



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	16MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TC)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/ata6614q-plqw-1

4.6 Voltage Regulator

The voltage regulator needs an external capacitor for compensation and for smoothing the disturbances from the microcontroller. It is recommended to use an electrolytic capacitor with $C \geq 1.8\mu\text{F}$ and a ceramic capacitor with $C = 100\text{nF}$. The values of these capacitors can be varied by the customer, depending on the application.

The main power dissipation of the IC is created from the VCC output current I_{VCC} , which is needed for the application.

Figure 4-11. VCC Voltage Regulator: Ramp-up and Undervoltage Detection

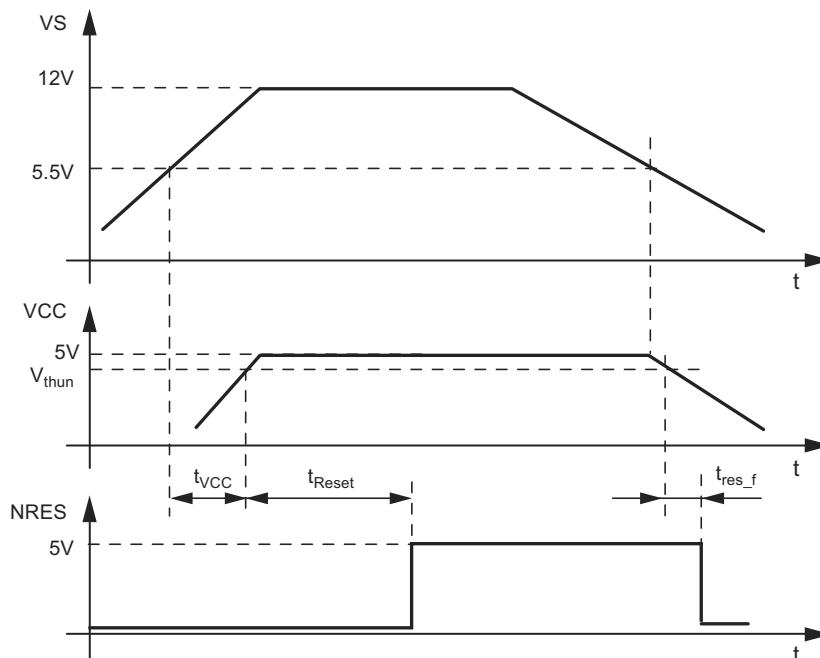
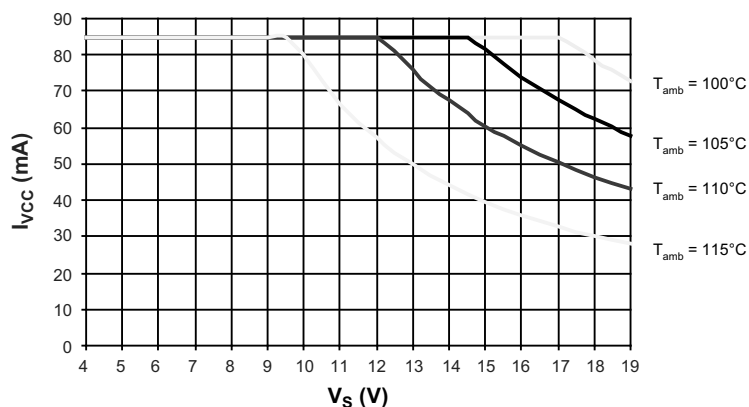


Figure 4-12. Power Dissipation: Safe Operating Area: VCC Output Current versus Supply Voltage VS at Different Ambient Temperatures



For programming purposes of the microcontroller it is potentially necessary to supply the VCC output via an external power supply while the VS Pin of the system basis chip is disconnected. This behavior will not affect the system basis chip.

4.8 Electrical Characteristics (Continued)

5V < V_S < 27V, -40°C < T_j < 150°C, unless otherwise specified. All values refer to GND pins

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
4	EN Input Pin								
4.1	Low-level voltage input		EN	V _{ENL}	-0.3		+0.8	V	A
4.2	High-level voltage input		EN	V _{ENH}	2		V _{CC} + 0.3V	V	A
4.3	Pull-down resistor	V _{EN} = V _{CC}	EN	R _{EN}	50	125	200	kΩ	A
4.4	Low-level input current	V _{EN} = 0V	EN	I _{EN}	-3		+3	μA	A
5	NTRIG Watchdog Input Pin								
5.1	Low-level voltage input		NTRIG	V _{NTRIGL}	-0.3		+0.8	V	A
5.2	High-level voltage input		NTRIG	V _{NTRIGH}	2		V _{CC} + 0.3V	V	A
5.3	Pull-up resistor	V _{NTRIG} = 0V	NTRIG	R _{NTRIG}	125	250	400	kΩ	A
5.4	High-level leakage current	V _{NTRIG} = V _{CC}	NTRIG	I _{NTRIG}	-3		+3	μA	A
6	Mode Input Pin								
6.1	Low-level voltage input		MODE	V _{MODEL}	-0.3		+0.8	V	A
6.2	High-level voltage input		MODE	V _{MODEH}	2		V _{CC} + 0.3V	V	A
6.3	High-level leakage current	V _{MODE} = V _{CC} or V _{MODE} = 0V	MODE	I _{MODE}	-3		+3	μA	A
7	INH Output Pin								
7.1	High-level voltage	I _{INH} = -15mA	INH	V _{INHH}	V _S - 0.75		V _S	V	A
7.2	Switch-on resistance between VS and INH		INH	R _{INH}		30	50	Ω	A
7.3	Leakage current	Sleep Mode V _{INH} = 0V/27V, V _S = 27V	INH	I _{INHL}	-3		+3	μA	A
8	LIN Bus Driver								
8.1	Driver recessive output voltage	Load1/Load2	LIN	V _{BUSrec}	0.9 × V _S		V _S	V	A
8.2	Driver dominant voltage	V _{VS} = 7V R _{load} = 500Ω	LIN	V _{LoSUP}			1.2	V	A
8.3	Driver dominant voltage	V _{VS} = 18V R _{load} = 500Ω	LIN	V _{HiSUP}			2	V	A
8.4	Driver dominant voltage	V _{VS} = 7.0V R _{load} = 1000Ω	LIN	V _{LoSUP_1k}	0.6			V	A
8.5	Driver dominant voltage	V _{VS} = 18V R _{load} = 1000Ω	LIN	V _{HiSUP_1k}	0.8			V	A
8.6	Pull-up resistor to VS	The serial diode is mandatory	LIN	R _{LIN}	20	30	47	kΩ	A
8.7	Voltage drop at the serial diodes	In pull-up path with R _{slave} I _{SerDiode} = 10mA	LIN	V _{SerDiode}	0.4		1.0	V	D
8.8	LIN current limitation V _{BUS} = V _{Batt_max}		LIN	I _{BUS_LIM}	70	120	200	mA	A

*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

Figure 5-3 shows the structure of the 32 general purpose working registers in the CPU.

