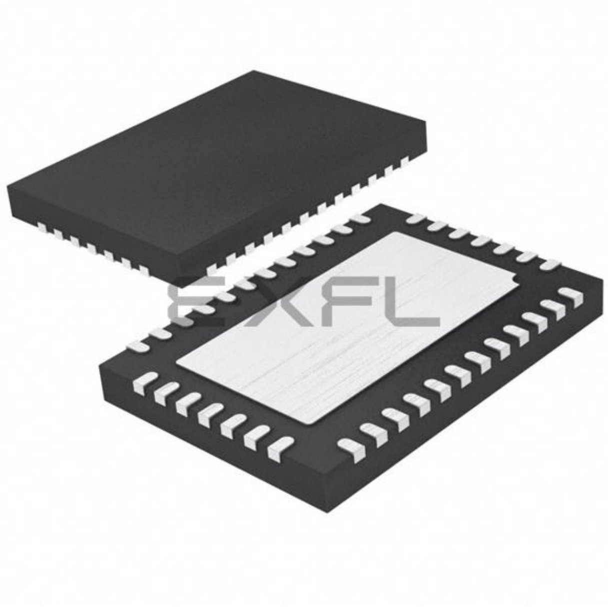


Welcome to E-XFL.COM

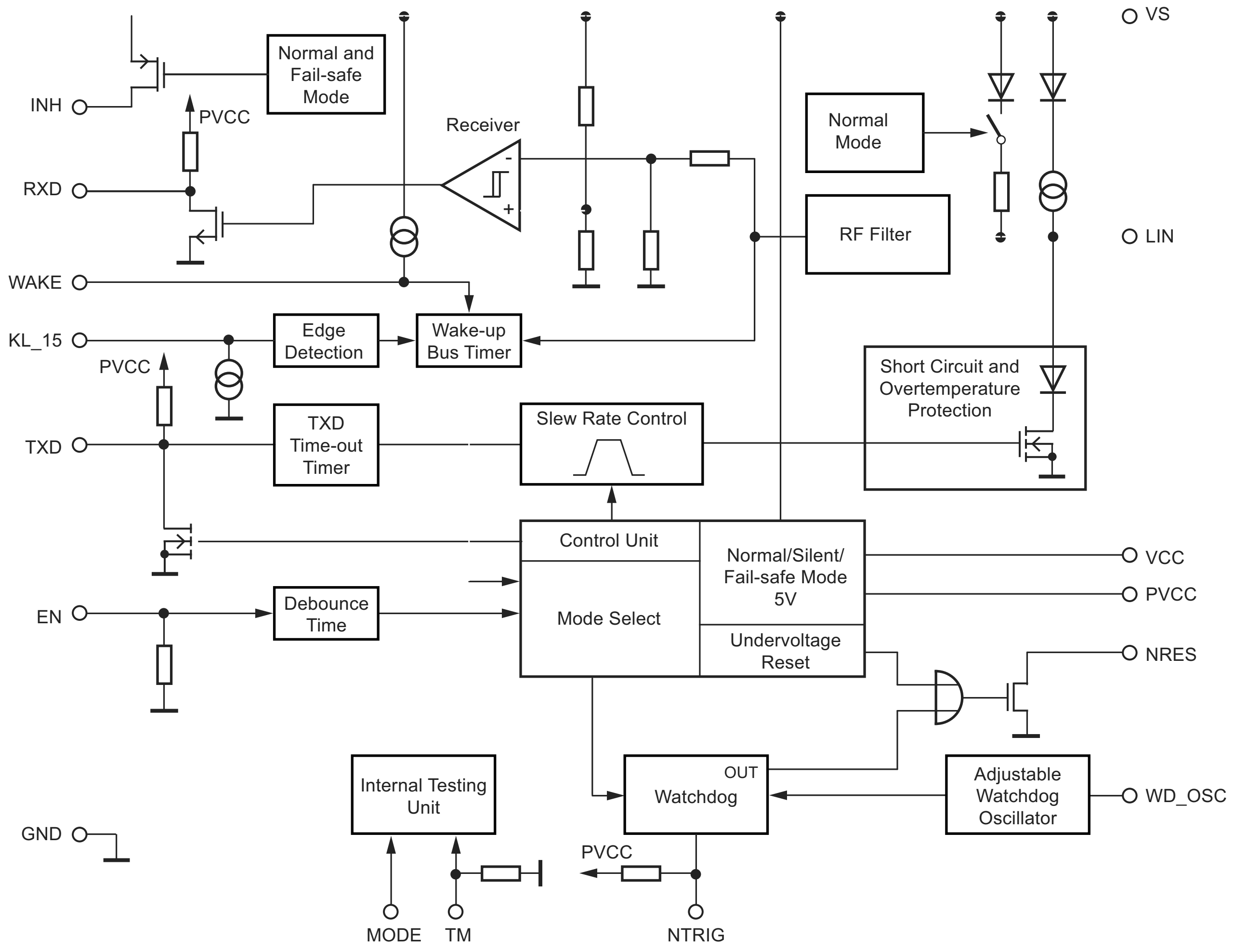
What is "Embedded - Microcontrollers"?

