



Welcome to [E-XFL.COM](http://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	16MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/atmel/atmega48-15at">https://www.e-xfl.com/product-detail/atmel/atmega48-15at</a>

## 4.4 Status Register

The status register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the status register is updated after all ALU operations, as specified in the instruction set reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR<sup>®</sup> status register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The bit copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. half carry is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit,  $S = N \oplus V$**

The S-bit is always an exclusive or between the negative flag N and the two’s complement overflow flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The two’s complement overflow flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

## 6.1.4 Asynchronous Timer Clock – $clk_{ASY}$

The asynchronous timer clock allows the asynchronous Timer/Counter to be clocked directly from an external clock or an external 32kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

## 6.1.5 ADC Clock – $clk_{ADC}$

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

## 6.2 Clock Sources

The device has the following clock source options, selectable by flash fuse bits as shown below. The clock from the selected source is input to the AVR<sup>®</sup> clock generator, and routed to the appropriate modules.

**Table 6-1. Device Clocking Options Select<sup>(1)</sup>**

Device Clocking Option	CKSEL3..0
Low power crystal oscillator	1111 - 1000
Full swing crystal oscillator	0111 - 0110
Low frequency crystal oscillator	0101 - 0100
Internal 128kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External clock	0000
Reserved	0001

Note: 1. For all fuses “1” means unprogrammed while “0” means programmed.

### 6.2.1 Default Clock Source

The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock. The startup time is set to maximum and time-out period enabled. (CKSEL = “0010”, SUT = “10”, CKDIV8 = “0”). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

### 6.2.2 Clock Startup Sequence

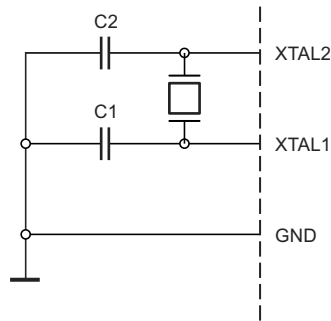
Any clock source needs a sufficient  $V_{CC}$  to start oscillating and a minimum number of oscillating cycles before it can be considered stable.

To ensure sufficient  $V_{CC}$ , the device issues an internal reset with a time-out delay ( $t_{TOUR}$ ) after the device reset is released by all other reset sources. Section 8. “System Control and Reset” on page 38 describes the start conditions for the internal reset. The delay ( $t_{TOUR}$ ) is timed from the watchdog oscillator and the number of cycles in the delay is set by the SUTx and CKSELx fuse bits. The selectable delays are shown in Table 6-2. The frequency of the watchdog oscillator is voltage dependent as shown in Section “” on page 285.

**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
0ms	0ms	0
4.1ms	4.3ms	4K (4,096)
65ms	69ms	8K (8,192)

**Figure 6-3. Crystal Oscillator Connections**



**Table 6-6. Start-up Times for the Full Swing Crystal Oscillator Clock Selection**

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ( $V_{CC} = 5.0V$ )	CKSEL0	SUT1..0
Ceramic resonator, fast rising power	258CK	$14CK + 4.1ms^{(1)}$	0	00
Ceramic resonator, slowly rising power	258CK	$14CK + 65ms^{(1)}$	0	01
Ceramic resonator, BOD enabled	1KCK	$14CK^{(2)}$	0	10
Ceramic resonator, fast rising power	1KCK	$14CK + 4.1ms^{(2)}$	0	11
Ceramic resonator, slowly rising power	1KCK	$14CK + 65ms^{(2)}$	1	00
Crystal Oscillator, BOD enabled	16KCK	14CK	1	01
Crystal Oscillator, fast rising power	16KCK	$14CK + 4.1ms$	1	10
Crystal Oscillator, slowly rising power	16KCK	$14CK + 65ms$	1	11

- Notes:
1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
  2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

**Table 9-2. Reset and Interrupt Vectors in ATmega88 (Continued)**

Vector No.	Program Address <sup>(2)</sup>	Source	Interrupt Definition
8	0x007	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x008	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x009	TIMER2 OVF	Timer/Counter2 overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 capture event
12	0x00B	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x00C	TIMER1 COMPB	Timer/coutner1 compare match B
14	0x00D	TIMER1 OVF	Timer/Counter1 overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x010	TIMER0 OVF	Timer/Counter0 overflow
18	0x011	SPI, STC	SPI serial transfer complete
19	0x012	USART, RX	USART Rx complete
20	0x013	USART, UDRE	USART, data register empty
21	0x014	USART, TX	USART, Tx complete
22	0x015	ADC	ADC conversion complete
23	0x016	EE READY	EEPROM ready
24	0x017	ANALOG COMP	Analog comparator
25	0x018	TWI	2-wire serial interface
26	0x019	SPM READY	Store program memory ready

- Notes:
1. When the BOOTRST fuse is programmed, the device will jump to the boot loader address at reset, see Section 24. “Boot Loader Support – Read-While-Write Self-Programming, ATmega88 and ATmega168” on page 229.
  2. When the IVSEL bit in MCUCR is set, interrupt vectors will be moved to the start of the boot flash section. The address of each interrupt vector will then be the address in this table added to the start address of the boot flash section.

Table 9-3 shows reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section while the interrupt vectors are in the boot section or vice versa.

**Table 9-3. Reset and Interrupt Vectors Placement in ATmega88<sup>(1)</sup>**

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot reset address + 0x001
0	0	Boot reset address	0x001
0	1	Boot reset address	Boot reset address + 0x001

- Note:
1. The boot reset address is shown in Table 24-6 on page 240. For the BOOTRST fuse “1” means unprogrammed while “0” means programmed.

## 10.2.2 Toggling the Pin

Writing a logic one to PIN<sub>xn</sub> toggles the value of PORT<sub>xn</sub>, independent on the value of DDR<sub>xn</sub>. Note that the SBI instruction can be used to toggle one single bit in a port.

## 10.2.3 Switching Between Input and Output

When switching between tri-state ({DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b00) and output high ({DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b11), an intermediate state with either pull-up enabled {DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b01) or output low ({DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b00) or the output high state ({DDR<sub>xn</sub>, PORT<sub>xn</sub>} = 0b11) as an intermediate step.