Figure 5-3. AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

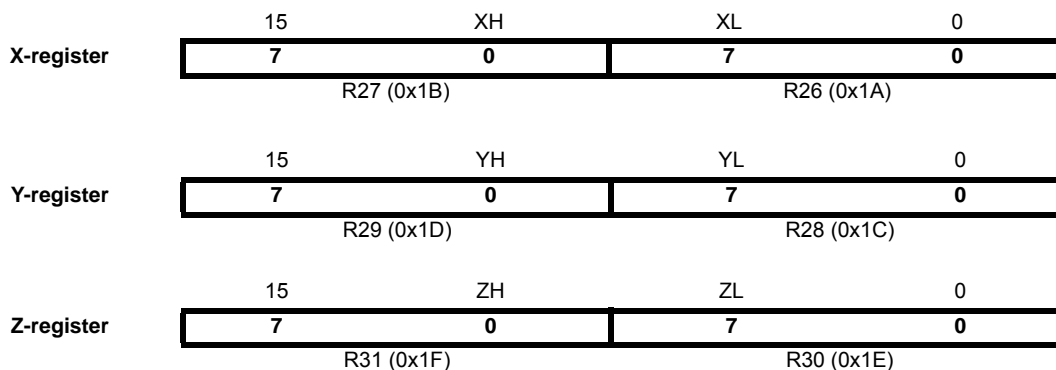
Most of the instructions operating on the register file have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 5-3, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user data space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

5.6.4.1 The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 5-4.

Figure 5-4. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

5.9.11 Register Description

5.9.11.1 SMCR – Sleep Mode Control Register

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	–	–	–	–	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 Res: Reserved Bits**

These bits are unused bits in the ATmega328P, and will always read as zero.

- **Bits 3..1 – SM2..0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the five available sleep modes as shown in Table 5-21.

Table 5-21. Sleep Mode Select

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 ⁽¹⁾
1	1	1	External standby ⁽¹⁾

Note: 1. Standby mode is only recommended for use with external crystals or resonators.

- **Bit 0 – 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 programmer's 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.

5.9.11.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – BODS: BOD Sleep**

The BODS bit must be written to logic one in order to turn off BOD during sleep, see Table 5-20 on page 58. Writing to the BODS bit is controlled by a timed sequence and an enable bit, BODSE in MCUCR. To disable BOD in relevant sleep modes, both BODS and BODSE must first be set to one. Then, to set the BODS bit, BODS must be set to one and BODSE must be set to zero within four clock cycles.

The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

- **Bit 5 – BODSE: BOD Sleep Enable**

BODSE enables setting of BODS control bit, as explained in BODS bit description. BOD disable is controlled by a timed sequence.

5.14.9.7 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the ATmega328P and will always read as zero.

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**

The OCF0B bit is set when a compare match occurs between the Timer/Counter and the data in OCR0B – output compare register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter compare B match interrupt Enable), and OCF0B are set, the Timer/Counter compare match interrupt is executed.

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a compare match occurs between the Timer/Counter0 and the data in OCR0A – output compare register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 compare match interrupt enable), and OCF0A are set, the Timer/Counter0 compare match interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 overflow interrupt enable), and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed. The setting of this flag is dependent of the WGM02:0 bit setting. Refer to Table 5-47 on page 109.

5.15.7.3 Using the Output Compare Unit

Since writing TCNT1 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT1 when using any of the output compare channels, independent of whether the Timer/Counter is running or not. If the value written to TCNT1 equals the OCR1x value, the compare match will be missed, resulting in incorrect waveform generation. Do not write the TCNT1 equal to TOP in PWM modes with variable TOP values. The compare match for the TOP will be ignored and the counter will continue to 0xFFFF. Similarly, do not write the TCNT1 value equal to BOTTOM when the counter is downcounting.

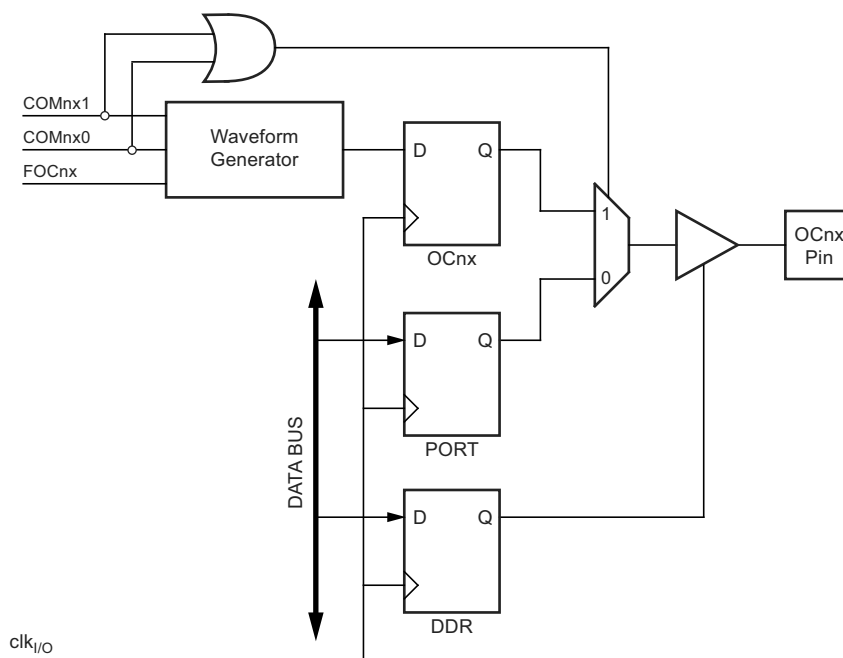
The setup of the OC1x should be performed before setting the data direction register for the port pin to output. The easiest way of setting the OC1x value is to use the force output compare (FOC1x) strobe bits in normal mode. The OC1x register keeps its value even when changing between waveform generation modes.

