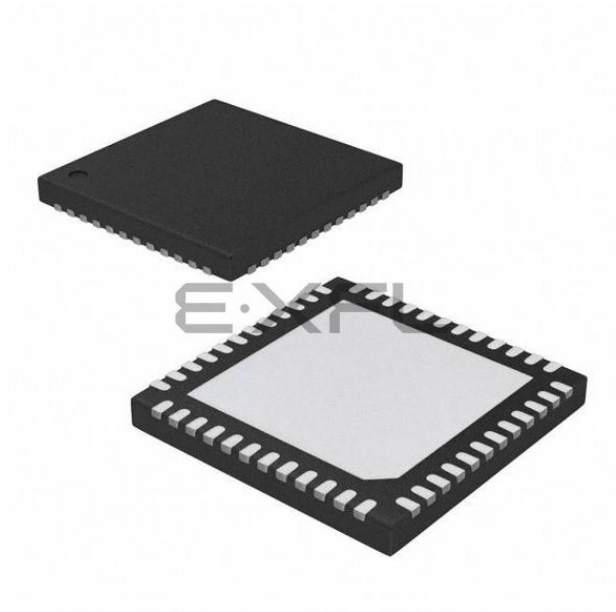


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K 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/ata6613c-plqw-1

1. Pin Configuration

Figure 1-1. Pinning QFN48, 7mm × 7mm

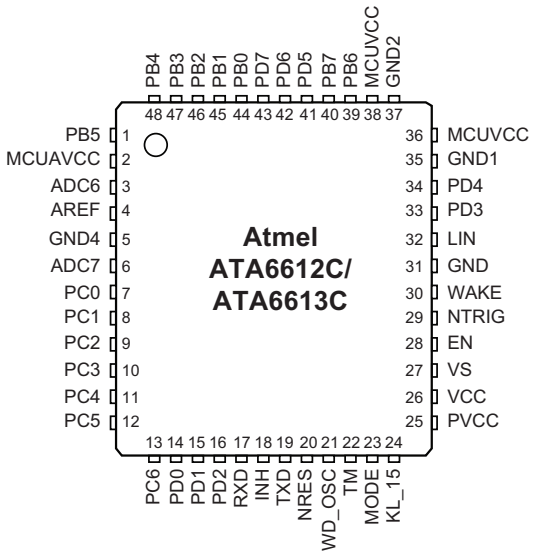


Table 1-1. Pin Description

Pin	Symbol	Function
1	PB5	Port B 5 I/O line (SCK / PCINT5)
2	MCUAVCC	Microcontroller ADC-unit supply voltage (referred to as AVCC pin in Section 5. “Microcontroller Block” on page 27 and Section 6. “2-wire Serial Interface Characteristics” on page 276)
3	ADC6	ADC input channel 6
4	AREF	Analog reference voltage input
5	GND4	Ground
6	ADC7	ADC input channel 7
7	PC0	Port C 0 I/O line (ADC0/PCINT8)
8	PC1	Port C 1 I/O line (ADC1/PCINT9)
9	PC2	Port C 2 I/O line (ADC2/PCINT10)
10	PC3	Port C 3 I/O line (ADC3/PCINT11)
11	PC4	Port C 4 I/O line (ADC4/SDA/PCINT12)
12	PC5	Port C 5 I/O line (ADC5/SCL/PCINT13)
13	PC6	Port C 6 I/O line (RESET/PCINT14)
14	PD0	Port D 0 I/O line (RXD/PCINT16)
15	PD1	Port D 1 I/O line (TXD/PCINT17)
16	PD2	Port D 2 I/O line (INT0/PCINT18)
17(1)	RXD	Receive data output
18(1)	INH	High side switch output for controlling an external voltage regulator
19(1)	TXD	Transmit data input / active low output after a local wake up request

Note: 1. This identifies the pins of the LIN SBC Atmel ATA6624

3.3.20.2 Silent Mode

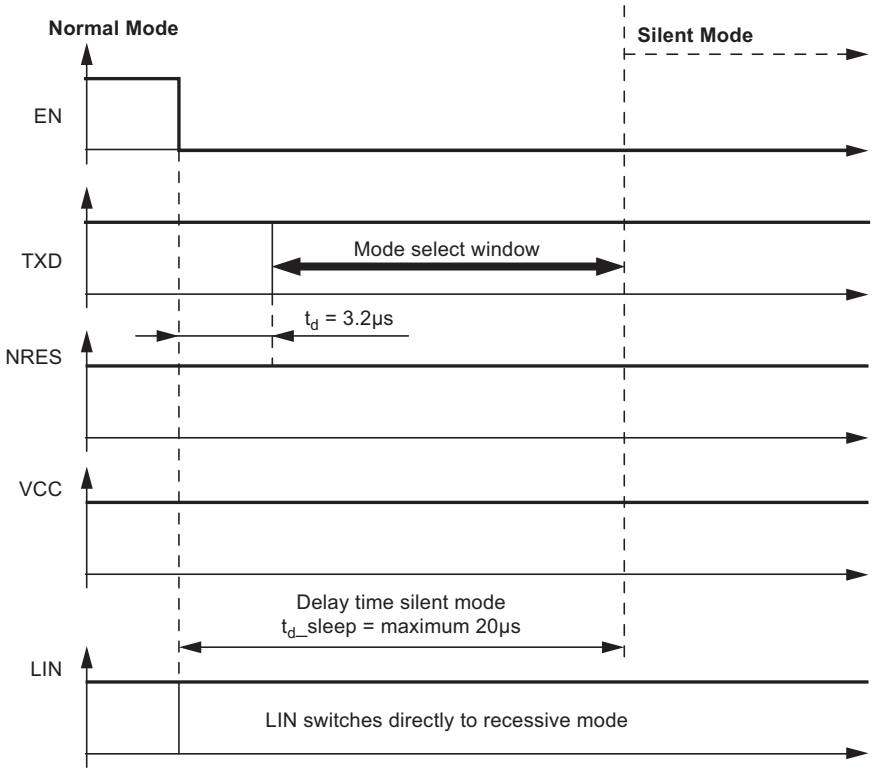
A falling edge at EN when TXD is high switches the IC into silent mode. The TXD signal has to be logic high during the mode select window (see Figure 3-3). The transmission path is disabled in silent mode. The overall supply current from V_{Batt} is a combination of the I_{VSsi} 57µA plus the VCC regulator output current I_{VCC}.

The internal slave termination between the LIN pin and the VS pin is disabled in silent mode, only a weak pull-up current (typically 10µA) between the LIN pin and the VS pin is present. silent mode can be activated independently from the actual level on the LIN, WAKE, or KL_15 pins.