Table 10-1 summarizes the control signals for the pin value.

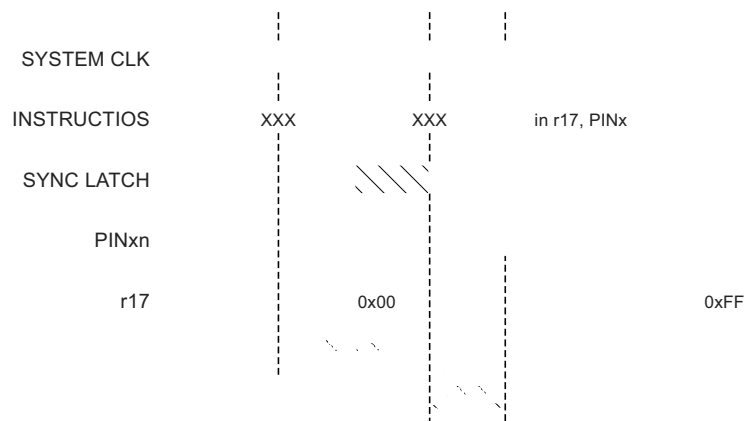
**Table 10-1. Port Pin Configurations**

DD <sub>xn</sub>	PORT <sub>xn</sub>	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (hi-Z)
0	1	0	Input	Yes	P <sub>xn</sub> will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (hi-Z)
1	0	X	Output	No	Output low (sink)
1	1	X	Output	No	Output high (source)

## 10.2.4 Reading the Pin Value

Independent of the setting of data direction bit DD<sub>xn</sub>, the port pin can be read through the PIN<sub>xn</sub> register bit. As shown in Figure 10-2 on page 58, the PIN<sub>xn</sub> register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 10-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted  $t_{pd,max}$  and  $t_{pd,min}$  respectively.

**Figure 10-3. Synchronization when Reading an Externally Applied Pin value**



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PIN<sub>xn</sub> register at the succeeding positive clock edge. As indicated by the two arrows  $t_{pd,max}$  and  $t_{pd,min}$ , a single signal transition on the pin will be delayed between  $\frac{1}{2}$  and  $1\frac{1}{2}$  system clock period depending upon the time of assertion.

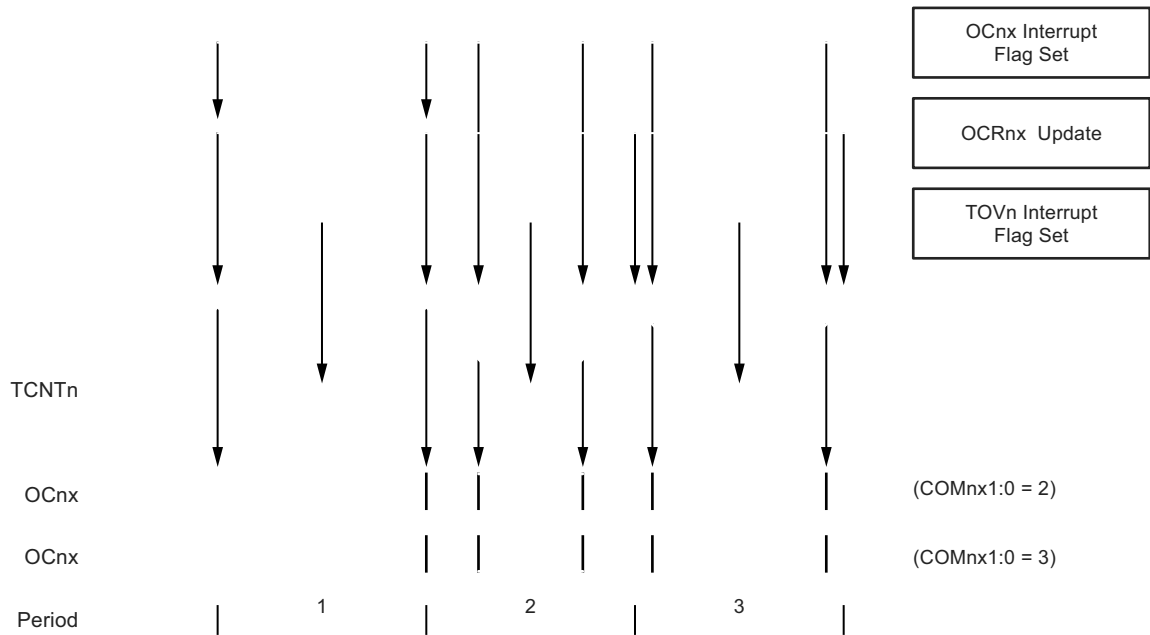
A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0x to toggle its logical level on each compare match (COM0x1:0 = 1). The waveform generated will have a maximum frequency of  $f_{OC0} = f_{clk\_I/O}/2$  when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

#### 12.6.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM02:0 = 1 or 5) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1, and OCR0A when WGM2:0 = 5. In non-inverting compare output mode, the output compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x while upcounting, and set on the compare match while downcounting. In inverting output compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 12-7. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

**Figure 12-7. Phase Correct PWM Mode, Timing Diagram**



The Timer/Counter overflow flag (TOV0) is set each time the counter reaches BOTTOM. The interrupt flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

The OCR2x register is double buffered when using any of the pulse width modulation (PWM) modes. For the normal and clear timer on compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR2x compare register to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR2x register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR2x buffer register, and if double buffering is disabled the CPU will access the OCR2x directly.

#### 15.4.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the force output compare (FOC2x) bit. Forcing compare match will not set the OCF2x flag or reload/clear the timer, but the OC2x pin will be updated as if a real compare match had occurred (the COM2x1:0 bits settings define whether the OC2x pin is set, cleared or toggled).

#### 15.4.2 Compare Match Blocking by TCNT2 Write

All CPU write operations to the TCNT2 register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR2x to be initialized to the same value as TCNT2 without triggering an interrupt when the Timer/Counter clock is enabled.

#### 15.4.3 Using the Output Compare Unit

Since writing TCNT2 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT2 when using the output compare channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT2 equals the OCR2x value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT2 value equal to BOTTOM when the counter is downcounting.

