: The PE2 pin can serve as an external output when the Timer/Counter1 compare matches. The PE2 pin has to be configured as an output (DDE2 set [one]) to serve this function. See “Timer/Counter1” on page 49 for further details. The OC1B pin is also the output pin for the PWM mode timer function.

- **ALE – Port E, Bit 1**

ALE: When the External Memory is enabled, the PE1 pin serves as the Dress Latch Enable. Note that enabling of External Memory will override both the direction and port value. See “Interface to External Memory” on page 84 for a detailed description.

- **ICP/INT2 – Port E, Bit 0**

ICP, Input Capture pin: The PE0 pin can serve as the Input Capture source for Timer/Counter 1. See page 54 for a detailed description.

INT2, External Interrupt source 2: The PE0 pin can serve as an external interrupt source to the MCU. See “Extended MCU Control Register – EMCUCR” on page 36 for further details.

# Memory Programming

## Boot Loader Support

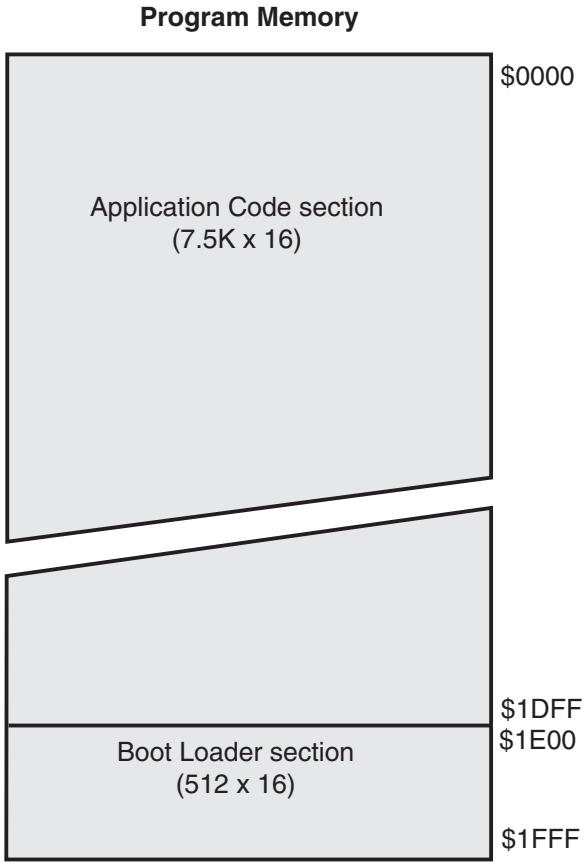
The ATmega161 provides a mechanism for downloading and uploading program code by the MCU itself. This feature allows flexible application software updates, controlled by the MCU using a Flash-resident Boot Loader program.

The ATmega161 Flash memory is organized in two main sections:

1. The Application Code section (address $0000 - $1DFF)

2. The Boot Loader section/Boot block (address $1E00 - $1FFF)

**Figure 74.** Memory Sections




The Boot Loader program can use any available data interface and associated protocol, such as UART serial bus interface, to input or output program code and write (program) that code into the Flash memory or read the code from the Program memory.

The program Flash memory is divided into pages that each contain 128 bytes. The Boot Loader Flash section occupies eight pages from $1E00 to $1FFF by 16-bit words.

The Store Program Memory (SPM) instruction can access the entire Flash, but it can only be executed from the Boot Loader Flash section. If no Boot Loader capability is needed, the entire Flash is available for application code. The ATmega161 has two separate sets of Boot Lock bits that can be set independently. This gives the user a unique flexibility to select different levels of protection. The user can elect to:

## Program Memory Lock bits

The ATmega161 MCU provides six Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 40. The Lock bits can only be erased to “1” with the Chip Erase command.

**Table 40.** Lock Bit Protection Modes [1]

| Memory Lock bits | | | |
|---|---|---|---|
| **LB Mode** | **LB1** | **LB2** | **Protection Type** |
| 1 | 1 | 1 | No memory lock features enabled |
| 2 | 0 | 1 | Further programming of the Flash and EEPROM is disabled in parallel and Serial Programming modes. The Fuse bits are locked in both Serial and Parallel Programming modes.[1] |
| 3 | 0 | 0 | Further programming and verification of the Flash and EEPROM is disabled in parallel and Serial Programming modes. The Fuse bits are locked in both Serial and Parallel Programming modes.[1] |
| **BLB0 Mode** | **BLB02** | **BLB01** | |
| 1 | 1 | 1 | No restrictions for SPM, LPM accessing the Application Code section |
| 2 | 1 | 0 | SPM is not allowed to write to the Application Code section. |
| 3 | 0 | 0 | SPM is not allowed to write to the Application Code section and LPM executing from Boot Loader section is not allowed to read from the Application Code section. |
| 4 | 0 | 1 | LPM executing from the Boot Loader section is not allowed to read from the Application Code section. |
| **BLB1 Mode** | **BLB12** | **BLB11** | |
| 1 | 1 | 1 | No restrictions for SPM, LPM accessing the Boot Loader section |
| 2 | 1 | 0 | SPM is not allowed to write the Boot Loader section. |
| 3 | 0 | 0 | SPM is not allowed to write to the Boot Loader section and LPM executing from the Application Code section is not allowed to read from the Boot Loader section. |
| 4 | 0 | 1 | LPM executing from the Application Code section is not allowed to read from the Boot Loader section. |

Note: 1. Program the Fuse bits before programming the Lock bits.

