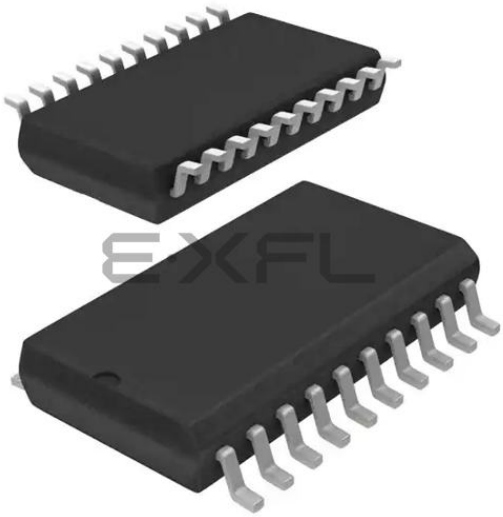


Welcome to E-XFL.COM

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	20MHz
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 ~ 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/microchip-technology/attiny861-20sur

C Code Example

```
_SEI();     /* set Global Interrupt Enable */
_SLEEP();   /* enter sleep, waiting for interrupt */
            /* note: will enter sleep before any pending interrupt(s) */
```

Note: See “Code Examples” on page 6.

4.7.1 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 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. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.

Table 10-4 and Table 10-5 relate the alternate functions of Port A to the overriding signals shown in Figure 10-5 on page 61.

Table 10-4. 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	$\overline{\text{ADC6D}}$	$\overline{\text{ADC5D}}$	$\overline{\text{ADC4D}}$	$\overline{\text{ADC3D}}$
DI	PCINT7	PCINT6	PCINT5	ICP0/PCINT4
AIO	ADC6, AIN0	ADC5, AIN1	ADC4, AIN2	ADC3

Table 10-5. Overriding Signals for Alternate Functions in PA3:PA0

Signal Name	PA3/AREF/ PCINT3	PA2/ADC2/INT1/ USCK/SCL/PCINT2	PA1/ADC1/DO/ PCINT1	PA0/ADC0/DI/SDA/ PCINT0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	USI_TWO_WIRE • USIPOS	0	USI_TWO_WIRE • USIPOS
DDOV	0	($\overline{\text{USI\_SCL\_HOLD}}$ + PORTB2) • DDB2 • USIPOS	0	($\overline{\text{SDA}}$ + $\overline{\text{PORTB0}}$) • DDRB0 • USIPOS
PVOE	0	USI_TWO_WIRE • DDRB2	USI_THREE_WIRE • USIPOS	USI_TWO_WIRE • DDRB0 • USIPOS
PVOV	0	0	DO • USIPOS	0
PTOE	0	USI_PTOE • USIPOS	0	0
DIEOE	PCINT3 • PCIE	PCINT2 • PCIE + INT1 + ADC2D + USISIE • USIPOS	PCINT1 • PCIE + ADC1D	PCINT0 • PCIE + ADC0D + USISIE • USIPOS
DIEOV	0	$\overline{\text{ADC2D}}$	ADC1D	ADC0D
DI	PCINT3	USCK/SCL/INT1/ PCINT2	PCINT1	DI/SDA/PCINT0
AIO	AREF	ADC2	ADC1	ADC0

- ADC10: ADC input Channel 10. Note that ADC input channel 10 uses analog power.
- PCINT15: Pin Change Interrupt source 15.

- **Port B, Bit 6 – ADC9/ T0/ INT0/ PCINT14**
  - ADC9: ADC input Channel 9. Note that ADC input channel 9 uses analog power.
  - T0: Timer/Counter0 counter source.
  - INT0: The PB6 pin can serve as an External Interrupt source 0.
  - PCINT14: Pin Change Interrupt source 14.

- **Port B, Bit 5 – XTAL2/ CLKO/ ADC8/ PCINT13**
  - XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - CLKO: The divided system clock can be output on the PB5 pin, if the CKOUT Fuse is programmed, regardless of the PORTB5 and DDB5 settings. It will also be output during reset.
  - OC1D Output Compare Match output: The PB5 pin can serve as an external output for the Timer/Counter1 Compare Match D when configured as an output (DDA1 set). The OC1D pin is also the output pin for the PWM mode timer function.
  - ADC8: ADC input Channel 8. Note that ADC input channel 8 uses analog power.
  - PCINT13: Pin Change Interrupt source 13.

- **Port B, Bit 4 – XTAL1/ CLKI/ OC1B/ ADC7/ PCINT12**
  - XTAL1/CLKI: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - $\overline{\text{OC1D}}$: Inverted Output Compare Match output: The PB4 pin can serve as an external output for the Timer/Counter1 Compare Match D when configured as an output (DDA0 set). The $\overline{\text{OC1D}}$ pin is also the inverted output pin for the PWM mode timer function.
  - ADC7: ADC input Channel 7. Note that ADC input channel 7 uses analog power.
  - PCINT12: Pin Change Interrupt source 12.