The setup of the OC2x should be performed before setting the data direction register for the port pin to output. The easiest way of setting the OC2x value is to use the force output compare (FOC2x) strobe bit in normal mode. The OC2x register keeps its value even when changing between waveform generation modes.

Be aware that the COM2x1:0 bits are not double buffered together with the compare value. Changing the COM2x1:0 bits will take effect immediately.

### 15.5 Compare Match Output Unit

The compare output mode (COM2x1:0) bits have two functions. The waveform generator uses the COM2x1:0 bits for defining the output compare (OC2x) state at the next compare match. Also, the COM2x1:0 bits control the OC2x pin output source. Figure 15-4 on page 123 shows a simplified schematic of the logic affected by the COM2x1:0 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 COM2x1:0 bits are shown. When referring to the OC2x state, the reference is for the internal OC2x register, not the OC2x pin.



- **Bit 0 – TCR2BUB: Timer/Counter Control Register2 Update Busy**

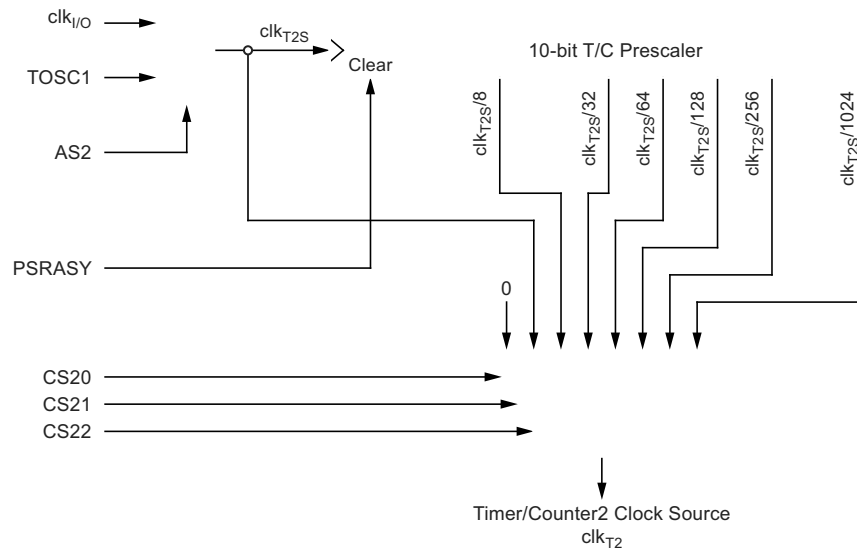
When Timer/Counter2 operates asynchronously and TCCR2B is written, this bit becomes set. When TCCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2B is ready to be updated with a new value.

If a write is performed to any of the five Timer/Counter2 registers while its update busy flag is set, the updated value might get corrupted and cause an unintentional interrupt to occur.

The mechanisms for reading TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B are different. When reading TCNT2, the actual timer value is read. When reading OCR2A, OCR2B, TCCR2A and TCCR2B the value in the temporary storage register is read.

## 15.10 Timer/Counter Prescaler

**Figure 15-12. Prescaler for Timer/Counter2**



The clock source for Timer/Counter2 is named  $clk_{T2S}$ .  $clk_{T2S}$  is by default connected to the main system I/O clock  $clk_{I/O}$ . By setting the AS2 bit in ASSR, Timer/Counter2 is asynchronously clocked from the TOSC1 pin. This enables use of Timer/Counter2 as a real time counter (RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from port C. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for Timer/Counter2. The oscillator is optimized for use with a 32.768kHz crystal. Applying an external clock source to TOSC1 is not recommended.

For Timer/Counter2, the possible prescaled selections are:  $clk_{T2S}/8$ ,  $clk_{T2S}/32$ ,  $clk_{T2S}/64$ ,  $clk_{T2S}/128$ ,  $clk_{T2S}/256$ , and  $clk_{T2S}/1024$ . Additionally,  $clk_{T2S}$  as well as 0 (stop) may be selected. Setting the PSRASY bit in GTCCR resets the prescaler. This allows the user to operate with a predictable prescaler.

## 17.9 USART Register Description

### 17.9.1 USART I/O Data Register n– UDRn

Bit	7	6	5	4	3	2	1	0	
	<b>RXB[7:0]</b>								UDRn (Read)
	<b>TXB[7:0]</b>								UDRn (Write)
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 USART transmit data buffer register and USART receive data buffer registers share the same I/O address referred to as USART data register or UDRn. The transmit data buffer register (TXB) will be the destination for data written to the UDRn register location. Reading the UDRn register location will return the contents of the receive data buffer register (RXB).

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the transmitter and set to zero by the receiver.

The transmit buffer can only be written when the UDREN flag in the UCSRnA register is set. Data written to UDRn when the UDREN flag is not set, will be ignored by the USART transmitter. When data is written to the transmit buffer, and the transmitter is enabled, the transmitter will load the data into the transmit shift register when the shift register is empty. Then the data will be serially transmitted on the TxDn pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use read-modify-write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

### 17.9.2 USART Control and Status Register n A – UCSRnA

Bit	7	6	5	4	3	2	1	0	
	<b>RXCn</b>	<b>TXCn</b>	<b>UDREN</b>	<b>FEn</b>	<b>DORn</b>	<b>UPEn</b>	<b>U2Xn</b>	<b>MPCMn</b>	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn flag can be used to generate a receive complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn flag can generate a transmit complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREN: USART Data Register Empty**

The UDREN flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREN is one, the buffer is empty, and therefore ready to be written. The UDREN flag can generate a data register empty interrupt (see description of the UDRIEn bit).

UDREN is set after a reset to indicate that the transmitter is ready.

- **Bit 4 – FEn: Frame Error**

This bit is set if the next character in the receive buffer had a frame error when received. I.e., when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDRn) is read. The FEn bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRnA.

After a repeated START condition (state 0x10) the 2-wire serial interface can access the same slave again, or a new slave without transmitting a STOP condition. Repeated START enables the master to switch between slaves, master transmitter mode and master receiver mode without losing control over the bus.

