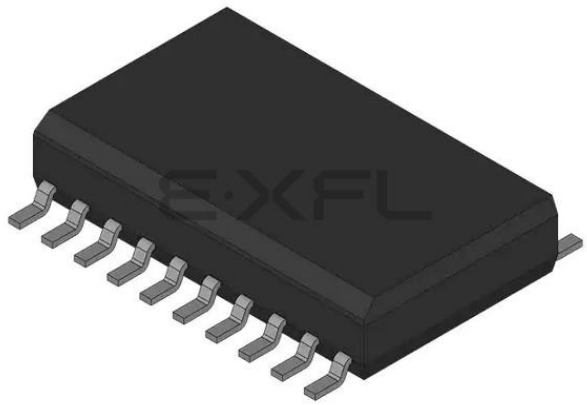


Welcome to E-XFL.COM

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	10MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	2KB (1K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/atmel/attiny261v-10su

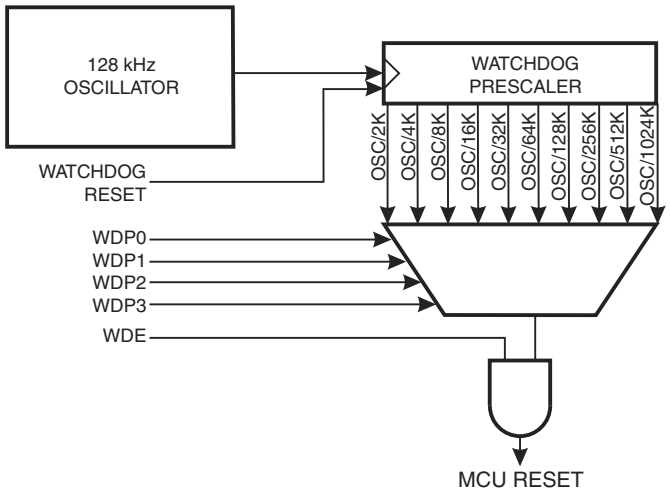
The Wathdog Timer can also be configured to generate an interrupt instead of a reset. This can be very helpful when using the Watchdog to wake-up from Power-down.

To prevent unintentional disabling of the Watchdog or unintentional change of time-out period, two different safety levels are selected by the fuse WDTON as shown in Table 8-1 Refer to “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 45 for details.

Table 8-1. WDT Configuration as a Function of the Fuse Settings of WDTON

WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-out
Unprogrammed	1	Disabled	Timed sequence	No limitations
Programmed	2	Enabled	Always enabled	Timed sequence

Figure 8-7. Watchdog Timer



8.3.1 Timed Sequences for Changing the Configuration of the Watchdog Timer

The sequence for changing configuration differs slightly between the two safety levels. Separate procedures are described for each level.

8.3.1.1 Safety Level 1

In this mode, the Watchdog Timer is initially disabled, but can be enabled by writing the WDE bit to one without any restriction. A timed sequence is needed when disabling an enabled Watchdog Timer. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, in the same operation, write the WDE and WDP bits as desired, but with the WDCE bit cleared.

8.3.1.2 Safety Level 2

In this mode, the Watchdog Timer is always enabled, and the WDE bit will always read as one. A timed sequence is needed when changing the Watchdog Time-out period. To change the Watchdog Time-out, the following procedure must be followed:

8.4 Register Description

8.4.1 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
0x34 (0x54)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bits 7:4 – Res: Reserved Bits**

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

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

8.4.2 WDTCR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
0x21 (0x41)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 7 – WDIF: Watchdog Timeout Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 – WDIE: Watchdog Timeout Interrupt Enable**

When this bit is written to one, WDE is cleared, and the I-bit in the Status Register is set, the Watchdog Time-out Interrupt is enabled. In this mode the corresponding interrupt is executed instead of a reset if a timeout in the Watchdog Timer occurs.

If WDE is set, WDIE is automatically cleared by hardware when a time-out occurs. This is useful for keeping the Watchdog Reset security while using the interrupt. After the WDIE bit is cleared,

the next time-out will generate a reset. To avoid the Watchdog Reset, WDIE must be set after each interrupt.

Table 8-2. Watchdog Timer Configuration

WDE	WDIE	Watchdog Timer State	Action on Time-out
0	0	Stopped	None
0	1	Running	Interrupt
1	0	Running	Reset
1	1	Running	Interrupt

- **Bit 4 – WDCE: Watchdog Change Enable**

This bit must be set when the WDE bit is written to logic zero. Otherwise, the Watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. Refer to the description of the WDE bit for a Watchdog disable procedure. This bit must also be set when changing the prescaler bits. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 45.

- **Bit 3 – WDE: Watchdog Enable**

When the WDE is written to logic one, the Watchdog Timer is enabled, and if the WDE is written to logic zero, the Watchdog Timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logic 0 to WDE. This disables the Watchdog.

In safety level 2, it is not possible to disable the Watchdog Timer, even with the algorithm described above. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 45.

In safety level 1, WDE is overridden by WDRF in MCUSR. See “MCUSR – MCU Status Register” on page 47 for description of WDRF. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared before disabling the Watchdog with the procedure described above. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

Note: If the watchdog timer is not going to be used in the application, it is important to go through a watchdog disable procedure in the initialization of the device. If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset, which in turn will lead to a new watchdog reset. To avoid this situation, the application software should always clear the WDRF flag and the WDE control bit in the initialization routine.

```
0x0003         rjmp   TIM1_COMPA       ; Timer1 CompareA Handler
0x0004         rjmp   TIM1_COMPB       ; Timer1 CompareB Handler
0x0005         rjmp   TIM1_OVF         ; Timer1 Overflow Handler
0x0006         rjmp   TIM0_OVF         ; Timer0 Overflow Handler
0x0007         rjmp   USI_START        ; USI Start Handler
0x0008         rjmp   USI_OVF          ; USI Overflow Handler
0x0009         rjmp   EE_RDY           ; EEPROM Ready Handler
0x000A         rjmp   ANA_COMP         ; Analog Comparator Handler
0x000B         rjmp   ADC              ; ADC Conversion Handler
0x000C         rjmp   WDT              ; WDT Interrupt Handler
0x000D         rjmp   EXT_INT1         ; IRQ1 Handler
0x000E         rjmp   TIM0_COMPA       ; Timer0 CompareA Handler
0x000F         rjmp   TIM0_COMPB       ; Timer0 CompareB Handler
0x0010         rjmp   TIM0_CAPT        ; Timer0 Capture Event Handler
0x0011         rjmp   TIM1_COMPD       ; Timer1 CompareD Handler
0x0012         rjmp   FAULT_PROTECTION ; Timer1 Fault Protection

0x0013  RESET: ldi    r16, low(RAMEND) ; Main program start
0x0014         ldi    r17, high(RAMEND); Tiny861 have also SPH
0x0015         out    SPL, r16         ; Set Stack Pointer to top of RAM
0x0016         out    SPH, r17         ; Tiny861 have also SPH
0x0017         sei                     ; Enable interrupts
0x0018         <instr>
...            ...
```

