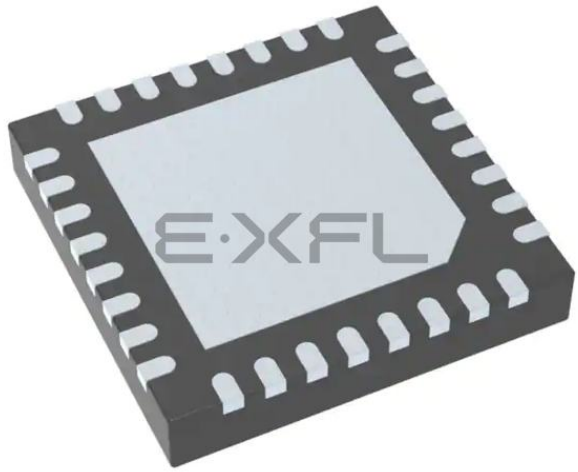


Welcome to E-XFL.COM

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	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny861-15md

7.6 Calibrated Internal RC Oscillator

By default, the internal RC oscillator provides an approximate 8.0MHz clock. Though voltage and temperature dependent, this clock can be very accurately calibrated by the user. See Table 23-1 on page 173 and Section 24.9 “Internal Oscillator Speed” on page 188 for more details. The device is shipped with the CKDIV8 fuse programmed. See Section 7.11 “System Clock Prescaler” on page 31 for more details.

This clock may be selected as the system clock by programming the CKSEL fuses as shown in Table 7-6. If selected, it will operate with no external components. During reset, hardware loads the pre-programmed calibration value into the OSCCAL register and thereby automatically calibrates the RC oscillator. The accuracy of this calibration is shown as factory calibration in Table 23-1 on page 173.

By changing the OSCCAL register from SW, see Section 7.12.1 “OSCCAL – Oscillator Calibration Register” on page 31, it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in Table 23-1 on page 173.

When this oscillator is used as the chip clock, the watchdog oscillator will still be used for the watchdog timer and for the reset time-out. For more information on the pre-programmed calibration value, see Section 22.4 “Calibration Byte” on page 158.

Table 7-6. Internal Calibrated RC Oscillator Operating Modes(1)(2)

Frequency Range (MHz)	CKSEL3..0
7.84 - 8.16	0010

Notes: 1. The device is shipped with this option selected.
2. If 8MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8.

When this oscillator is selected, start-up times are determined by the SUT fuses as shown in Table 7-7.

Table 7-7. Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6CK	14CK	BOD enabled
01	6CK	14CK + 4ms	Fast rising power
10(1)	6CK	14CK + 64ms	Slowly rising power
11	Reserved		

Note: 1. The device is shipped with this option selected.

8. Power Management and Sleep Modes

The high performance and industry leading code efficiency makes the AVR® microcontrollers an ideal choice for low power applications.

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

8.1 Sleep Modes

Figure 7-1 on page 24 presents the different clock systems in the Atmel® ATtiny261/461/861, and their distribution. The figure is helpful in selecting an appropriate sleep mode. Table 8-1 shows the different sleep modes and their wake up sources.

Table 8-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes

	Active Clock Domains					Oscillators	Wake-up Sources					
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{PCK}	Main Clock Source Enabled	INT0, INT1 and Pin Change	SPM/EEPROM Ready	ADC	WDT Interrupt	USI Interrupt	Other I/O
Idle			X	X	X	X	X	X	X	X	X	X
ADC noise reduction				X		X	X(1)	X	X	X	X	
Power-down							X(1)			X	X	
Standby							X(1)			X	X	

Note: 1. For INT0 and INT1, only level interrupt.

To enter any of the three sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM1..0 bits in the MCUCR register select which sleep mode (idle, ADC noise reduction, power-down, or standby) will be activated by the SLEEP instruction. See Table 8-2 on page 37 for a summary.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

8.2 Idle Mode

When the SM1..0 bits are written to 00, the SLEEP instruction makes the MCU enter idle mode, stopping the CPU but allowing analog comparator, ADC, Timer/Counter, watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the timer overflow. If wake-up from the 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. If the ADC is enabled, a conversion starts automatically when this mode is entered.

8.3 ADC Noise Reduction Mode

When the SM1..0 bits are written to 01, the SLEEP instruction makes the MCU enter ADC noise reduction mode, stopping the CPU but allowing the ADC, the external interrupts, and the watchdog to continue operating (if enabled). This sleep mode halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart form the ADC conversion complete interrupt, only an external reset, a watchdog reset, a brown-out reset, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or a pin change interrupt can wake up the MCU from ADC noise reduction mode.