If an under voltage condition occurs, NRES is switched to low, and the IC changes its state to fail-safe mode.

A voltage lower than the LIN pre_wake detection VLINL at the LIN pin activates the internal LIN receiver and switches on the internal slave termination between the LIN pin and the V_S pin.

Figure 3-3. Switch to Silent Mode



4. Electrical Characteristics (Continued)

5V < V_S < 27V, –40°C < T_{case} < 125°C, –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*
8.8	LIN current limitation $V_{BUS} = V_{Batt\_max}$		LIN	I_{BUS_LIM}	40	120	200	mA	A
8.9	Input leakage current at the receiver including pull-up resistor as specified	Input leakage current Driver off V_{BUS} = 0V V_{Batt} = 12V	LIN	$I_{BUS\_PAS\_dom}$	–1	–0.35		mA	A
8.10	Leakage current LIN recessive	Driver off 8V < V_{Batt} < 18V 8V < V_{BUS} < 18V $V_{BUS} \geq V_{Batt}$	LIN	$I_{BUS\_PAS\_rec}$		10	20	µA	A
8.11	Leakage current when control unit disconnected from ground. Loss of local ground must not affect communication in the residual network.	$GND_{Device} = V_S$ V_{Batt} = 12V 0V < V_{BUS} < 18V	LIN	$I_{BUS\_NO\_gnd}$	–10	+0.5	+10	µA	A
8.12	Leakage current at a disconnected battery. Node has to sustain the current that can flow under this condition. Bus must remain operational under this condition.	V_{Batt} disconnected V_{SUP_Device} = GND 0V < V_{BUS} < 18V	LIN	$I_{BUS\_NO\_bat}$		0.1	2	µA	A
9	LIN Bus Receiver								
9.1	Center of receiver threshold	V_{BUS_CNT} = (V_{th_dom} + V_{th_rec})/2	LIN	V_{BUS_CNT}	0.475 × V_S	0.5 × V_S	0.525 × V_S	V	A
9.2	Receiver dominant state	V_{EN} = 5V	LIN	V_{BUSdom}			0.4 × V_S	V	A
9.3	Receiver recessive state	V_{EN} = 5V	LIN	V_{BUSrec}	0.6 × V_S			V	A
9.4	Receiver input hysteresis	$V_{hys} = V_{th\_rec} – V_{th\_dom}$	LIN	V_{BUShys}	0.028 × V_S	0.1 × V_S	0.175 × V_S	V	A
9.5	Pre_Wake detection LIN High-level input voltage		LIN	V_{LINH}	V_S – 2V		V_S + 0.3V	V	A
9.6	Pre_Wake detection LIN Low-level input voltage	Activates the LIN receiver	LIN	V_{LINL}	–27		V_S – 3.3V	V	A

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

5.4.4 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® 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 ⊕ 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.

When applying an external clock, it is required to avoid sudden changes in the applied clock frequency to ensure stable operation of the MCU. A variation in frequency of more than 2% from one clock cycle to the next can lead to unpredictable behavior. If changes of more than 2% is required, ensure that the MCU is kept in reset during the changes.

Note that the system clock prescaler can be used to implement run-time changes of the internal clock frequency while still ensuring stable operation. Refer to Section 5.6.11 “System Clock Prescaler” on page 54 for details.

5.6.9 Clock Output Buffer

The device can output the system clock on the CLKO pin. To enable the output, the CKOUT Fuse has to be programmed. This mode is suitable when the chip clock is used to drive other circuits on the system. The clock also will be output during reset, and the normal operation of I/O pin will be overridden when the fuse is programmed. Any clock source, including the internal RC oscillator, can be selected when the clock is output on CLKO. If the system clock prescaler is used, it is the divided system clock that is output.

5.6.10 Timer/Counter Oscillator

The device can operate its Timer/Counter2 from an external 32.768kHz watch crystal or a external clock source. The Timer/Counter oscillator pins (TOSC1 and TOSC2) are shared with XTAL1 and XTAL2. This means that the Timer/Counter oscillator can only be used when an internal RC oscillator is selected as system clock source. See Figure 5-12 on page 48 for crystal connection.

Applying an external clock source to TOSC1 requires EXTCLK in the ASSR register written to logic one. See Section 5.15.9 “Asynchronous Operation of the Timer/Counter” on page 153 for further description on selecting external clock as input instead of a 32kHz crystal.

5.6.11 System Clock Prescaler

The Atmel® ATA6612C/ATA6613C has a system clock prescaler, and the system clock can be divided by setting the Section 5.6.11.1 “Clock Prescale Register – CLKPR” on page 54. This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low. This can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals. $clk_{I/O}$, clk_{ADC}, clk_{CPU}, and clk_{FLASH} are divided by a factor as shown in Table 5-20 on page 62.

When switching between prescaler settings, the system clock prescaler ensures that no glitches occurs in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting. The ripple counter that implements the prescaler runs at the frequency of the undivided clock, which may be faster than the CPU's clock frequency. Hence, it is not possible to determine the state of the prescaler - even if it were readable, and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. From the time the CLKPS values are written, it takes between T1 + T2 and
$T1 + 2 \times T2$
before the new clock frequency is active. In this interval, 2 active clock edges are produced. Here, T1 is the previous clock period, and T2 is the period corresponding to the new prescaler setting.

To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS bits:

1. Write the clock prescaler change enable (CLKPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE.

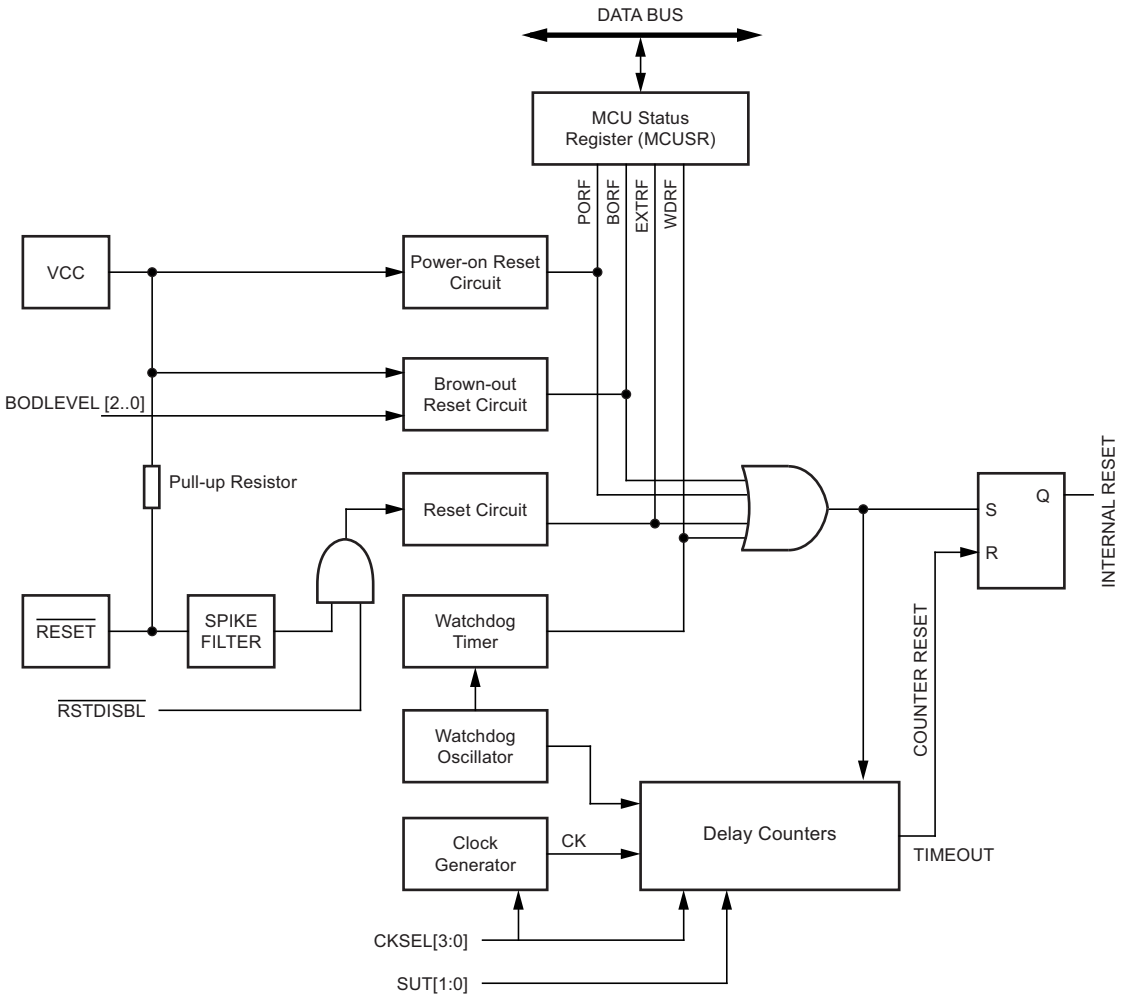
Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

5.6.11.1 Clock Prescale Register – CLKPR

Bit	7	6	5	4	3	2	1	0	
	CLKPCE	**–**	**–**	**–**	**CLKPS3**	**CLKPS2**	**CLKPS1**	**CLKPS0**	**CLKPR**
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7 – CLKPCE: Clock Prescaler Change Enable**
  The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

Figure 5-15. Reset Logic



5.8.3 Power-on Reset

A power-on reset (POR) pulse is generated by an on-chip detection circuit. The detection level is defined in Table 5-20 on page 62. 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.

Table 5-29 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 5-29. Reset and Interrupt Vectors Placement in ATA6613C(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot reset address + 0x0002
0	0	Boot reset address	0x001
0	1	Boot reset address	Boot reset address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 5-108 on page 251. For the BOOTRST fuse “1” means unprogrammed while “0” means programmed.

The most typical and general program setup for the reset and interrupt vector addresses in ATA6613C is:

```
Address      Labels Code                 Comments
0x0000              rjmp   RESET        ; Reset Handler
0x0002              rjmp   EXT_INT0     ; IRQ0 Handler
0x0004              rjmp   EXT_INT1     ; IRQ1 Handler
0x0006              rjmp   PCINT0       ; PCINT0 Handler
0x0008              rjmp   PCINT1       ; PCINT1 Handler
0x000A              rjmp   PCINT2       ; PCINT2 Handler
0x000C              rjmp   WDT          ; Watchdog Timer Handler
0x000E              rjmp   TIM2_COMPA   ; Timer2 Compare A Handler
0x0010              rjmp   TIM2_COMPB   ; Timer2 Compare B Handler
0x0012              rjmp   TIM2_OVF     ; Timer2 Overflow Handler
0x0014              rjmp   TIM1_CAPT    ; Timer1 Capture Handler
0x0016              rjmp   TIM1_COMPA   ; Timer1 Compare A Handler
0x0018              rjmp   TIM1_COMPB   ; Timer1 Compare B Handler
0x001A              rjmp   TIM1_OVF     ; Timer1 Overflow Handler
0x001C              rjmp   TIM0_COMPA   ; Timer0 Compare A Handler
0x001E              rjmp   TIM0_COMPB   ; Timer0 Compare B Handler
0x0020              rjmp   TIM0_OVF     ; Timer0 Overflow Handler
0x0022              rjmp   SPI_STC      ; SPI Transfer Complete Handler
0x0024              rjmp   USART_RXC    ; USART, RX Complete Handler
0x0026              rjmp   USART_UDRE   ; USART, UDR Empty Handler
0x0028              rjmp   USART_TXC    ; USART, TX Complete Handler
0x002A              rjmp   ADC          ; ADC Conversion Complete Handler
0x002C              rjmp   EE_RDY       ; EEPROM Ready Handler
0x002E              rjmp   ANA_COMP     ; Analog Comparator Handler
0x0030              rjmp   TWI          ; 2-wire Serial Interface Handler
0x0032              rjmp   SPM_RDY      ; Store Program Memory Ready Handler
;
0x0033       RESET: ldi    r16, high(RAMEND); Main program start
0x0034              out    SPH,r16      ; Set Stack Pointer to top of RAM
0x0035              ldi    r16, low(RAMEND)
0x0036              out    SPL,r16
0x0037              sei                 ; Enable interrupts
0x0038              <instr> xxx
...    ...    ...    ...
```

Table 5-40. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/OC2B/INT1/PCINT19	PD2/INT0/PCINT18	PD1/TXD/PCINT17	PD0/RXD/PCINT16
PUOE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 × $\overline{PUD}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B ENABLE	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	INT1 ENABLE + PCINT19 × PCIE2	INT0 ENABLE + PCINT18 × PCIE1	PCINT17 × PCIE2	PCINT16 × PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT RXD
AIO	–	–	–	–

5.10.4 Register Description for I/O Ports

5.10.4.1 The Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
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	

5.10.4.2 The Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
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	

5.10.4.3 The Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

5.10.4.4 The Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

5.10.4.5 The Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figure 5-35 shows the same timing data, but with the prescaler enabled.