9.2 External Interrupts

The External Interrupts are triggered by the INT0 or INT1 pin or any of the PCINT15:0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0, INT1 or PCINT15:0 pins are configured as outputs. This feature provides a way of generating a software interrupt. Pin change interrupts PCI will trigger if any enabled PCINT15:0 pin toggles. The PCMSK Register control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT15:0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the MCU Control Register – MCUCR. When the INT0 interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in “Clock Subsystems” on page 24.

9.2.1 Low Level Interrupt

A low level interrupt on INT0 is detected asynchronously. This means that the interrupt source can be used for waking the part also from sleep modes other than Idle (the I/O clock is halted in all sleep modes except Idle).

Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no inter-

11. Timer/Counter0

11.1 Features

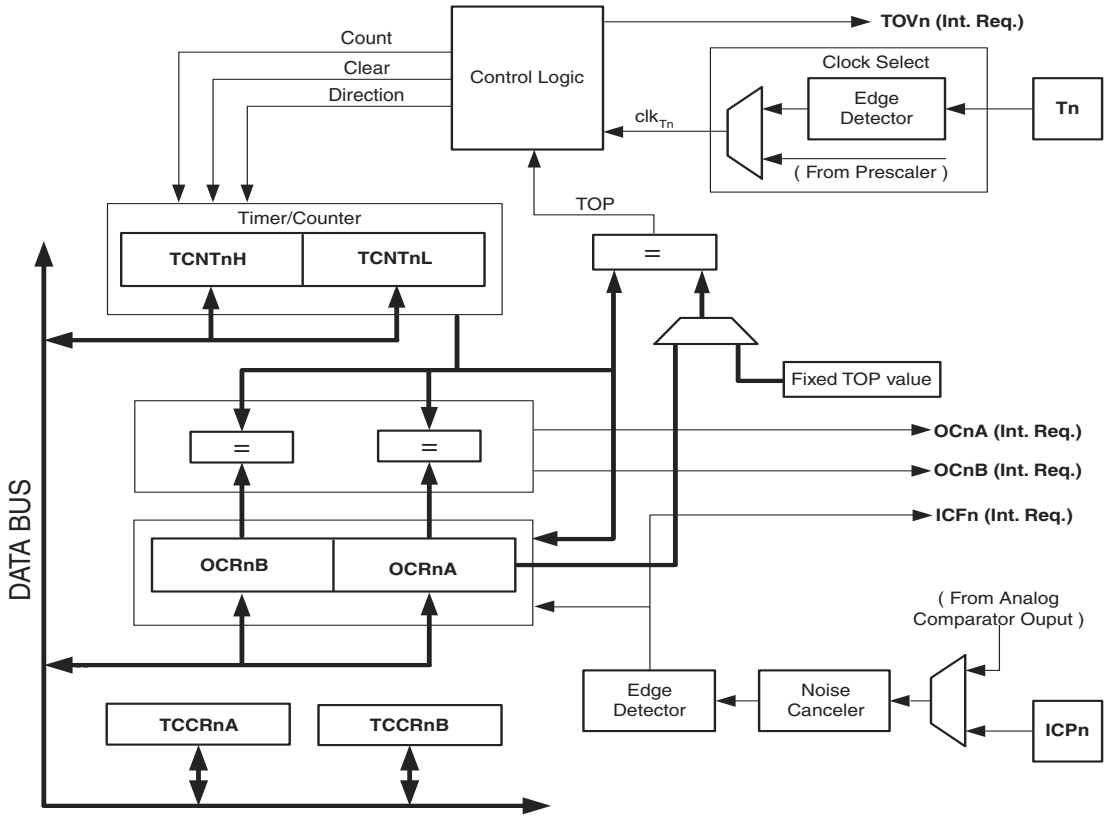
- **Clear Timer on Compare Match (Auto Reload)**
- **One Input Capture unit**
- **Four Independent Interrupt Sources (TOV0, OCF0A, OCF0B, ICF0)**
- **8-bit Mode with Two Independent Output Compare Units**
- **16-bit Mode with One Independent Output Compare Unit**

11.2 Overview

Timer/Counter0 is a general purpose 8/16-bit Timer/Counter module, with two/one Output Compare units and Input Capture feature.

The general operation of Timer/Counter0 is described in 8/16-bit mode. A simplified block diagram of the 8/16-bit Timer/Counter is shown in Figure 11-1. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. For actual placement of I/O pins, refer to “Pinout ATtiny261/461/861 and ATtiny261V/461V/861V” on page 2. Device-specific I/O Register and bit locations are listed in the “Register Description” on page 84.

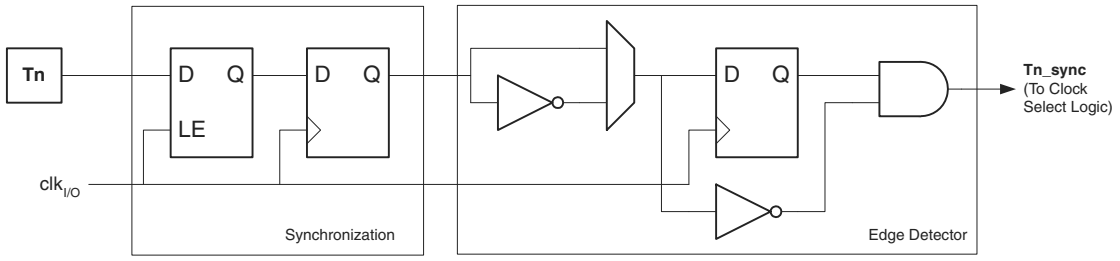
Figure 11-1. 8-/16-bit Timer/Counter Block Diagram



11.2.1 Registers

The Timer/Counter0 Low Byte Register (TCNT0L) and Output Compare Registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request (abbreviated Int.Req. in Figure 11-1) signals are all

Figure 11-3. T0 Pin Sampling



The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T0 pin to the counter is updated.