11.1.2 GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	INT1	INT0	PCIE1	PCIE0	–	–	–	–	GIMSK
Read/Write	R/W	R/W	R/W	R/w	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 control0 bits 1/0 (ISC01 and ISC00) in the MCU control register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or 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 the INT1 interrupt vector.

- **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 control register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or 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 the INT0 interrupt vector.

- **Bit 5 – PCIE1: Pin Change Interrupt Enable**

When the PCIE1 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT7..0 or PCINT15..12 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI interrupt vector. PCINT7..0 and PCINT15..12 pins are enabled individually by the PCMSK0 and PCMSK1 register.

- **Bit 4 – PCIE0: Pin Change Interrupt Enable**

When the PCIE0 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT11..8 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI interrupt vector. PCINT11..8 pins are enabled individually by the PCMSK1 register.

- **Bits 3..0 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATtiny261/461/861 and will always read as zero.

11.1.3 GIFR – General Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x3A (0x5A)	INT1	INTF0	PCIF	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7– INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

- **Bit 5 – PCIF: Pin Change Interrupt Flag**

When a logic change on any PCINT15 pin triggers an interrupt request, PCIF becomes set (one). If the I-bit in SREG and the PCIE bit in GIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bits 4:0 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATtiny261/461/861 and will always read as zero.

11.1.4 PCMSK0 – Pin Change Mask Register A

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	**PCINT7**	**PCINT6**	**PCINT5**	**PCINT4**	**PCINT3**	**PCINT2**	**PCINT1**	**PCINT0**	**PCMSK0**
Read/Write	R/W	R/W	R/W	R/w	R/W	R/W	R/W	R/W	
Initial Value	1	1	0	0	1	0	0	0	

- **Bits 7:0 – PCINT7:0: Pin Change Enable Mask 7..0**

Each PCINT7:0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

11.1.5 PCMSK1 – Pin Change Mask Register B

Bit	7	6	5	4	3	2	1	0	
0x22 (0x42)	**PCINT15**	**PCINT14**	**PCINT13**	**PCINT12**	**PCINT11**	**PCINT10**	**PCINT9**	**PCINT8**	**PCMSK1**
Read/Write	R/W	R/W	R/W	R/w	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

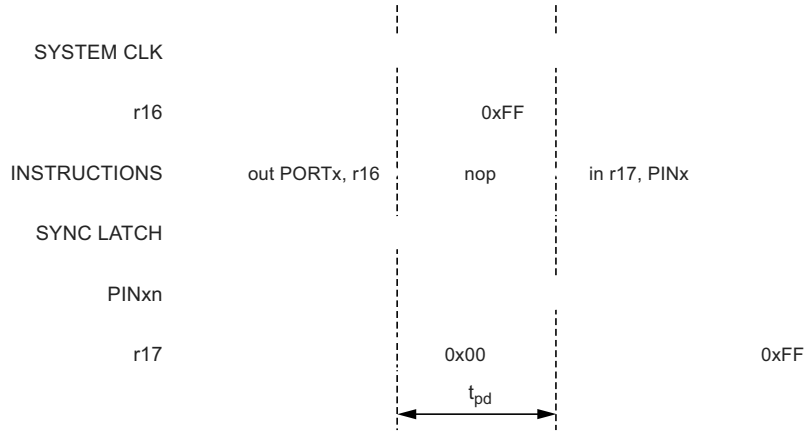
- **Bits 7:0 – PCINT15:8: Pin Change Enable Mask 15..8**

Each PCINT15:8 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT11:8 is set and the PCIE0 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin, and if PCINT15:12 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT15:8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn register at the succeeding positive clock edge. As indicated by the two arrows tpd, max and tpd, min, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 12-4. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is one system clock period.

Figure 12-4. Synchronization when Reading a Software Assigned Pin Value



- **Port A, Bit 2 - ADC2/INT1/USCK/SCL/PCINT2**

ADC2: Analog to digital converter, channel 2.

INT1: The PA2 pin can serve as an external interrupt source 1.

USCK: Three-wire mode universal serial interface clock.

SCL: Two-wire mode serial clock for USI two-wire mode.

PCINT2: Pin change interrupt source 2.

- **Port A, Bit 1 - ADC1/DO/PCINT1**

ADC1: Analog to digital converter, channel 1.