Figure 5-35. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

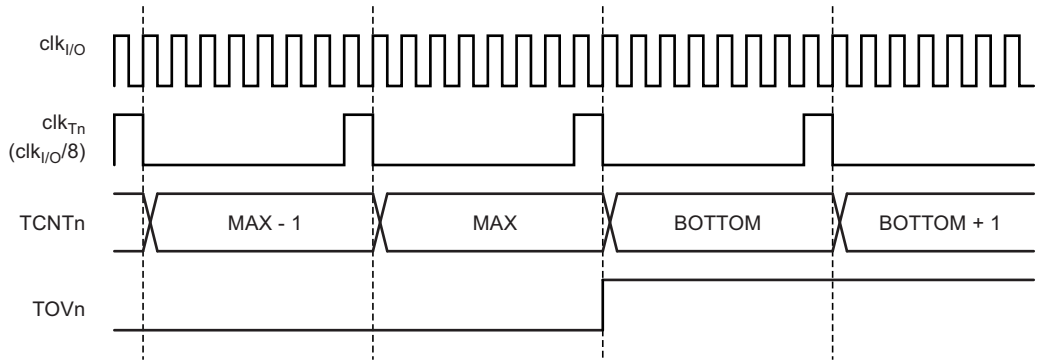


Figure 5-36 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 5-36. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk\_I/O}/8$)

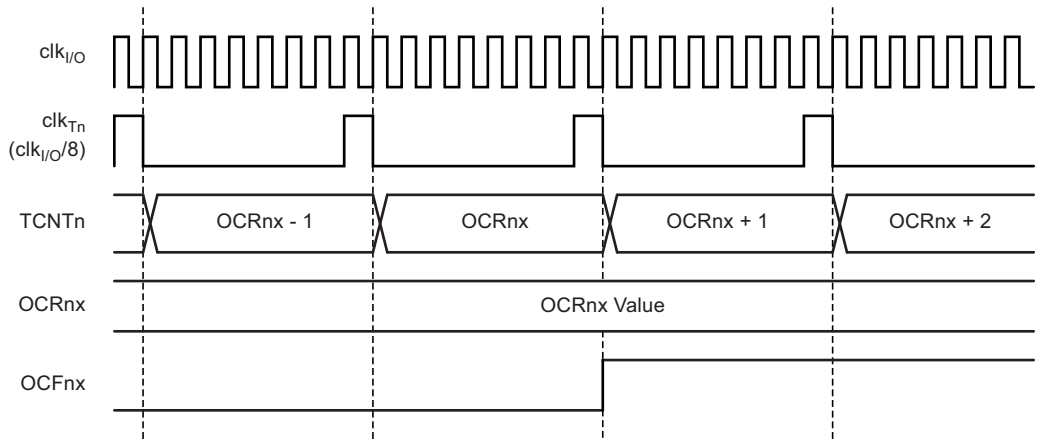


Figure 5-37 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 5-37. Timer/Counter Timing Diagram, Clear Timer on Compare Match Mode, with Prescaler ($f_{clk\_I/O}/8$)

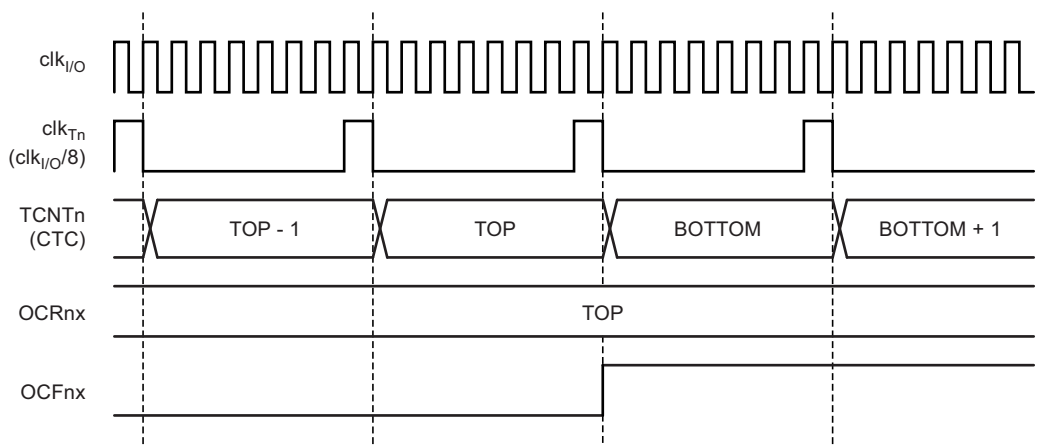
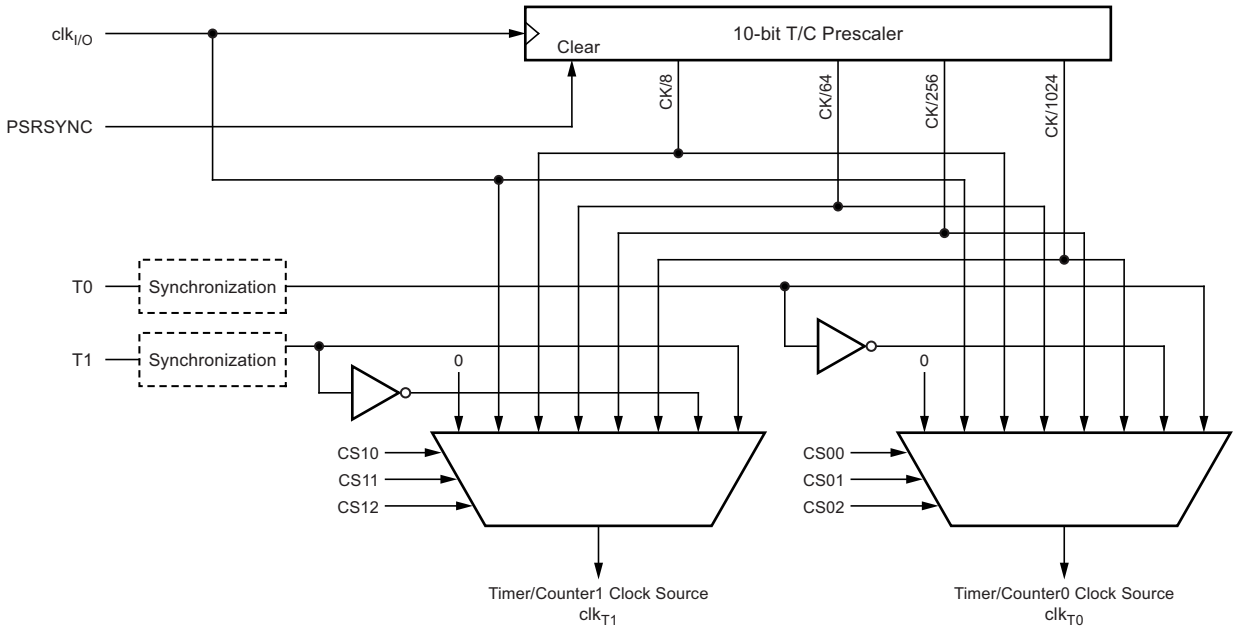


Figure 5-39. Prescaler for Timer/Counter0 and Timer/Counter1(1)



Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 5-38 on page 112.