- **Port B, Bit 3 – OC1B/ PCINT11**
  - OC1B, Output Compare Match output: The PB3 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.
  - PCINT11: Pin Change Interrupt source 11.

- **Port B, Bit 2 – SCK/ USCK/ SCL/ $\overline{\text{OC1B}}$/ PCINT10**
  - USCK: Three-wire mode Universal Serial Interface Clock.
  - SCL: Two-wire mode Serial Clock for USI Two-wire mode.
  - $\overline{\text{OC1B}}$: Inverted Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B when configured as an output (DDB2 set). The $\overline{\text{OC1B}}$ pin is also the inverted output pin for the PWM mode timer function.
  - PCINT10: Pin Change Interrupt source 10.

visible in the Timer Interrupt Flag Register (TIFR). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK). TIFR and TIMSK are not shown in the figure.

In 16-bit mode one more 8-bit register is available, the Timer/Counter0 High Byte Register (TCNT0H). Also, in 16-bit mode, there is only one output compare unit as the two Output Compare Registers, OCR0A and OCR0B, are combined to one, 16-bit Output Compare Register, where OCR0A contains the low byte and OCR0B contains the high byte of the word. When accessing 16-bit registers, special procedures described in section “Accessing Registers in 16-bit Mode” on page 80 must be followed.

11.2.2 Definitions

Many register and bit references in this section are written in general form. A lower case “n” replaces the Timer/Counter number, in this case 0. A lower case “x” replaces the Output Compare Unit, in this case Compare Unit A or Compare Unit B. However, when using the register or bit defines in a program, the precise form must be used, i.e. TCNT0L for accessing Timer/Counter0 counter value, and so on.

The definitions in Table 11-1 are also used extensively throughout the document.

Table 11-1. Definitions

Constant	Description
BOTTOM	The counter reaches BOTTOM when it becomes 0x00
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255)
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A Register. The assignment depends on the mode of operation

11.3 Clock Sources

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic is controlled by the Clock Select (CS02:0) bits located in the Timer/Counter Control Register 0 B (TCCR0B), and controls which clock source and edge the Timer/Counter uses to increment its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

11.3.1 Prescaler

The Timer/Counter can be clocked directly by the system clock (by setting the CSn2:0 = 1). This provides the fastest operation, with a maximum Timer/Counter clock frequency equal to system clock frequency ($f_{CLK\_I/O}$). Alternatively, one of four taps from the prescaler can be used as a clock source.

See Figure 11-2 for an illustration of the prescaler unit.

11.7.1 Normal, 8-bit Mode

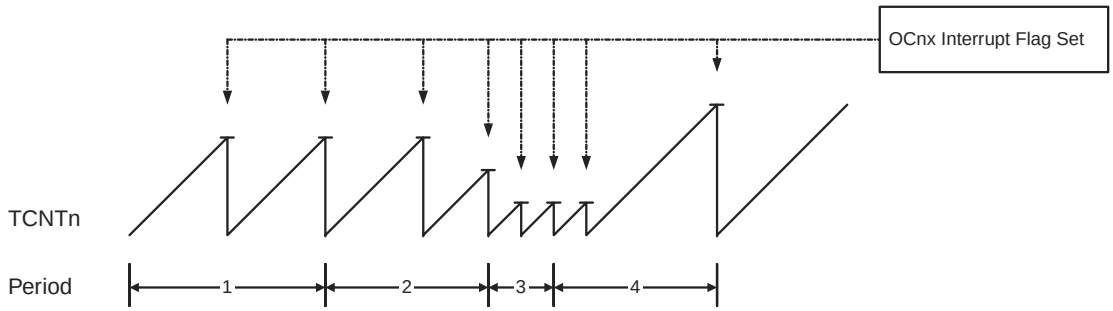
In Normal 8-bit mode (see Table 11-3), the counter (TCNT0L) is incrementing until it overruns when it passes its maximum 8-bit value (MAX = 0xFF) and then restarts from the bottom (0x00). The Overflow Flag (TOV0) is set in the same timer clock cycle as when TCNT0L becomes zero. The TOV0 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal 8-bit mode, a new counter value can be written anytime. The Output Compare Unit can be used to generate interrupts at some given time.

11.7.2 Clear Timer on Compare Match (CTC) 8-bit Mode

In Clear Timer on Compare or CTC mode, see Table 11-3 on page 77, the OCR0A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches the OCR0A. The OCR0A defines the top value for the counter, hence also its resolution. This mode allows greater control of the Compare Match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 11-6. The counter value (TCNT0) increases until a Compare Match occurs between TCNT0 and OCR0A, and then counter (TCNT0) is cleared.

