

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega88v-10pu

dependent as shown in ["ATmega48/88/168 Typical Characteristics – Preliminary Data" on page 307](#).

Table 6-2. Number of Watchdog Oscillator Cycles

Typ Time-out ($V_{CC} = 5.0V$)	Typ Time-out ($V_{CC} = 3.0V$)	Number of Cycles
0 ms	0 ms	0
4.1 ms	4.3 ms	4K (4,096)
65 ms	69 ms	8K (8,192)

Main purpose of the delay is to keep the AVR in reset until it is supplied with minimum V_{CC} . The delay will not monitor the actual voltage and it will be required to select a delay longer than the V_{CC} rise time. If this is not possible, an internal or external Brown-Out Detection circuit should be used. A BOD circuit will ensure sufficient V_{CC} before it releases the reset, and the time-out delay can be disabled. Disabling the time-out delay without utilizing a Brown-Out Detection circuit is not recommended.

The oscillator is required to oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock, and keeps the internal reset active for a given number of clock cycles. The reset is then released and the device will start to execute. The recommended oscillator start-up time is dependent on the clock type, and varies from 6 cycles for an externally applied clock to 32K cycles for a low frequency crystal.

The start-up sequence for the clock includes both the time-out delay and the start-up time when the device starts up from reset. When starting up from Power-save or Power-down mode, V_{CC} is assumed to be at a sufficient level and only the start-up time is included.

6.3 Low Power Crystal Oscillator

Pins XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in [Figure 6-2](#). Either a quartz crystal or a ceramic resonator may be used.

This Crystal Oscillator is a low power oscillator, with reduced voltage swing on the XTAL2 output. It gives the lowest power consumption, but is not capable of driving other clock inputs, and may be more susceptible to noise in noisy environments. In these cases, refer to the ["Full Swing Crystal Oscillator" on page 29](#).

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in [Table 6-3](#). For ceramic resonators, the capacitor values given by the manufacturer should be used.

8.0.5 Brown-out Detection

ATmega48/88/168 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL Fuses. The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

Table 8-2. BODLEVEL Fuse Coding⁽¹⁾

BODLEVEL 2..0 Fuses	Min V _{BOT}	Typ V _{BOT}	Max V _{BOT}	Units
111	BOD Disabled			
110	1.7 ⁽²⁾	1.8	2.0 ⁽²⁾	V
101	2.5 ⁽²⁾	2.7	2.9 ⁽²⁾	
100	4.1 ⁽²⁾	4.3	4.5 ⁽²⁾	
011	Reserved			
010				
001				
000				

- Notes:
- V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{BOT}$ during the production test. This guarantees that a Brown-Out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 110 and BODLEVEL = 101 for ATmega48V/88V/168V, and BODLEVEL = 101 and BODLEVEL = 101 for ATmega48/88/168.
 - Min/Max values applicable for ATmega48.

Table 8-3. Brown-out Characteristics

Symbol	Parameter	Min	Typ	Max	Units
V_{HYST}	Brown-out Detector Hysteresis		50		mV
t_{BOD}	Min Pulse Width on Brown-out Reset		2		μs

When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in [Figure 8-5](#)), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in [Figure 8-5](#)), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in [Table 8-1](#).

Figure 12-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk_I/O}/8$)

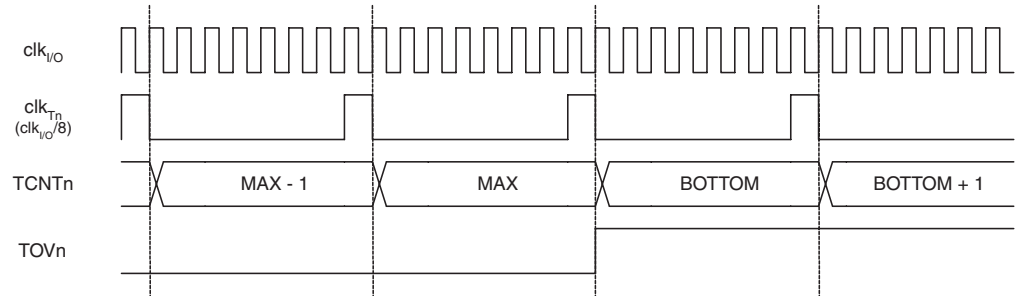


Figure 12-10 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 12-10. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk_I/O}/8$)

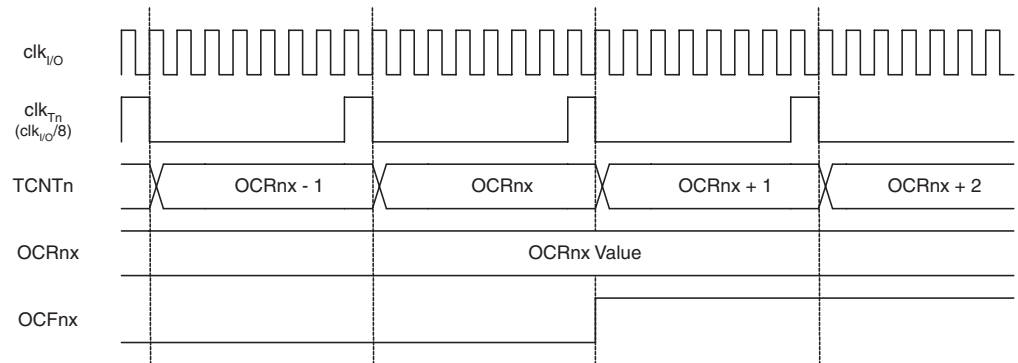


Figure 12-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 12-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk_I/O}/8$)