5.13.4 General Timer/Counter Control Register – GTCCR

Bit	7	6	5	4	3	2	1	0	
	TSM	–	–	–	–	–	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TSM: Timer/Counter Synchronization Mode**

Writing the TSM bit to one activates the Timer/Counter synchronization mode. In this mode, the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware, and the Timer/Counters start counting simultaneously.

- **Bit 0 – PSRSYNC: Prescaler Reset**

When this bit is one, Timer/Counter1 and Timer/Counter0 prescaler will be reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers.

5.14 16-bit Timer/Counter1 with PWM

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement. The main features are:

- True 16-bit design (i.e., allows 16-bit PWM)
- Two independent output compare units
- Double buffered output compare registers
- One input capture unit
- Input capture noise canceler
- Clear timer on compare match (auto reload)
- Glitch-free, phase correct pulse width modulator (PWM)

The following code examples show how to do an atomic read of the TCNT1 register contents. Reading any of the OCR1A/B or ICR1 registers can be done by using the same principle.

Assembly Code Example[1]

```
TIM16_ReadTCNT1:
        ; Save global interrupt flag
        in      r18,SREG
        ; Disable interrupts
        cli
        ; Read TCNT1 into r17:r16
        in      r16,TCNT1L
        in      r17,TCNT1H
        ; Restore global interrupt flag
        out     SREG,r18
        ret
```

C Code Example[1]

```
unsigned int TIM16_ReadTCNT1( void )
{
        unsigned char sreg;
        unsigned int i;
        /* Save global interrupt flag */
        sreg = SREG;
        /* Disable interrupts */
        _CLI();
        /* Read TCNT1 into i */
        i = TCNT1;
        /* Restore global interrupt flag */
        SREG = sreg;
        return i;
}
```

Note: 1. The example code assumes that the part specific header file is included.
For I/O registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

5.17.9 USART Register Description

5.17.9.1 USART I/O Data Register n– UDRn

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn (Read)
	TXB[7:0]								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.

5.17.9.2 USART Control and Status Register n A – UCSRnA

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	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.

- **Bit 3 – DORn: Data OverRun**

This bit is set if a data overrun condition is detected. A data overrun occurs when the receive buffer is full (two characters), it is a new character waiting in the receive shift register, and a new start bit is detected. This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 1 - UCPHAn: Clock Phase**

The UCPHAn bit setting determine if data is sampled on the leasing edge (first) or tailing (last) edge of XCKn. Refer to the section Section 5.18.3 “SPI Data Modes and Timing” on page 187 for details.

- **Bit 0 - UCPOLn: Clock Polarity**

The UCPOLn bit sets the polarity of the XCKn clock. The combination of the UCPOLn and UCPHAn bit settings determine the timing of the data transfer. Refer to the section Section 5.18.3 “SPI Data Modes and Timing” on page 187 for details.

USART MSPIM baud rate registers - UBRRnL and UBRRnH

The function and bit description of the baud rate registers in MSPI mode is identical to normal USART operation (see Section 5.17.9.5 “USART Baud Rate Registers – UBRRnL and UBRRnH” on page 183).

5.18.7 AVR USART MSPIM versus AVR SPI

The USART in MSPIM mode is fully compatible with the AVR® SPI regarding:

- Master mode timing diagram.
- The UCPOLn bit functionality is identical to the SPI CPOL bit.
- The UCPHAn bit functionality is identical to the SPI CPHA bit.
- The UDORDn bit functionality is identical to the SPI DORD bit.

However, since the USART in MSPIM mode reuses the USART resources, the use of the USART in MSPIM mode is somewhat different compared to the SPI. In addition to differences of the control register bits, and that only master operation is supported by the USART in MSPIM mode, the following features differ between the two modules:

- The USART in MSPIM mode includes (double) buffering of the transmitter. The SPI has no buffer.
- The USART in MSPIM mode receiver includes an additional buffer level.
- The SPI WCOL (Write Collision) bit is not included in USART in MSPIM mode.
- The SPI double speed mode (SPI2X) bit is not included. However, the same effect is achieved by setting UBRRn accordingly.
- Interrupt timing is not compatible.
- Pin control differs due to the master only operation of the USART in MSPIM mode.

A comparison of the USART in MSPIM mode and the SPI pins is shown in Table 5-87.

Table 5-87. Comparison of USART in MSPIM Mode and SPI Pins.

USART_MSPIM	SPI	Comment
TxDn	MOSI	Master out only
RxDn	MISO	Master in only
XCKn	SCK	(Functionally identical)
(N/A)	$\overline{SS}$	Not supported by USART in MSPIM

- **Bit 1 – Res: Reserved Bit**

This bit is a reserved bit and will always read as zero.

- **Bit 0 – TWIE: TWI Interrupt Enable**

When this bit is written to one, and the I-bit in SREG is set, the TWI interrupt request will be activated for as long as the TWINT flag is high.

5.19.6.3 TWI Status Register – TWSR

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7..3 – TWS: TWI Status**

These 5 bits reflect the status of the TWI logic and the 2-wire serial bus. The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the status bits. This makes status checking independent of prescaler setting. This approach is used in this datasheet, unless otherwise noted.

- **Bit 2 – Res: Reserved Bit**

This bit is reserved and will always read as zero.

- **Bits 1..0 – TWPS: TWI Prescaler Bits**

These bits can be read and written, and control the bit rate prescaler.

Table 5-89. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

To calculate bit rates, see Section 5.19.5.2 “Bit Rate Generator Unit” on page 198. The value of TWPS1..0 is used in the equation.