Figure 11-6. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the Compare Match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the Compare Match can occur. As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

11.7.3 Normal, 16-bit Mode

In 16-bit mode, see Table 11-3 on page 77, the counter (TCNT0H/L) is a incrementing until it overruns when it passes its maximum 16-bit value (MAX = 0xFFFF) and then restarts from the bottom (0x0000). The Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0H/L becomes zero. The TOV0 Flag in this case behaves like a 17th bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. There are no special

The dedicated Dead Time prescaler in front of the Dead Time Generator can divide the Timer/Counter1 clock (PCK or CK) by 1, 2, 4 or 8 providing a large range of dead times that can be generated. The Dead Time prescaler is controlled by two bits DTPS11 and DTPS10 from the Dead Time Prescaler register. These bits define the division factor of the Dead Time prescaler. The division factors are given in Table 12-16.

Table 12-16. Division factors of the Dead Time prescaler

DTPS11	DTPS10	Prescaler divides the T/C1 clock by
0	0	1x (no division)
0	1	2x
1	0	4x
1	1	8x

- **Bits 3:0 – CS13, CS12, CS11, CS10: Clock Select Bits 3, 2, 1, and 0**

The Clock Select bits 3, 2, 1, and 0 define the prescaling source of Timer/Counter1.

Table 12-17. Timer/Counter1 Prescaler Select

CS13	CS12	CS11	CS10	Asynchronous Clocking Mode	Synchronous Clocking Mode
0	0	0	0	T/C1 stopped	T/C1 stopped
0	0	0	1	PCK	CK
0	0	1	0	PCK/2	CK/2
0	0	1	1	PCK/4	CK/4
0	1	0	0	PCK/8	CK/8
0	1	0	1	PCK/16	CK/16
0	1	1	0	PCK/32	CK/32
0	1	1	1	PCK/64	CK/64
1	0	0	0	PCK/128	CK/128
1	0	0	1	PCK/256	CK/256
1	0	1	0	PCK/512	CK/512
1	0	1	1	PCK/1024	CK/1024
1	1	0	0	PCK/2048	CK/2048
1	1	0	1	PCK/4096	CK/4096
1	1	1	0	PCK/8192	CK/8192
1	1	1	1	PCK/16384	CK/16384

The Stop condition provides a Timer Enable/Disable function.

the corresponding interrupt handling vector. Alternatively, OCF1A is cleared, after synchronization clock cycle, by writing a logic one to the flag. When the I-bit in SREG, OCIE1A, and OCF1A are set (one), the Timer/Counter1 A compare match interrupt is executed.

- **Bit 5 – OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between Timer/Counter1 and the data value in OCR1B - Output Compare Register 1A. OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1B is cleared, after synchronization clock cycle, by writing a logic one to the flag. When the I-bit in SREG, OCIE1B, and OCF1B are set (one), the Timer/Counter1 B compare match interrupt is executed.

- **Bit 2 – TOV1: Timer/Counter1 Overflow Flag**

In Normal Mode and Fast PWM Mode the TOV1 bit is set (one) each time the counter reaches TOP at the same clock cycle when the counter is reset to BOTTOM. In Phase and Frequency Correct PWM Mode the TOV1 bit is set (one) each time the counter reaches BOTTOM at the same clock cycle when zero is clocked to the counter.

The bit TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared, after synchronization clock cycle, by writing a logical one to the flag. When the SREG I-bit, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflow interrupt is executed.

12.12.15 DT1 – Timer/Counter1 Dead Time Value

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	**DT1H3**	**DT1H2**	**DT1H1**	**DT1H0**	**DT1L3**	**DT1L2**	**DT1L1**	**DT1L0**	**DT1**
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 dead time value register is an 8-bit read/write register.

The dead time delay of all Timer/Counter1 channels are adjusted by the dead time value register, DT1. The register consists of two fields, DT1H3:0 and DT1L3:0, one for each complementary output. Therefore a different dead time delay can be adjusted for the rising edge of OC1x and the rising edge of $\overline{\text{OC1x}}$.

- **Bits 7:4 – DT1H3:DT1H0: Dead Time Value for OC1x Output**

The dead time value for the OC1x output. The dead time delay is set as a number of the prescaled timer/counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

- **Bits 3:0 – DT1L3:DT1L0: Dead Time Value for $\overline{\text{OC1x}}$ Output**

The dead time value for the $\overline{\text{OC1x}}$ output. The dead time delay is set as a number of the prescaled timer/counter clocks. The minimum dead time is zero and the maximum dead time is the prescaled time/counter clock period multiplied by 15.

Table 13-2. Relations between the USICS1:0 and USICLK Setting (Continued)

USICS1	USICS0	USICLK	USI Data Register Clock Source	4-bit Counter Clock Source
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 13-2).

