

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

E·XFI

Product Status	Active
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	32
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VFQFN Exposed Pad
Supplier Device Package	44-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega16-16mur

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

#### Power Management and Sleep Modes

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

To enter any of the six sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the MCUCR Register select which sleep mode (Idle, ADC Noise Reduction, Power-down, Power-save, Standby, or Extended Standby) will be activated by the SLEEP instruction. See Table 13 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, it executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. If a Reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Figure 11 on page 24 presents the different clock systems in the ATmega16, and their distribution. The figure is helpful in selecting an appropriate sleep mode.

#### MCU Control Register – MCUCR

The MCU Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	_
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
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, 5, 4 - SM2..0: Sleep Mode Select Bits 2, 1, and 0

These bits select between the six available sleep modes as shown in Table 13.

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby <sup>(1)</sup>
1	1	1	Extended Standby <sup>(1)</sup>

#### Table 13. Sleep Mode Select

Note: 1. Standby mode and Extended Standby mode are only available with external crystals or resonators.

#### • Bit 6 – SE: Sleep Enable

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.



# **Standby Mode** When the SM2..0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

## **Extended Standby** Mode When the SM2..0 bits are 111 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Extended Standby mode. This mode is identical to Power-save mode with the exception that the Oscillator is kept running. From Extended Standby mode, the device wakes up in six clock cycles..

	Active Clock domains			Oscilla	Wake-up Sources								
Sleep Mode	clk <sub>CPU</sub>	clk <sub>FLASH</sub>	clk <sub>io</sub>	clk <sub>ADC</sub>	clk <sub>ASY</sub>	Main Clock Source Enabled	Timer Osc. Enabled	INT2 INT1 INT0	TWI Address Match	Timer 2	SPM / EEPROM Ready	ADC	Other I/O
Idle			Х	Х	Х	Х	X <sup>(2)</sup>	Х	Х	Х	Х	Х	Х
ADC Noise Redu- ction				x	х	Х	X <sup>(2)</sup>	X <sup>(3)</sup>	Х	x	Х	x	
Power Down								Х <sup>(3)</sup>	х				
Power Save					X <sup>(2)</sup>		X <sup>(2)</sup>	X <sup>(3)</sup>	Х	X <sup>(2)</sup>			
Standby <sup>(1)</sup>						Х		X <sup>(3)</sup>	Х				
Exten- ded Standby <sup>(1)</sup>					X <sup>(2)</sup>	Х	X <sup>(2)</sup>	X <sup>(3)</sup>	Х	X <sup>(2)</sup>			

Table 14. Active Clock Domains and Wake L	Jp Sources in the Different Sleep Modes
---	---

Notes: 1. External Crystal or resonator selected as clock source.

2. If AS2 bit in ASSR is set.

3. Only INT2 or level interrupt INT1 and INT0.



#### **Power-on Reset**

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 15. The POR is activated whenever  $V_{CC}$  is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after  $V_{CC}$  rise. The RESET signal is activated again, without any delay, when  $V_{CC}$  decreases below the detection level.





#### Figure 17. MCU Start-up, RESET Extended Externally





#### **External Reset**

An External Reset is generated by a low level on the RESET pin. Reset pulses longer than the minimum pulse width (see Table 15) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage –  $V_{RST}$  – on its positive edge, the delay counter starts the MCU after the Time-out period t<sub>TOUT</sub> has expired.





# **Brown-out Detection** ATmega16 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V<sub>CC</sub> level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT} + V_{HYST}/2$ and $V_{BOT} = V_{BOT} - V_{HYST}/2$ .

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and  $V_{CC}$  decreases to a value below the trigger level ( $V_{BOT-}$  in Figure 19), the Brown-out Reset is immediately activated. When  $V_{CC}$  increases above the trigger level ( $V_{BOT+}$  in Figure 19), 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 15.