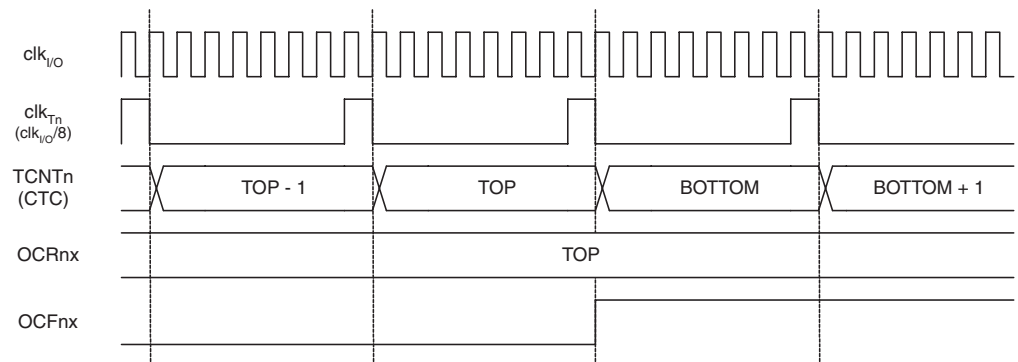


Table 12-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 12-4. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 121 for more details.

• **Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 12-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 12-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Table 12-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 12-6. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at TOP
1	1	Set OC0B on Compare Match, clear OC0B at TOP

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 94 for more details.

15.1.1 Registers

The Timer/Counter (TCNT2) and Output Compare Register (OCR2A and OCR2B) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR2). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK2). TIFR2 and TIMSK2 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or asynchronously clocked from the TOSC1/2 pins, as detailed later in this section. The asynchronous operation is controlled by the Asynchronous Status Register (ASSR). The Clock Select logic block controls which clock source the Timer/Counter uses to increment (or decrement) 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_{T2}).

The double buffered Output Compare Register (OCR2A and OCR2B) are compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins (OC2A and OC2B). See Section “15.4” on page 140. for details. The compare match event will also set the Compare Flag (OCF2A or OCF2B) which can be used to generate an Output Compare interrupt request.

15.1.2 Definitions

Many register and bit references in this document are written in general form. A lower case “n” replaces the Timer/Counter number, in this case 2. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT2 for accessing Timer/Counter2 counter value and so on.

The definitions in Table 15-1 are also used extensively throughout the section.

Table 15-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes zero (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 OCR2A Register. The assignment is dependent on the mode of operation.

15.2 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal synchronous or an external asynchronous clock source. The clock source clk_{T2} is by default equal to the MCU clock, clk_{IO} . When the AS2 bit in the ASSR Register is written to logic one, the clock source is taken from the Timer/Counter Oscillator connected to TOSC1 and TOSC2. For details on asynchronous operation, see “Asynchronous Status Register – ASSR” on page 156. For details on clock sources and prescaler, see “Timer/Counter Prescaler” on page 158.

15.3 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 15-2 shows a block diagram of the counter and its surrounding environment.

Table 15-4 shows the COM2A1:0 bit functionality when the WGM22:0 bits are set to phase correct PWM mode.

Table 15-4. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC2A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match when up-counting. Set OC2A on Compare Match when down-counting.
1	1	Set OC2A on Compare Match when up-counting. Clear OC2A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 146 for more details.

• **Bits 5:4 – COM2B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC2B) behavior. If one or both of the COM2B1:0 bits are set, the OC2B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC2B pin must be set in order to enable the output driver.

When OC2B is connected to the pin, the function of the COM2B1:0 bits depends on the WGM22:0 bit setting. Table 15-5 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to a normal or CTC mode (non-PWM).

Table 15-5. Compare Output Mode, non-PWM Mode

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Toggle OC2B on Compare Match
1	0	Clear OC2B on Compare Match
1	1	Set OC2B on Compare Match

Table 15-6 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to fast PWM mode.

Table 15-6. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match, set OC2B at TOP
1	1	Set OC2B on Compare Match, clear OC2B at TOP

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 146 for more details.

17.2.3 External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to [Figure 17-2](#) for details.

External clock input from the XCKn pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCKn clock frequency is limited by the following equation:

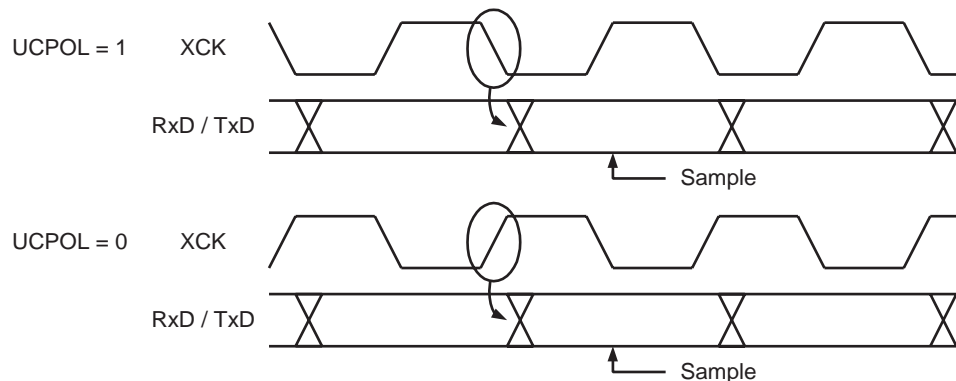
$$f_{XCK} < \frac{f_{OSC}}{4}$$

Note that f_{OSC} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

17.2.4 Synchronous Clock Operation

When synchronous mode is used ($UMSELn = 1$), the XCKn pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on RxDn) is sampled at the opposite XCKn clock edge of the edge the data output (TxDn) is changed.

Figure 17-3. Synchronous Mode XCKn Timing.