"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, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TC)
Mounting Type	Surface Mount
Package / Case	38-VFQFN Exposed Pad
Supplier Device Package	38-VQFN (5x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/ata6616c-p3pw

Figure 3-1. Block Diagram



3.6 Watchdog

The watchdog anticipates a trigger signal from the microcontroller at the NTRIG (negative edge) input within a time window of T_{wd}. The trigger signal must exceed a minimum time $t_{trigmin}$ > 200ns. If a triggering signal is not received, a reset signal will be generated at output NRES. After a watchdog reset, the IC starts with the lead time. The timing basis of the watchdog is provided by the internal oscillator. Its time period, T_{osc}, is adjustable via the external resistor R_{wd_osc} (34kΩ to 120kΩ).

During Silent or sleep mode the watchdog is switched off to reduce current consumption.

The minimum time for the first watchdog pulse is required after the undervoltage reset at NRES disappears. It is defined as lead time t_d. After wake up from Sleep or silent mode, the lead time t_d starts with the negative edge of the RXD output.

3.6.1 Typical Timing Sequence with R_{WD_OSC} = 51kΩ

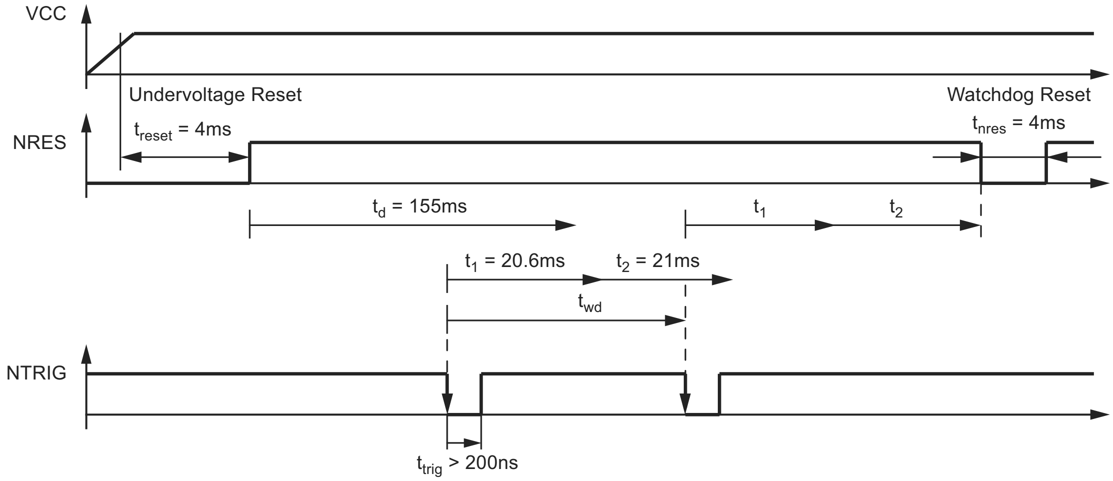
The trigger signal T_{wd} is adjustable between 20ms and 64ms using the external resistor R_{WD_OSC}.

For example, with an external resistor of R_{WD_OSC} = 51kΩ ±1%, the typical parameters of the watchdog are as follows:

$t_{osc} = 0.405 \times R_{WD\_OSC} - 0.0004 \times (R_{WD\_OSC})^2$ (R_{WD_OSC} in kΩ; t_{osc} in µs)
t_{OSC} = 19.6µs due to 51kΩ
$t_d = 7895 \times 19.6\mu s = 155ms$
$t_1 = 1053 \times 19.6\mu s = 20.6ms$
$t_2 = 1105 \times 19.6\mu s = 21.6ms$
t_{nres} = constant = 4 ms