Be aware that the COM1x1:0 bits are not double buffered together with the compare value. Changing the COM1x1:0 bits will take effect immediately.

5.15.8 Compare Match Output Unit

The compare output mode (COM1x1:0) bits have two functions. The waveform generator uses the COM1x1:0 bits for defining the output compare (OC1x) state at the next compare match. Secondly the COM1x1:0 bits control the OC1x pin output source. Figure 5-42 shows a simplified schematic of the logic affected by the COM1x1:0 bit setting. The I/O registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM1x1:0 bits are shown. When referring to the OC1x state, the reference is for the internal OC1x register, not the OC1x pin. If a system reset occur, the OC1x register is reset to “0”.

Figure 5-42. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the output compare (OC1x) from the waveform generator if either of the COM1x1:0 bits are set. However, the OC1x pin direction (input or output) is still controlled by the data direction register (DDR) for the port pin. The data direction register bit for the OC1x pin (DDR_OC1x) must be set as output before the OC1x value is visible on the pin. The port override function is generally independent of the waveform generation mode, but there are some exceptions. Refer to Table 5-50, Table 5-51 and Table 5-52 for details.

The design of the output compare pin logic allows initialization of the OC1x state before the output is enabled. Note that some COM1x1:0 bit settings are reserved for certain modes of operation. See Section 5.15.11 “Register Description” on page 131.

The COM1x1:0 bits have no effect on the input capture unit.

5.15.10 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T1}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt flags are set, and when the OCR1x register is updated with the OCR1x buffer value (only for modes utilizing double buffering). Figure 5-47 shows a timing diagram for the setting of OCF1x.

Figure 5-47. Timer/Counter Timing Diagram, Setting of OCF1x, no Prescaling

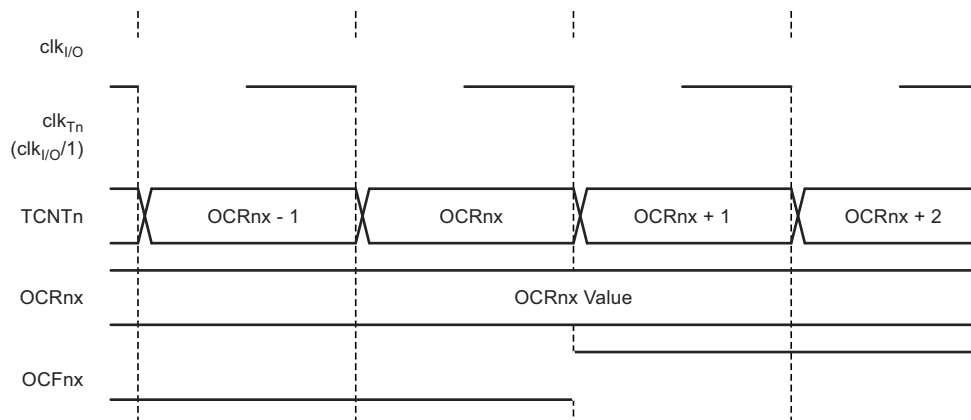
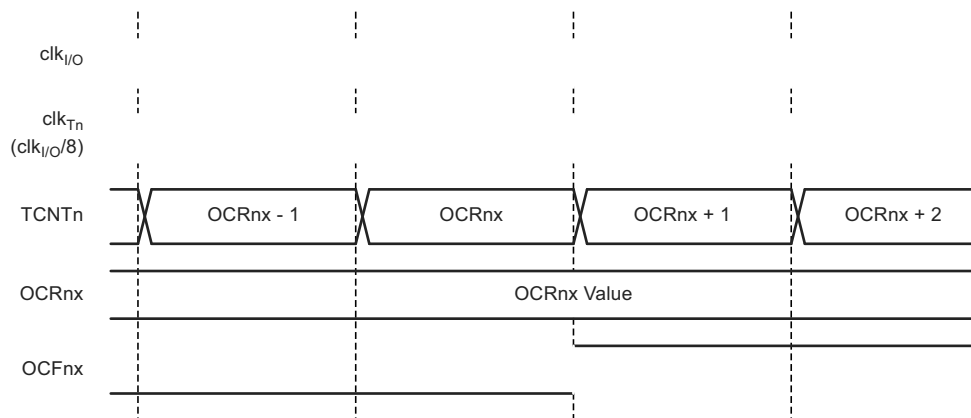


Figure 5-48 shows the same timing data, but with the prescaler enabled.

Figure 5-48. Timer/Counter Timing Diagram, Setting of OCF1x, with Prescaler ($f_{\text{clk}_{I/O}}/8$)



5.15.11.9 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the ATmega328P, and will always read as zero.

- **Bit 5 – ICF1: Timer/Counter1, Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the input capture register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the input capture interrupt vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the ATmega328P, and will always read as zero.

- **Bit 2 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register B (OCR1B).

Note that a forced output compare (FOC1B) strobe will not set the OCF1B flag.

OCF1B is automatically cleared when the output compare match B Interrupt Vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 1 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register A (OCR1A).

Note that a forced output compare (FOC1A) strobe will not set the OCF1A flag.

OCF1A is automatically cleared when the output compare match A interrupt vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter1, Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 flag is set when the timer overflows. Refer to Table 5-53 on page 132 for the TOV1 flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

5.16 Timer/Counter0 and Timer/Counter1 Prescalers

Section 5.14 “8-bit Timer/Counter0 with PWM” on page 98 and Section 5.15 “16-bit Timer/Counter1 with PWM” on page 113 share the same prescaler module, but the Timer/Counter can have different prescaler settings. The description below applies to both Timer/Counter1 and Timer/Counter0.