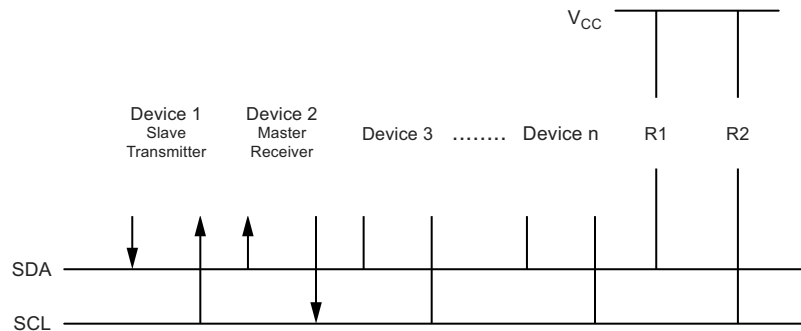
**Table 19-5. Status Codes for Master Receiver Mode**

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x08	A START condition has been transmitted	Load SLA+R	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received
0x10	A repeated START condition has been transmitted	Load SLA+R	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received SLA+W will be transmitted Logic will switch to master transmitter mode
		Load SLA+W	0	0	1	X	
0x38	Arbitration lost in SLA+R or NOT ACK bit	No TWDR action	0	0	1	X	2-wire serial bus will be released and not addressed Slave mode will be entered A START condition will be transmitted when the bus becomes free
		No TWDR action	0	0	1	X	
0x40	SLA+R has been transmitted; ACK has been received	No TWDR action	0	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	0	0	1	1	
0x48	SLA+R has been transmitted; NOT ACK has been received	No TWDR action or	1	0	1	X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO flag will be reset
		No TWDR action or	0	1	1	X	
		No TWDR action	1	1	1	X	
0x50	Data byte has been received; ACK has been returned	Read data byte or	0	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		Read data byte	0	0	1	1	
0x58	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	1	X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO flag will be reset
		Read data byte	0	1	1	X	
		Read data byte	1	1	1	X	

## 19.8.4 Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 19-18). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

**Figure 19-18. Data Transfer in Slave Transmitter Mode**



To initiate the slave transmitter mode, TWAR and TWCR must be initialized as follows:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
value	Device's Own Slave Address							

The upper seven bits are the address to which the 2-wire serial interface will respond when addressed by a master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "1" (read), the TWI will operate in ST mode, otherwise SR mode is entered. After its own slave address and the write bit have been received, the TWINT flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in Table 19-7 on page 199. The slave transmitter mode may also be entered if arbitration is lost while the TWI is in the master mode (see state 0xB0).

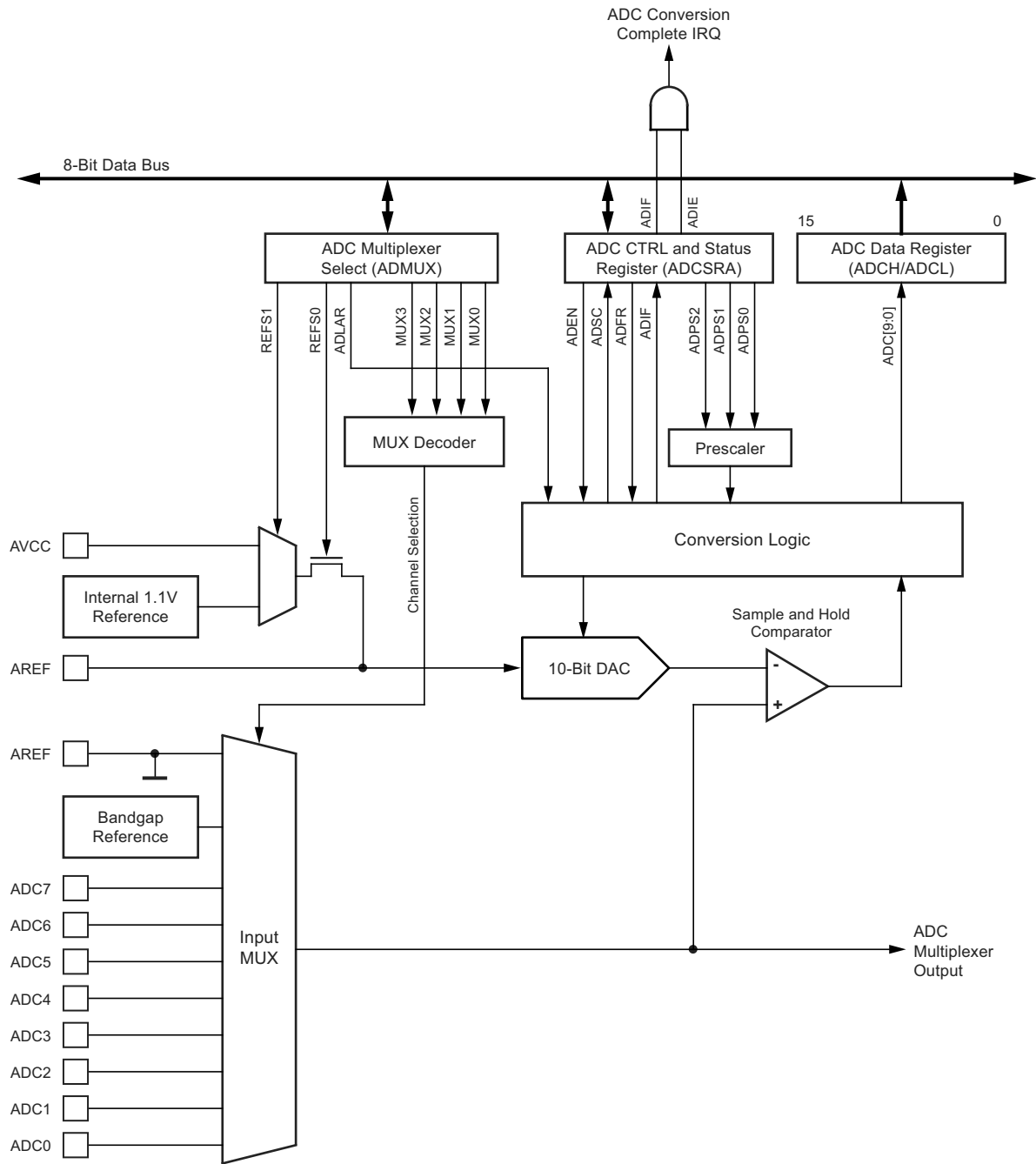
If the TWEA bit is written to zero during a transfer, the TWI will transmit the last byte of the transfer. State 0xC0 or state 0xC8 will be entered, depending on whether the master receiver transmits a NACK or ACK after the final byte. The TWI is switched to the not addressed slave mode, and will ignore the master if it continues the transfer. Thus the master receiver receives all "1" as serial data. State 0xC8 is entered if the master demands additional data bytes (by transmitting ACK), even though the Slave has transmitted the last byte (TWEA zero and expecting NACK from the master).