5.19.6.4 TWI Data Register – TWDR

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

In transmit mode, TWDR contains the next byte to be transmitted. In receive mode, the TWDR contains the last byte received. It is writable while the TWI is not in the process of shifting a byte. This occurs when the TWI interrupt flag (TWINT) is set by hardware. Note that the data register cannot be initialized by the user before the first interrupt occurs. The data in TWDR remains stable as long as TWINT is set. While data is shifted out, data on the bus is simultaneously shifted in. TWDR always contains the last byte present on the bus, except after a wake up from a sleep mode by the TWI interrupt. In this case, the contents of TWDR is undefined. In the case of a lost bus arbitration, no data is lost in the transition from master to slave. Handling of the ACK bit is controlled automatically by the TWI logic, the CPU cannot access the ACK bit directly.

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

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

5.21.5 ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC noise reduction and Idle mode. To make use of this feature, the following procedure should be used:

a. Make sure that the ADC is enabled and is not busy converting. Single conversion mode must be selected and the ADC conversion complete interrupt must be enabled.

b. Enter ADC noise reduction mode (or idle mode). The ADC will start a conversion once the CPU has been halted.

c. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC conversion complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC conversion complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than idle mode and ADC noise reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption.

5.21.5.1 Analog Input Circuitry

The analog input circuitry for single ended channels is illustrated in Figure 5-107 An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 kΩ or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the S/H capacitor, with can vary widely. The user is recommended to only use low impedant sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

Signal components higher than the nyquist frequency ($f_{ADC}/2$) should not be present for either kind of channels, to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 5-107. Analog Input Circuitry

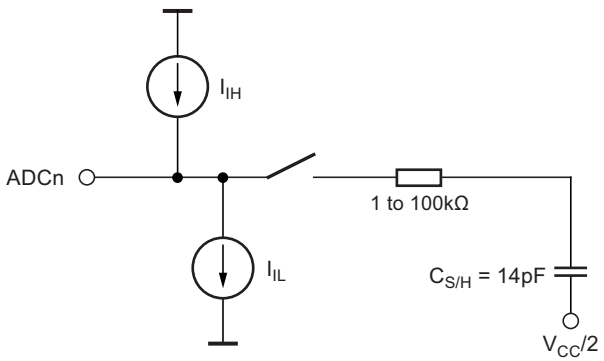
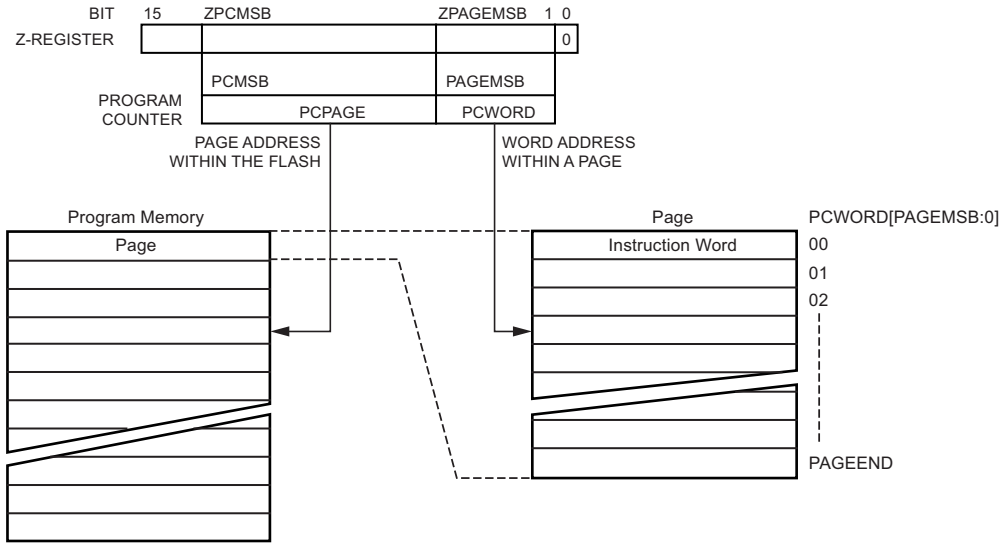


Figure 5-116. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 5-116 are listed in Table 5-110 on page 251.

5.23.7 Self-Programming the Flash

The program memory is updated in a page by page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the page erase command or between a page erase and a page write operation:

Alternative 1, fill the buffer before a page erase

- Fill temporary page buffer
- Perform a page erase
- Perform a page write

Alternative 2, fill the buffer after page erase

- Perform a page erase
- Fill temporary page buffer
- Perform a page write

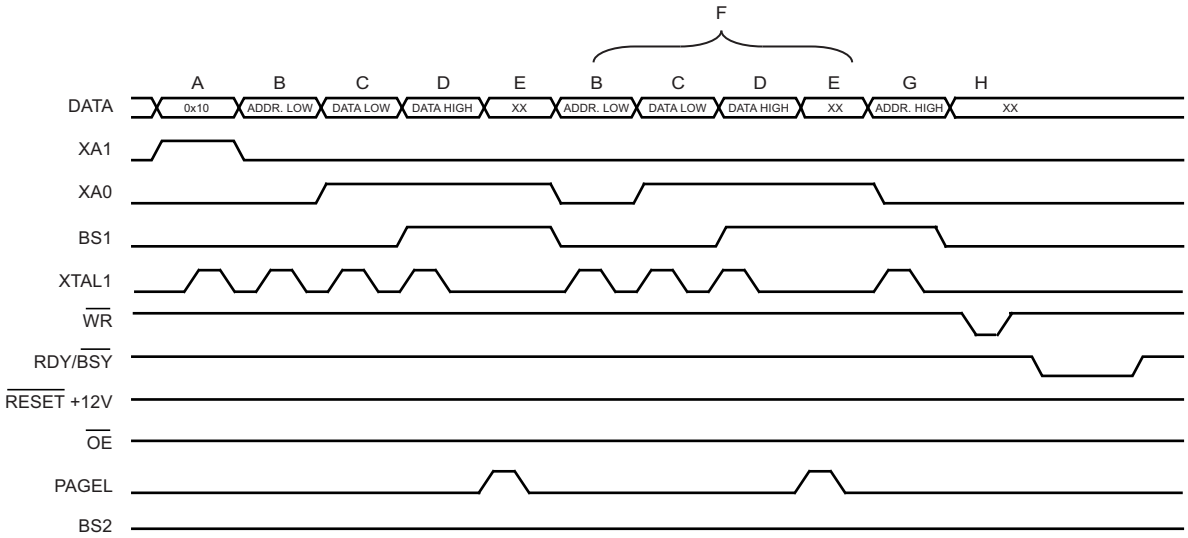
If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be rewritten. When using alternative 1, the boot loader provides an effective read-modify-write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the page erase and page write operation is addressing the same page. See Section 5.23.7.12 “Simple Assembly Code Example for a Boot Loader” on page 249 for an assembly code example.