Enabling and disabling of the clock input must be done when T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

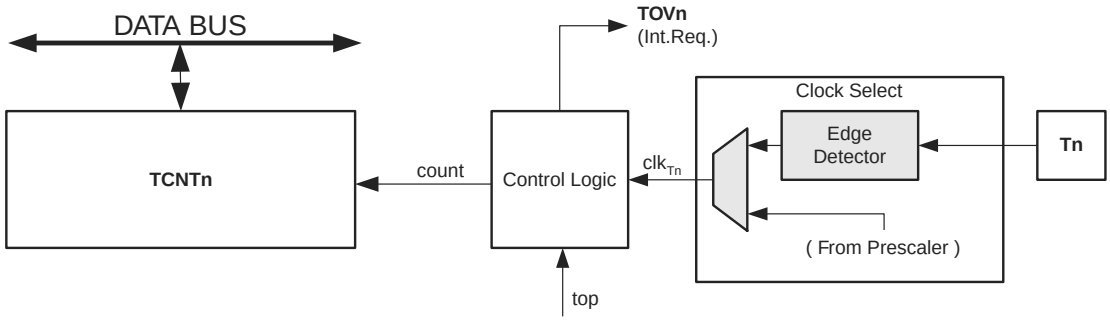
Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk\_I/O}/2$) given a 50/50% duty cycle. Since the edge detector uses sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk\_I/O}/2.5$.

An external clock source can not be prescaled.

11.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 11-4 shows a block diagram of the counter and its surroundings.

Table 11-2. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
clk_{Tn}	Timer/Counter clock, referred to as clk_{T0} in the following.
top	Signalize that TCNT0 has reached maximum value.

The counter is incremented at each timer clock (clk_{T0}) until it passes its TOP value and then restarts from BOTTOM. The counting sequence is determined by the setting of the CTC0 bit located in the Timer/Counter Control Register (TCCR0A). For more details about counting sequences, see “Modes of Operation” on page 77. clk_{T0} can be generated from an external or

cases to consider in the Normal mode, a new counter value can be written anytime. The Output Compare Unit can be used to generate interrupts at some given time.

11.7.4 8-bit Input Capture Mode

The Timer/Counter0 can also be used in an 8-bit Input Capture mode, see Table 11-3 on page 77 for bit settings. For full description, see the section “Input Capture Unit” on page 75.

11.7.5 16-bit Input Capture Mode

The Timer/Counter0 can also be used in a 16-bit Input Capture mode, see Table 11-3 on page 77 for bit settings. For full description, see the section “Input Capture Unit” on page 75.

11.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set. Figure 11-7 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value.

Figure 11-7. Timer/Counter Timing Diagram, no Prescaling

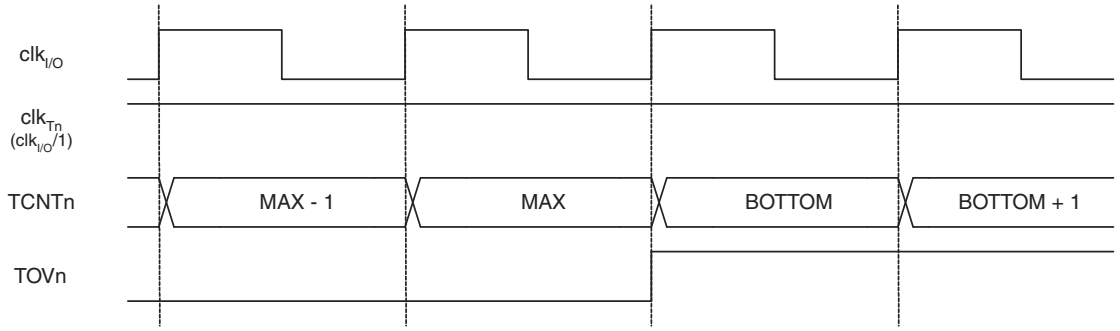


Figure 11-8 shows the same timing data, but with the prescaler enabled.

Figure 11-8. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

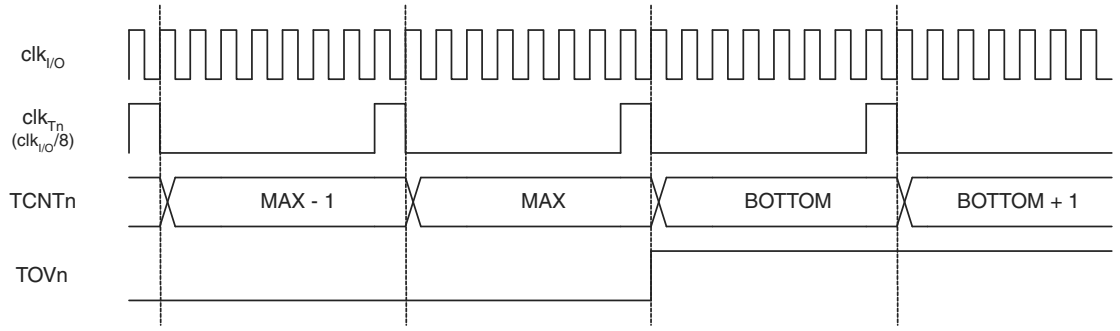


Figure 11-9 on page 80 shows the setting of OCF0A and OCF0B in Normal mode.

11.10 Register Description

11.10.1 TCCR0A – Timer/Counter0 Control Register A

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	TCW0	ICEN0	ICNC0	ICES0	ACIC0	–	–	CTC0	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TCW0: Timer/Counter0 Width**

When this bit is written to one 16-bit mode is selected as described Figure 11-7 on page 79. Timer/Counter0 width is set to 16-bits and the Output Compare Registers OCR0A and OCR0B are combined to form one 16-bit Output Compare Register. Because the 16-bit registers TCNT0H/L and OCR0B/A are accessed by the AVR CPU via the 8-bit data bus, special procedures must be followed. These procedures are described in section “Accessing Registers in 16-bit Mode” on page 80.