## Fuse bits

The ATmega161 has six Fuse bits: BOOTRST, SPIEN, SUT, and CKSEL [2:0].

- When BOOTRST is programmed (“0”), the Reset Vector is set to address $1E00, which is the first address location in the Boot Loader section of the Flash. If the BOOTRST is unprogrammed (“1”), the Reset Vector is set to address $0000. Default value is unprogrammed (“1”).
- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading is enabled. Default value is programmed (“0”). The SPIEN Fuse is not accessible in Serial Programming mode.
- The SUT Fuse changes the start-up times. Default value is unprogrammed (“1”).

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

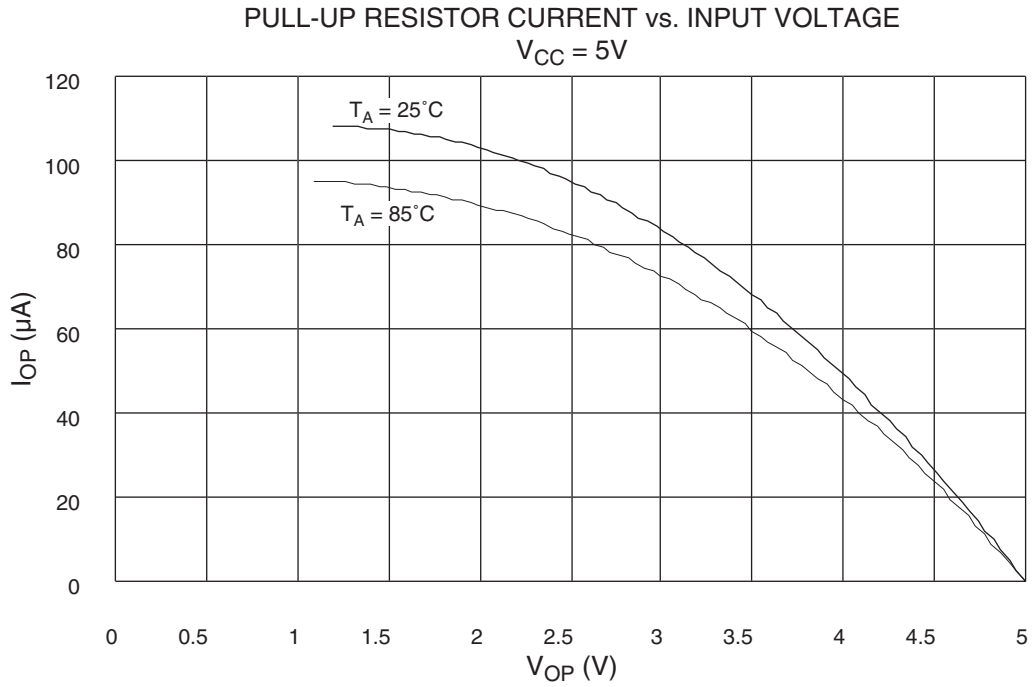
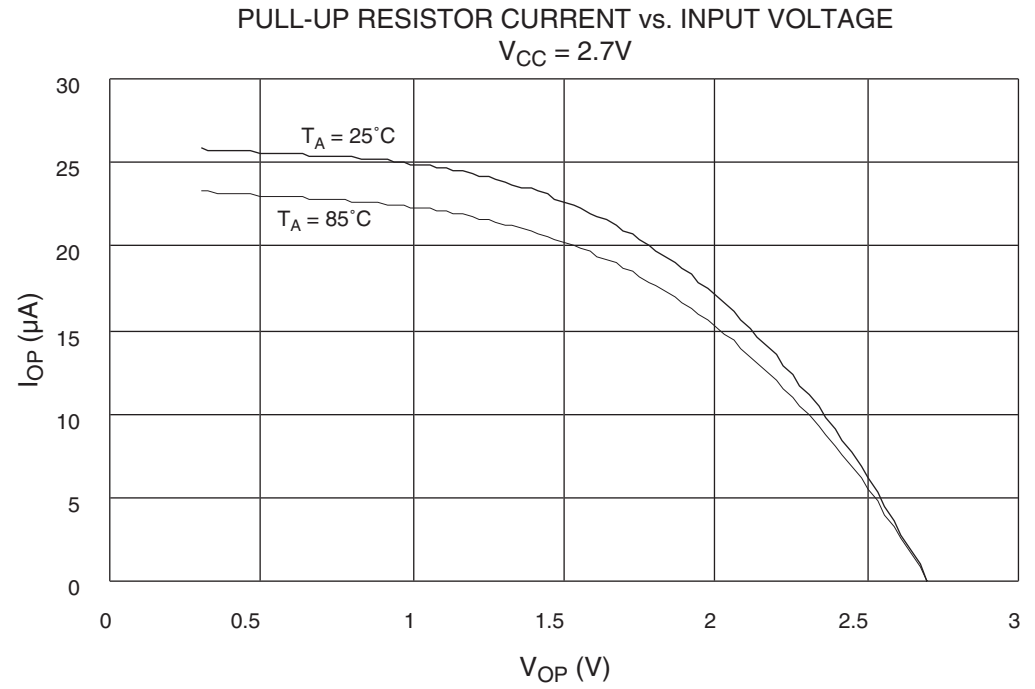



**Figure 92.** Pull-up Resistor Current vs. Input Voltage



**1. Store Program Memory Instruction May Fail**

At certain frequencies and voltages, the store program memory (SPM) instruction may fail.

**Problem Fix/Workaround**

Avoid using the SPM instruction.