The UCPOLn bit UCRSC selects which XCKn clock edge is used for data sampling and which is used for data change. As [Figure 17-3](#) shows, when UCPOLn is zero the data will be changed at rising XCKn edge and sampled at falling XCKn edge. If UCPOLn is set, the data will be changed at falling XCKn edge and sampled at rising XCKn edge.

17.3 Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

17.6.3 Receive Complete Flag and Interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXCn) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled (RXENn = 0), the receive buffer will be flushed and consequently the RXCn bit will become zero.

When the Receive Complete Interrupt Enable (RXCIEn) in UCSRnB is set, the USART Receive Complete interrupt will be executed as long as the RXCn Flag is set (provided that global interrupts are enabled). When interrupt-driven data reception is used, the receive complete routine must read the received data from UDRn in order to clear the RXCn Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

17.6.4 Receiver Error Flags

The USART Receiver has three Error Flags: Frame Error (FEn), Data OverRun (DORn) and Parity Error (UPEn). All can be accessed by reading UCSRnA. Common for the Error Flags is that they are located in the receive buffer together with the frame for which they indicate the error status. Due to the buffering of the Error Flags, the UCSRnA must be read before the receive buffer (UDRn), since reading the UDRn I/O location changes the buffer read location. Another equality for the Error Flags is that they can not be altered by software doing a write to the flag location. However, all flags must be set to zero when the UCSRnA is written for upward compatibility of future USART implementations. None of the Error Flags can generate interrupts.

The Frame Error (FEn) Flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The FEn Flag is zero when the stop bit was correctly read (as one), and the FEn Flag will be one when the stop bit was incorrect (zero). This flag can be used for detecting out-of-sync conditions, detecting break conditions and protocol handling. The FEn Flag is not affected by the setting of the USBSn bit in UCSRnC since the Receiver ignores all, except for the first, stop bits. For compatibility with future devices, always set this bit to zero when writing to UCSRnA.

The Data OverRun (DORn) Flag indicates data loss due to a receiver buffer full condition. A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the Receive Shift Register, and a new start bit is detected. If the DORn Flag is set there was one or more serial frame lost between the frame last read from UDRn, and the next frame read from UDRn. For compatibility with future devices, always write this bit to zero when writing to UCSRnA. The DORn Flag is cleared when the frame received was successfully moved from the Shift Register to the receive buffer.

The Parity Error (UPEn) Flag indicates that the next frame in the receive buffer had a Parity Error when received. If Parity Check is not enabled the UPEn bit will always be read zero. For compatibility with future devices, always set this bit to zero when writing to UCSRnA. For more details see ["Parity Bit Calculation" on page 173](#) and ["Parity Checker" on page 181](#).

17.6.5 Parity Checker

The Parity Checker is active when the high USART Parity mode (UPMn1) bit is set. Type of Parity Check to be performed (odd or even) is selected by the UPMn0 bit. When enabled, the Parity Checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit from the serial frame. The result of the check is stored in the receive buffer together with the received data and stop bits. The Parity Error (UPEn) Flag can then be read by software to check if the frame had a Parity Error.

- **Bits 7..0 – TWD: TWI Data Register**

These eight bits constitute the next data byte to be transmitted, or the latest data byte received on the 2-wire Serial Bus.

19.6.5 TWI (Slave) Address Register – TWAR

Bit	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
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	0	

The TWAR should be loaded with the 7-bit Slave address (in the seven most significant bits of TWAR) to which the TWI will respond when programmed as a Slave Transmitter or Receiver, and not needed in the Master modes. In multi master systems, TWAR must be set in masters which can be addressed as Slaves by other Masters.

The LSB of TWAR is used to enable recognition of the general call address (0x00). There is an associated address comparator that looks for the slave address (or general call address if enabled) in the received serial address. If a match is found, an interrupt request is generated.

- **Bits 7..1 – TWA: TWI (Slave) Address Register**

These seven bits constitute the slave address of the TWI unit.

- **Bit 0 – TWGCE: TWI General Call Recognition Enable Bit**

If set, this bit enables the recognition of a General Call given over the 2-wire Serial Bus.

19.6.6 TWI (Slave) Address Mask Register – TWAMR

Bit	7	6	5	4	3	2	1	0	
	TWAM[6:0]							-	TWAMR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..1 – TWAM: TWI Address Mask**

The TWAMR can be loaded with a 7-bit Slave Address mask. Each of the bits in TWAMR can mask (disable) the corresponding address bits in the TWI Address Register (TWAR). If the mask bit is set to one then the address match logic ignores the compare between the incoming address bit and the corresponding bit in TWAR. [Figure 19-10](#) shown the address match logic in detail.

23.0.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write “00000101” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written to zero during this operation.

- The CPU is halted during the Page Write operation.