DO: Three-wire mode universal serial interface data output. Three-wire mode data output overrides PORTA1 value and it is driven to the port when data direction bit DDA1 is set. PORTA1 still enables the pull-up, if the direction is input and PORTA1 is set.

PCINT1: Pin change interrupt source 1.

- **Port A, Bit 0 - ADC0/DI/SDA/PCINT0**

ADC0: Analog to digital converter, channel 0.

DI: Data input in USI three-wire mode. USI three-wire mode does not override normal port functions, so pin must be configure as an input for DI function.

SDA: Two-wire mode serial interface data.

PCINT0: Pin change interrupt source 0.

Table 12-7 and Table 12-8 relate the alternate functions of port A to the overriding signals shown in Figure 12-5 on page 57.

Table 12-7. Overriding Signals for Alternate Functions in PA7..PA4

Signal Name	PA7/ADC6/AIN0/ PCINT7	PA6/ADC5/AIN1/ PCINT6	PA5/ADC4/AIN2/ PCINT5	PA4/ADC3/ICP0/ PCINT4
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	0	0	0	0
DIEOE	PCINT7 × PCIE + ADC6D	PCINT6 × PCIE + ADC5D	PCINT5 × PCIE + ADC4D	PCINT4 × PCIE + ADC3D
DIEOV	ADC6D	ADC5D	ADC4D	ADC3D
DI	PCINT7	PCINT6	PCINT5	ICP0/PCINT4
AIO	ADC6, AIN0	ADC5, AIN1	ADC4, AIN2	ADC3

The following code examples show how to access the 16-bit timer registers assuming that no interrupts updates the temporary register. The same principle can be used directly for accessing the OCR0A/B registers.

Assembly Code Example

```
...
; Set TCNT0 to 0x01FF
ldi    r17,0x01
ldi    r16,0xFF
out    TCNT0H,r17
out    TCNT0L,r16
; Read TCNT0 into r17:r16
in     r16,TCNT0L
in     r17,TCNT0H
...
```

C Code Example

```
unsigned int i;
...
/* Set TCNT0 to 0x01FF */
TCNT0H = 0x01;
TCNT0L = 0xff;

/* Read TCNT0 into i */
i = TCNT0L;
i |= ((unsigned int)TCNT0H << 8);
...
```

Note: 1. The example code assumes that the part specific header file is included.
For I/O registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

The assembly code example returns the TCNT0H/L value in the r17:r16 register pair.

It is important to notice that accessing 16-bit registers are atomic operations. If an interrupt occurs between the two instructions accessing the 16-bit register, and the interrupt code updates the temporary register by accessing the same or any other of the 16-bit timer registers, then the result of the access outside the interrupt will be corrupted. Therefore, when both the main code and the interrupt code update the temporary register, the main code must disable the interrupts during the 16-bit access.

The following code examples show how to do an atomic read of the TCNT0 register contents. Reading any of the OCR0 register can be done by using the same principle.

.

Assembly Code Example

```
TIM0_ReadTCNT0:
       ; Save global interrupt flag
       in     r18,SREG
       ; Disable interrupts
       cli
       ; Read TCNT0 into r17:r16
       in     r16,TCNT0L
       in     r17,TCNT0H
       ; Restore global interrupt flag
       out    SREG,r18
       ret
```