Figure 19. Brown-out Reset During Operation





#### External Interrupts

The External Interrupts are triggered by the INT0, INT1, and INT2 pins. Observe that, if enabled, the interrupts will trigger even if the INT0..2 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level (INT2 is only an edge triggered interrupt). This is set up as indicated in the specification for the MCU Control Register – MCUCR – and MCU Control and Status Register – MCUCSR. When the external interrupt is enabled and is configured as level triggered (only INT0/INT1), the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 and INT1 requires the presence of an I/O clock, described in "Clock Systems and their Distribution" on page 24. Low level interrupts on INT0/INT1 and the edge interrupt on INT2 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock. The period of the Watchdog Oscillator is 1 µs (nominal) at 5.0V and 25°C. The frequency of the Watchdog Oscillator is voltage dependent as shown in "Electrical Characteristics" on page 291. The MCU will wake up if the input has the required level during this sampling or if it is held until the end of the start-up time. The start-up time is defined by the SUT Fuses as described in "System Clock and Clock Options" on page 24. If the level is sampled twice by the Watchdog Oscillator clock but disappears before the end of the start-up time, the MCU will still wake up, but no interrupt will be generated. The required level must be held long enough for the MCU to complete the wake up to trigger the level interrupt.

**MCU Control Register** The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

Bit	7	6	5	4	3	2	1	0	_
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
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 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-bit and the corresponding interrupt mask in the GICR are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 34. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 34.	Interrupt 1	Sense	Contro
-----------	-------------	-------	--------

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.



whether  $clk_{T0}$  is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC0. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 76.

The Timer/Counter Overflow (TOV0) Flag is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

**Output Compare Unit** The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0). Whenever TCNT0 equals OCR0, the comparator signals a match. A match will set the Output Compare Flag (OCF0) at the next timer clock cycle. If enabled (OCIE0 = 1 and Global Interrupt Flag in SREG is set), the Output Compare Flag generates an output compare interrupt. The OCF0 Flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0 Flag can be cleared by software by writing a logical one to its I/O bit location. The waveform generator uses the match signal to generate an output according to operating mode set by the WGM01:0 bits and Compare Output mode (COM01:0) bits. The max and bottom signals are used by the waveform generator for handling the special cases of the extreme values in some modes of operation (See "Modes of Operation" on page 76.).

Figure 29 shows a block diagram of the output compare unit.



Figure 29. Output Compare Unit, Block Diagram

The OCR0 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 OCR0 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.



Modes of Operation	The mode of operation, that is, the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM01:0) and Compare Output mode (COM01:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM01:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM01:0 bits control whether the output should be set, cleared, or toggled at a compare match (See "Compare Match Output Unit" on page 74.).
	For detailed timing information refer to Figure 34, Figure 35, Figure 36 and Figure 37 in "Timer/Counter Timing Diagrams" on page 81.
Normal Mode	The simplest mode of operation is the normal mode (WGM01:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 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 mode, a new counter value can be written anytime.
	The output compare unit can be used to generate interrupts at some given time. Using the output compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.
Clear Timer on Compare Match (CTC) Mode	In Clear Timer on Compare or CTC mode (WGM01:0 = 2), the OCR0 Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches the OCR0. The OCR0 defines the top value for the counter, hence also its

(TCNT0) matches the OCR0. The OCR0 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 31. The counter value (TCNT0) increases until a compare match occurs between TCNT0 and OCR0, and then counter (TCNT0) is cleared.

Figure 31. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0 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 PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk\_l/O}}{N \cdot (1 + TOP)}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1x is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR1x equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COM1x1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC1A to toggle its logical level on each compare match (COM1A1:0 = 1). This applies only if OCR1A is used to define the TOP value (WGM13:0 = 15). The waveform generated will have a maximum frequency of  $f_{OC1A} = f_{clk\_I/O}/2$  when OCR1A is set to zero (0x0000). This feature is similar to the OC1A toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

**Phase Correct PWM** Mode
The phase correct Pulse Width Modulation or phase correct PWM mode (WGM13:0 = 1,2,3,10, or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x 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.

The PWM resolution for the phase correct PWM mode can be fixed to 8-bit, 9-bit, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In phase correct PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 1, 2, or 3), the value in ICR1 (WGM13:0 = 10), or the value in OCR1A (WGM13:0 = 11). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 47. The figure shows phase correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 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 TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.



#### Timer/Counter Timing Diagrams

The following figures show the Timer/Counter in Synchronous mode, and the timer clock ( $clk_{T2}$ ) is therefore shown as a clock enable signal. In Asynchronous mode,  $clk_{I/O}$  should be replaced by the Timer/Counter Oscillator clock. The figures include information on when Interrupt Flags are set. Figure 60 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.