While TWEA is zero, the TWI does not respond to its own slave address. However, the 2-wire serial bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the 2-wire serial bus.

In all sleep modes other than idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the 2-wire serial bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock will low during the wake up and until the TWINT flag is cleared (by writing it to one). Further data transmission will be carried out as normal, with the AVR® clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the 2-wire serial interface data register – TWDR does not reflect the last byte present on the bus when waking up from these sleep modes.

**Figure 21-1. Analog to Digital Converter Block Schematic Operation**



The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally,  $AV_{CC}$  or an internal 1.1V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel is selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. The ADC is enabled by setting the ADC enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

## 21.6 ADC Conversion Result

After the conversion is complete (ADIF is high), the conversion result can be found in the ADC result registers (ADCL, ADCH).

For single ended conversion, the result is:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

where  $V_{IN}$  is the voltage on the selected input pin and  $V_{REF}$  the selected voltage reference (see Table 21-2 and Table 21-3 on page 218). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

### 21.6.1 ADC Multiplexer Selection Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in Table 21-2. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

**Table 21-2. Voltage Reference Selections for ADC**

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal $V_{ref}$ turned off
0	1	$AV_{CC}$ with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

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

The ADLAR bit affects the presentation of the ADC conversion result in the ADC data register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC data register immediately, regardless of any ongoing conversions. For a complete description of this bit, see Section 21.6.3 “The ADC Data Register – ADCL and ADCH” on page 219.

- **Bit 4 – Res: Reserved Bit**

This bit is an unused bit in the Atmel® ATmega48/88/168, and will always read as zero.

- **Bits 3:0 – MUX3:0: Analog Channel Selection Bits**

The value of these bits selects which analog inputs are connected to the ADC. See Table 21-3 on page 218 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

## 26.2 DC Characteristics (Continued)

$T_A = -40^\circ\text{C}$  to  $+125^\circ\text{C}$ ,  $V_{CC} = 2.7\text{V}$  to  $5.5\text{V}$  (unless otherwise noted)

Parameter	Condition	Symbol	Min. <sup>(5)</sup>	Typ.	Max. <sup>(5)</sup>	Unit
Input high voltage, except XTAL1 and RESET pins	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	$V_{IH}$	$0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Input high voltage, XTAL1 pin	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	$V_{IH1}$	$0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Input high voltage, RESET pin	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	$V_{IH2}$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Output low voltage <sup>(3)</sup>	$I_{OL} = 20\text{mA}$ , $V_{CC} = 5\text{V}$ $I_{OL} = 5\text{mA}$ , $V_{CC} = 3\text{V}$	$V_{OL}$			0.8 0.5	V
Output high voltage <sup>(4)</sup>	$I_{OH} = -20\text{mA}$ , $V_{CC} = 5\text{V}$ $I_{OH} = -10\text{mA}$ , $V_{CC} = 3\text{V}$	$V_{OH}$	4.1 2.3			V
Input leakage current I/O pin	$V_{CC} = 5.5\text{V}$ , pin low (absolute value)	$I_{IL}$			50	nA
Input leakage current I/O pin	$V_{CC} = 5.5\text{V}$ , pin high (absolute value)	$I_{IH}$			50	nA
Reset pull-up resistor	$V_{CC} = 5.0\text{V}$ , $V_{in} = 0\text{V}$	$R_{RST}$	30		60	k $\Omega$
I/O pin pull-up resistor		$R_{PU}$	20		50	k $\Omega$

- Notes:
- “Max” means the highest value where the pin is guaranteed to be read as low
  - “Min” means the lowest value where the pin is guaranteed to be read as high
  - 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:  
Atmel ATmega48:  
1] The sum of all IOL, for ports C0 - C5, should not exceed 70mA.  
2] The sum of all IOL, for ports C6, D0 - D4, should not exceed 70mA.  
3] The sum of all IOL, for ports B0 - B7, D5 - D7, should not exceed 70mA.  
ATmega88/168:  
1] The sum of all IOL, for ports C0 - C5, should not exceed 100mA.  
2] The sum of all IOL, for ports C6, D0 - D4, should not exceed 100mA.  
3] The sum of all IOL, for ports B0 - B7, D5 - D7, should not exceed 100mA.  
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.
  - 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 70mA.  
2] The sum of all IOH, for ports C6, D0 - D4, should not exceed 70mA.  
3] The sum of all IOH, for ports B0 - B7, D5 - D7, should not exceed 70mA.  
ATmega88/168:  
1] The sum of all IOH, for ports C0 - C5, should not exceed 100mA.  
2] The sum of all IOH, for ports C6, D0 - D4, should not exceed 100mA.  
3] The sum of all IOH, for ports B0 - B7, D5 - D7, should not exceed 100mA.  
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.
  - All DC characteristics contained in this datasheet are based on actual ATmega88 microcontrollers characterization.
  - Values with Section 7.7.1 “Power Reduction Register - PRR” on page 35 enabled (0xEF).