5.16.1 Internal Clock Source

The Timer/Counter can be clocked directly by the system clock (by setting the CSn2:0 = 1). This provides the fastest operation, with a maximum Timer/Counter clock frequency equal to system clock frequency ($f_{CLK_I/O}$). Alternatively, one of four taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either $f_{CLK_I/O}/8$, $f_{CLK_I/O}/64$, $f_{CLK_I/O}/256$, or $f_{CLK_I/O}/1024$.

5.16.2 Prescaler Reset

The prescaler is free running, i.e., operates independently of the clock select logic of the Timer/Counter, and it is shared by Timer/Counter1 and Timer/Counter0. Since the prescaler is not affected by the Timer/Counter's clock select, the state of the prescaler will have implications for situations where a prescaled clock is used. One example of prescaling artifacts occurs when the timer is enabled and clocked by the prescaler ($6 > CSn2:0 > 1$). The number of system clock cycles from when the timer is enabled to the first count occurs can be from 1 to N+1 system clock cycles, where N equals the prescaler divisor (8, 64, 256, or 1024).

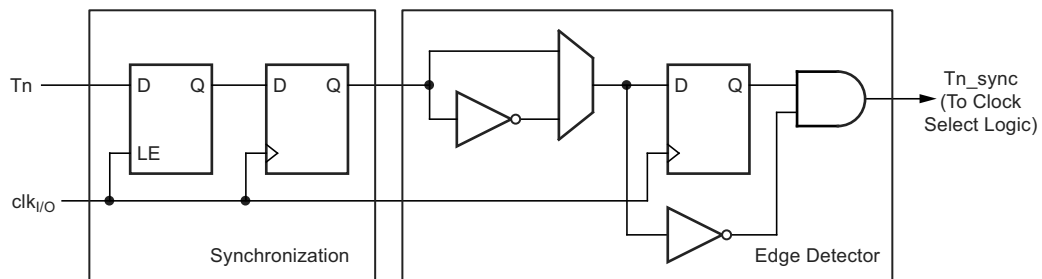
It is possible to use the prescaler reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect the prescaler period for all Timer/Counter it is connected to.

5.16.3 External Clock Source

An external clock source applied to the T1/T0 pin can be used as Timer/Counter clock (clk_{T1}/clk_{T0}). The T1/T0 pin is sampled once every system clock cycle by the pin synchronization logic. The synchronized (sampled) signal is then passed through the edge detector. Figure 5-51 shows a functional equivalent block diagram of the T1/T0 synchronization and edge detector logic. The registers are clocked at the positive edge of the internal system clock ($clk_{I/O}$). The latch is transparent in the high period of the internal system clock.

The edge detector generates one clk_{T1}/clk_{T0} pulse for each positive ($CSn2:0 = 7$) or negative ($CSn2:0 = 6$) edge it detects.

Figure 5-51. T1/T0 Pin Sampling



The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated.

Enabling and disabling of the clock input must be done when T1/T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

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

An external clock source can not be prescaled.

Figure 5-62 shows the setting of OCF2A in all modes except CTC mode.

Figure 5-62. Timer/Counter Timing Diagram, Setting of OCF2A, with Prescaler ($f_{clk_I/O}/8$)

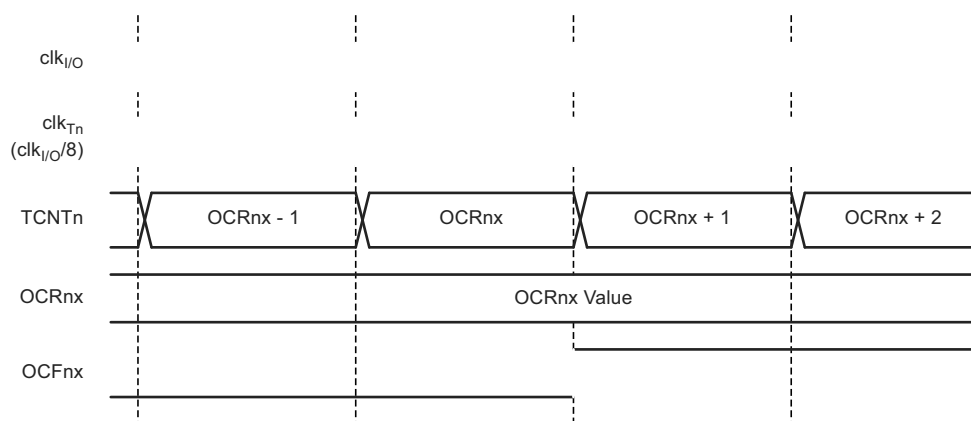
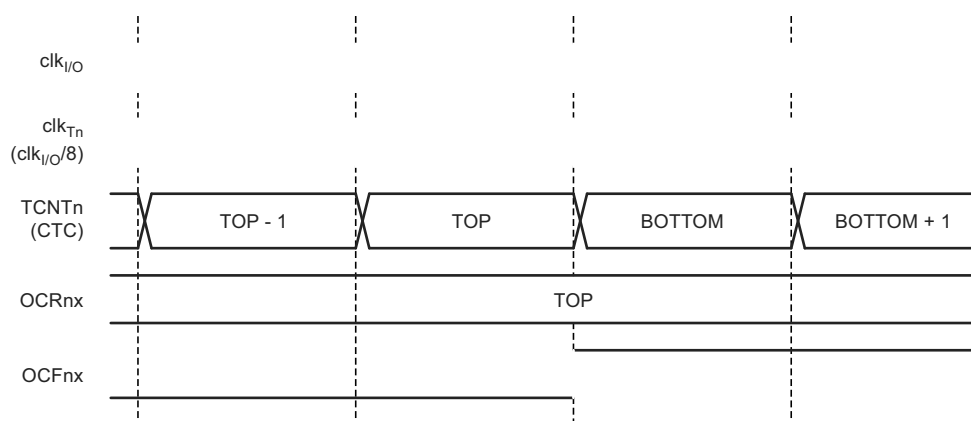


Figure 5-63 shows the setting of OCF2A and the clearing of TCNT2 in CTC mode.