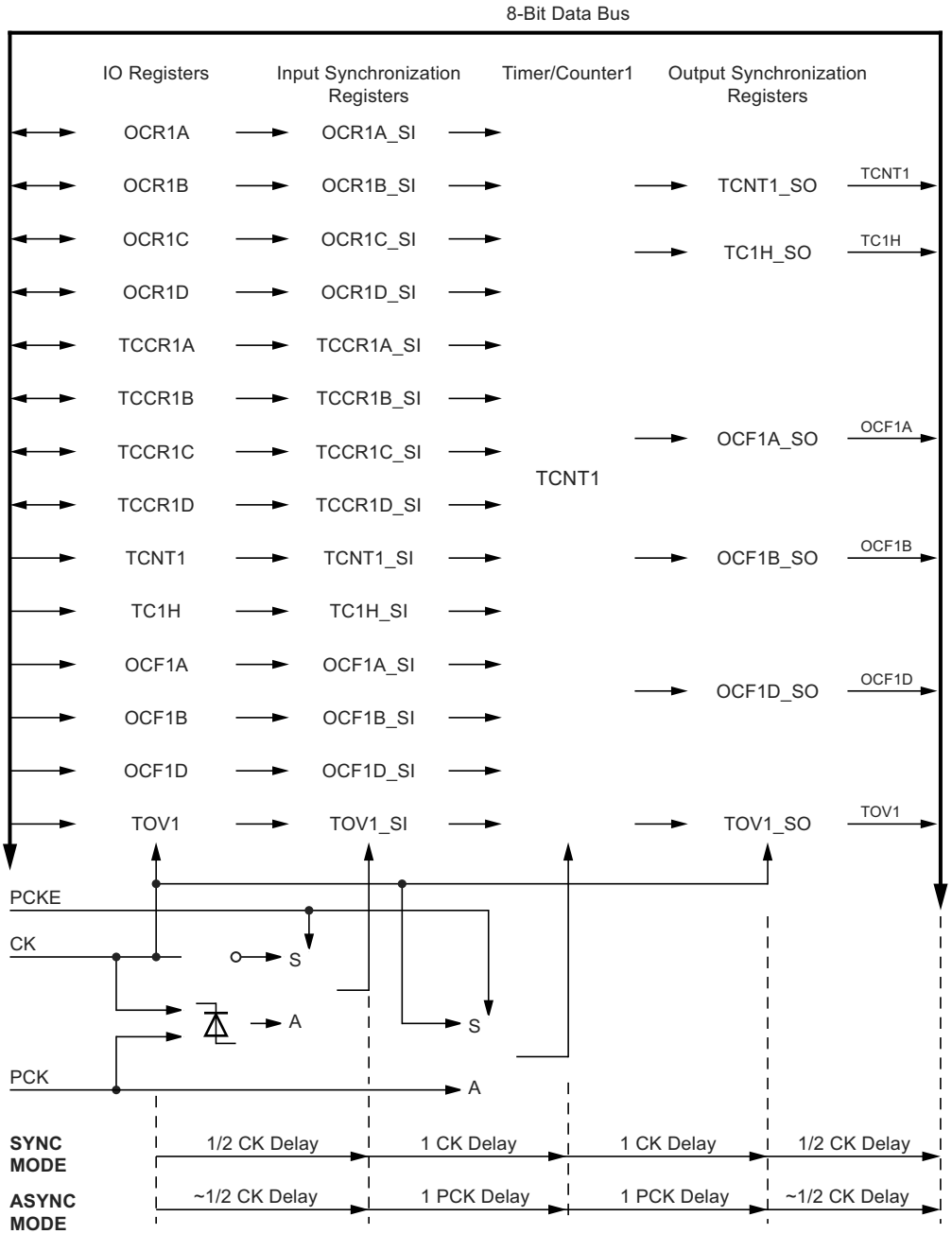
C Code Example

```
unsigned int TIM0_ReadTCNT0( void )
{
unsigned char sreg;
unsigned int i;
/* Save global interrupt flag */
sreg = SREG;
/* Disable interrupts */
_CLI();
/* Read TCNT0 into i */
i = TCNT0L;
i |= ((unsigned int)TCNT0H << 8);
/* Restore global interrupt flag */
SREG = sreg;
return i;
}
```

Note: 1. The example code assumes that the part specific header file is included.
For I/O registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

The assembly code example returns the TCNT0H/L value in the r17:r16 register pair.

Figure 16-2. Timer/Counter1 Synchronization Register Block Diagram



16.11.11 OCR1D – Timer/Counter1 Output Compare Register D

Bit	7	6	5	4	3	2	1	0	
0x2A (0x4A)	MSB							LSB	OCR1D
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 register D is an 8-bit read/write register.

The Timer/Counter output compare register D contains data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in TCCR1A. A compare match does only occur if Timer/Counter1 counts to the OCR1D value. A software write that sets TCNT1 and OCR1D to the same value does not generate a compare match.

A compare match will set the compare interrupt flag OCF1D after a synchronization delay following the compare event.

Note that, if 10-bit accuracy is used special procedures must be followed when accessing the internal 10-bit output compare registers via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105.

16.11.12 TIMSK – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x39 (0x59)	OCIE1D	OCIE1A	OCIE1B	OCIE0A	OCIE0B	TOIE1	TOIE0	TICIE0	TIMSK
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- OCIE1D: Timer/Counter1 Output Compare Interrupt Enable**

When the OCIE1D bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 compare matchD, interrupt is enabled. The corresponding interrupt at vector $010 is executed if a compare matchD occurs. The compare flag in Timer/Counter1 is set (one) in the Timer/Counter interrupt flag register.