5.23.7.1 Performing Page Erase by SPM

To execute page erase, set up the address in the Z-pointer, write “X0000011” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer will be ignored during this operation.

- Page erase to the RWW section: The NRWW section can be read during the page erase.
- Page erase to the NRWW section: The CPU is halted during the operation.

Figure 5-119. Programming the Flash Waveforms(1)



Note: 1. “XX” is do not care. The letters refer to the programming description above.

5.24.7.5 Programming the EEPROM

The EEPROM is organized in pages (see Table 5-125 on page 258). When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to Section 5.24.7.4 “Programming the Flash” on page 259 for details on command, address and data loading):

1. A: Load command “0001 0001”.
2. G: Load address high byte (0x00 - 0xFF).
3. B: Load address low byte (0x00 - 0xFF).
4. C: Load data (0x00 - 0xFF).
5. E: Latch data (give PAGEL a positive pulse).

K: Repeat 3 through 5 until the entire buffer is filled.

L: Program EEPROM page

1. Set BS1 to “0”.
2. Give $\overline{WR}$ a negative pulse. This starts programming of the EEPROM page. RDY/$\overline{BSY}$ goes low.
3. Wait until to RDY/$\overline{BSY}$ goes high before programming the next page (see Figure 5-120 on page 262 for signal waveforms).

Figure 6-11. Output High Voltage versus Output High Current (V_{CC} = 5V)

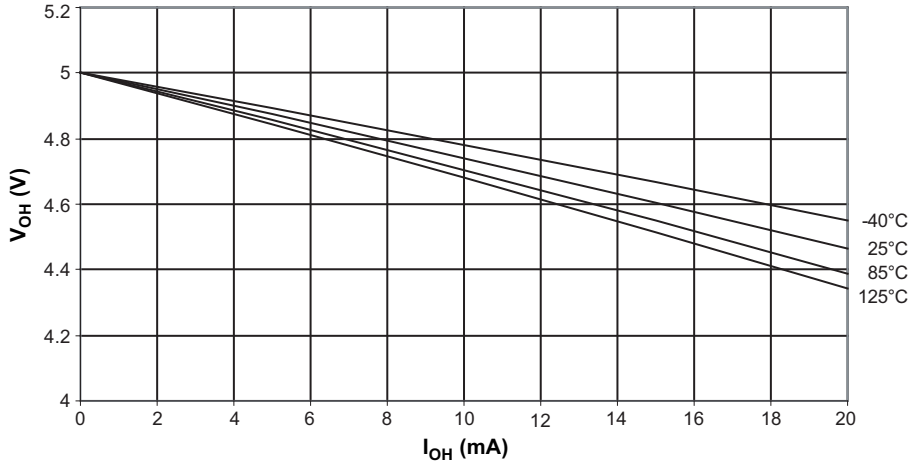


Figure 6-12. Output High Voltage versus Output High Current (V_{CC} = 3V)

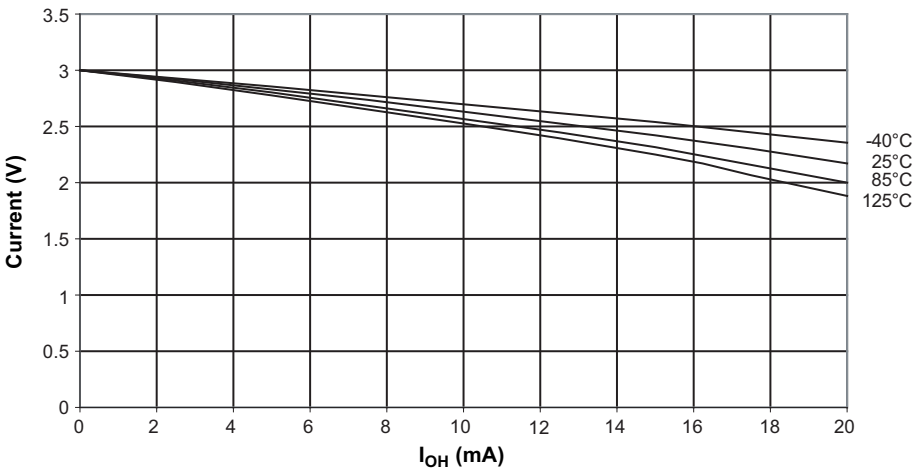


Figure 6-13. Reset Pull-Up Resistor Current versus Reset Pin Voltage (V_{CC} = 5V)