Figure 5-63. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk_I/O}/8$)



5.17.9 Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the timer registers TCNT2, OCR2x, and TCCR2x might be corrupted. A safe procedure for switching clock source is:
 - a. Disable the Timer/Counter2 interrupts by clearing OCIE2x and TOIE2.
 - b. Select clock source by setting AS2 as appropriate.
 - c. Write new values to TCNT2, OCR2x, and TCCR2x.
 - d. To switch to asynchronous operation: Wait for TCN2xUB, OCR2xUB, and TCR2xUB.
 - e. Clear the Timer/Counter2 interrupt flags.
 - f. Enable interrupts, if needed.
- The CPU main clock frequency must be more than four times the oscillator frequency.
- When writing to one of the registers TCNT2, OCR2x, or TCCR2x, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the five mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2x write in progress. To detect that a transfer to the destination register has taken place, the asynchronous status register – ASSR has been implemented.

5.19.10.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
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	1	1	0	

- **Bits 7:6 – UMSELn1:0 USART Mode Select**

These bits select the mode of operation of the USARTn as shown in Table 5-72.

Table 5-72. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM) ⁽¹⁾

Note: 1. See Section 5.20 “USART in SPI Mode” on page 187 for full description of the master SPI mode (MSPIM) operation.

- **Bits 5:4 – UPMn1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and compare it to the UPMn setting. If a mismatch is detected, the UPEn Flag in UCSRnA will be set.

Table 5-73. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, even parity
1	1	Enabled, odd parity

- **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the transmitter. The receiver ignores this setting.

Table 5-74. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (character SiZe) in a frame the receiver and transmitter use.

Table 5-79. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	$f_{osc} = 8.0000\text{MHz}$				$f_{osc} = 11.0592\text{MHz}$				$f_{osc} = 14.7456\text{MHz}$			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	—	—	2	-7.8%	1	-7.8%	3	-7.8%
1M	—	—	0	0.0%	—	—	—	—	0	-7.8%	1	-7.8%
Max. ⁽¹⁾	0.5Mbps		1Mbps		691.2kbps		1.3824Mbps		921.6kbps		1.8432Mbps	

Note: 1. UBRRn = 0, Error = 0.0%

Table 5-80. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

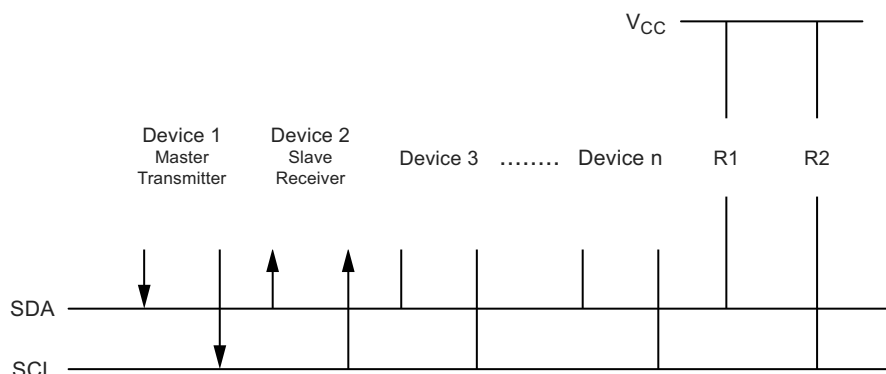
Baud Rate (bps)	$f_{osc} = 16.0000\text{MHz}$			
	U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	-0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	-0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	-0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	-3.5%	16	2.1%
230.4k	3	8.5%	8	-3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max. ⁽¹⁾	1Mbps		2Mbps	

Note: 1. UBRRn = 0, Error = 0.0%

5.21.7.1 Master Transmitter Mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 5-87). In order to enter a master mode, a START condition must be transmitted. The format of the following address packet determines whether master transmitter or master receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 5-87. Data Transfer in Master Transmitter Mode



A START condition is sent by writing the following value to TWCR:

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

TWEN must be set to enable the 2-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be written to one to clear the TWINT flag. The TWI will then test the 2-wire serial bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT flag is set by hardware, and the status code in TWSR will be 0x08 (see Table 5-87). In order to enter MT mode, SLA+W must be transmitted. This is done by writing SLA+W to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

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

When SLA+W have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in master mode are 0x18, 0x20, or 0x38. The appropriate action to be taken for each of these status codes is detailed in Table 5-87.

When SLA+W has been successfully transmitted, a data packet should be transmitted. This is done by writing the data byte to TWDR. TWDR must only be written when TWINT is high. If not, the access will be discarded, and the write collision bit (TWWC) will be set in the TWCR register. After updating TWDR, the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

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

This scheme is repeated until the last byte has been sent and the transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

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

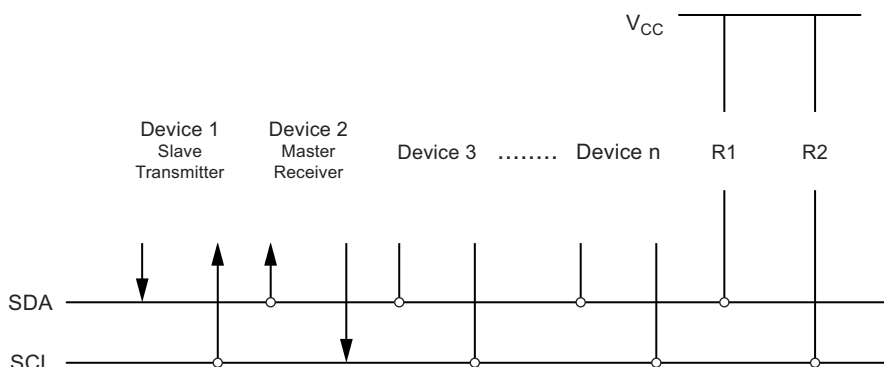
A REPEATED START condition is generated by writing the following value to TWCR:

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

5.21.7.4 Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 5-93). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 5-93. 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 5-90. 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 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.

5.26.8.6 Prevent Reading the RWW Section During Self-programming

During self-programming (either page erase or page write), the RWW section is always blocked for reading. The user software itself must prevent that this section is addressed during the self programming operation. The RWWSB in the SPMCSR will be set as long as the RWW section is busy. During self-programming the interrupt vector table should be moved to the BLS as described in Section 5.10.8 “Watchdog Timer” on page 68, or the interrupts must be disabled. Before addressing the RWW section after the programming is completed, the user software must clear the RWWSB by writing the RWWSRE. See Section 5.26.8.13 “Simple Assembly Code Example for a Boot Loader” on page 258 for an example.