Figure 28-7. Output Low Voltage versus Output Low Current ( $V_{CC} = 3V$ )

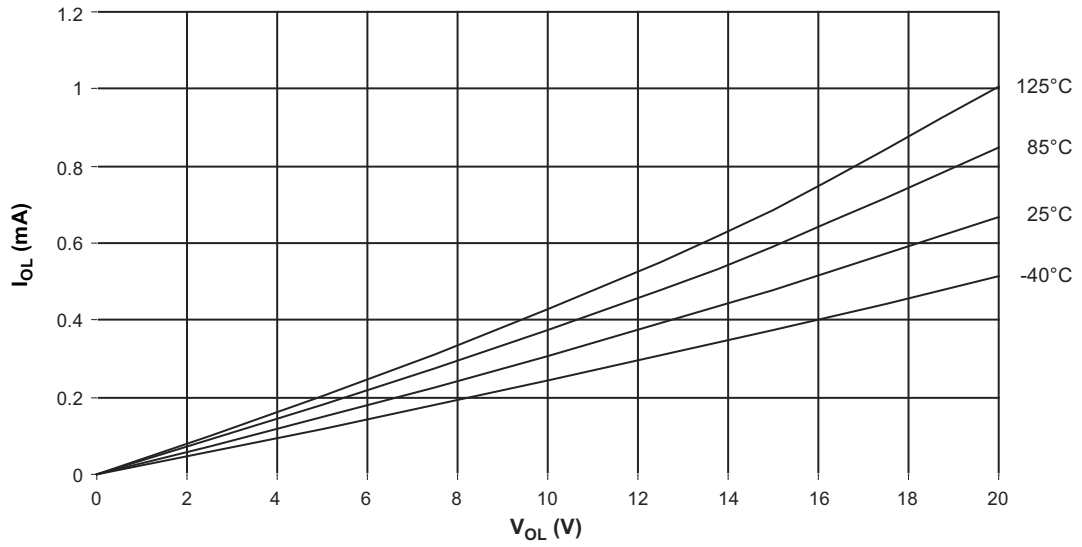
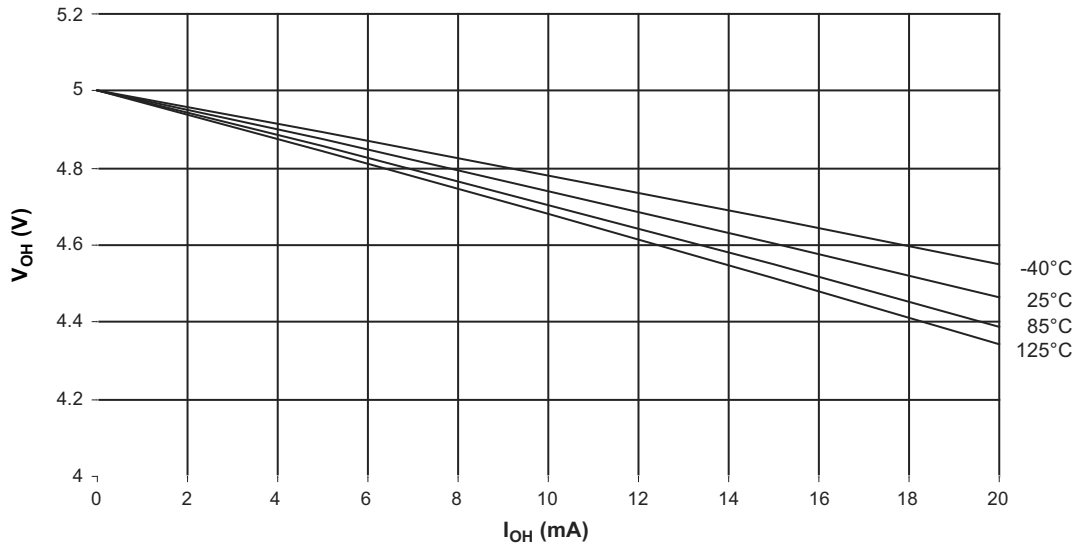


Figure 28-8. Output High Voltage versus Output High Current ( $V_{CC} = 5V$ )