Figure 61 shows the same timing data, but with the prescaler enabled.

Figure 61. Timer/Counter Timing Diagram, with Prescaler (f<sub>clk I/O</sub>/8)



Figure 62 shows the setting of OCF2 in all modes except CTC mode.



#### Timer/Counter Prescaler

Figure 64. 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  $clk_{1O}$ . 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 32.768 kHz 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 PSR2 bit in SFIOR resets the prescaler. This allows the user to operate with a predictable prescaler.

#### Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	_
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	SFIOR
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 1 – PSR2: Prescaler Reset Timer/Counter2

When this bit is written to one, the Timer/Counter2 prescaler will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. This bit will always be read as zero if Timer/Counter2 is clocked by the internal CPU clock. If this bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset.



#### **Data Modes**

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 67 and Figure 68. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing Table 56 and Table 57, as done below:

Table 59. CPOL and CPHA Functiona	ality
-----------------------------------	-------

	Leading Edge	Trailing Edge	SPI Mode
CPOL = 0, CPHA = 0	Sample (Rising)	Setup (Falling)	0
CPOL = 0, CPHA = 1	Setup (Rising)	Sample (Falling)	1
CPOL = 1, CPHA = 0	Sample (Falling)	Setup (Rising)	2
CPOL = 1, CPHA = 1	Setup (Falling)	Sample (Rising)	3





Figure 68. SPI Transfer Format with CPHA = 1





The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control Registers are shared by all units. The clock generation logic consists of synchronization logic for external clock input used by synchronous Slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by Synchronous Transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The receiver units are used for asynchronous data reception. In addition to the recovery units, the receiver includes a parity checker, control logic, a Shift Register and a two level receive buffer (UDR). The receiver supports the same frame formats as the transmitter, and can detect frame error, data overrun and parity errors.

**AVR USART vs. AVR** The USART is fully compatible with the AVR UART regarding:

UART - Compatibility . Bit locations inside all USART Registers

- Baud Rate Generation
- Transmitter Operation
- Transmit Buffer Functionality
- Receiver Operation

However, the receive buffering has two improvements that will affect the compatibility in some special cases:

- A second Buffer Register has been added. The two Buffer Registers operate as a circular FIFO buffer. Therefore the UDR must only be read once for each incoming data! More important is the fact that the Error Flags (FE and DOR) and the 9th data bit (RXB8) are buffered with the data in the receive buffer. Therefore the status bits must always be read before the UDR Register is read. Otherwise the error status will be lost since the buffer state is lost.
- The receiver Shift Register can now act as a third buffer level. This is done by allowing the received data to remain in the serial Shift Register (see Figure 69) if the Buffer Registers are full, until a new start bit is detected. The USART is therefore more resistant to Data OverRun (DOR) error conditions.

The following control bits have changed name, but have same functionality and register location:

- CHR9 is changed to UCSZ2
- OR is changed to DOR
- **Clock Generation** The clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous mode. The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation. Double Speed (Asynchronous mode only) is controlled by the U2X found in the UCSRA Register. When using Synchronous mode (UMSEL = 1), the Data Direction Register for the XCK pin (DDR\_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using Synchronous mode.

Figure 70 shows a block diagram of the clock generation logic.



figure), to decide if a valid start bit is received. If two or more of these three samples have logical high levels (the majority wins), the start bit is rejected as a noise spike and the receiver starts looking for the next high to low-transition. If however, a valid start bit is detected, the clock recovery logic is synchronized and the data recovery can begin. The synchronization process is repeated for each start bit.

#### Asynchronous Data Recovery

When the receiver clock is synchronized to the start bit, the data recovery can begin. The data recovery unit uses a state machine that has 16 states for each bit in normal mode and 8 states for each bit in Double Speed mode. Figure 74 shows the sampling of the data bits and the parity bit. Each of the samples is given a number that is equal to the state of the recovery unit.