After ramping up the battery voltage, the 5V regulator is switched on. The reset output NRES stays low for the time t_{reset} (typically 4ms), then it switches to high, and the watchdog waits for the trigger sequence from the microcontroller. The lead time, t_d, follows the reset and is t_d = 155ms. In this time, the first watchdog pulse from the microcontroller is required. If the trigger pulse NTRIG occurs during this time, the time t_1 starts immediately. If no trigger signal occurs during the time t_d, a watchdog reset with t_{NRES} = 4ms will reset the microcontroller after t_d = 155ms. The times t_1 and t_2 have a fixed relationship between each other. A triggering signal from the microcontroller is anticipated within the time frame of t_2 = 21.6ms. To avoid false triggering from glitches, the trigger pulse must be longer than $t_{TRIG,min}$ > 200ns. This slope serves to restart the watchdog sequence. If the triggering signal fails in this open window t_2, the NRES output will be drawn to ground. A triggering signal during the closed window t_1 immediately switches NRES to low.

Figure 3-9. Timing Sequence with R_{WD_OSC} = 51kΩ



4.4.3.4 Erase

To erase a byte, the address must be written to EEAR. If the EEPMn bits are 0b01, writing the EEPE (within four cycles after EEMPE is written) will trigger the erase operation only (programming time is given in Table 1). The EEPE bit remains set until the erase operation completes. While the device is busy programming, it is not possible to do any other EEPROM operations.

4.4.3.5 Write

To write a location, the user must write the address into EEAR and the data into EEDR. If the EEPMn bits are 0b10, writing the EEPE (within four cycles after EEMPE is written) will trigger the write operation only (programming time is given in Table 1-1). The EEPE bit remains set until the write operation completes. If the location to be written has not been erased before write, the data that is stored must be considered as lost. While the device is busy with programming, it is not possible to do any other EEPROM operations.

The calibrated oscillator is used to time the EEPROM accesses. Make sure the oscillator frequency is within the requirements described in Section 4.5.5.1 “OSCCAL – Oscillator Calibration Register” on page 58.

The following code examples show one assembly and one C function for erase, write, or atomic write of the EEPROM. The examples assume that interrupts are controlled (e.g., by disabling interrupts globally) so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_write:
      ; Wait for completion of previous write
      sbic    EECR,EEPE
      rjmp   EEPROM_write
      ; Set Programming mode
      ldi    r16, (0<<EEPM1)|(0<<EEPM0)
      out    EECR, r16
      ; Set up address (r18:r17) in address register
      out    EEARH, r18
      out    EEARL, r17
      ; Write data (r16) to data register
      out    EEDR, r16
      ; Write logical one to EEMPE
      sbi    EECR,EEMPE
      ; Start eeprom write by setting EEPE
      sbi    EECR,EEPE
      ret
```

C Code Example