23.1 Addressing the Flash During Self-Programming

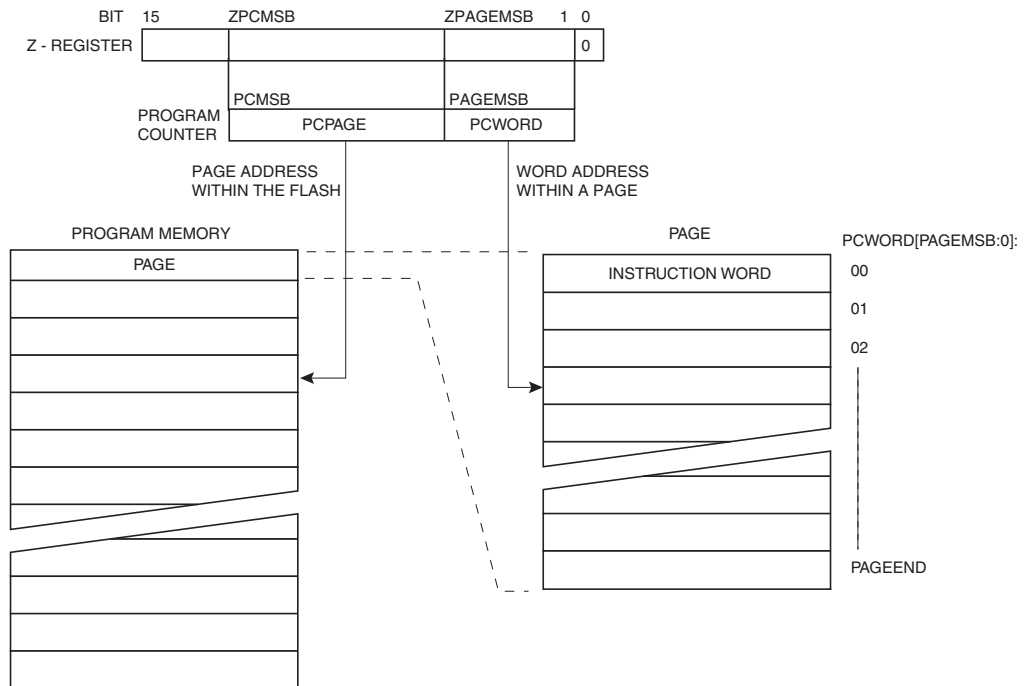
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
	7	6	5	4	3	2	1	0

Since the Flash is organized in pages (see [Table 25-8 on page 284](#)), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in [Figure 24-3](#). Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the software addresses the same page in both the Page Erase and Page Write operation.

The LPM instruction uses the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 23-1. Addressing the Flash During SPM⁽¹⁾



Note: 1. The different variables used in [Figure 24-3](#) are listed in [Table 25-8 on page 284](#).

24.3 Read-While-Write and No Read-While-Write Flash Sections

Whether the CPU supports Read-While-Write or if the CPU is halted during a Boot Loader software update is dependent on which address that is being programmed. In addition to the two sections that are configurable by the BOOTSZ Fuses as described above, the Flash is also divided into two fixed sections, the Read-While-Write (RWW) section and the No Read-While-Write (NRWW) section. The limit between the RWW- and NRWW sections is given in [Table 24-7 on page 277](#) and [Figure 24-2 on page 267](#). The main difference between the two sections is:

- When erasing or writing a page located inside the RWW section, the NRWW section can be read during the operation.
- When erasing or writing a page located inside the NRWW section, the CPU is halted during the entire operation.

Note that the user software can never read any code that is located inside the RWW section during a Boot Loader software operation. The syntax “Read-While-Write section” refers to which section that is being programmed (erased or written), not which section that actually is being read during a Boot Loader software update.

24.3.1 RWW – Read-While-Write Section

If a Boot Loader software update is programming a page inside the RWW section, it is possible to read code from the Flash, but only code that is located in the NRWW section. During an ongoing programming, the software must ensure that the RWW section never is being read. If the user software is trying to read code that is located inside the RWW section (i.e., by a call/jmp/lpm or an interrupt) during programming, the software might end up in an unknown state. To avoid this, the interrupts should either be disabled or moved to the Boot Loader section. The Boot Loader section is always located in the NRWW section. The RWW Section Busy bit (RWWSB) in the Store Program Memory Control and Status Register (SPMCSR) will be read as logical one as long as the RWW section is blocked for reading. After a programming is completed, the RWWSB must be cleared by software before reading code located in the RWW section. [See Section “24.5.1” on page 269.](#) for details on how to clear RWWSB.

24.3.2 NRWW – No Read-While-Write Section

The code located in the NRWW section can be read when the Boot Loader software is updating a page in the RWW section. When the Boot Loader code updates the NRWW section, the CPU is halted during the entire Page Erase or Page Write operation.

Table 24-1. Read-While-Write Features

Which Section does the Z-pointer Address during the Programming?	Which Section can be read during Programming?	CPU Halted?	Read-While-Write Supported?
RWW Section	NRWW Section	No	Yes
NRWW Section	None	Yes	No

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min. ⁽⁵⁾	Typ.	Max. ⁽⁵⁾	Units
R_{RST}	Reset Pull-up Resistor		30		60	$k\Omega$
R_{PU}	I/O Pin Pull-up Resistor		20		50	$k\Omega$
I_{CC}	Power Supply Current ⁽⁶⁾	Active 1MHz, $V_{CC} = 2\text{V}$ (ATmega48/88/168V)			0.55	mA
		Active 4MHz, $V_{CC} = 3\text{V}$ (ATmega48/88/168L)			3.5	mA
		Active 8MHz, $V_{CC} = 5\text{V}$ (ATmega48/88/168)			12	mA
		Idle 1MHz, $V_{CC} = 2\text{V}$ (ATmega48/88/168V)		0.25	0.5	mA
		Idle 4MHz, $V_{CC} = 3\text{V}$ (ATmega48/88/168L)			1.5	mA
		Idle 8MHz, $V_{CC} = 5\text{V}$ (ATmega48/88/168)			5.5	mA
	Power-down mode	WDT enabled, $V_{CC} = 3\text{V}$		<8	15	μA
		WDT disabled, $V_{CC} = 3\text{V}$		<1	2	μA
V_{ACIO}	Analog Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$		<10	40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$	-50		50	nA
t_{ACID}	Analog Comparator Propagation Delay	$V_{CC} = 2.7\text{V}$ $V_{CC} = 4.0\text{V}$		750 500		ns