- **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.

13.5.5 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 and will always read 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.

- 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.

assuming calibration at room temperature. Better accuracies are achieved by using two temperature points for calibration.

Table 15-2. Temperature vs. Sensor Output Voltage (Typical Case)

Temperature	-40 °C	+25 °C	+85 °C
ADC	230 LSB	300 LSB	370 LSB

The values described in Table 15-2 are typical values. However, due to process variation the temperature sensor output voltage varies from one chip to another. To be capable of achieving more accurate results the temperature measurement can be calibrated in the application software. The sofware calibration can be done using the formula:

$$T = k * [(ADCH << 8) \mid ADCL] + T_{OS}$$

where ADCH and ADCL are the ADC data registers, k is the fixed slope coefficient and T_{OS} is the temperature sensor offset. Typically, k is very close to 1.0 and in single-point calibration the coefficient may be omitted. Where higher accuracy is required the slope coefficient should be evaluated based on measurements at two temperatures.

15.13 Register Description

15.13.1 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	

- **Bits 7:6 – REFS1:REFS0: Voltage Reference Selection Bits**

These bits together with the REFS2 bit from the ADC Control and Status Register B (ADCSRB) select the voltage reference for the ADC, as shown in Table 15-3.

Table 15-3. Voltage Reference Selections for ADC

REFS2	REFS1	REFS0	Voltage Reference 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

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). Also note, that when these bits are changed, the next conversion will take 25 ADC clock cycles.

Table 15-4. Input Channel Selections (Continued)

MUX5:0	Single-Ended Input	Differential Input		Gain
		Positive	Negative	
100000	N/A	ADC0(PA0)	ADC1(PA1)	20x/32x
100001		ADC0(PA0)	ADC1(PA1)	1x/8x
100010		ADC1(PA1	ADC0(PA0)	20x/32x
100011		ADC1(PA1)	ADC0(PA0)	1x/8x
100100	N/A	ADC1(PA1)	ADC2(PA2)	20x/32x
100101		ADC1(PA1)	ADC2(PA2)	1x/8x
100110		ADC2(PA2	ADC1(PA1)	20x/32x
100111		ADC2(PA2)	ADC1(PA1)	1x/8x
101000	N/A	ADC2(PA2)	ADC0(PA0)	20x/32x
101001		ADC2(PA2)	ADC0(PA0)	1x/8x
101010		ADC0(PA0)	ADC2(PA2)	20x/32x
101011		ADC0(PA0)	ADC2(PA2)	1x/8x
101100	N/A	ADC4(PA5)	ADC5(PA6)	20x/32x
101101		ADC4(PA5)	ADC5(PA6)	1x/8x
101110		ADC5(PA6)	ADC4(PA5)	20x/32x
101111		ADC5(PA6)	ADC4(PA5)	1x/8x
110000	N/A	ADC5(PA6)	ADC6(PA7)	20x/32x
110001		ADC5(PA6)	ADC6(PA7)	1x/8x
110010		ADC6(PA7)	ADC5(PA6)	20x/32x
110011		ADC6(PA7)	ADC5(PA6)	1x/8x
110100	N/A	ADC6(PA7)	ADC4(PA5)	20x/32x
110101		ADC6(PA7)	ADC4(PA5)	1x/8x
110110		ADC4(PA5)	ADC6(PA7)	20x/32x
110111		ADC4(PA5)	ADC6(PA7)	1x/8x
111000	N/A	ADC0(PA0)	ADC0(PA0)	20x/32x
111001		ADC0(PA0)	ADC0(PA0)	1x/8x
111010		ADC1(PA1)	ADC1(PA1)	20x/32x
111011		ADC2(PA2)	ADC2(PA2)	20x/32x
111100	N/A	ADC4(PA5)	ADC4(PA5)	20x/32x
111101		ADC5(PA6)	ADC5(PA6)	20x/32x
111110		ADC6(PA7)	ADC6(PA7)	20x/32x
111111	ADC11 (1)	N/A	N/A	N/A

Note: 1. Temperature sensor

To read the Fuse High Byte (FHB), simply replace the address in the Z-pointer with 0x0003 and repeat the procedure above. If successful, the contents of the destination register are as follows.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

Refer to Table 18-4 on page 171 for detailed description and mapping of the Fuse High Byte.

To read the Fuse Extended Byte (FEB), replace the address in the Z-pointer with 0x0002 and repeat the previous procedure. If successful, the contents of the destination register are as follows.

Bit	7	6	5	4	3	2	1	0
Rd	FEB7	FEB6	FEB5	FEB4	FEB3	FEB2	FEB1	FEB0

Refer to Table 18-3 on page 171 for detailed description and mapping of the Fuse Extended Byte.