```
void EEPROM_write(unsigned char ucAddress, unsigned char ucData)
{
      /* Wait for completion of previous write */
      while(EECR & (1<<EEPE))
      ;
      /* Set Programming mode */
      EECR = (0<<EEPM1)|(0<<EEPM0);
      /* Set up address and data registers */
      EEAR = ucAddress;
      EEDR = ucData;
      /* Write logical one to EEMPE */
      EECR |= (1<<EEMPE);
      /* Start eeprom write by setting EEPE */
      EECR |= (1<<EEPE);
}
```

Table 4-14. Clock Prescaler Select

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

4.5.5.3 CLKCSR – Clock Control and Status Register

Bit	7	6	5	4	3	2	1	0	
	CLKCCE	–	–	CLKRDY	CLKC3	CLKC2	CLKC1	CLKC0	CLKCSR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – CLKCCE: Clock Control Change Enable**

The CLKCCE bit must be written to logic one to enable change of the CLKCSR bits. The CLKCCE bit is only updated when the other bits in CLKCSR are simultaneously written to zero. CLKCCE is cleared by hardware four cycles after it is written or when the CLKCSR bits are written. Rewriting the CLKCCE bit within this time-out period does neither extend the time-out period, nor clear the CLKCCE bit.

- **Bits 6:5 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATtiny87/167 and will always read as zero.

- **Bits 4 – CLKRDY: Clock Ready Flag**

This flag is the output of the '*Clock Availability*' logic.
This flag is cleared by the '*Request for Clock Availability*' command or '*Enable Clock Source*' command being entered.
It is set when '*Clock Availability*' logic confirms that the (selected) clock is running and is stable. The delay from the request and the flag setting is not fixed, it depends on the clock start-up time, the clock frequency and, of course, if the clock is alive.
The user's code has to differentiate between '***no_clock_signal***' and '***clock_signal_not_yet_available***' condition.

- **Bits 3:0 – CLKC3:0: Clock Control Bits 3 - 0**

These bits define the command to provide to the '*Clock Switch'* module. The special write procedure must be followed to change the CLKC3..0 bits (See "Bit 7 – CLKCCE: Clock Control Change Enable" on page 60.).

1. Write the Clock Control Change Enable (CLKCCE) bit to one and all other bits in CLKCSR to zero.
2. Within 4 cycles, write the desired value to CLKCSR register while clearing CLKCCE bit.

Interrupts should be disabled when setting CLKCSR register in order not to disturb the procedure.

4.6.3 Idle Mode

When the SM1..0 bits are written to 00, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing the SPI, analog comparator, ADC, USI start condition, asynchronous Timer/Counter, watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the SPI interrupts. If wake-up from the analog comparator interrupt is not required, the analog comparator can be powered down by setting the ACD bit in the analog comparator control and status register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

4.6.4 ADC Noise Reduction Mode

When the SM1..0 bits are written to 01, the SLEEP instruction makes the MCU enter ADC noise reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the USI start condition, the asynchronous Timer/Counter and the watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC conversion complete interrupt, only an external reset, a watchdog system reset, a watchdog interrupt, a brown-out reset, a USI start condition interrupt, an asynchronous Timer/Counter interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or INT1 or a pin change interrupt can wake up the MCU from ADC noise reduction mode.

4.6.5 Power-down Mode

When the SM1..0 bits are written to 10, the SLEEP instruction makes the MCU enter power-down mode. In this mode, the external oscillator is stopped, while the external interrupts, the USI start condition, and the Watchdog continue operating (if enabled). Only an external reset, a watchdog system reset, a watchdog interrupt, a brown-out reset, the USI start condition interrupt, an external level interrupt on INT0 or INT1, or a pin change interrupt can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

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. Refer to Section 4.9 "External Interrupts" on page 79 for details.