- **Bit 6 – ICEN0: Input Capture Mode Enable**

When this bit is written to onem, the Input Capture Mode is enabled.

- **Bit 5 – ICNC0: Input Capture Noise Canceler**

Setting this bit activates the Input Capture Noise Canceler. When the noise canceler is activated, the input from the Input Capture Pin (ICP0) is filtered. The filter function requires four successive equal valued samples of the ICP0 pin for changing its output. The Input Capture is therefore delayed by four System Clock cycles when the noise canceler is enabled.

- **Bit 4 – ICES0: Input Capture Edge Select**

This bit selects which edge on the Input Capture Pin (ICP0) that is used to trigger a capture event. When the ICES0 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES0 bit is written to one, a rising (positive) edge will trigger the capture. When a capture is triggered according to the ICES0 setting, the counter value is copied into the Input Capture Register. The event will also set the Input Capture Flag (ICF0), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

- **Bit 3 - ACIC0: Analog Comparator Input Capture Enable**

When written logic one, this bit enables the input capture function in Timer/Counter0 to be triggered by the Analog Comparator. The comparator output is in this case directly connected to the input capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter0 Input Capture interrupt. When written logic zero, no connection between the Analog Comparator and the input capture function exists. To make the comparator trigger the Timer/Counter0 Input Capture interrupt, the TICIE0 bit in the Timer Interrupt Mask Register (TIMSK) must be set.

- **Bits 2:1 – Res: Reserved Bits**

These bits are reserved and will always read zero.

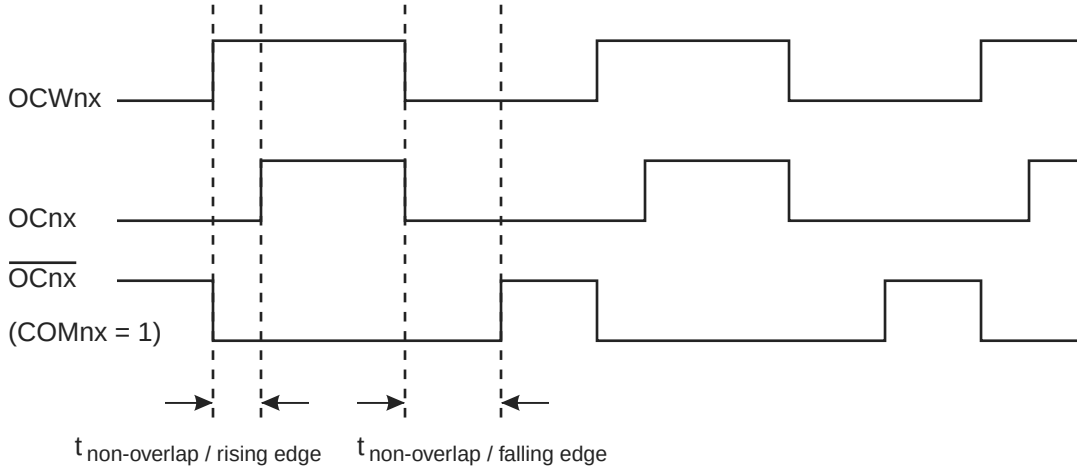
- **Bit 0 – CTC0: Waveform Generation Mode**

This bit controls the counting sequence of the counter, the source for maximum (TOP) counter value, see Figure 11-7 on page 79. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter) and Clear Timer on Compare Match (CTC) mode (see “Modes of Operation” on page 77).

zero. The outputs OC1x and $\overline{OC1x}$ are inverted, if the PWM Inversion Mode bit PWM1X is set. This will also cause both outputs to be high during the dead time.

The length of the counting period is user adjustable by selecting the dead time prescaler setting by using the DTPS11:10 control bits, and selecting then the dead time value in I/O register DT1. The DT1 register consists of two 4-bit fields, DT1H and DT1L that control the dead time periods of the PWM output and its' complementary output separately in terms of the number of prescaled dead time generator clock cycles. Thus the rising edge of OC1x and $\overline{OC1x}$ can have different dead time periods as the $t_{non\text{-}overlap\ /\ rising\ edge}$ is adjusted by the 4-bit DT1H value and the $t_{non\text{-}overlap\ /\ falling\ edge}$ is adjusted by the 4-bit DT1L value.

Figure 12-9. The Complementary Output Pair, COM1x1:0 = 1



12.7 Compare Match Output Unit

The Compare Output Mode (COM1x1:0) bits have two functions. The Waveform Generator uses the COM1x1:0 bits for defining the inverted or non-inverted Waveform Output (OCW1x) at the next Compare Match. Also, the COM1x1:0 bits control the OC1x and $\overline{OC1x}$ pin output source. Figure 12-10 on page 98 shows a simplified schematic of the logic affected by the COM1x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM1x1:0 bits are shown.

In Normal Mode (non-PWM) the Dead Time Generator is disabled and it is working like a synchronizer: the Output Compare (OC1x) is delayed from the Waveform Output (OCW1x) by one timer clock cycle. Whereas in Fast PWM Mode and in Phase and Frequency Correct PWM Mode when the COM1x1:0 bits are set to “01” both the non-inverted and the inverted Output Compare output are generated, and an user programmable Dead Time delay is inserted for these complementary output pairs (OC1x and $\overline{OC1x}$). The functionality in PWM modes is similar to Normal mode when any other COM1x1:0 bit setup is used. When referring to the OC1x state, the reference is for the Output Compare output (OC1x) from the Dead Time Generator, not the OC1x pin. If a system reset occur, the OC1x is reset to “0”.