17.7 Preventing Flash Corruption

During periods of low V_{CC}, the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
2. Keep the AVR core in Power-down sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

17.8 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. Table 17-1 shows the typical programming time for Flash accesses from the CPU.

Table 17-1. SPM Programming Time(1)

Symbol	Min Programming Time	Max Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.7 ms	4.5 ms

Note: 1. Minimum and maximum programming time is per individual operation.

After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

Table 18-9. Pin Mapping Serial Programming

Symbol	Pins	I/O	Description
MOSI	PB0	I	Serial Data in
MISO	PB1	O	Serial Data out
SCK	PB2	I	Serial Clock

Note: In Table 18-9, above, the pin mapping for SPI programming is listed. Not all parts use the SPI pins dedicated for the internal SPI interface.

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the Serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and EEPROM arrays into 0xFF.

Depending on CKSEL Fuses, a valid clock must be present. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

- Low:> 2 CPU clock cycles for f_{ck} < 12 MHz, 3 CPU clock cycles for f_{ck} >= 12 MHz
- High:> 2 CPU clock cycles for f_{ck} < 12 MHz, 3 CPU clock cycles for f_{ck} >= 12 MHz

18.6.1 Serial Programming Algorithm

When writing serial data to the ATtiny261/461/861, data is clocked on the rising edge of SCK. When reading, data is clocked on the falling edge of SCK. See Figure 19-4 and Figure 19-5 for timing details.

To program and verify the ATtiny261/461/861 in the Serial Programming mode, the following sequence is recommended (see four byte instruction formats in Table 18-11):

1. Power-up sequence:
   Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ and SCK are set to “0”. In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse after SCK has been set to '0'. The duration of the pulse must be at least t_{RST} (the minimum pulse width on $\overline{\text{RESET}}$ pin, see Table 19-4 on page 190) plus two CPU clock cycles.
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync. the second byte (0x53), will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the 0x53 did not echo back, give $\overline{\text{RESET}}$ a positive pulse and issue a new Programming Enable command.
4. The Flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 5 LSB of the address and data together with the Load Program memory Page instruction. To ensure correct loading of the page, the data low byte must be loaded before data high byte is applied for a given address. The Program memory Page is stored by loading the Write Program memory Page instruction with the 6 MSB of the address. If polling (RDY/BSY) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page. (See Table 18-10.) Accessing the serial programming

Table 18-13. 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

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in Table 18-14.

Table 18-14. 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

When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The different Commands are shown in Table 18-15.

Table 18-15. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse bits
0010 0000	Write Lock bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes and Calibration byte
0000 0100	Read Fuse and Lock bits
0000 0010	Read Flash
0000 0011	Read EEPROM

18.7.2 Entering Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5V between V_{CC} and GND.
2. Set $\overline{RESET}$ to “0” and toggle XTAL1 at least six times.
3. Set Prog_enable pins listed in Table 18-13 on page 179 to “0000” and wait at least 100 ns.
4. Apply 11.5 - 12.5V to $\overline{RESET}$. Any activity on Prog_enable pins within 100 ns after +12V has been applied to $\overline{RESET}$, will cause the device to fail entering programming mode.
5. Wait at least 50 µs before sending a new command.

3. Load Data Low Byte:
   a. Set XA1, XA0 to “01”. This enables data loading.
   b. Set DATA = Data low byte (0x00 - 0xFF).
   c. Give XTAL1 a positive pulse. This loads the data byte.
4. Load Data High Byte:
   a. Set BS1 to “1”. This selects high data byte.
   b. Keep XA1, XA0 at “01”. This enables data loading.
   c. Set DATA = Data high byte (0x00 - 0xFF).
   d. Give XTAL1 a positive pulse. This loads the data byte.
5. Repeat steps 2 to 4 until the entire buffer is filled or until all data within the page is loaded.
6. Load Address High byte:
   a. Set XA1, XA0 to “00”. This enables address loading.
   b. Set BS1 to “1”. This selects high address.
   c. Set DATA = Address high byte (0x00 - 0xFF).
   d. Give XTAL1 a positive pulse. This loads the address high byte.
7. Program Page:
   a. Give $\overline{\text{WR}}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{\text{BSY}}$ goes low.
   b. Wait until RDY/$\overline{\text{BSY}}$ goes high.
8. Repeat steps 2 to 7 until the entire Flash is programmed or until all data has been programmed.
9. End Page Programming:
   a. Set XA1, XA0 to “10”. This enables command loading.
   b. Set DATA to “0000 0000”. This is the command for No Operation.
   c. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in Figure 18-4. Note that if less than eight bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address low byte are used to address the page when performing a Page Write.

18.7.8 Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to “Programming the Flash” on page 180 for details on Command and Address loading):

1. A: Load Command “0000 0011”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. Set $\overline{OE}$ to “0”, and BS1 to “0”. The EEPROM Data byte can now be read at DATA.
5. Set $\overline{OE}$ to “1”.