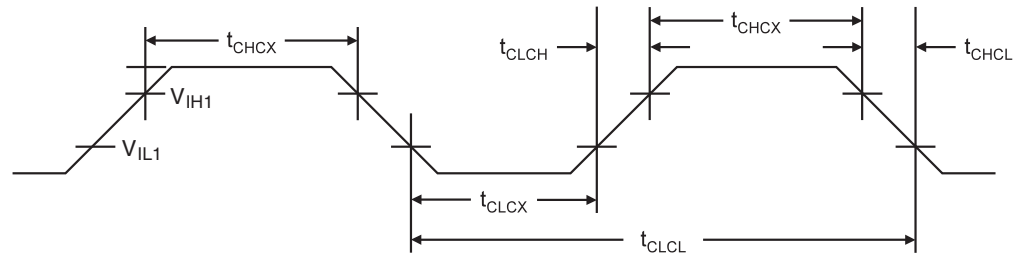
- Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low
 2. "Min" means the lowest value where the pin is guaranteed to be read as high
 3. Although each I/O port can sink more than the test conditions (20mA at $V_{CC} = 5\text{V}$, 10mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
ATmega48:
1] The sum of all IOL, for ports C0 - C5, should not exceed 100 mA.
2] The sum of all IOL, for ports C6, D0 - D4, should not exceed 100 mA.
3] The sum of all IOL, for ports B0 - B7, D5 - D7, should not exceed 100 mA.
ATmega88/168:
1] The sum of all IOL, for ports C0 - C5, should not exceed 100 mA.
2] The sum of all IOL, for ports C6, D0 - D4, should not exceed 100 mA.
3] The sum of all IOL, for ports B0 - B7, D5 - D7, should not exceed 100 mA.
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
 4. Although each I/O port can source more than the test conditions (20mA at $V_{CC} = 5\text{V}$, 10mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
ATmega48:
1] The sum of all IOH, for ports C0 - C5, should not exceed 100 mA.
2] The sum of all IOH, for ports C6, D0 - D4, should not exceed 100 mA.
3] The sum of all IOH, for ports B0 - B7, D5 - D7, should not exceed 100 mA.
ATmega88/168:
1] The sum of all IOH, for ports C0 - C5, should not exceed 100 mA.
2] The sum of all IOH, for ports C6, D0 - D4, should not exceed 100 mA.
3] The sum of all IOH, for ports B0 - B7, D5 - D7, should not exceed 100 mA.

If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

5. All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon
6. Values with “Power Reduction Register - PRR” disabled (0x00).

26.3 External Clock Drive Waveforms

Figure 26-1. External Clock Drive Waveforms



26.4 External Clock Drive

Table 26-1. External Clock Drive

Symbol	Parameter	V _{CC} =1.8-5.5V		V _{CC} =2.7-5.5V		V _{CC} =4.5-5.5V		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
1/t _{CLCL}	Oscillator Frequency	0	4	0	10	0	20	MHz
t _{CLCL}	Clock Period	250		100		50		ns
t _{CHCX}	High Time	100		40		20		ns
t _{CLCX}	Low Time	100		40		20		ns
t _{CLCH}	Rise Time		2.0		1.6		0.5	μs
t _{CHCL}	Fall Time		2.0		1.6		0.5	μs
Δt _{CLCL}	Change in period from one clock cycle to the next		2		2		2	%

Note: All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

26.5 Maximum Speed vs. V_{CC}

Maximum frequency is dependent on V_{CC}. As shown in [Figure 26-2](#) and [Figure 26-3](#), the Maximum Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V and between 2.7V < V_{CC} < 4.5V.

Figure 26-2. Maximum Frequency vs. V_{CC} , ATmega48V/88V/168V

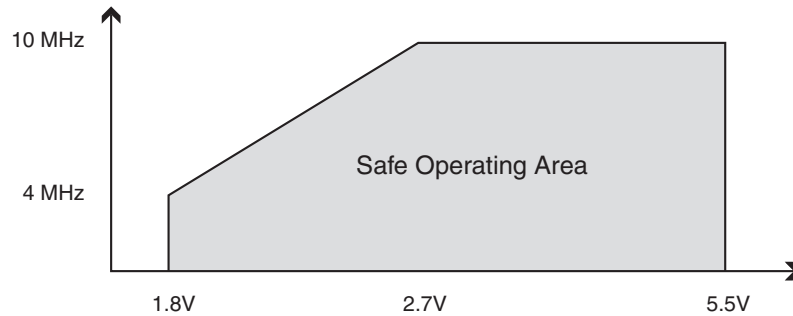
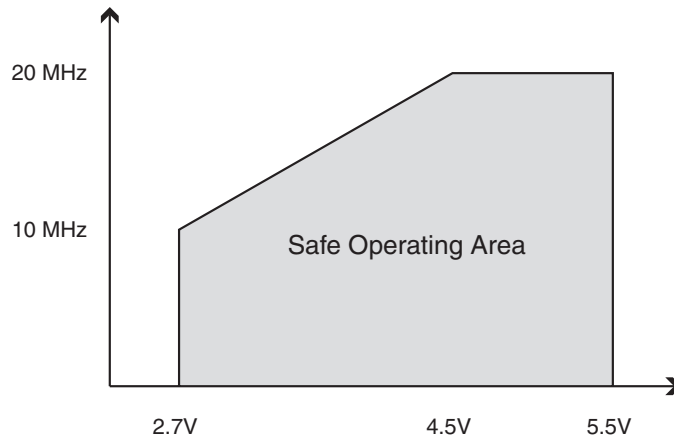


Figure 26-3. Maximum Frequency vs. V_{CC} , ATmega48/88/168



26.6 2-wire Serial Interface Characteristics

Table 26-2 describes the requirements for devices connected to the 2-wire Serial Bus. The ATmega48/88/168 2-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 26-4.