- **Bit 6 - OCIE1A: Timer/Counter1 Output Compare Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 compare matchA, interrupt is enabled. The corresponding interrupt at vector $003 is executed if a compare matchA occurs. The compare flag in Timer/Counter1 is set (one) in the Timer/Counter interrupt flag register.

- **Bit 5 - OCIE1B: Timer/Counter1 Output Compare Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 compare matchB, interrupt is enabled. The corresponding interrupt at vector $009 is executed if a compare matchB occurs. The compare flag in Timer/Counter1 is set (one) in the Timer/Counter interrupt flag register.

- **Bit 2 - TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt (at vector $004) is executed if an overflow in Timer/Counter1 occurs. The overflow flag (Timer1) is set (one) in the Timer/Counter interrupt flag register - TIFR.

- **Bit 3:2 – USICS1:0: Clock Source Select**

These bits set the clock source for the USI data register and counter. The data output latch ensures that the output is changed at the opposite edge of the sampling of the data input (DI/SDA) when using external clock source (USCK/SCL). When software strobe or Timer/Counter0 compare match clock option is selected, the output latch is transparent and therefore the output is changed immediately. Clearing the USICS1:0 bits enables software strobe option. When using this option, writing a one to the USICLK bit clocks both the USI data register and the counter. For external clock source (USICS1 = 1), the USICLK bit is no longer used as a strobe, but selects between external clocking and software clocking by the USITC strobe bit.

Table 17-2 on page 128 shows the relationship between the USICS1..0 and USICLK setting and clock source used for the USI data register and the 4-bit counter.

Table 17-2. Relations between the USICS1..0 and USICLK Setting

USICS1	USICS0	USICLK	USI Data Register Clock Source	4-bit Counter Clock Source
0	0	0	No Clock	No Clock
0	0	1	Software clock strobe (USICLK)	Software clock strobe (USICLK)
0	1	X	Timer/Counter0 compare match	Timer/Counter0 compare match
1	0	0	External, positive edge	External, both edges
1	1	0	External, negative edge	External, both edges
1	0	1	External, positive edge	Software clock strobe (USITC)
1	1	1	External, negative edge	Software clock strobe (USITC)

- **Bit 1 – USICLK: Clock Strobe**

Writing a one to this bit location strobes the USI data register to shift one step and the counter to increment by one, provided that the USICS1..0 bits are set to zero and by doing so the software clock strobe option is selected. The output will change immediately when the clock strobe is executed, i.e., in the same instruction cycle. The value shifted into the USI data register is sampled the previous instruction cycle. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1), the USICLK function is changed from a clock strobe to a clock select register. Setting the USICLK bit in this case will select the USITC strobe bit as clock source for the 4-bit counter (see Table 17-2).

- **Bit 0 – USITC: Toggle Clock Port Pin**

Writing a one to this bit location toggles the USCK/SCL value either from 0 to 1, or from 1 to 0. The toggling is independent of the setting in the data direction register, but if the PORT value is to be shown on the pin the DDB2 must be set as output (to one). This feature allows easy clock generation when implementing master devices. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1) and the USICLK bit is set to one, writing to the USITC strobe bit will directly clock the 4-bit counter. This allows an early detection of when the transfer is done when operating as a master device.

17.5.5 USIPP – USI Pin Position

Bit	7	6	5	4	3	2	1	0	
0x11 (0x31)	-	-	-	-	-	-	-	**USIPOS**	**USIPP**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:1 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny261/461/861 and always reads as zero.

- **Bit 0 – USIPOS: USI Pin Position**

Setting this bit to one changes the USI pin position. As default pins PB2..PB0 are used for the USI pin functions, but when writing this bit to one the USIPOS bit is set the USI pin functions are on pins PA2..PA0.

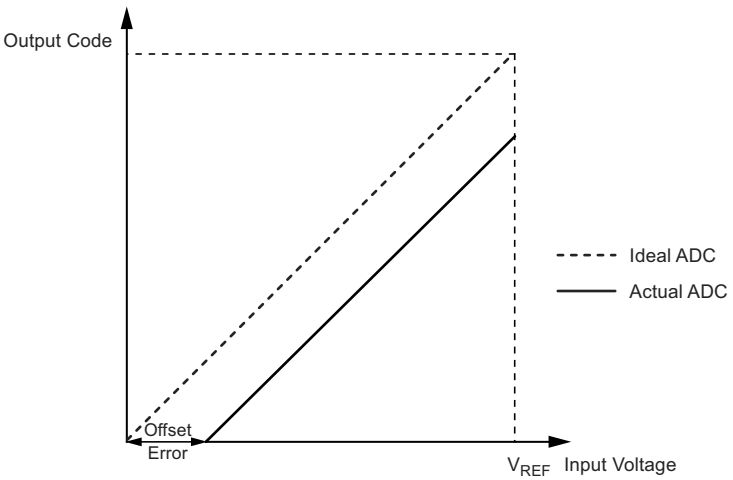
19.7.3 ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^n-1.