15. ADC – Analog to Digital Converter

15.1 Features

- **10-bit Resolution**
- **1.0 LSB Integral Non-linearity**
- **± 2 LSB Absolute Accuracy**
- **13µs Conversion Time**
- **15 kSPS at Maximum Resolution**
- **11 Multiplexed Single Ended Input Channels**
- **16 Differential input pairs**
- **15 Differential input pairs with selectable gain**
- **Temperature Sensor Input Channel**
- **Optional Left Adjustment for ADC Result Readout**
- **0 - V_{CC} ADC Input Voltage Range**
- **Selectable 1.1V / 2.56V ADC Voltage Reference**
- **Free Running or Single Conversion Mode**
- **ADC Start Conversion by Auto Triggering on Interrupt Sources**
- **Interrupt on ADC Conversion Complete**
- **Sleep Mode Noise Cancele**
- **Unipolar / Bipolar Input Mode**
- **Input Polarity Reversal Mode**

15.2 Overview

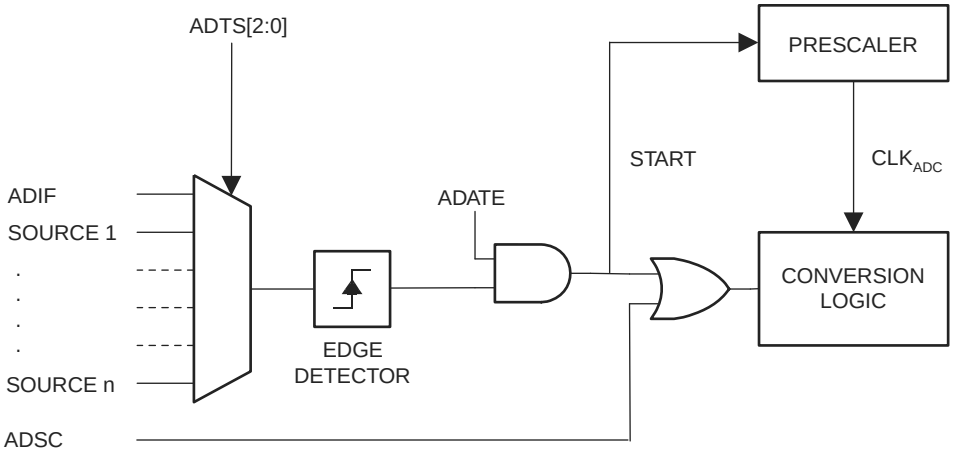
The ATtiny261/461/861 features a 10-bit successive approximation ADC. The ADC is connected to a 11-channel Analog Multiplexer which allows 16 differential voltage input combinations and 11 single-ended voltage inputs constructed from the pins PA7:PA0 or PB7:PB4. The differential input is equipped with a programmable gain stage, providing amplification steps of 1x, 8x, 20x or 32x on the differential input voltage before the A/D conversion. The single-ended voltage inputs refer to 0V (GND).

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 15-1 on page 143.

Internal reference voltages of nominally 1.1V or 2.56V are provided On-chip. The Internal reference voltage of 2.56V, can optionally be externally decoupled at the AREF (PA3) pin by a capacitor, for better noise performance. Alternatively, V_{CC} can be used as reference voltage for single ended channels. There is also an option to use an external voltage reference and turn-off the internal voltage reference. These options are selected using the REFS2:0 bits of the ADCSRB and ADMUX registers.

conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 15-2. ADC Auto Trigger Logic



Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

15.5 Prescaling and Conversion Timing

By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

See Figure 15-3 on page 146.

- When ADATE or ADEN is cleared.
- During conversion, minimum one ADC clock cycle after the trigger event.
- After a conversion, before the Interrupt Flag used as trigger source is cleared.

When updating ADMUX in one of these conditions, the new settings will affect the next ADC conversion.

15.6.1 ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

15.6.2 ADC Voltage Reference

The voltage reference for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either V_{CC}, or internal 1.1V / 2.56V voltage reference, or external AREF pin. The first ADC conversion result after switching voltage reference source may be inaccurate, and the user is advised to discard this result.

15.7 ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode. This reduces noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

- Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled.
- Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
- If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not automatically be turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption.

0x3FF represents the selected voltage reference minus one LSB. The result is presented in one-sided form, from 0x3FF to 0x000.

15.11.2 Unipolar Differential Conversion

If differential channels and an unipolar input mode are used, the result is

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot 1024}{V_{REF}} \cdot GAIN$$

where V_{POS} is the voltage on the positive input pin, V_{NEG} the voltage on the negative input pin, and V_{REF} the selected voltage reference (see Table 15-3 on page 155 and Table 15-4 on page 157). The voltage on the positive pin must always be larger than the voltage on the negative pin or otherwise the voltage difference is saturated to zero. The result is presented in one-sided form, from 0x000 (0d) to 0x3FF (+1023d). The GAIN is either 1x, 8x, 20x or 32x.

15.11.3 Bipolar Differential Conversion

As default the ADC converter operates in the unipolar input mode, but the bipolar input mode can be selected by writting the BIN bit in the ADCSRB to one. In the bipolar input mode two-sided voltage differences are allowed and thus the voltage on the negative input pin can also be larger than the voltage on the positive input pin. If differential channels and a bipolar input mode are used, the result is

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot 512}{V_{REF}} \cdot GAIN$$

where V_{POS} is the voltage on the positive input pin, V_{NEG} the voltage on the negative input pin, and V_{REF} the selected voltage reference. The result is presented in two's complement form, from 0x200 (-512d) through 0x000 (+0d) to 0x1FF (+511d). The GAIN is either 1x, 8x, 20x or 32x.

However, if the signal is not bipolar by nature (9 bits + sign as the 10th bit), this scheme loses one bit of the converter dynamic range. Then, if the user wants to perform the conversion with the maximum dynamic range, the user can perform a quick polarity check of the result and use the unipolar differential conversion with selectable differential input pair. When the polarity check is performed, it is sufficient to read the MSB of the result (ADC9 in ADCH). If the bit is one, the result is negative, and if this bit is zero, the result is positive.

15.12 Temperature Measurement

The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC11 channel. Selecting the ADC11 channel by writing the MUX5:0 bits in ADMUX register to "111111" enables the temperature sensor. The internal 1.1V voltage reference must also be selected for the ADC voltage reference source in the temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

The measured voltage has a linear relationship to the temperature as described in Table 15-2 The sensitivity is approximately 1 LSB / °C and the accuracy depends on the method of user calibration. Typically, the measurement accuracy after a single temperature calibration is ±10°C,

When designing a system where debugWIRE will be used, the following must be observed:

- Pull-Up resistor on the dW/(RESET) line must be in the range of 10k to 20 kΩ. However, the pull-up resistor is optional.
- Connecting the RESET pin directly to V_{CC} will not work.
- Capacitors inserted on the RESET pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

16.4 Software Break Points

debugWIRE supports Program memory Break Points by the AVR Break instruction. Setting a Break Point in AVR Studio® will insert a BREAK instruction in the Program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the Program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a Break Point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of Break Points will therefore reduce the Falsh Data retention. Devices used for debugging purposes should not be shipped to end customers.