5.26.8.7 Setting the Boot Loader Lock Bits by SPM

To set the boot loader lock bits and general lock bits, write the desired data to R0, write “X0001001” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR.

Bit	7	6	5	4	3	2	1	0
R0	1	1	BLB12	BLB11	BLB02	BLB01	LB2	LB1

See Table 5-103 and Table 5-104 for how the different settings of the boot loader bits affect the flash access.

If bits 5..0 in R0 are cleared (zero), the corresponding lock bit will be programmed if an SPM instruction is executed within four cycles after BLBSET and SELFPRGEN are set in SPMCSR. The Z-pointer is don't care during this operation, but for future compatibility it is recommended to load the Z-pointer with 0x0001 (same as used for reading the IO_{ck} bits). For future compatibility it is also recommended to set bits 7 and 6 in R0 to “1” when writing the lock bits. When programming the lock bits the entire flash can be read during the operation.

5.26.8.8 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to flash. Reading the fuses and lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEPE) in the EECR register and verifies that the bit is cleared before writing to the SPMCSR register.

5.26.8.9 Reading the Fuse and Lock Bits from Software

It is possible to read both the fuse and lock bits from software. To read the lock bits, load the Z-pointer with 0x0001 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits are set in SPMCSR, the value of the lock bits will be loaded in the destination register. The BLBSET and SELFPRGEN bits will auto-clear upon completion of reading the lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When BLBSET and SELFPRGEN are cleared, LPM will work as described in the instruction set manual.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	BLB12	BLB11	BLB02	BLB01	LB2	LB1

The algorithm for reading the fuse low byte is similar to the one described above for reading the lock bits. To read the fuse low byte, load the Z-pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the fuse low byte (FLB) will be loaded in the destination register as shown below. Refer to Table 5-115 on page 264 for a detailed description and mapping of the fuse low byte.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Similarly, when reading the fuse high byte, load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the fuse high byte (FHB) will be loaded in the destination register as shown below. Refer to Table 5-117 on page 265 for detailed description and mapping of the fuse high byte.

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

5.28 Electrical Characteristics

All DC/AC characteristics contained in this datasheet are based on characterization of ATmega328P AVR® microcontroller manufactured in an automotive process technology.

5.28.1 DC Characteristics

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

Parameter	Condition	Symbol	Min.	Typ.	Max.	Unit
Input low voltage, except XTAL1 and RESET pin	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	V_{IL}	-0.5		$0.3V_{CC}^{(1)}$	V
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 low voltage, XTAL1 pin	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	V_{IL1}	-0.5		$0.1V_{CC}^{(1)}$	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 low voltage, RESET pin	$V_{CC} = 2.7\text{V} - 5.5\text{V}$	V_{IL2}	-0.5		$0.1V_{CC}^{(1)}$	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 ⁽³⁾	$I_{OL} = 20\text{mA}$, $V_{CC} = 5\text{V}$ $I_{OL} = 5\text{mA}$, $V_{CC} = 3\text{V}$	V_{OL}			0.8 0.5	V
Output high voltage ⁽⁴⁾	$I_{OH} = -20\text{mA}$, $V_{CC} = 5\text{V}$ $I_{OH} = -10\text{mA}$, $V_{CC} = 3\text{V}$	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}			1	μA
Input leakage current I/O pin	$V_{CC} = 5.5\text{V}$, pin high (absolute value)	I_{IH}			1	μA
Reset pull-up resistor		R_{RST}	30		60	$\text{k}\Omega$
I/O pin pull-up resistor		R_{PU}	20		50	$\text{k}\Omega$
Analog comparator input offset voltage	$0.4\text{V} < V_{in} < V_{CC} - 0.5$ (absolute value)	V_{ACIO}		10	40	mV
Analog comparator input leakage current	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$	I_{ACLK}	-50		50	nA

- 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 $V_{CC} = 5\text{V}$, 10 mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
ATmega328P:
1] The sum of all I_{OL} , for ports C0 - C5, should not exceed 100mA.
2] The sum of all I_{OL} , for ports B0 - B5, D5 - D7, XTAL1, XTAL2 should not exceed 100mA.
3] The sum of all I_{OL} , for ports D0 - D4, should not exceed 100mA.
If I_{OL} exceeds the test condition, V_{OL} 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:
ATmega328P:
1] The sum of all I_{OH} , for ports C0 - C5, D0 - D4, should not exceed 150mA.
2] The sum of all I_{OH} , for ports B0 - B5, D5 - D7, XTAL1, XTAL2 should not exceed 150mA.
If I_{OH} exceeds the test condition, V_{OH} may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

5.29.1.3 Pin Pull-Up

Figure 5-139. I/O Pin Pull-up Resistor Current versus Input Voltage ($V_{CC} = 5V$)