When waking up from power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL fuses that define the reset time-out period, as described in Section 4.5.2 "Clock Sources" on page 47.

4.6.6 Power-save Mode

When the SM1..0 bits are written to 11, the SLEEP instruction makes the MCU enter power-save mode. This mode is identical to power-down, with one exception:

If Timer/Counter0 is clocked asynchronously, i.e., the AS0 bit in ASSR is set, Timer/Counter0 will run during sleep. The device can wake up from either timer overflow or output compare event from Timer/Counter0 if the corresponding Timer/Counter0 interrupt enable bits are set in TIMSK0, and the global interrupt enable bit in SREG is set.

If the asynchronous timer is **NOT** clocked asynchronously, power-down mode is recommended instead of power-save mode because the contents of the registers in the asynchronous timer should be considered undefined after wake-up in power-save mode if AS0 is 0.

This sleep mode basically halts all clocks except clk_{ASY}, allowing operation only of asynchronous modules, including Timer/Counter0 if clocked asynchronously.

4.6.7 Power Reduction Register

The power reduction register (PRR), see Section 4.6.9.3 "PRR – Power Reduction Register" on page 66, provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped.

The alternate pin configuration is as follows:

- **PCINT7/ADC7/AIN1/XREF/AREF – Port A, Bit7**

PCINT7: Pin change interrupt, source 7.

ADC7: Analog to digital converter, channel 7.

AIN1: Analog comparator positive input. This pin is directly connected to the positive input of the analog comparator.

XREF: Internal voltage reference output. The internal voltage reference 2.56V or 1.1V is output when XREFEN is set and if either 2.56V or 1.1V is used as reference for ADC conversion. When XREF output is enabled, the pin port pull-up and digital output driver are turned off.

AREF: External voltage reference input for ADC. The pin port pull-up and digital output driver are disabled when the pin is used as an external voltage reference input for ADC or as when the pin is only used to connect a bypass capacitor for the voltage reference of the ADC.

- **PCINT6/ADC6/AIN0/$\overline{\text{SS}}$ – Port A, Bit6**

PCINT6: Pin change interrupt, source 6.

ADC6: Analog to digital converter, channel 6.

AIN0: Analog comparator negative input. This pin is directly connected to the negative input of the analog comparator.

$\overline{\text{SS}}$: SPI slave select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDA6. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDA6. When the pin is forced to be an input, the pull-up can still be controlled by the PORTA6 bit.

- **PCINT5/ADC5/T1/USCK/SCL/SCK – Port A, Bit5**

PCINT5: Pin change interrupt, source 5.

ADC5: Analog to digital converter, channel 5.

T1: Timer/Counter1 clock input.

USCK: Three-wire mode USI clock input.

SCL: Two-wire mode USI clock input.

SCK: SPI master clock output, slave clock input pin. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDA5. When the SPI is enabled as a master, the data direction of this pin is controlled by DDA5. When the pin is forced to be an input, the pull-up can still be controlled by the PORTA5 bit.

- **PCINT4/ADC4/ICP1/DI/SDA/MOSI – Port A, Bit 4**

PCINT4: Pin change interrupt, source 4.

ADC4: Analog to digital converter, channel 4.

ICP1: Timer/Counter1 input capture trigger. The PA3 pin can act as an input capture pin for Timer/Counter1.

DI: Three-wire mode USI data input. USI three-wire mode does not override normal port functions, so pin must be configure as an input for DI function.

SDA: Two-wire mode serial interface (USI) data input / output.

MOSI: SPI master output / slave input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDA3. When the SPI is enabled as a master, the data direction of this pin is controlled by DDA3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTA3 bit.

- **PCINT3/ADC3/ISRC/INT1 – Port A, Bit 3**

PCINT3: Pin Change Interrupt, source 3.

ADC3: Analog to digital converter, channel 3.