16.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as External Reset (RESET). An External Reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e., when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O Registers via the debugger (AVR Studio). See the debugWIRE documentation for detailed description of the limitations.

The debugWIRE interface is asynchronous, which means that the debugger needs to synchronize to the system clock. If the system clock is changed by software (e.g. by writing CLKPS bits) communication via debugWIRE may fail. Also, clock frequencies below 100 kHz may cause communication problems.

A programmed DWEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

16.6 Register Description

The following section describes the registers used with the debugWire.

16.6.1 DWDR – debugWire Data Register

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	DWDR[7:0]								DWDR
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 DWDR Register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

18.7.3 Considerations for Efficient Programming

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

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF, that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address high byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

18.7.4 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 and/or EEPROM are reprogrammed.

1. Load Command “Chip Erase”:
   a. Set XA1, XA0 to “10”. This enables command loading.
   b. Set BS1 to “0”.
   c. Set DATA to “1000 0000”. This is the command for Chip Erase.
   d. Give XTAL1 a positive pulse. This loads the command.
   e. Give $\overline{WR}$ a negative pulse. This starts the Chip Erase. RDY/$\overline{BSY}$ goes low.
   f. Wait until RDY/$\overline{BSY}$ goes high before loading a new command.

Note: The EEPROM memory is preserved during Chip Erase if the EESAVE Fuse is programmed.

18.7.5 Programming the Flash

The Flash is organized in pages, see Table 18-7 on page 173. When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory (see Figure 18-5 for signal waveforms):

1. Load Command “Write Flash”:
   a. Set XA1, XA0 to “10”. This enables command loading.
   b. Set BS1 to “0”.
   c. Set DATA to “0001 0000”. This is the command for Write Flash.
   d. Give XTAL1 a positive pulse. This loads the command.
2. Load Address Low byte:
   a. Set XA1, XA0 to “00”. This enables address loading.
   b. Keep BS1 at “0”. This selects low address.
   c. Set DATA = Address low byte (0x00 - 0xFF).
   d. Give XTAL1 a positive pulse. This loads the address low byte.

19.6 ADC Characteristics

Table 19-7. ADC Characteristics, Single Ended Channels. T = -40°C to +85°C

Symbol	Parameter	Condition	Min(1)	Typ(1)	Max(1)	Units
	Resolution				10	Bits
	Absolute accuracy (Including INL, DNL, and Quantization, Gain and Offset Errors)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		2		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 1 MHz		3		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz Noise Reduction Mode		1.5		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 1 MHz Noise Reduction Mode		2.5		LSB
	Integral Non-Linearity (INL) (Accuracy after Offset and Gain Calibration)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		1		LSB
	Differential Non-linearity (DNL)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		0.5		LSB
	Gain Error	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		2.5		LSB
	Offset Error	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		1.5		LSB
	Conversion Time	Free Running Conversion	13		260	µs
	Clock Frequency		50		1000	kHz
AV_{CC}	Analog Supply Voltage		V_{CC} - 0.3		V_{CC} + 0.3	V
A_{REF}	External Voltage Reference	Single Ended Conversions	2.0		AVCC	V
		Differential Conversions	2.0		AVCC - 1.0	V
V_{IN}	Input Voltage	Single Ended Conversions	GND		V_{REF}	
		Differential Conversions	0		AV_{CC} (2)	V
	Input Bandwidth	Single Ended Conversions		38.5		kHz
		Differential Conversions		4		
V_{INT}	Internal 1.1V Reference		1.0	1.1	1.2	V
	Internal 2.56V Reference (2)		2.3	2.56	2.8	V
R_{REF}	Reference Input Resistance			35		kΩ
R_{AIN}	Analog Input Resistance			100		MΩ
	ADC Conversion Output		0		1023	LSB

Note: 1. Values are guidelines, only.