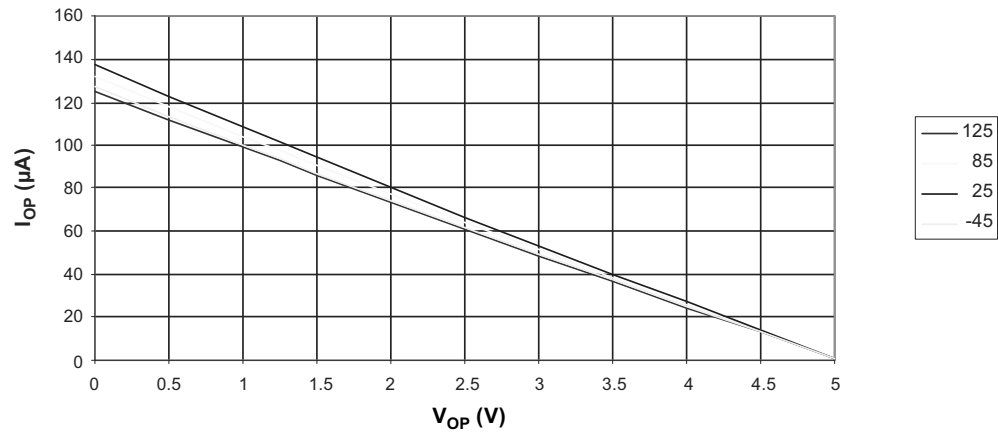
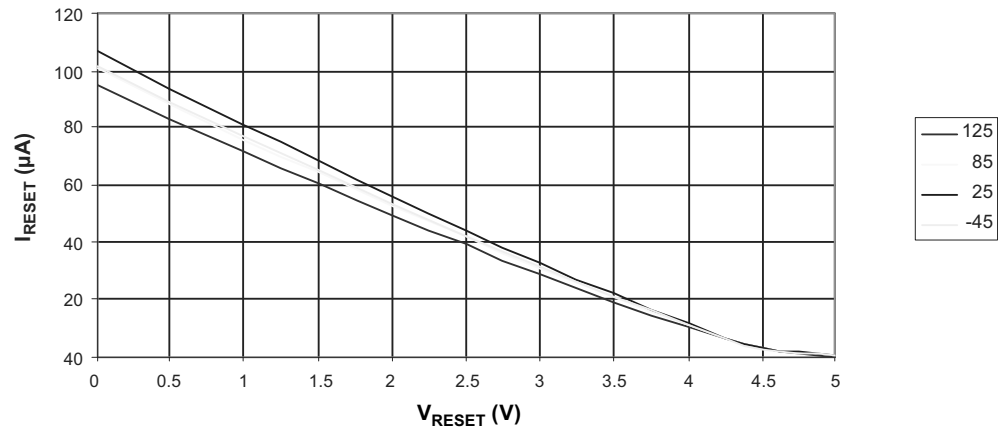


Figure 5-140. Reset Pull-up Resistor Current versus Reset Pin Voltage ($V_{CC} = 5V$)



5.29.1.4 Pin Driver Strength

Figure 5-141. I/O Pin Output Voltage versus Sink Current ($V_{CC} = 3V$)

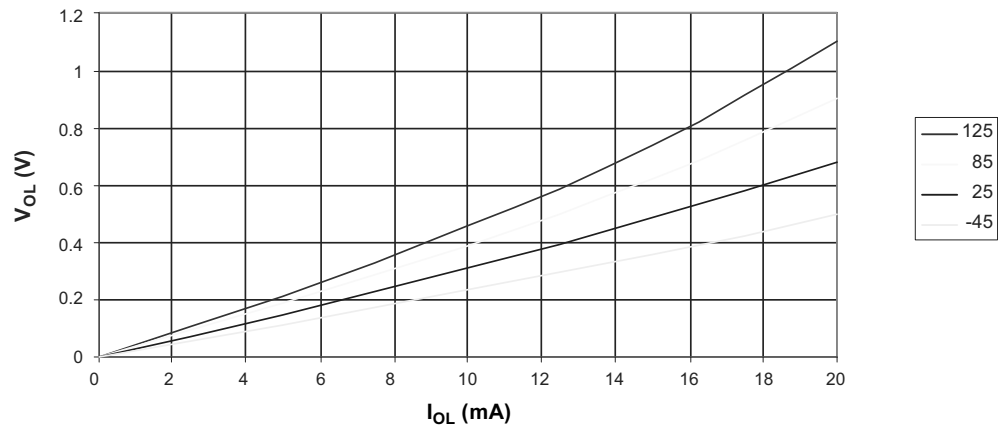
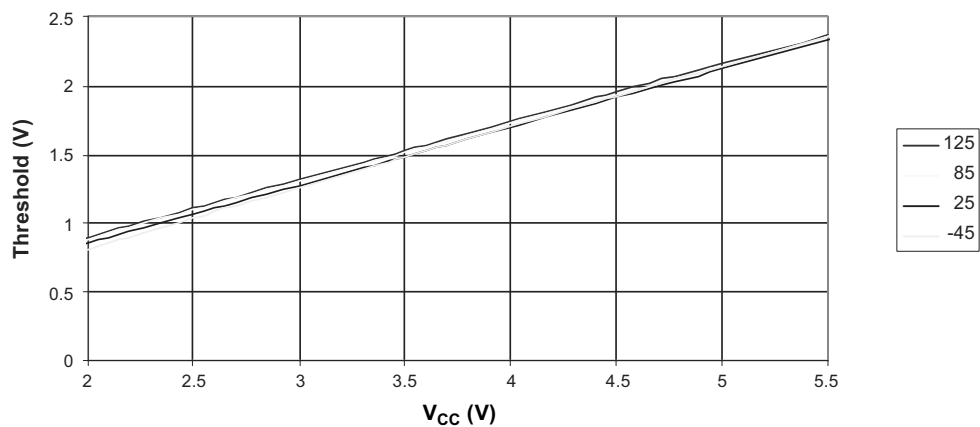


Figure 5-148. Reset Input Threshold Voltage versus V_{CC} (V_{IL} , I/O Pin read as '0')



5.29.1.6 BOD Threshold

Figure 5-149. BOD Thresholds versus Temperature (BODLEVEL is 2.7V)

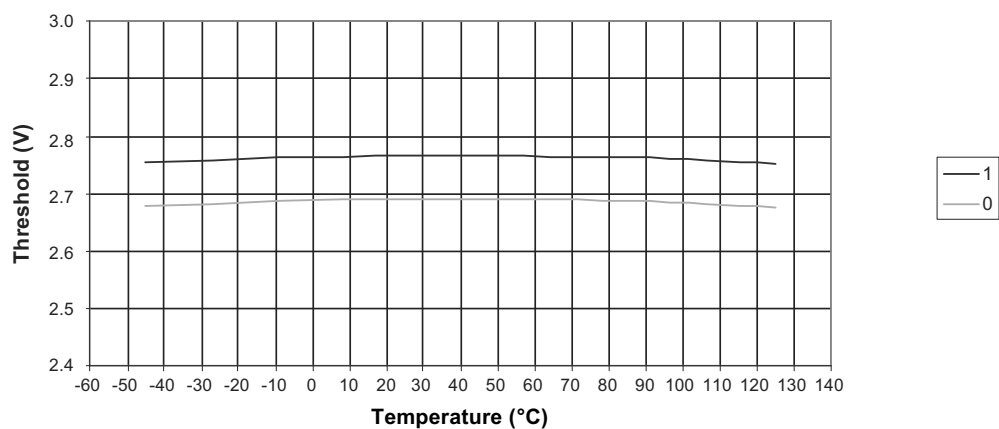


Figure 5-150. BOD Thresholds versus Temperature (BODLEVEL is 4.3V)