Several parameters describe the deviation from the ideal behavior:

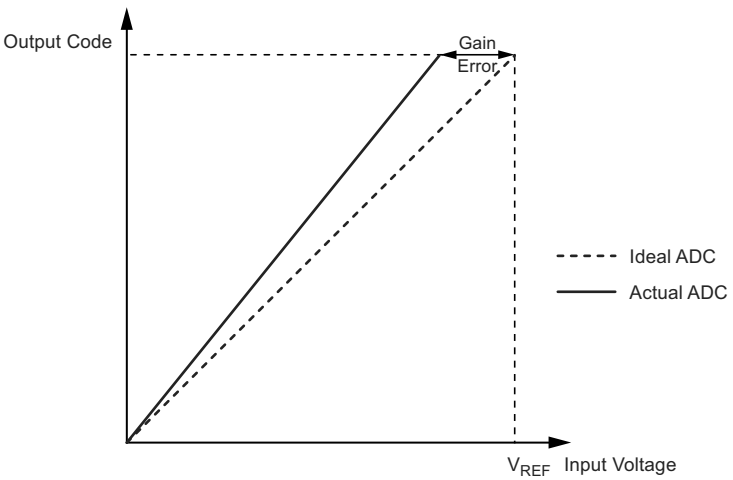
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 19-9. Offset Error



- Gain error: After adjusting for offset, the gain error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB.

Figure 19-10. Gain Error



The values described in Table 19-2 are typical values. However, due to the process variation the temperature sensor output varies from one chip to another.

Table 19-2. Temperature versus Sensor Output Voltage (Typical Case): Example ADC Values

Temperature / °C	–40°C	+25 °C	+125 °C
	0x00F6	0x0144	0c01B8

19.9.1 Manufacturing Calibration

Calibration values determined during test are available in the signature row.

The temperature in degrees Celsius can be calculated using the formula:

$$T = \frac{([\langle ADCH << 8\rangle \ I\ ADCL] - \langle 273 + 25 - TS\_OFFSET\rangle) \times 128}{TS\_GAIN} + 25$$

Where:

a. ADCH & ADCL are the ADC data registers,

b. is the temperature sensor gain

c. TS_OFFSET is the temperature sensor offset correction term

TS_GAIN is the unsigned fixed point 8-bit temperature sensor gain factor in 1/128th units stored in the signature row.

TS_OFFSET is the signed twos complement temperature sensor offset reading stored in the signature row.

The table below summarizes the parameter signature row address vs product.

Table 19-3. Parameter Signature Row Address versus Product

	ATtiny261	ATtiny461	ATtiny861
TS_OFFSET	0x1F	0x05	0x05
TS_GAIN	0x1E	0x07	0x07

The following code example allows to read Signature Row data

```
.equ TS_GAIN = 0x0007
.equ TS_OFFSET = 0x0005
       LDI R30,LOW(TS_GAIN)
       LDI R31,HIGH (TS_GAIN)
       RCALL Read_signature_row
       MOV R17,R16                      ; Save R16 result
       LDI R30,LOW(TS_OFFSET)
       LDI R31,HIGH (TS_OFFSET)
       RCALL Read_signature_row
       ; R16 holds TS_OFFSET and R17 holds TS_GAIN
       Read_signature_row:
       IN R16,SPMCSR                    ; Wait for SPMEN ready
       SBRC R16,SPMEN                   ; Exit loop here when SPMCSR is free
       RJMP Read_signature_row
       LDI R16,((1<<SIGRD)|(1<<SPMEN))  ; We need to set SIGRD and SPMEN together
       OUT SPMCSR,R16                   ; and execute the LPM within 3 cycles
       LPM R16,Z
       RET
```

19.10 Register Description

19.10.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	**REFS1**	**REFS0**	**ADLAR**	**MUX4**	**MUX3**	**MUX2**	**MUX1**	**MUX0**	**ADMUX**
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 7:6 – REFS1:REFS0: Voltage Reference Selection Bits**

These bits and the REFS2 bit from the ADC control and status register B (ADCSRB) select the voltage reference for the ADC, as shown in Table 19-4. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSR is set). Whenever these bits are changed, the next conversion will take 25 ADC clock cycles. If active channels are used, using AVCC or an external AREF higher than (AVCC - 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin.