2. V_{DIFF} must be below V_{REF}.

Figure 20-8. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

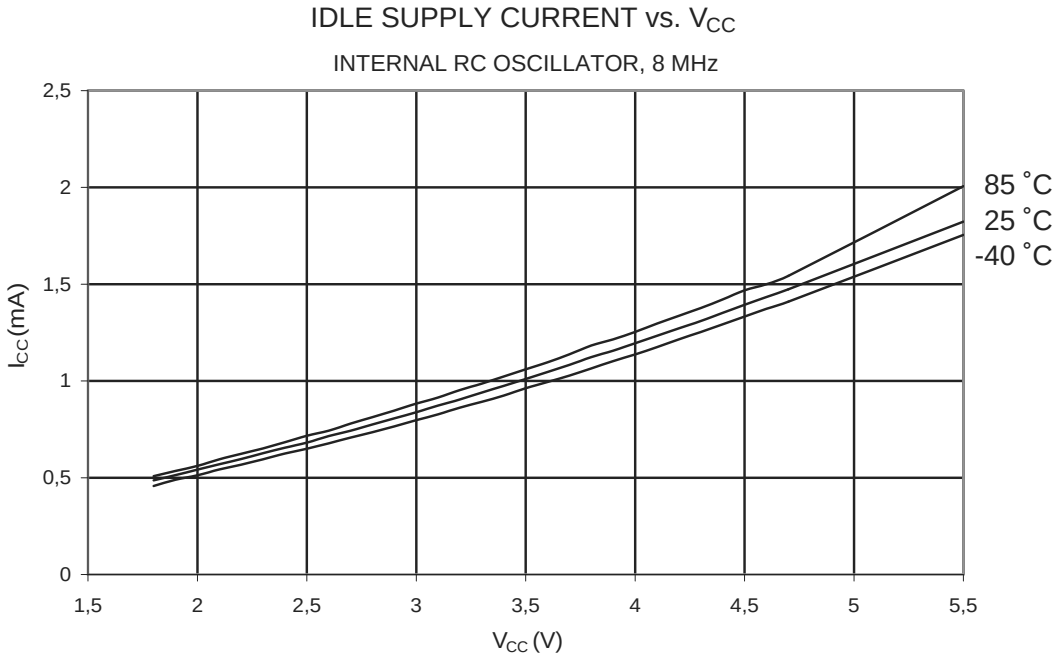


Figure 20-9. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

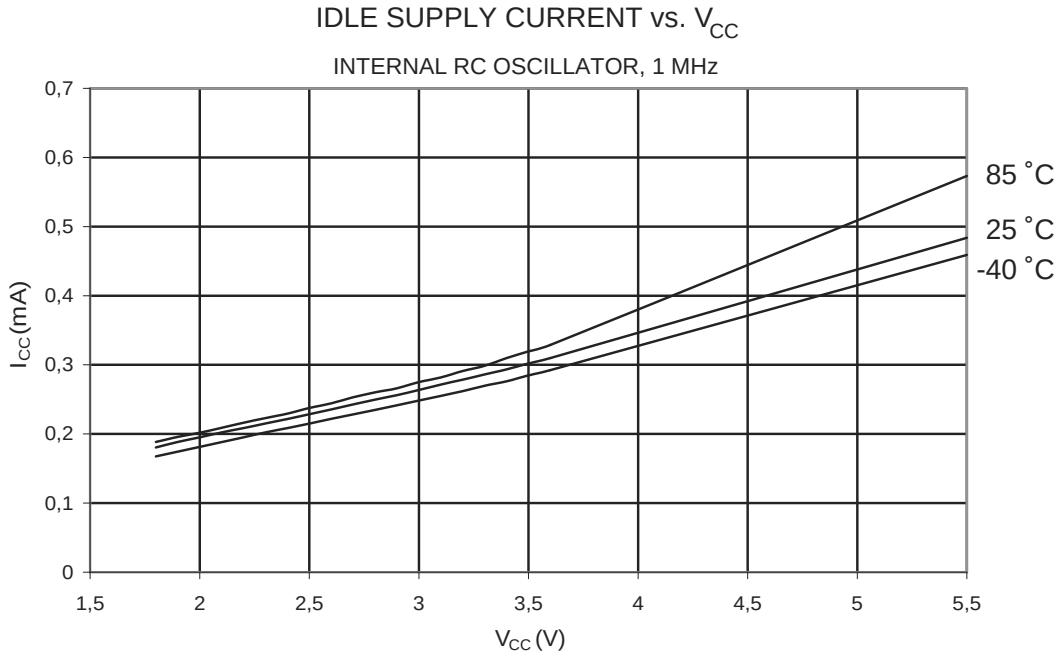
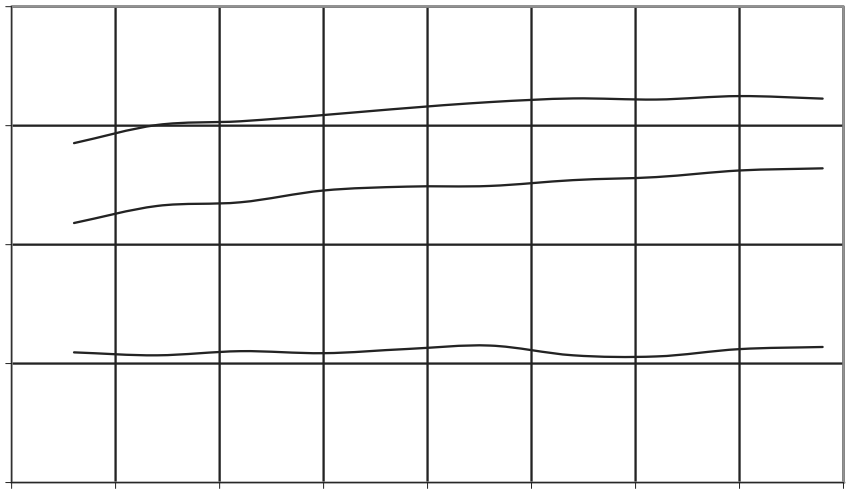


Figure 20-36. Bandgap Voltage vs. Supply Voltage (V_{CC}).