## 28.1.5 Internal Oscillator Speed

Figure 28-19. Watchdog Oscillator Frequency versus  $V_{CC}$

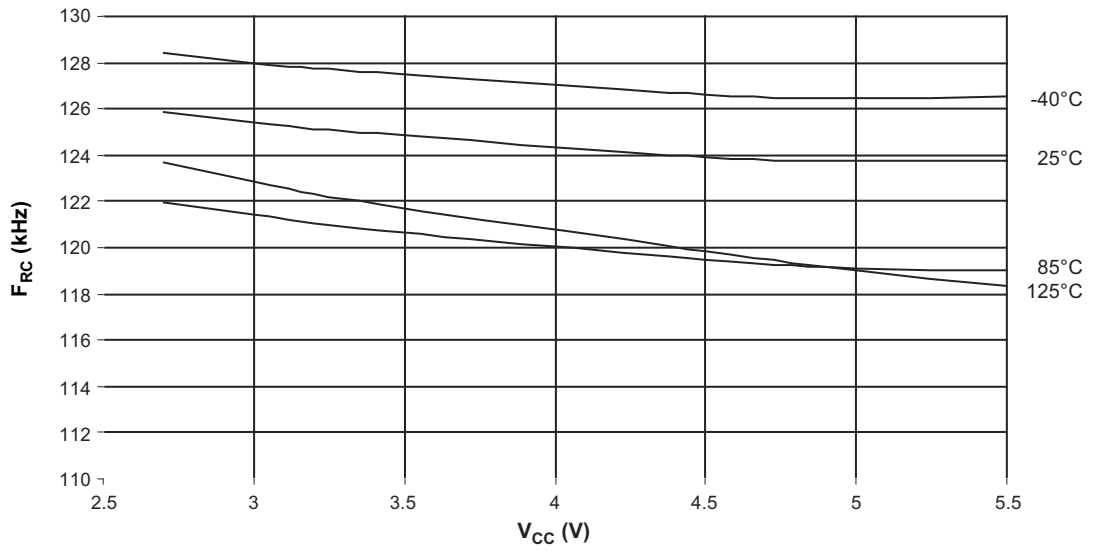


Figure 28-20. Calibrated 8MHz RC Oscillator Frequency versus Temperature

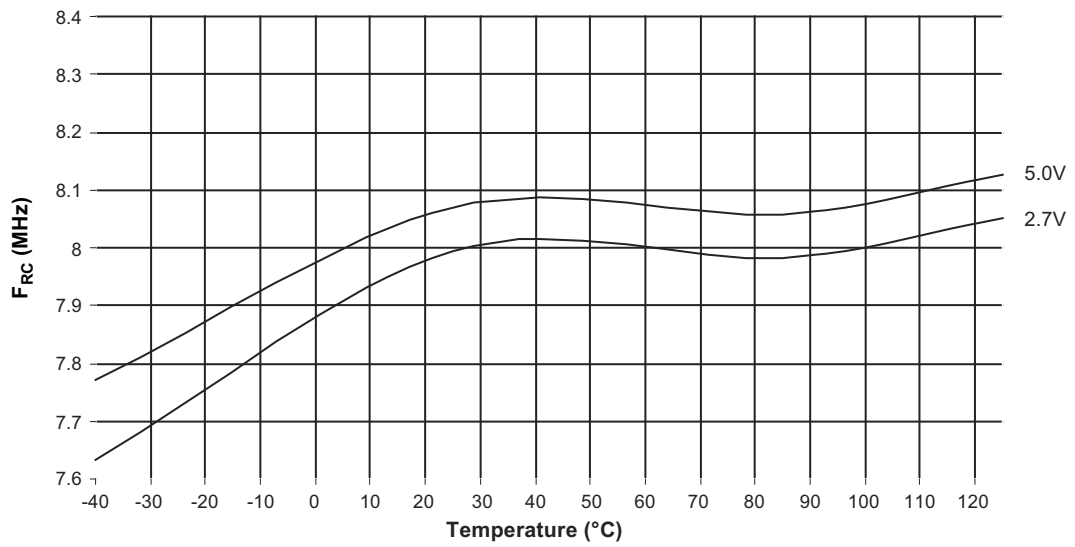


Figure 28-25. BOD Threshold versus Temperature (BODLEVEL is 2.7V)

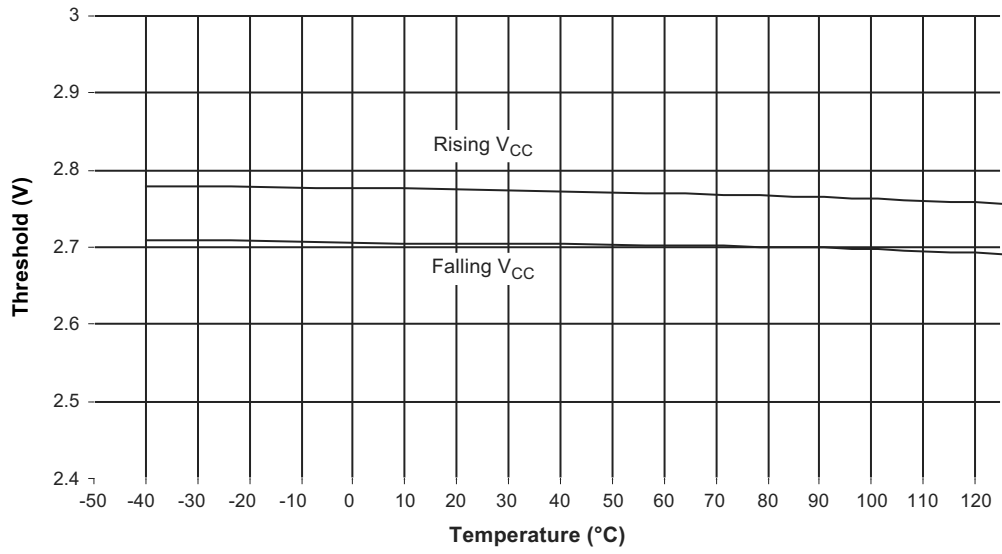
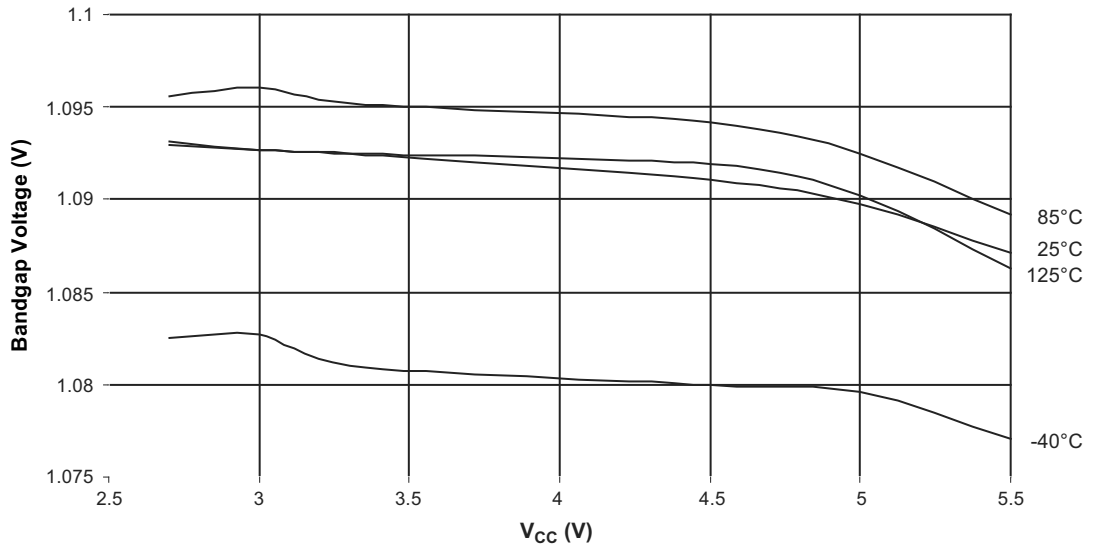