18.7.9 Programming the Fuse Low Bits

The algorithm for programming the Fuse Low bits is as follows (refer to “Programming the Flash” on page 180 for details on Command and Data loading):

1. A: Load Command “0100 0000”.
2. C: Load Data Low Byte. Bit n = “0” programs and bit n = “1” erases the Fuse bit.
3. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

18.7.10 Programming the Fuse High Bits

The algorithm for programming the Fuse High bits is as follows (refer to “Programming the Flash” on page 180 for details on Command and Data loading):

1. A: Load Command “0100 0000”.
2. C: Load Data Low Byte. Bit n = “0” programs and bit n = “1” erases the Fuse bit.
3. Set BS1 to “1” and BS2 to “0”. This selects high data byte.
4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
5. Set BS1 to “0”. This selects low data byte.

18.7.11 Programming the Extended Fuse Bits

The algorithm for programming the Extended Fuse bits is as follows (refer to “Programming the Flash” on page 180 for details on Command and Data loading):

1. 1. A: Load Command “0100 0000”.
2. 2. C: Load Data Low Byte. Bit n = “0” programs and bit n = “1” erases the Fuse bit.
3. 3. Set BS1 to “0” and BS2 to “1”. This selects extended data byte.
4. 4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
5. 5. Set BS2 to “0”. This selects low data byte.

19.7 Serial Programming Characteristics

Figure 19-4. Serial Programming Waveforms

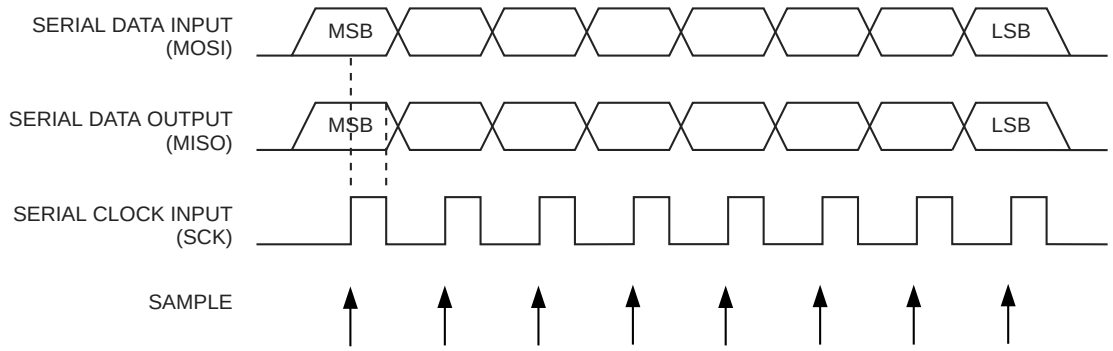


Figure 19-5. Serial Programming Timing

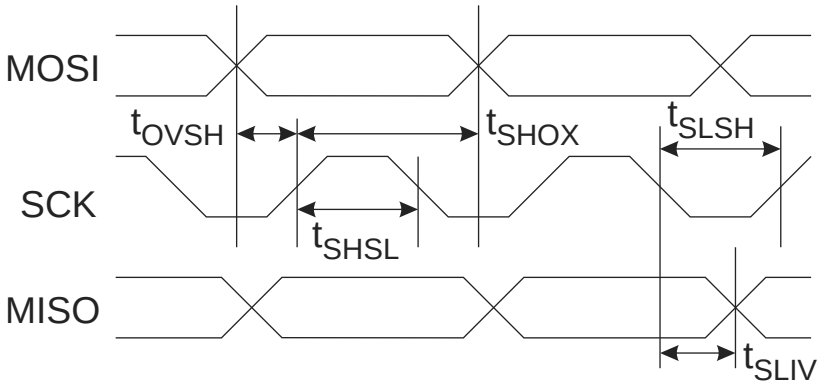
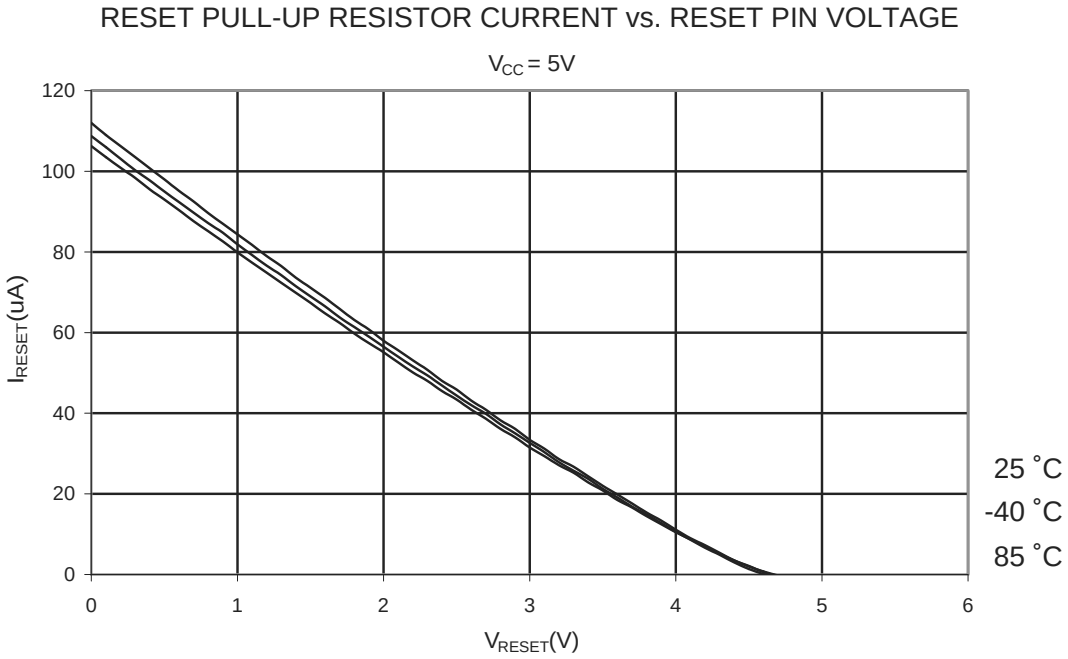