ISCR: Current source output pin. While current is sourced by the current source module, the user can use the analog to digital converter channel 4 (ADC4) to measure the pin voltage.

INT1: External interrupt, source 1. The PA4 pin can serve as an external interrupt source.

Table 4-25 and Table 4-26 relate the alternate functions of Port A to the overriding signals shown in Figure 4-29 on page 89.

Table 4-25. Overriding Signals for Alternate Functions in PA7..PA4

Signal Name	PA7/PCINT7/ ADC7/AIN1 /XREF/AREF	PA6/PCINT6/ ADC6/AIN0/$\overline{\text{SS}}$	PA5/PCINT5/ADC5/ T1/USCK/SCL/SCK	PA4/PCINT4/ADC4/ ICP1/DI/SDA/MOSI
PUOE	0	SPE & $\overline{\text{MSTR}}$	SPE & $\overline{\text{MSTR}}$	SPE & $\overline{\text{MSTR}}$
PUOV	0	PORTA6 & $\overline{\text{PUD}}$	PORTA5 & $\overline{\text{PUD}}$	PORTA4 & $\overline{\text{PUD}}$
DDOE	0	SPE & $\overline{\text{MSTR}}$	(SPE & $\overline{\text{MSTR}}$) | (USI_2_WIRE & USIPOS)	(SPE & $\overline{\text{MSTR}}$) | (USI_2_WIRE & USIPOS)

DDOV	0	0	(USI_SCL_HOLD | $\overline{\text{PORTA5}}$)
			& DDRA6
				{ (SPE & $\overline{\text{MSTR}}$) ?
				(0) :
				($\overline{\text{USI\_SHIFTOUT}}$ | $\overline{\text{PORTA4}}$)
				& DDRA4) }

PVOE	0	0	(SPE & MSTR) | (USI_2_WIRE & USIPOS & DDRA5)	(SPE & MSTR) | (USI_2_WIRE & USIPOS & DDRA4)

PVOV	0	0	{ (SPE & MSTR) ?
			(SCK_OUTPUT) :
			~ (USI_2_WIRE & USIPOS
			& DDRA5) }
				{ (SPE & MSTR) ?
				(MOSI_OUTPUT) :
				~ (USI_2_WIRE & USIPOS
				& DDRA4) }

PTOE	0	0	USI_PTOE & USIPOS	0
DIEOE	ADC7D | (PCIE0 & PCMSK07)	ADC6D | (PCIE0 & PCMSK06)	ADC5D | (USISIE & USIPOS) | (PCIE0 & PCMSK05)	ADC4D | (USISIE & USIPOS) | (PCIE0 & PCMSK04)
DIEOV	PCIE0 & PCMSK07	PCIE0 & PCMSK06	(USISIE & USIPOS) | (PCIE0 & PCMSK05)	(USISIE & USIPOS) | (PCIE0 & PCMSK04)
DI	PCINT7	PCINT6 -/- $\overline{\text{SS}}$	PCINT5 -/- T1 -/- USCK -/- SCL -/- SCK	PCINT4 -/- ICP1 -/- DI -/- SDA -/- MOSI
AIO	ADC7 -/- AIN1 -/- XREF -/- AREF	ADC6 -/- AIN0	ADC5	ADC4

Table 4-26. Overriding Signals for Alternate Functions in PA3..PA0

Signal Name	PA3/PCINT3/ADC3/ ISRC/INT1	PA2/PCINT2/ADC2/ OC0A/DO/MISO	PA1/PCINT1/ADC1/ TXD/TXLIN	PA0/PCINT0/ADC0/ RXD/RXLIN
PUOE	0	SPE **&** $\overline{\text{MSTR}}$	LIN_TX_ENABLE	LIN_RX_ENABLE
PUOV	PORTA3 **&** $\overline{\text{PUD}}$	PORTA2 **&** $\overline{\text{PUD}}$	**{** (LIN_TX_ENABLE) **?** (0) **:** (PORTA1 **&** $\overline{\text{PUD}}$) **}**	PORTA0 **&** $\overline{\text{PUD}}$
DDOE	0	SPE **&** $\overline{\text{MSTR}}$	LIN_TX_ENABLE	LIN_RX_ENABLE
DDOV	0	0	LIN_TX_ENABLE	0
PVOE	0	(SPE **&** MSTR) **|** $\overline{\text{(USI\_2\_WIRE}}$ **&** USI_3_WIRE **&** USIPOS) **|** OC0A	LIN_TX_ENABLE	0
PVOV	0	**{** (SPE **&** MSTR) **?** (MISO_OUTPUT) **:** (($\overline{\text{USI\_2\_WIRE}}$ **&** USI_3_WIRE **&** USIPOS) **?** (USI_SHIFTOUT) **:** (OC0A)) **}**	**{** (LIN_TX_ENABLE) **?** (LIN_TX) **:** (0) **}**	0
PTOE	0	0	0	0
DIEOE	ADC3D **|** INT1_ENABLE **|** (PCIE0 **&** PCMSK03)	ADC2D **|** (PCIE0 **&** PCMSK02)	ADC1D **|** (PCIE0 **&** PCMSK01)	ADC0D **|** (PCIE0 **&** PCMSK00)
DIEOV	INT1_ENABLE **|** (PCIE0 **&** PCMSK03)	PCIE0 **&** PCMSK02	PCIE0 **&** PCMSK01	PCIE0 **&** PCMSK00
DI	PCINT3 **-/-** INT1	PCINT2 **-/-** MISO	PCINT1	PCINT0
AIO	ADC3 **-/-** ISRC	ADC2	ADC1	ADC0

The alternate pin configuration is as follows:

- **PCINT15/ADC10/OC1BX/$\overline{\text{RESET}}$/dW – Port B, Bit 7**

PCINT15: Pin change interrupt, source 15.

ADC10: Analog to digital converter, channel 10.

OC1BX: Output compare and PWM output B-X for Timer/Counter1. The PB7 pin has to be configured as an output (DDB7 set (one)) to serve this function. The OC1BX pin is also the output pin for the PWM mode timer function (c.f. OC1BX bit of TCCR1D register).

$\overline{\text{RESET}}$: Reset input pin. When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on power-on reset and brown-out reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin. If PB7 is used as a reset pin, DDB7, PORTB7 and PINB7 will all read 0.

dW: When the debugWIRE enable (DWEN) fuse is programmed and Lock bits are unprogrammed, the RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

- **PCINT14/ADC9/OC1AX/INT0 – Port B, Bit 6**

PCINT14: Pin change interrupt, source 14.

ADC9: Analog to digital converter, channel 9.