Table 19-4. Voltage Reference Selections for ADC

REFS2	REFS1	REFS0	Voltage Reference (V_{REF}) Selection
X	0	0	V_{CC} used as voltage reference, disconnected from AREF
X	0	1	External voltage reference at AREF pin, internal voltage reference turned off
0	1	0	Internal 1.1V voltage reference
0	1	1	Reserved
1	1	0	Internal 2.56V voltage reference without external bypass capacitor, disconnected from AREF
1	1	1	Internal 2.56V voltage reference with external bypass capacitor at AREF pin

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC data register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC data register immediately, regardless of any ongoing conversions. For a complete description of this bit, see Section 19.10.3 “ADCL and ADCH – The ADC Data Register” on page 147.

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

These bits and the MUX5 bit from the ADC control and status register B (ADCSRB) select which combination of analog inputs are connected to the ADC. In case of differential input, gain selection is also made with these bits. Selecting the same pin as both inputs to the differential gain stage enables offset measurements. Selecting the single-ended channel ADC11 enables the temperature sensor. Refer to Table 19-5 on page 145 for details. If these bits are changed during a conversion, the change will not go into effect until this conversion is complete (ADIF in ADCSRA is set).

- **Bits 2:0 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected Interrupt Flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to free running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC interrupt flag is set.

Table 19-7. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free running mode
0	0	1	Analog comparator
0	1	0	External interrupt request 0
0	1	1	Timer/Counter0 compare match A
1	0	0	Timer/Counter0 overflow
1	0	1	Timer/Counter0 compare match B
1	1	0	Timer/Counter1 overflow
1	1	1	Watchdog interrupt request

19.10.5 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	ADC6D	ADC5D	ADC4D	ADC3D	AREFD	ADC2D	ADC1D	ADC0D	DIDR0
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	

- **Bits 7:4,2:0 – ADC6D:ADC0D: ADC6:0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC7:0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

- **Bit 3 – AREFD: AREF Digital Input Disable**

When this bit is written logic one, the digital input buffer on the AREF pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the AREF pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

19.10.6 DIDR1 – Digital Input Disable Register 1

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	ADC10D	ADC9D	ADC8D	ADC7D	-				DIDR1
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 – ADC10D..ADC7D: ADC10..7 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC10:7 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

22.2 Fuse Bytes

The ATtiny261/461/861 has three fuse bytes. Table 22-3, Table 22-4 and Table 22-5 describe briefly the functionality of all the fuses and how they are mapped into the fuse bytes. Note that the fuses are read as logical zero, “0”, if they are programmed.

Table 22-3. Fuse Extended Byte

Fuse High Byte	Bit No	Description	Default Value
	7	-	1 (unprogrammed)
	6	-	1 (unprogrammed)
	5	-	1 (unprogrammed)
	4	-	1 (unprogrammed)
	3	-	1 (unprogrammed)
	2	-	1 (unprogrammed)
	1	-	1 (unprogrammed)
SELFPRGEN	0	Self-programming enable	1 (unprogrammed)

Table 22-4. Fuse High Byte

Fuse High Byte	Bit No	Description	Default Value
RSTDISBL(1)	7	External reset disable	1 (unprogrammed)
DWEN(2)	6	DebugWIRE enable	1 (unprogrammed)
SPIEN(3)	6	Enable serial program and data downloading	0 (programmed, SPI prog. enabled)
WDTON(4)	4	Watchdog timer always on	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the chip erase	1 (unprogrammed, EEPROM not preserved)
BODLEVEL2(5)	2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL1(5)	1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL0(5)	0	Brown-out detector trigger level	1 (unprogrammed)

Notes:
1. See Section 12.3.1 “Alternate Functions of Port B” on page 59 for description of RSTDISBL and DWEN fuses.
2. DWEN must be unprogrammed when lock bit security is required. See Section 22.1 “Program And Data Memory Lock Bits” on page 156.
3. The SPIEN fuse is not accessible in SPI programming mode.
4. See Section 9.10.2 “WDTCR – Watchdog Timer Control Register” on page 44 for details.
5. See Table 23-4 on page 174 for BODLEVEL fuse decoding.
6. When programming the RSTDISBL fuse, high-voltage serial programming has to be used to change fuses to perform further programming.