Table 19-8. Serial Programming Characteristics, T_A = -40°C to +85°C, V_{CC} = 1.8 - 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency (ATtiny261V/461V/861V)	0		4	MHz
t_{CLCL}	Oscillator Period (ATtiny261V/461V/861V)	250			ns
$1/t_{CLCL}$	Oscillator Frequency (ATtiny261/461/861, V_{CC} = 4.5 - 5.5V)	0		20	MHz
t_{CLCL}	Oscillator Period (ATtiny261/461/861, V_{CC} = 4.5 - 5.5V)	50			ns
t_{SHSL}	SCK Pulse Width High	$2\ t_{CLCL}$ (1)			ns
t_{SLSH}	SCK Pulse Width Low	$2\ t_{CLCL}$ (1)			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	$2\ t_{CLCL}$			ns
t_{SLIV}	SCK Low to MISO Valid			100	ns

Note: 1. $2\ t_{CLCL}$ for f_{ck} < 12 MHz, $3\ t_{CLCL}$ for f_{ck} >= 12 MHz

Figure 20-18. Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5V)



20.6 Pin Driver Strength

Figure 20-19. I/O Pin Output Voltage vs. Sink Current (V_{CC} = 3V)

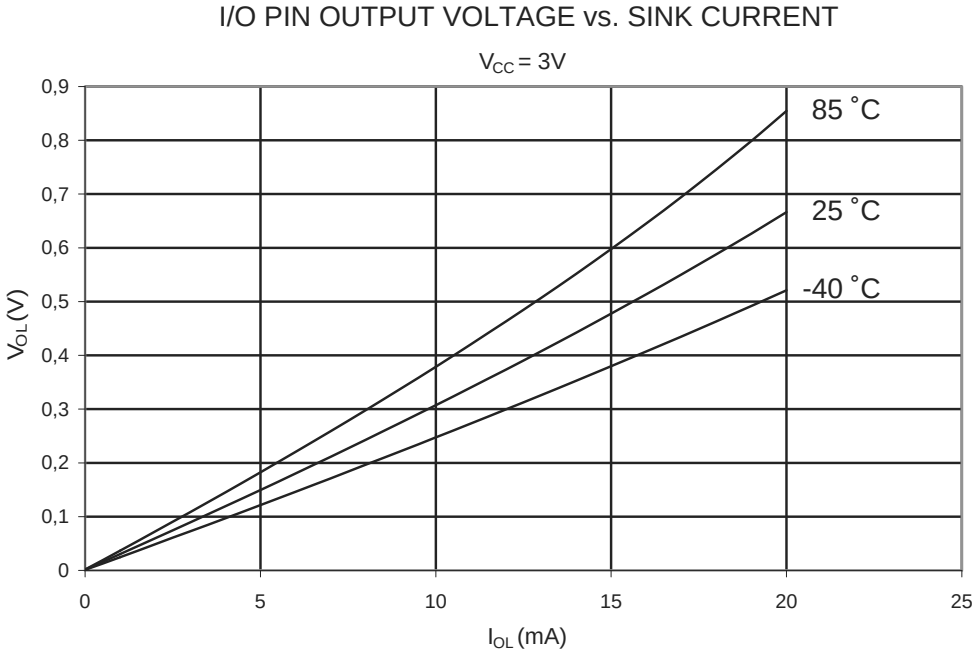


Figure 20-44. Brownout Detector Current vs. V_{CC}

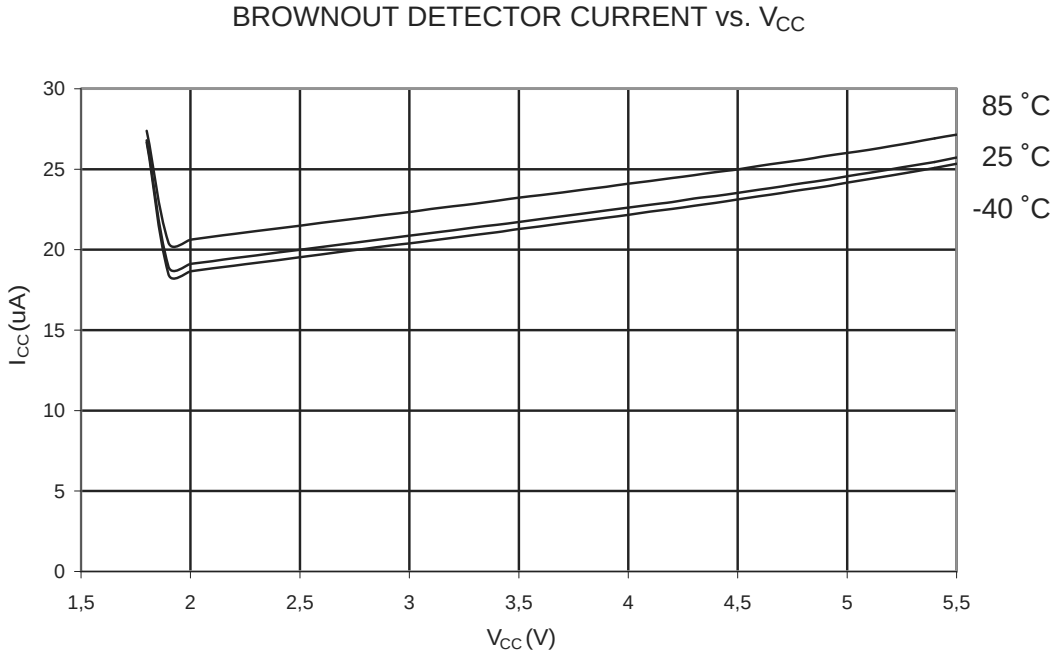
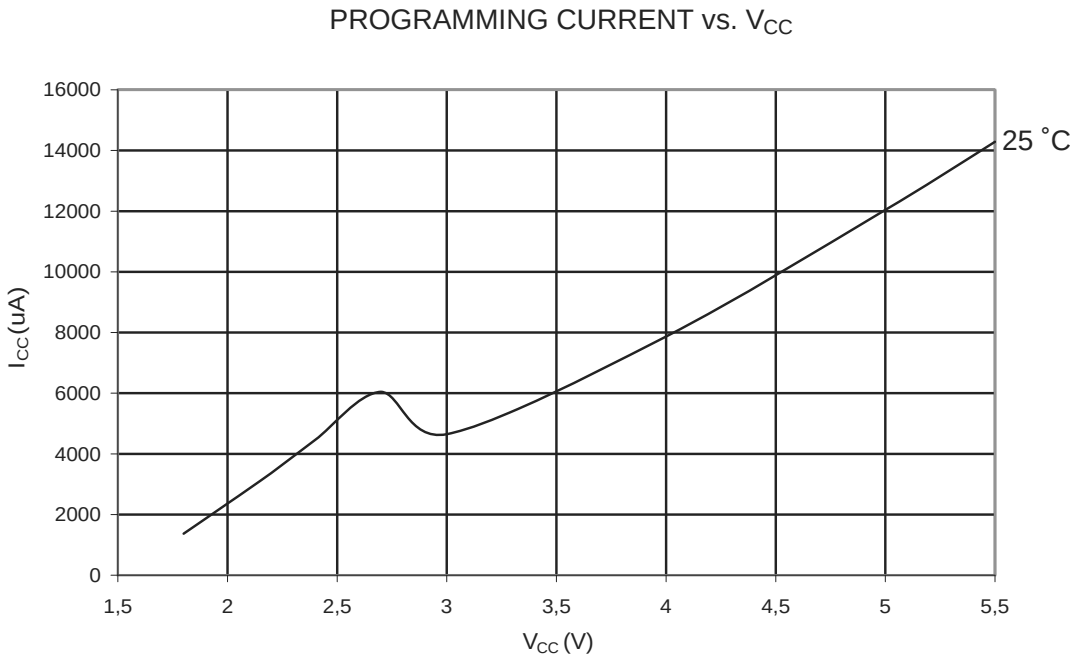


Figure 20-45. Programming Current vs. V_{CC}



- "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.