Table 26-2. 2-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	$0.3 V_{CC}$	V
V_{IH}	Input High-voltage		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{hys}^{(1)}$	Hysteresis of Schmitt Trigger Inputs		$0.05 V_{CC}^{(2)}$	–	V
$V_{OL}^{(1)}$	Output Low-voltage	3 mA sink current	0	0.4	V
$t_r^{(1)}$	Rise Time for both SDA and SCL		$20 + 0.1C_b^{(3)(2)}$	300	ns
$t_{of}^{(1)}$	Output Fall Time from V_{IHmin} to V_{ILmax}	$10 \text{ pF} < C_b < 400 \text{ pF}^{(3)}$	$20 + 0.1C_b^{(3)(2)}$	250	ns
$t_{SP}^{(1)}$	Spikes Suppressed by Input Filter		0	$50^{(2)}$	ns
I_i	Input Current each I/O Pin	$0.1V_{CC} < V_i < 0.9V_{CC}$	-10	10	μA

Figure 27-2. Active Supply Current vs. Frequency (1 - 24 MHz)

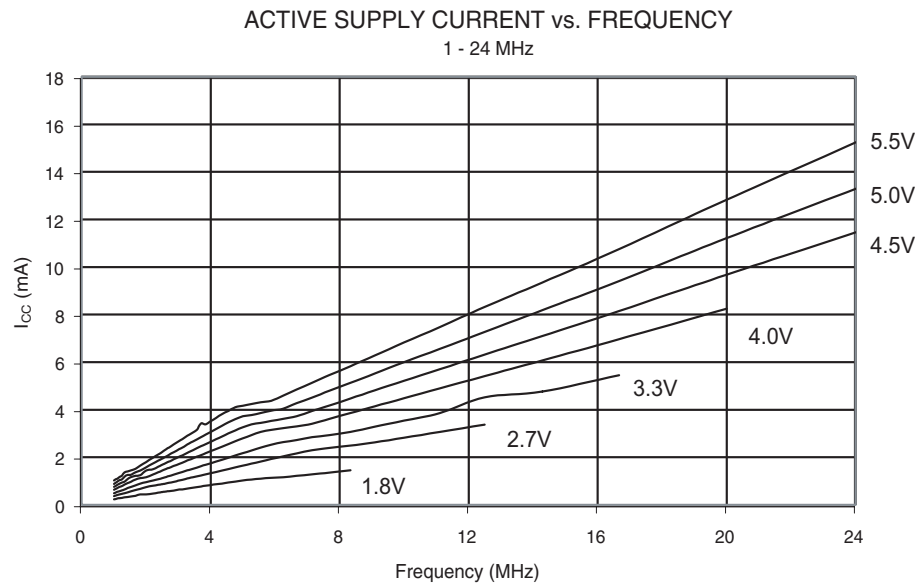


Figure 27-3. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

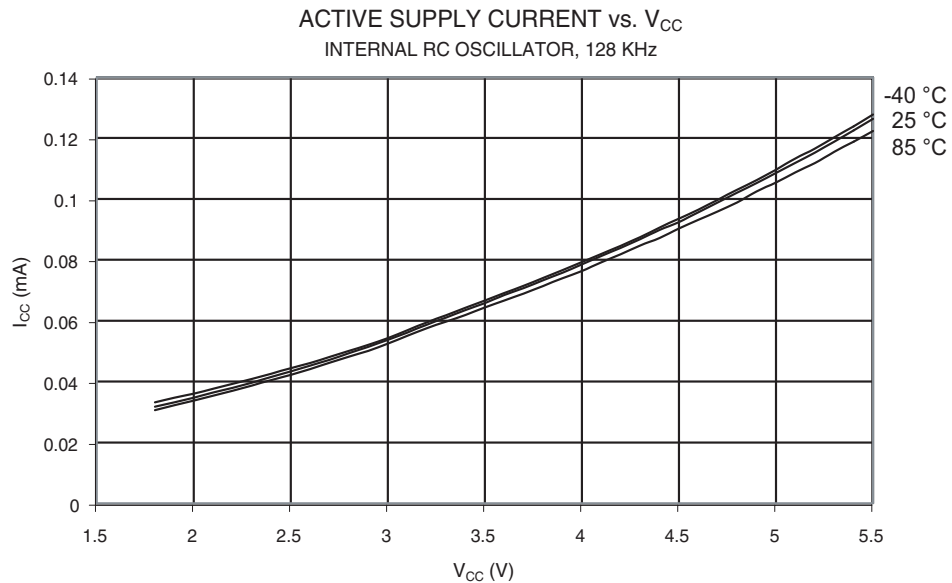


Figure 27-4. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

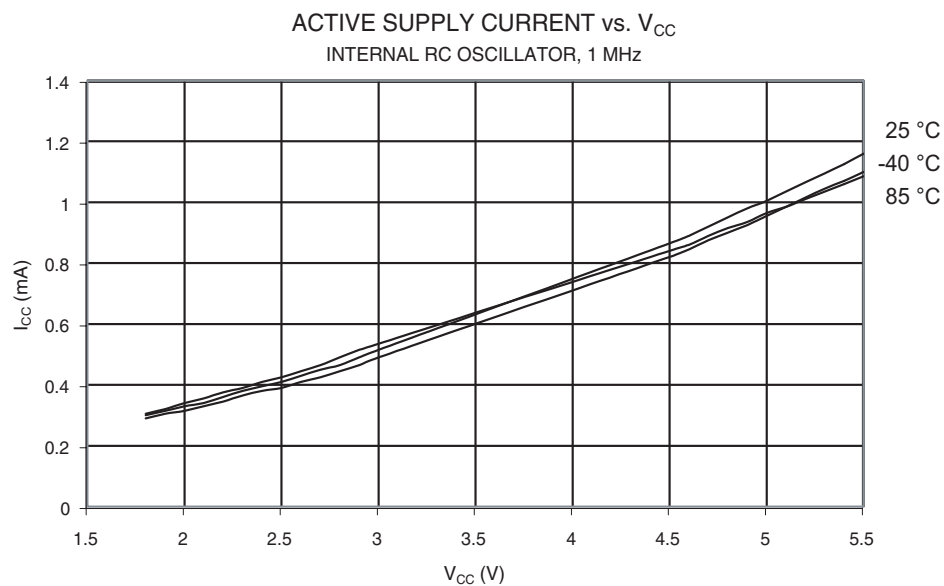


Figure 27-5. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

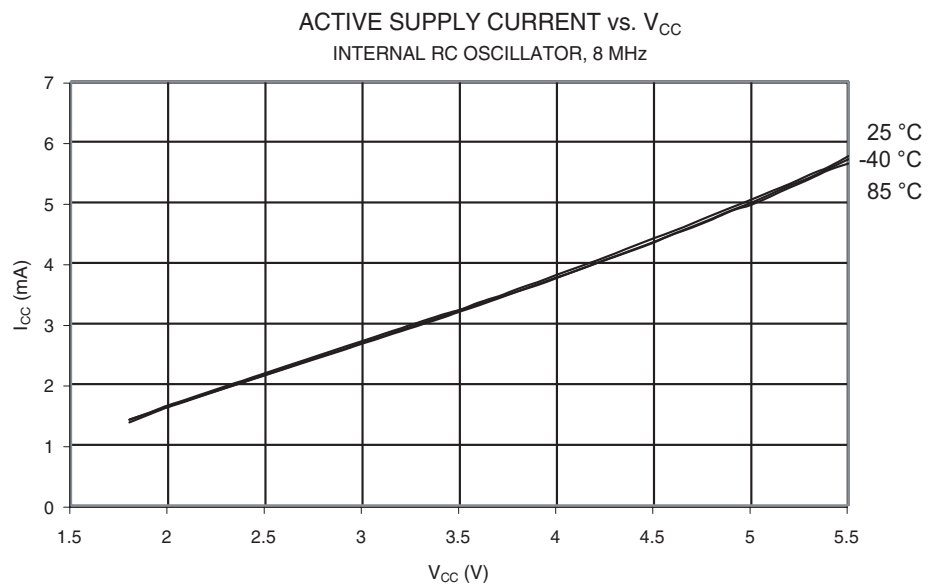
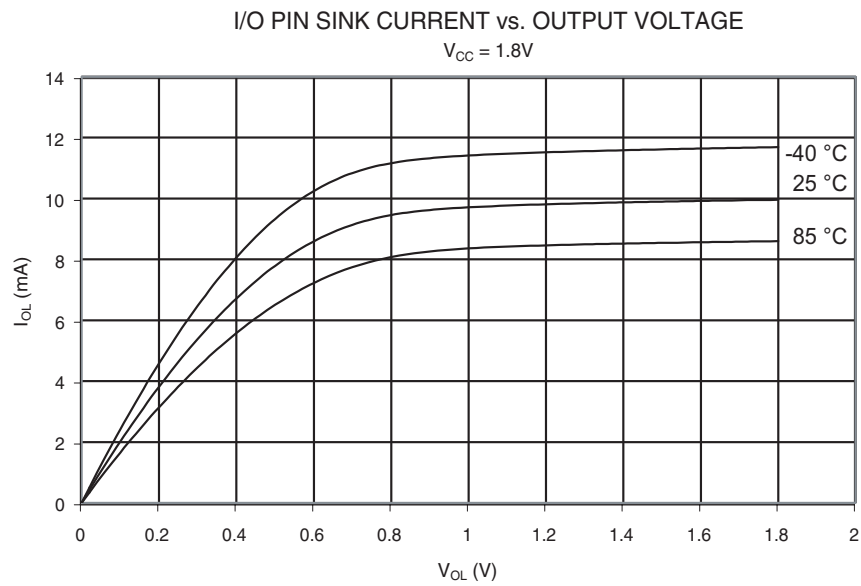
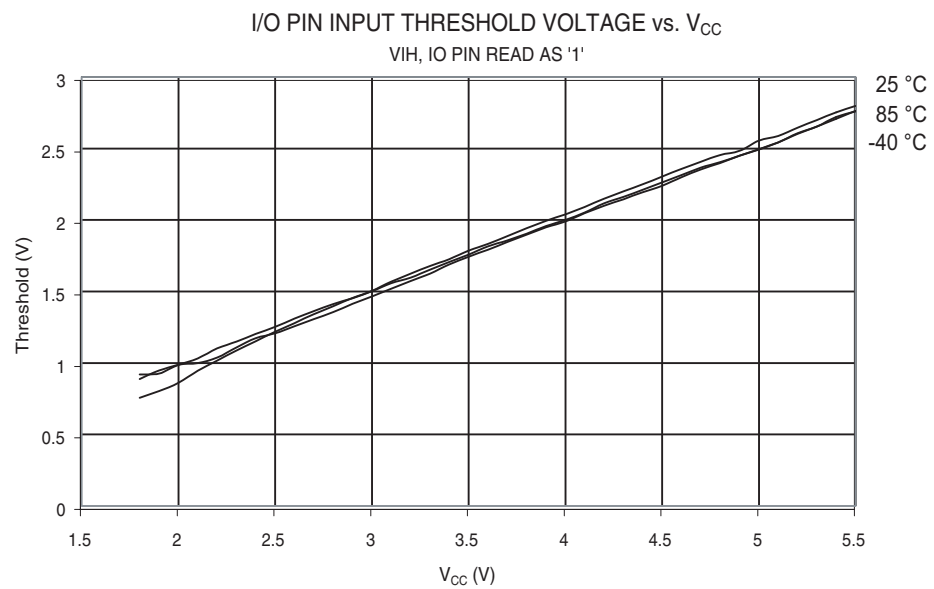