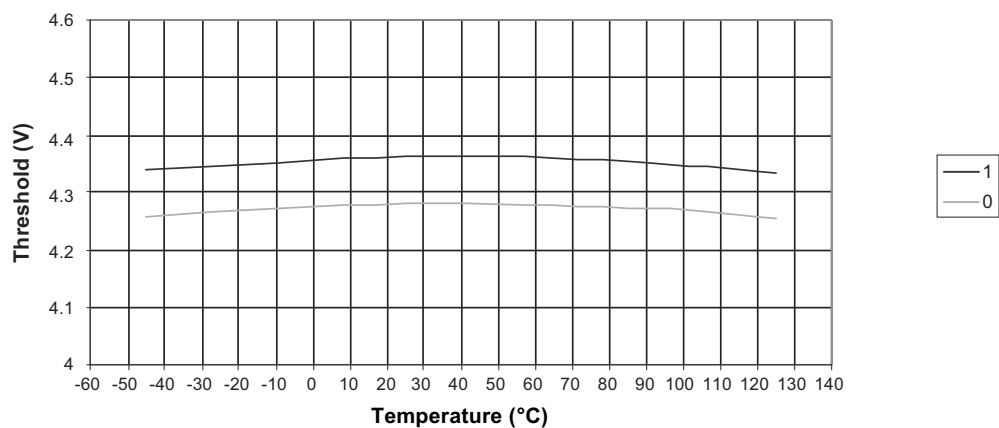
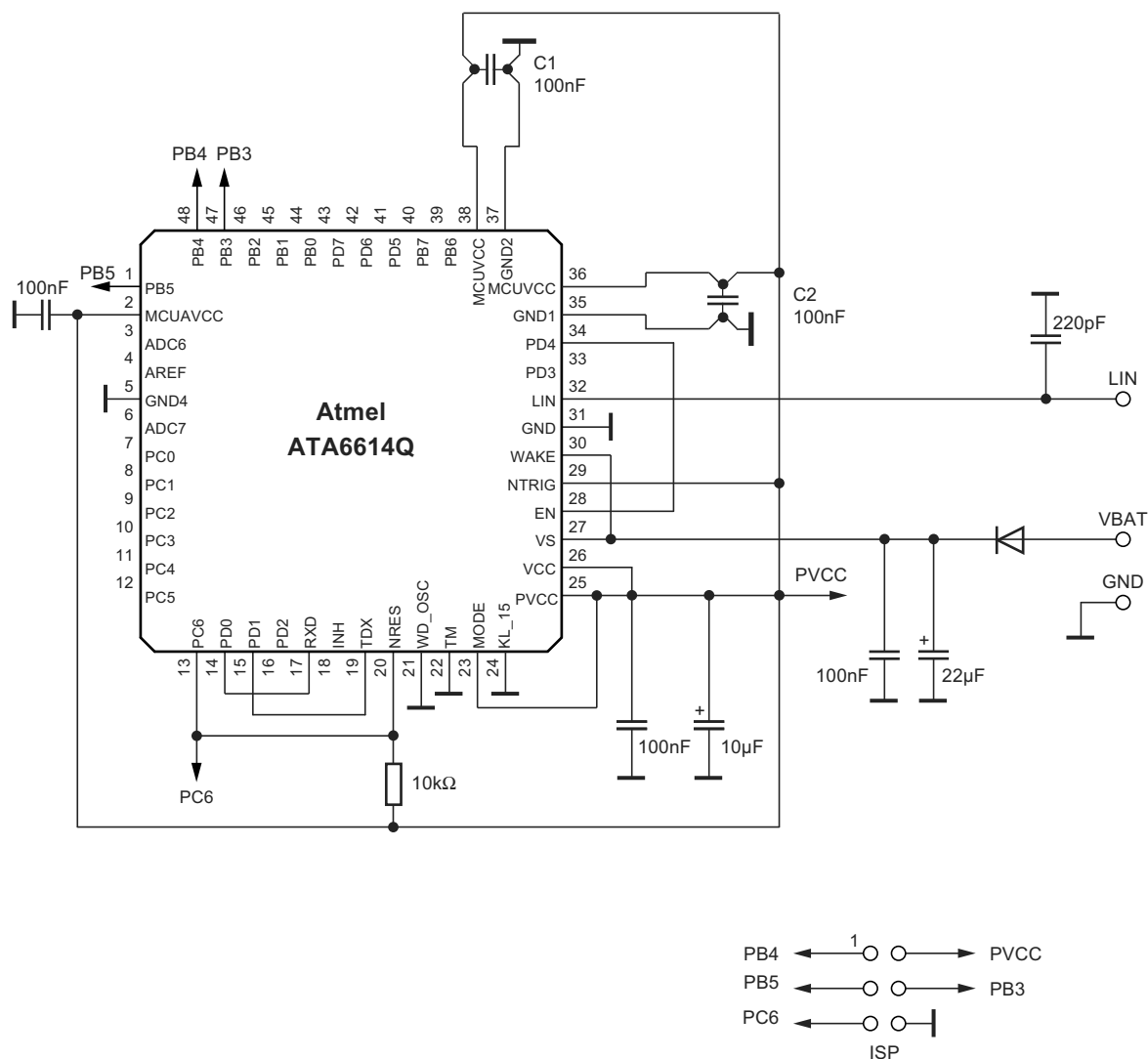


Figure 6-4. LIN Slave Application with Minimum External Components



Note: All open pins of the SiP can be used for application-specific purposes.
 AVR®: Internal clock, no ADC application, TXD, RXD, NRES and EN connected for LIN slave. The connection between the LIN-SBC and the AVR requires the software being programmed correspondingly.
 SBC: LIN slave operation with 5V regulator, no watchdog, no local wake-up.
 RF emissions: Best results for RF emissions will be achieved by connecting the blocking capacitors of the microcontroller supply (C1 and C2) between the microcontroller pins and the GND/PVCC line. See also Figure 6-4.