Figure 74. Sampling of Data and Parity Bit

The decision of the logic level of the received bit is taken by doing a majority voting of the logic value to the three samples in the center of the received bit. The center samples are emphasized on the figure by having the sample number inside boxes. The majority voting process is done as follows: If two or all three samples have high levels, the received bit is registered to be a logic 1. If two or all three samples have low levels, the received bit is registered to be a logic 0. This majority voting process acts as a low pass filter for the incoming signal on the RxD pin. The receiver process is then repeated until a complete frame is received. Including the first stop bit. Note that the receiver only uses the first stop bit of a frame.

Figure 75 shows the sampling of the stop bit and the earliest possible beginning of the start bit of the next frame.

Figure 75. Stop Bit Sampling and Next Start Bit Sampling



The same majority voting is done to the stop bit as done for the other bits in the frame. If the stop bit is registered to have a logic 0 value, the Frame Error (FE) Flag will be set.

A new high to low transition indicating the start bit of a new frame can come right after the last of the bits used for majority voting. For Normal Speed mode, the first low level sample can be at point marked (A) in Figure 75. For Double Speed mode the first low level must be delayed to (B). (C) marks a stop bit of full length. The early start bit detection influences the operational range of the receiver.







**Boundary-scan Chain** The Boundary-scan Chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having Off-chip connections.

See "Boundary-scan Chain" on page 232 for a complete description.

**Boundary-scan Specific JTAG Instructions** The instruction register is 4-bit wide, supporting up to 16 instructions. Listed below are the JTAG instructions useful for Boundary-scan operation. Note that the optional HIGHZ instruction is not implemented, but all outputs with tri-state capability can be set in high-impedant state by using the AVR\_RESET instruction, since the initial state for all port pins is tri-state.

As a definition in this datasheet, the LSB is shifted in and out first for all Shift Registers.

The OPCODE for each instruction is shown behind the instruction name in hex format. The text describes which Data Register is selected as path between TDI and TDO for each instruction.

**EXTEST; \$0** Mandatory JTAG instruction for selecting the Boundary-scan Chain as Data Register for testing circuitry external to the AVR package. For port-pins, Pull-up Disable, Output Control, Output Data, and Input Data are all accessible in the scan chain. For Analog circuits having Off-chip connections, the interface between the analog and the digital logic is in the scan chain. The contents of the latched outputs of the Boundary-scan chain is driven out as soon as the JTAG IR-register is loaded with the EXTEST instruction.

The active states are:

- Capture-DR: Data on the external pins are sampled into the Boundary-scan Chain.
- Shift-DR: The Internal Scan Chain is shifted by the TCK input.
- Update-DR: Data from the scan chain is applied to output pins.
- **IDCODE; \$1** Optional JTAG instruction selecting the 32-bit ID-register as Data Register. The ID-register consists of a version number, a device number and the manufacturer code chosen by JEDEC. This is the default instruction after power-up.

The active states are:

- Capture-DR: Data in the IDCODE-register is sampled into the Boundary-scan Chain.
- Shift-DR: The IDCODE scan chain is shifted by the TCK input.

## SAMPLE\_PRELOAD; Mandatory JTAG instruction for pre-loading the output latches and talking a snap-shot of the input/output pins without affecting the system operation. However, the output latches are not connected to the pins. The Boundary-scan Chain is selected as Data Register.



Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/BSY	PD1	0	0: Device is busy programming, 1: Device is ready for new command
ŌĒ	PD2	Ι	Output Enable (Active low)
WR	PD3	Ι	Write Pulse (Active low)
BS1	PD4	I	Byte Select 1 ("0" selects Low byte, "1" selects High byte)
XA0	PD5	Ι	XTAL Action Bit 0
XA1	PD6	Ι	XTAL Action Bit 1
PAGEL	PD7	Ι	Program Memory and EEPROM data Page Load
BS2	PA0	I	Byte Select 2 ("0" selects Low byte, "1" selects 2'nd High byte)
DATA	PB7-0	I/O	Bidirectional Data bus (Output when $\overline{OE}$ is low)

Table 109.	Pin	Name	Mapping
------------	-----	------	---------

#### Table 110. Pin Values used to Enter Programming Mode

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

#### Table 111. 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

#### Table 112. 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







- Notes: 1. If the device is clocked by the Internal Oscillator, it is no need to connect a clock source to the XTAL1 pin.
  - 2.  $V_{CC}$  -0.3V < AVCC <  $V_{CC}$  +0.3V, however, AVCC should always be within 2.7V 5.5V

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 \$FF.