OC1AX: Output compare and PWM output A-X for Timer/Counter1. The PB6 pin has to be configured as an output (DDB6 set (one)) to serve this function. The OC1AX pin is also the output pin for the PWM mode timer function (c.f. OC1AX bit of TCCR1D register).

INT0: External Interrupt0 input. The PB6 pin can serve as an external interrupt source.

- **PCINT13/ADC8/OC1BW/XTAL2/CLKO – Port B, Bit 5**

PCINT13: Pin change interrupt, source 13.

ADC8: Analog to digital converter, channel 8.

OC1BW: Output compare and PWM output B-W for Timer/Counter1. The PB5 pin has to be configured as an output (DDB5 set (one)) to serve this function. The OC1BW pin is also the output pin for the PWM mode timer function (c.f. OC1BW bit of TCCR1D register).

XTAL2: Chip clock oscillator pin 2. Used as clock pin for crystal oscillator or low-frequency crystal oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

CLKO: Divided system clock output. The divided system clock can be output on the PB5 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB5 and DDB5 settings. It will also be output during reset.

- **PCINT12/OC1AW/XTAL1/CLKI – Port B, Bit 4**

PCINT12: Pin change interrupt, source 12.

OC1AW: Output compare and PWM output A-W for Timer/Counter1. The PB4 pin has to be configured as an output (DDB4 set (one)) to serve this function. The OC1AW pin is also the output pin for the PWM mode timer function (c.f. OC1AW bit of TCCR1D register).

XTAL1: Chip clock oscillator pin 1. Used for all chip clock sources except internal calibrated RC oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

CLKI: External clock input. When used as a clock pin, the pin can not be used as an I/O pin.

Note: If PB4 is used as a clock pin (XTAL1 or CLKI), DDB4, PORTB4 and PINB4 will all read 0.

Table 4-28 and Table 4-29 relate the alternate functions of Port B to the overriding signals shown in Figure 4-29 on page 89.

Table 4-28. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/PCINT15/ADC10/ OC1BX/$\overline{\text{RESET}}$/dW	PB6/PCINT14/ADC9/ OC1AX/INT0	PB5/PCINT13/ADC8/ OC1BW/XTAL2/CLKO	PB4/PCINT12/ OC1AW/XTAL1/CLKI
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC1B_ENABLE & OC1BX	OC1A_ENABLE & OC1AX	OC1B_ENABLE & OC1BW	OC1A_ENABLE & OC1AW
PVOV	OC1B	OC1A	OC1B	OC1A
PTOE	0	0	0	0
DIEOE	ADC10D | (PCIE1 & PCMSK15)	ADC9D | INT0_ENABLE | (PCIE1 & PCMSK14)	ADC8D | (PCIE1 & PCMSK13)	(PCIE1 & PCMSK13)
DIEOV	PCIE1 & PCMSK15	INT0_ENABLE | (PCIE1 & PCMSK14)	PCIE1 & PCMSK13	1
DI	PCINT15	PCINT14 -/- INT1	PCINT13	PCINT12
AIO	RESET -/- ADC10 -/-	ADC9 -/- ISRC	ADC8 -/- XTAL2	XTAL1 -/- CLKI

Table 4-29. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/PCINT11/ OC1BV	PB2/PCINT10/ OC1AV/USCK/SCL	PB1/PCINT9/ OC1BU/DO	PB0/IPCINT8/ OC1AU/DI/SDA
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	(USI_2_WIRE & $\overline{\text{USIPOS}}$)	0	(USI_2_WIRE & $\overline{\text{USIPOS}}$)
DDOV	0	(USI_SCL_HOLD | $\overline{\text{PORTB2}}$) & DDRB2	0	($\overline{\text{USI\_SHIFTOUT}}$ | $\overline{\text{PORTB0}}$) & DDRB0)
PVOE	OC1B_ENABLE & OC1BV	(USI_2_WIRE & $\overline{\text{USIPOS}}$ & DDRB2) | (OC1A_ENABLE & OC1AV)	($\overline{\text{USI\_2\_WIRE}}$ & USI_3_WIRE & $\overline{\text{USIPOS}}$) | (OC1B_ENABLE & OC1BU)	(USI_2_WIRE & $\overline{\text{USIPOS}}$ & DDRB0) | (OC1A_ENABLE & OC1AU)
PVOV	OC1B	{ (USI_2_WIRE & $\overline{\text{USIPOS}}$ & DDRB2) ? (0) : (OC1A) }	{ ($\overline{\text{USI\_2\_WIRE}}$ & USI_3_WIRE & $\overline{\text{USIPOS}}$) ? (USI_SHIFTOUT) : (OC1B) }	{ (USI_2_WIRE & $\overline{\text{USIPOS}}$ & DDRB0) ? (0) : (OC1A) }
PTOE	0	USI_PTOE & $\overline{\text{USIPOS}}$	0	0
DIEOE	PCIE1 & PCMSK11	(USISIE & $\overline{\text{USIPOS}}$) | (PCIE1 & PCMSK10)	PCIE1 & PCMSK9	(USISIE & $\overline{\text{USIPOS}}$) | (PCIE1 & PCMSK8)
DIEOV	1	(USISIE & $\overline{\text{USIPOS}}$) | (PCIE1 & PCMSK10)	1	(USISIE & $\overline{\text{USIPOS}}$) | (PCIE1 & PCMSK8)
DI	PCINT11	PCINT10 -/- USCK -/- SCL	PCINT9	PCINT8 -/- DI -/- SDA
AIO	0	0	0	0

4.10.4 Register Description for I/O Ports

4.10.4.1 Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

4.10.4.2 Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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	

4.10.4.3 Port A Input Pins Register – PINA