Figure 22-1. Parallel Programming

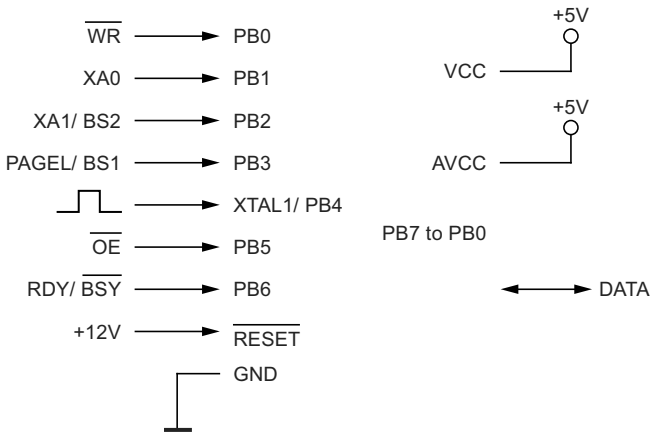


Table 22-9. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
$\overline{WR}$	PB0	I	Write pulse (active low).
XA0	PB1	I	XTAL action bit 0
XA1/BS2	PB2	I	XTAL action bit 1. Byte select 2 (“0” selects low byte, “1” selects 2’nd high byte).
PAGEL/BS1	PB3	I	Byte select 1 (“0” selects low byte, “1” selects high byte). Program memory and EEPROM data page load.
$\overline{OE}$	PB5	I	Output enable (active low).
RDY/$\overline{BSY}$	PB6	O	0: Device is busy programming, 1: Device is ready for new command.
DATA I/O	PA7-PA0	I/O	Bi-directional data bus (output when $\overline{OE}$ is low).

Table 22-10. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL/BS1	Prog_enable[3]	0
XA1/BS2	Prog_enable[2]	0
XA0	Prog_enable[1]	0
WR	Prog_enable[0]	0

Table 22-11. 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

Figure 24-2. Active Supply Current versus Frequency (1 - 16MHz)

24.2 Idle Supply Current

Figure 24-3. Idle Supply Current versus Frequency (1 - 16MHz)

Figure 24-10. I/O Pin Output Voltage versus Source Current (V_{CC} = 3V)

Figure 24-11. I/O Pin Output Voltage versus Source Current (V_{CC} = 5V)

24.7 Pin Threshold and Hysteresis

Figure 24-12. I/O Pin Input Threshold Voltage versus V_{CC} (V_{IH}, I/O Pin Read as '1')

25. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x15 (0x35)	TCCR0A	TCW0	ICEN0	ICNC0	ICES0	ACIC0			CTC0	80
0x14 (0x34)	TCNT0H	Timer/Counter0 counter register high byte								81
0x13 (0x33)	OCR0A	Timer/Counter0 output compare register A								81
0x12 (0x32)	OCR0B	Timer/Counter0 output compare register B								81
0x11 (0x31)	USIPP								USIPOS	128
0x10 (0x30)	USIBR	USI buffer register								126
0x0F (0x2F)	USIDR	USI data register								125
0x0E (0x2E)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	126
0x0D (0x2D)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	127
0x0C (0x2C)	GPIOR2	General purpose I/O register 2								23
0x0B (0x2B)	GPIOR1	General purpose I/O register 1								23
0x0A (0x2A)	GPIOR0	General purpose I/O register 0								23
0x09 (0x29)	ACSRB	HSEL	HLEV				ACM2	ACM1	ACM0	131
0x08 (0x28)	ACSRA	ACD	ACBG	ACO	ACI	ACIE	ACME	ACIS1	ACIS0	129
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	144
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	146
0x05 (0x25)	ADCH	ADC data register high byte								147
0x04 (0x24)	ADCL	ADC data register low byte								147
0x03 (0x23)	ADCSRB	BIN	GSEL		REFS2	MUX5	ADTS2	ADTS1	ADTS0	148
0x02 (0x22)	DIDR1	ADC10D	ADC9D	ADC8D	ADC7D					149
0x01 (0x21)	DIDR0	ADC6D	ADC5D	ADC4D	ADC3D	AREFD	ADC2D	ADC1D	ADC0D	149
0x00 (0x20)	TCCR1E	–	-	OC1OE5	OC1OE4	OC1OE3	OC1OE2	OC1OE1	OC1OE0	115

Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operation the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.