20.9 Internal Oscillator Speed

Figure 20-37. Watchdog Oscillator Frequency vs. V_{CC}

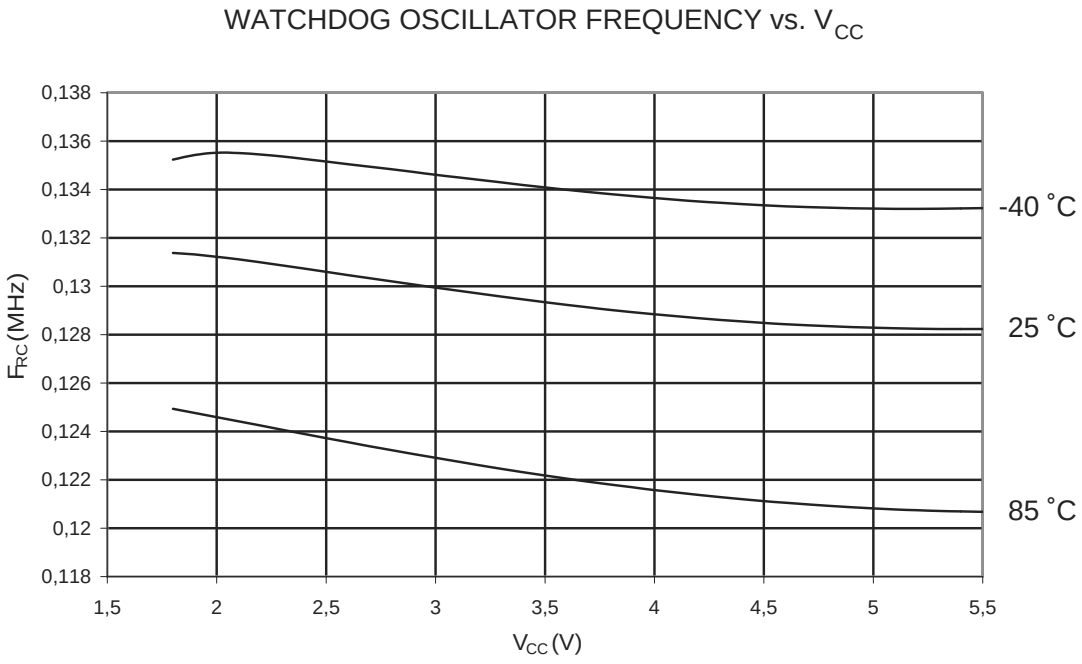


Figure 20-42. AREF External Reference Current vs. V_{CC}

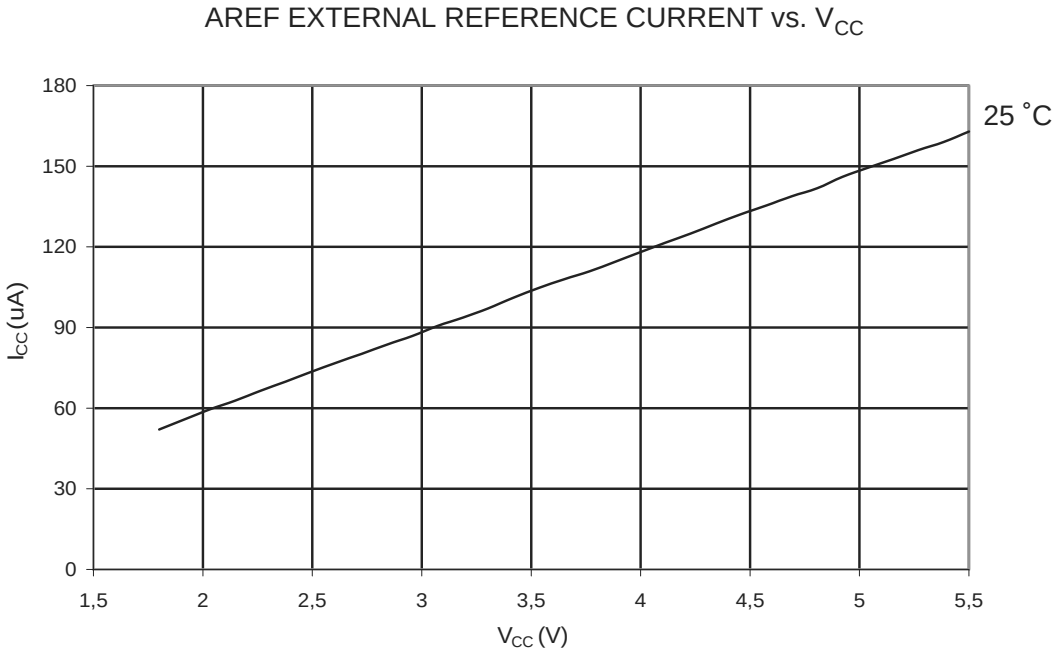
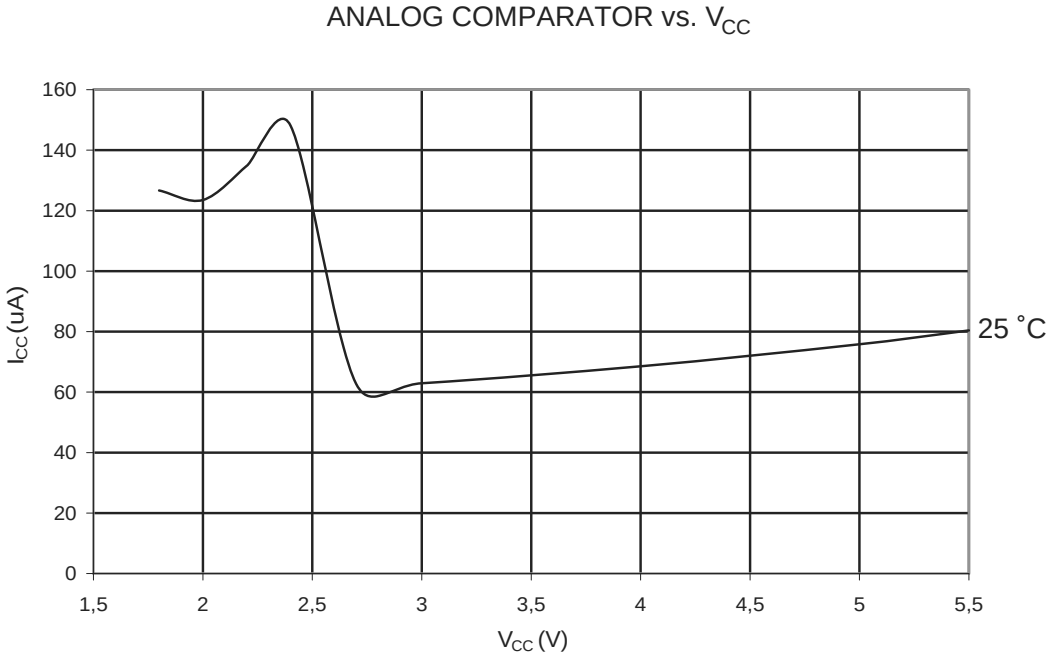


Figure 20-43. Analog Comparator vs. V_{CC}



- "Reset Pin Output Voltage vs. Sink Current (VCC = 3V)" on page 209
- "Reset Pin Output Voltage vs. Sink Current (VCC = 3V)" on page 209
- "Reset Pin Output Voltage vs. Sink Current (VCC = 3V)" on page 209
- "Reset Pin Output Voltage vs. Sink Current (VCC = 3V)" on page 209
- "Bandgap Voltage vs. Supply Voltage (VCC)." on page 216

6. Updated Figures:
   - "Block Diagram" on page 4
   - "Clock Distribution" on page 24
7. Added Table:
   - "Capacitance for Low-Frequency Crystal Oscillator" on page 29
8. Updated Tables:
   - "Start-up Times for the Internal Calibrated RC Oscillator Clock Selection" on page 28
   - "Start-up Times for the 128 kHz Internal Oscillator" on page 29
   - "Active Clock Domains and Wake-up Sources in Different Sleep Modes" on page 36
   - "Serial Programming Characteristics, TA = -40°C to +85°C, VCC = 1.8 - 5.5V (Unless Otherwise Noted)" on page 193
9. Updated Register Descriptions:
   - "TCCR1A – Timer/Counter1 Control Register A" on page 112
   - "TCCR1C – Timer/Counter1 Control Register C" on page 117
   - "ADMUX – ADC Multiplexer Selection Register" on page 155
10. Updated assembly program example in section "Write" on page 17.
11. Updated "DC Characteristics. TA = -40°C to +85°C, VCC = 1.8V to 5.5V (unless otherwise noted)." on page 187.

26.5 Rev. 2588B – 11/06

1. Updated "Ordering Information" on page 227.
2. Updated "Packaging Information" on page 231.

26.6 Rev. 2588A – 10/06

1. Initial Revision.