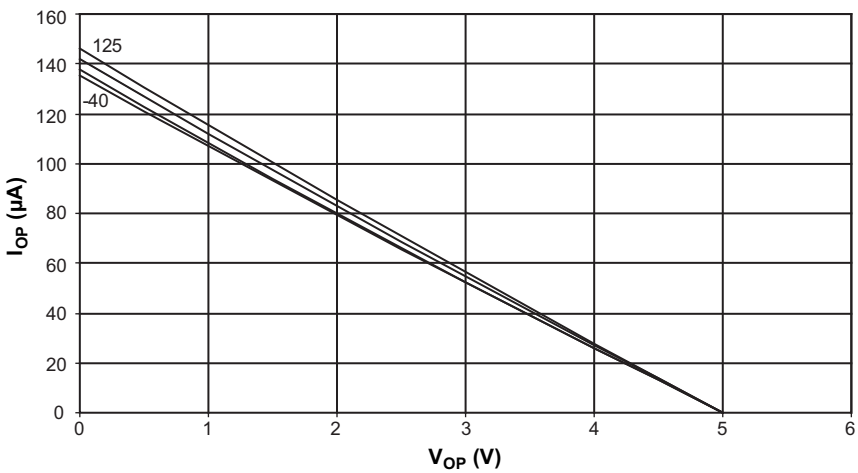
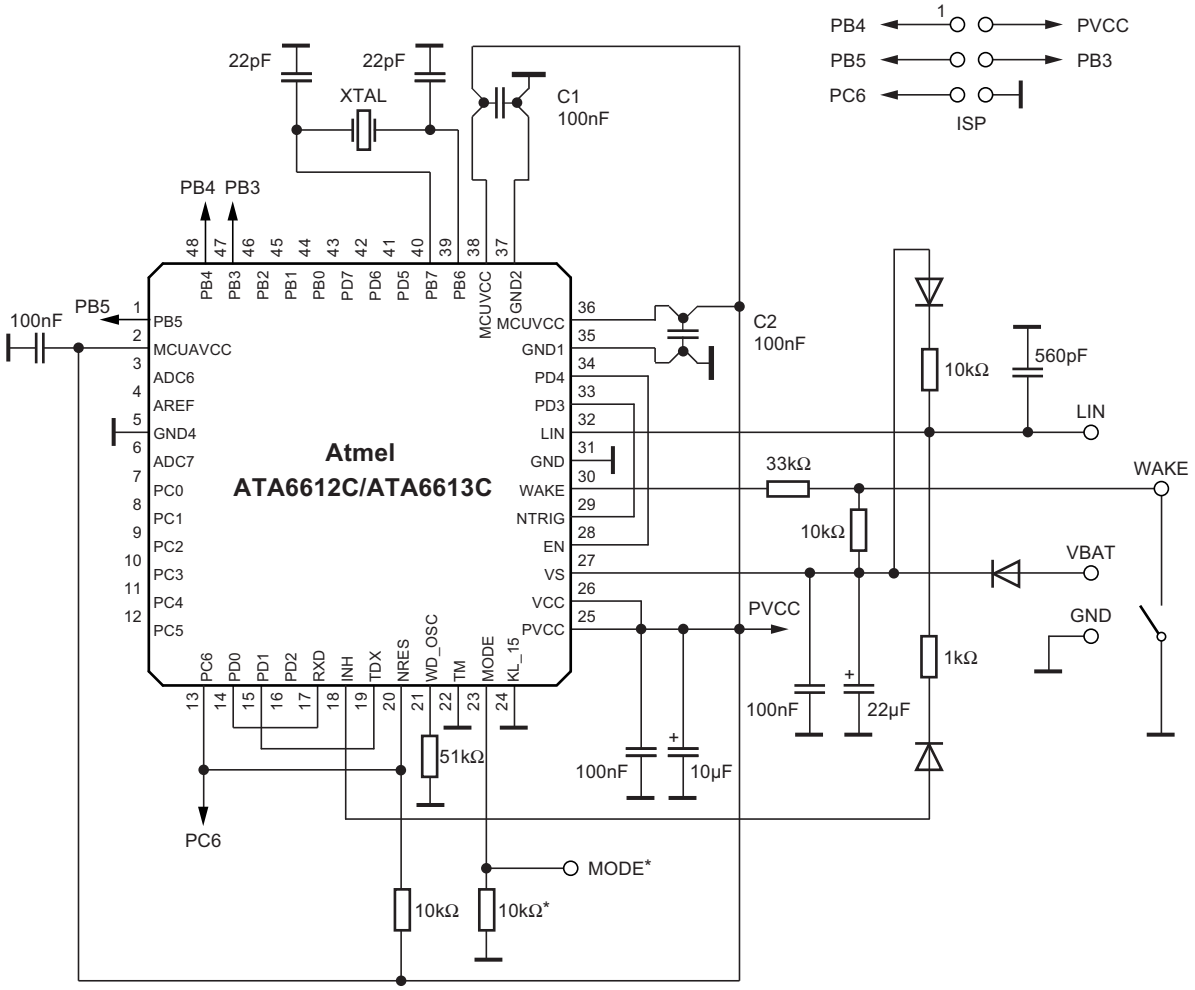


Figure 7-3. Typical LIN Master Application
LIN Master Pull-up Switched Off during Sleep Mode



* The MODE pin can be connected directly to GND, if it is not needed to disable the Watchdog

Note: All open pins of the SiP can be used for application-specific purposes.
AVR: TXD, RXD, NRES and EN connected for LIN Master. The connection between the LIN-SBC and the AVR requires the software being programmed correspondingly. Analog digital converter not active; system clock from external crystal.
LIN-SBC: Master application with 5V regulator and watchdog, 1k master resistance connected via diode and INH output to VBAT, local wake up via pin WAKE.
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 7-3.

11. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

Revision No.	History
9111L-AUTO-11/14	• Put datasheet in the latest template • Section 8 “Ordering Information” on page 306 updated • Section 9 “Package Information” on page 306 updated
9111K-AUTO-03/13	• Section 2 “Pin Configuration” on pages 2 to 3 updated • Table 3-1 “Maximum Ratings of the SIP” on page 4 updated • Table 3-2 “Maximum Ratings of the LIN-SBC” on pages 4 to 5 added • Table 3-3 “Maximum Ratings of the Microcontroller” on page 5 added • Figure 8-1 “Typical LIN Slave Application” on page 351 updated • Figure 8-2 “Typical LIN Master Application” on page 352 updated • Figure 8-3 “LIN Slave Application with Minimum External Components” on page 353 updated • Figure 8-4 “Typical LIN Master Application LIN Master Pull-up Switched Off during Sleep Mode” on page 354 updated
9111J-AUTO-11/12	• ATA6612P/ATA6613P renamed in ATA6612C/ATA6613C
9111I-AUTO-02/12	• General Features on page 1 changed • Section 3.1 “Features” on page 5 changed • Section 3.3 “Functional Description” on pages 7 to 19 changed • Section 4 “Absolute Maximum Ratings” on page 20 changed • Section 5 “Electrical Characteristics” on page 21 changed • Section 8 “Application” on pages 355 to 358 changed
9111H-AUTO-01/11	• Section 3.1 “Features” on page 5 changed • Section 3.3.3 “Ground Pin” on page 7 changed • Section 3.3.12 “Mode Input Pin (Mode)” on page 8 changed • Figure 3.2 “Modes of Operation” on page 10 changed • Section 3.3.20.4 “Fail-safe Mode” on page 13 changed • Section 3.3.23 “Voltage Regulator” on pages 16 to 17 changed • Section 6 “Electrical Characteristics” on pages 21 to 26 changed
9111G-AUTO-05/10	• Table 2-2 “Maximum Ratings of the SiP” on page 4 changed • Section 3.1 “Features” on page 5 changed • Section 3.2 “Description” on page 5 changed • Section 3.3.1 “Physical Layer Compatibility” on page 7 changed • Section 3.3.6 “Bus Pin LIN” on page 7 changed • Section 3.3.8 “TX Dominant Time-out Function” on page 8 changed • Section 3.3.10 “Enable Input Pin (EN)” on page 8 changed • Section 3.3.14 “KL_15 Pin” on page 9 changed • Section 3.3.20 “Modes of Operation” on pages 10 to 14 changed • Section 3.3.21 “Wake-up Scenarios from Silent to Sleep Mode on page 15 changed • Section 3.3.23 “Voltage Regulator” on page 16 changed • Section 6 “Electrical Characteristics” on page 23 changed • Section 7.7.7.1 “Power Reduction Register” on page 66 changed