Figure 28-26. Bandgap Voltage versus  $V_{CC}$



## 29. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xB8)	TWBR	2-wire serial interface bit rate register								182
(0xB7)	Reserved	-	-	-	-	-	-	-	-	
(0xB6)	ASSR	-	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	135
(0xB5)	Reserved	-	-	-	-	-	-	-	-	
(0xB4)	OCR2B	Timer/Counter2 output compare register B								132
(0xB3)	OCR2A	Timer/Counter2 output compare register A								132
(0xB2)	TCNT2	Timer/Counter2 (8-bit)								132
(0xB1)	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20	131
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20	129
(0xAF)	Reserved	-	-	-	-	-	-	-	-	
(0xAE)	Reserved	-	-	-	-	-	-	-	-	
(0xAD)	Reserved	-	-	-	-	-	-	-	-	
(0xAC)	Reserved	-	-	-	-	-	-	-	-	
(0xAB)	Reserved	-	-	-	-	-	-	-	-	
(0xAA)	Reserved	-	-	-	-	-	-	-	-	
(0xA9)	Reserved	-	-	-	-	-	-	-	-	
(0xA8)	Reserved	-	-	-	-	-	-	-	-	
(0xA7)	Reserved	-	-	-	-	-	-	-	-	
(0xA6)	Reserved	-	-	-	-	-	-	-	-	
(0xA5)	Reserved	-	-	-	-	-	-	-	-	
(0xA4)	Reserved	-	-	-	-	-	-	-	-	
(0xA3)	Reserved	-	-	-	-	-	-	-	-	
(0xA2)	Reserved	-	-	-	-	-	-	-	-	
(0xA1)	Reserved	-	-	-	-	-	-	-	-	
(0xA0)	Reserved	-	-	-	-	-	-	-	-	
(0x9F)	Reserved	-	-	-	-	-	-	-	-	
(0x9E)	Reserved	-	-	-	-	-	-	-	-	
(0x9D)	Reserved	-	-	-	-	-	-	-	-	
(0x9C)	Reserved	-	-	-	-	-	-	-	-	
(0x9B)	Reserved	-	-	-	-	-	-	-	-	
(0x9A)	Reserved	-	-	-	-	-	-	-	-	
(0x99)	Reserved	-	-	-	-	-	-	-	-	
(0x98)	Reserved	-	-	-	-	-	-	-	-	
(0x97)	Reserved	-	-	-	-	-	-	-	-	
(0x96)	Reserved	-	-	-	-	-	-	-	-	
(0x95)	Reserved	-	-	-	-	-	-	-	-	

- Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  2. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
  3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVR®, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
  4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48/88/168 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
  5. Only valid for Atmel® ATmega88/168

## 33. Errata ATmega48

The revision letter in this section refers to the revision of the Atmel® ATmega48 device.

### 33.1 Rev. E

- Interrupts may be lost when writing the timer registers in the asynchronous timer

1. Interrupts may be lost when writing the timer registers in the asynchronous timer

If one of the timer registers which is synchronized to the asynchronous Timer2 clock is written in the cycle before an overflow interrupt occurs, the interrupt may be lost.

#### **Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 control register, TCCR2, or output compare register, OCR2.

### 33.2 Rev. D

- Interrupts may be lost when writing the timer registers in the asynchronous timer

- POR sensitivity with Vcc ramp up from a very low supply voltage

1. Interrupts may be lost when writing the timer registers in the asynchronous timer

If one of the timer registers which is synchronized to the asynchronous Timer2 clock is written in the cycle before an overflow interrupt occurs, the interrupt may be lost.

#### **Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 control register, TCCR2, or output compare register, OCR2.

2. POR sensitivity with Vcc ramp up from a very low supply voltage

If Vcc ramp up from a stable 150mV to 300mV plateau, the power on reset (POR) may not reset the device properly.

#### **Problem Fix/Workaround**

None.

### 33.3 Rev. A

- Interrupts may be lost when writing the timer registers in the asynchronous timer

- POR sensitivity with Vcc ramp up from a very low supply voltage

1. Interrupts may be lost when writing the timer registers in the asynchronous timer

If one of the timer registers which is synchronized to the asynchronous Timer2 clock is written in the cycle before an overflow interrupt occurs, the interrupt may be lost.

#### **Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 control register, TCCR2, or output compare register, OCR2.

2. POR sensitivity with Vcc ramp up from a very low supply voltage

If Vcc ramp up from a stable 150mV to 300mV plateau, the power on reset (POR) may not reset the device properly.

#### **Problem Fix/Workaround**

None.

Note: Please note from datasheet 7530F-AVR-09/07 we introduce a new errata numbering scheme (errata Rev E of datasheet 7530E-AVR-03/07 is equivalent to errata rev D of datasheet 7530F-AVR-09/07)

8.	Minimizing Power Consumption	36
8.1	System Control and Reset	38
8.2	Resetting the AVR	38
8.3	Reset Sources	38
8.4	Power-on Reset	39
8.5	External Reset	41
8.6	Brown-out Detection	41
8.7	Watchdog System Reset	42
8.8	MCU Status Register – MCUSR	43
8.9	Internal Voltage Reference	43
8.10	Watchdog Timer	44
9.	Interrupts	48
9.1	Interrupt Vectors in ATmega48	48
9.2	Interrupt Vectors in ATmega88	49
9.3	Interrupt Vectors in ATmega168	52
10.	I/O-Ports	57
10.1	Introduction	57
10.2	Ports as General Digital I/O	58
10.3	Alternate Port Functions	62
10.4	Register Description for I/O Ports	71
11.	External Interrupts	73
11.1	External Interrupt Control Register A – EICRA	73
11.2	External Interrupt Mask Register – EIMSK	74
11.3	External Interrupt Flag Register – EIFR	74
11.4	Pin Change Interrupt Control Register - PCICR	75
11.5	Pin Change Interrupt Flag Register - PCIFR	75
11.6	Pin Change Mask Register 2 – PCMSK2	76
11.7	Pin Change Mask Register 1 – PCMSK1	76
11.8	Pin Change Mask Register 0 – PCMSK0	76
12.	8-bit Timer/Counter0 with PWM	77
12.1	Overview	77
12.2	Timer/Counter Clock Sources	78
12.3	Counter Unit	78
12.4	Output Compare Unit	79
12.5	Compare Match Output Unit	80
12.6	Modes of Operation	81
12.7	Timer/Counter Timing Diagrams	85
12.8	8-bit Timer/Counter Register Description	87
13.	Timer/Counter0 and Timer/Counter1 Prescalers	92
13.1	Internal Clock Source	92
13.2	Prescaler Reset	92
13.3	External Clock Source	92
14.	16-bit Timer/Counter1 with PWM	94
14.1	Overview	94
14.2	Accessing 16-bit Registers	96
14.3	Timer/Counter Clock Sources	99
14.4	Counter Unit	100
14.5	Input Capture Unit	101
14.6	Output Compare Units	102
14.7	Compare Match Output Unit	104