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  $f_{ck} \ge$  12 MHz

High: > 2 CPU clock cycles for  $f_{ck}$  < 12 MHz, 3 CPU clock cycles for  $f_{ck} \ge$  12 MHz

When writing serial data to the ATmega16, data is clocked on the rising edge of SCK.

When reading data from the ATmega16, data is clocked on the falling edge of SCK. See Figure 138 for timing details.

To program and verify the ATmega16 in the SPI Serial Programming mode, the following sequence is recommended (See four byte instruction formats in Figure 116 on page 276):

- Power-up sequence: Apply power between V<sub>CC</sub> and GND while 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, RESET must be given a positive pulse of at least two CPU clock cycles duration after SCK has been set to "0".
- 2. Wait for at least 20 ms and enable SPI Serial Programming by sending the Programming Enable serial instruction to pin MOSI.
- 3. The SPI Serial Programming instructions will not work if the communication is out of synchronization. When in sync. the second byte (\$53), 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 \$53 did not echo back, give RESET a positive pulse and issue a new Programming Enable command.



2466T-AVR-07/10

**SPI Serial** 

Algorithm

Programming

5] The sum of all IOH, for ports D3 - D7, should not exceed 100 mA.

6] The sum of all IOH, for ports C0 - C7, 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. Minimum  $V_{CC}$  for Power-down is 2.5V.



#### External Clock Drive

Table 118. External Clock Drive<sup>(1)</sup>

		V <sub>CC</sub> = 2.7V to 5.5V		V <sub>CC</sub> = 4.5V to 5.5V		
Symbol	Parameter	Min	Max	Min	Max	Units
1/t <sub>CLCL</sub>	Oscillator Frequency	0	8	0	16	MHz
t <sub>CLCL</sub>	Clock Period	125		62.5		
t <sub>CHCX</sub>	High Time	50		25		ns
t <sub>CLCX</sub>	Low Time	50		25		
t <sub>CLCH</sub>	Rise Time		1.6		0.5	
t <sub>CHCL</sub>	Fall Time		1.6		0.5	μs
$\Delta t_{CLCL}$	Change in period from one clock cycle to the next		2		2	%

t<sub>CLCX</sub>

t<sub>CLCL</sub>

Note: 1. Refer to "External Clock" on page 31 for details.

Table 119.	External RC Oscillato	or, Typical Frequencie	$es(V_{CC} = 5)$
------------	-----------------------	------------------------	------------------

<b>R [k</b> Ω] <sup>(1)</sup>	C [pF]	<b>f</b> <sup>(2)</sup>
33	22	650 kHz
10	22	2.0 MHz

Notes: 1. R should be in the range  $3 k\Omega - 100 k\Omega$ , and C should be at least 20 pF.

2. The frequency will vary with package type and board layout.







Figure 181. I/O Pin Sink Current vs. Output Voltage ( $V_{CC} = 2.7V$ )









#### Electrical Characteristics 291

Absolute Maximum Ratings\* 291 DC Characteristics 291 External Clock Drive Waveforms 293 External Clock Drive 293 Two-wire Serial Interface Characteristics 294 SPI Timing Characteristics 295 ADC Characteristics 297

#### ATmega16 Typical Characteristics 299

#### **Register Summary 331**

#### Instruction Set Summary 333

#### **Ordering Information 336**

#### Packaging Information 337

44A 337 40P6 338 44M1 339

#### Errata 340

ATmega16(L) Rev. M 340 ATmega16(L) Rev. L 340 ATmega16(L) Rev. K 341 ATmega16(L) Rev. J 342 ATmega16(L) Rev. I 343 ATmega16(L) Rev. H 344

#### Datasheet Revision History 345

Rev. 2466T-07/10 345 Rev. 2466S-05/09 345 Rev. 2466R-06/08 345 Rev. 2466Q-05/08 345 Rev. 2466P-08/07 345 Rev. 2466P-03/07 345 Rev. 2466N-10/06 346 Rev. 2466M-04/06 346 Rev. 2466L-06/05 346 Rev. 2466L-06/05 346 Rev. 2466L-10/04 346 Rev. 2466I-10/04 346 Rev. 2466H-12/03 347 Rev. 2466F-02/03 347