Figure 27-27. I/O Pin Sink Current vs. Output Voltage ($V_{CC} = 1.8V$)



27.9 Pin Thresholds and Hysteresis

Figure 27-28. I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin Read As '1')



27.10 BOD Thresholds and Analog Comparator Offset

Figure 27-33. BOD Thresholds vs. Temperature (BODLEVEL Is 4.0V)

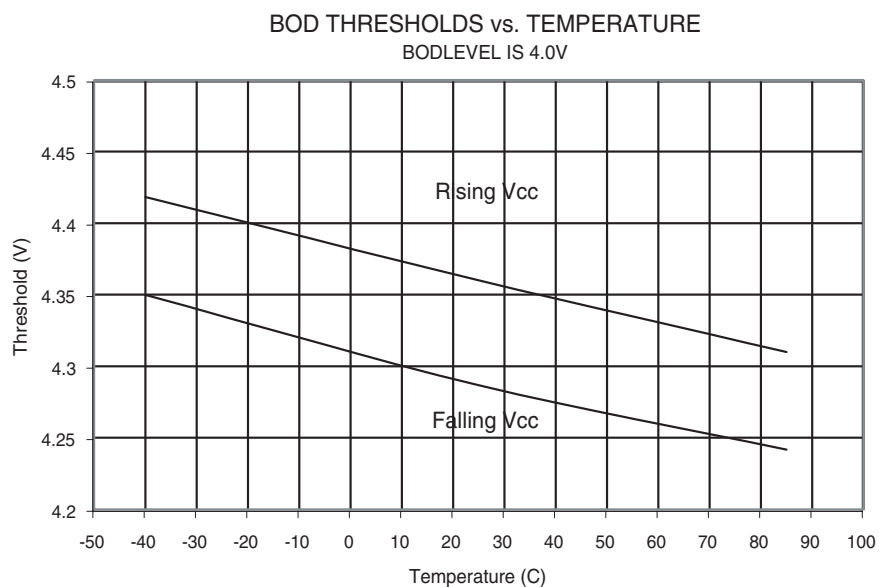
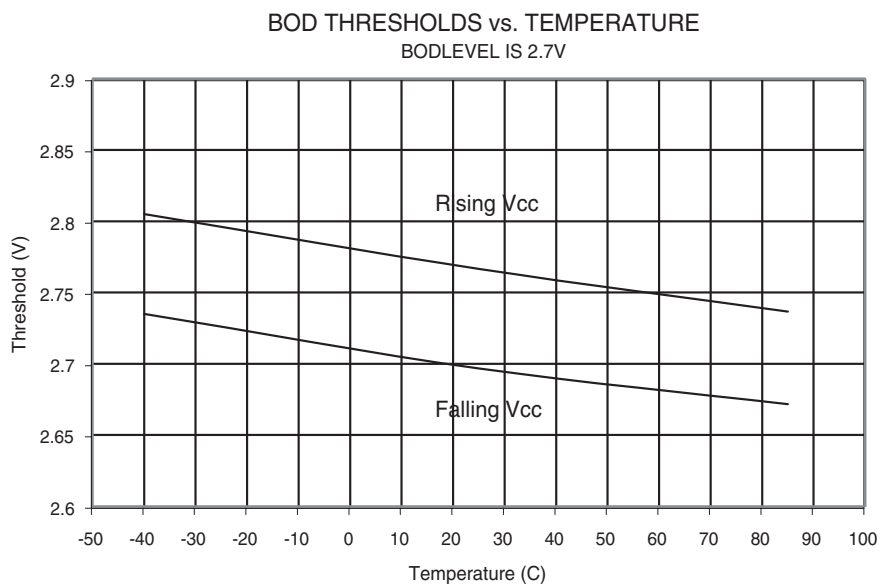


Figure 27-34. BOD Thresholds vs. Temperature (BODLEVEL Is 2.7V)



The Asynchronous oscillator does not stop when entering power down mode. This leads to higher power consumption than expected.

Problem fix / Workaround

Manually disable the asynchronous timer before entering power down.

32.2 Errata ATmega88

The revision letter in this section refers to the revision of the ATmega88 device.

32.2.1 Rev. A

- **Writing to EEPROM does not work at low Operating Voltages**
- **Part may hang in reset**

1. Writing to EEPROM does not work at low operating voltages

Writing to the EEPROM does not work at low voltages.

Problem Fix/Workaround

Do not write the EEPROM at voltages below 4.5 Volts.

This will be corrected in rev. B.

2. Part may hang in reset

Some parts may get stuck in a reset state when a reset signal is applied when the internal reset state-machine is in a specific state. The internal reset state-machine is in this state for approximately 10 ns immediately before the part wakes up after a reset, and in a 10 ns window when altering the system clock prescaler. The problem is most often seen during In-System Programming of the device. There are theoretical possibilities of this happening also in run-mode. The following three cases can trigger the device to get stuck in a reset-state:

- Two succeeding resets are applied where the second reset occurs in the 10ns window before the device is out of the reset-state caused by the first reset.
- A reset is applied in a 10 ns window while the system clock prescaler value is updated by software.
- Leaving SPI-programming mode generates an internal reset signal that can trigger this case.

The two first cases can occur during normal operating mode, while the last case occurs only during programming of the device.

Problem Fix/Workaround

The first case can be avoided during run-mode by ensuring that only one reset source is active. If an external reset push button is used, the reset start-up time should be selected such that the reset line is fully debounced during the start-up time.

The second case can be avoided by not using the system clock prescaler.

The third case occurs during In-System programming only. It is most frequently seen when using the internal RC at maximum frequency.

If the device gets stuck in the reset-state, turn power off, then on again to get the device out of this state.

32.2.2 Rev. D

No errata.