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

4.10.4.4 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	

4.10.4.5 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	

4.10.4.6 Port B Input Pins Register – PINB

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

The Timer/Counter (TCNT0) and output compare register (OCR0A) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the timer interrupt flag register (TIFR0). All interrupts are individually masked with the timer interrupt mask register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or asynchronously clocked from the XTAL1/2 pins, as detailed later in this section. The asynchronous operation is controlled by the asynchronous status register (ASSR). The clock select logic block controls which clock source the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk_{T0}).

The double buffered output compare register (OCR0A) is compared with the Timer/Counter value at all times. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the output compare pin (OC0A). Section 4.11.5 “Output Compare Unit” on page 104 for details. The compare match event will also set the compare flag (OCF0A) which can be used to generate an output compare interrupt request.

4.11.2.1 Definitions

The following definitions are used extensively throughout the section:

BOTTOM The counter reaches the BOTTOM when it becomes zero (0x00).

MAX The counter reaches its MAXimum when it becomes 0xFF (decimal 255).

TOP The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A Register. The assignment is dependent on the mode of operation.

4.11.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal synchronous or an external asynchronous clock source. The clock source is selected by the clock select logic which is controlled by the clock select (CS02:0) bits located in the Timer/Counter control register (TCCR0).The clock source clk_{T0} is by default equal to the MCU clock, $clk_{I/O}$. When the AS0 bit in the ASSR Register is written to logic one, the clock source is taken from the Timer/Counter Oscillator connected to XTAL1 and XTAL2 or directly from XTAL1. For details on asynchronous operation, see Section 4.11.11.5 “Asynchronous Status Register – ASSR” on page 116. For details on clock sources and prescaler, see Section 4.11.10 “Timer/Counter0 Prescaler” on page 112.

4.11.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 4-31 shows a block diagram of the counter and its surrounding environment.

Figure 4-31. Counter Unit Block Diagram

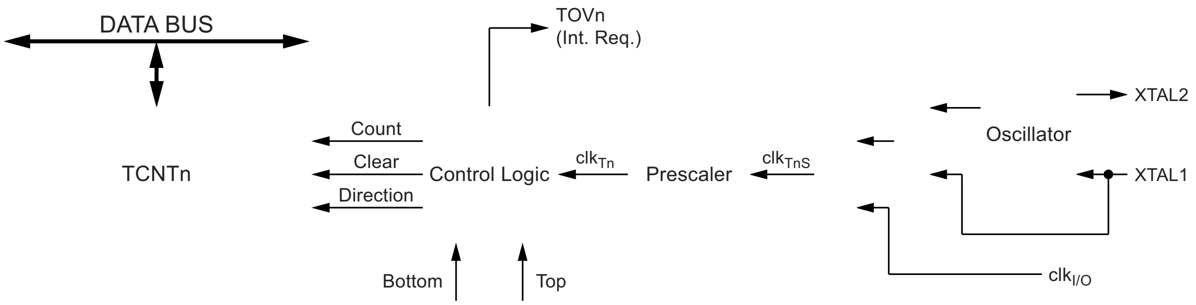


Table 4-37 shows the COM1A/B1:0 bit functionality when the WGM13:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

Table 4-37. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM(1)

OC1Ai OC1Bi	COM1A1 COM1B1	COM1A0 COM1B0	Description
0	x	x	Normal port operation, OC1A/OC1B disconnected.
1	0	0	

1	0	1	WGM13=0: Normal port operation, OC1A/OC1B disconnected.
			WGM13=1: Toggle OC1A on compare match, OC1B reserved.

1	1	0	Clear OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when downcounting.
1	1	1	Set OC1A/OC1B on compare match when up-counting. Clear OC1A/OC1B on compare match when downcounting.

Note: 1. A special case occurs when OC1A/OC1B equals TOP and COM1A1/COM1B1 is set. See Section 4.13.9.4 “Phase Correct PWM Mode” on page 136 for more details.

- **Bit 3:2 – Reserved Bits**

These bits are reserved for future use.

- **Bit 1:0 – WGM11:0: Waveform Generation Mode**

Combined with the WGM13:2 bits found in the TCCR1B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 4-38. Modes of operation supported by the Timer/Counter unit are: normal mode (counter), clear timer on compare match (CTC) mode, and three types of pulse width modulation (PWM) modes (see Section 4.13.9 “Modes of Operation” on page 133).

Table 4-38. Waveform Generation Mode Bit Description(1)

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1A/B at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, phase correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, phase correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, phase correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	TOP	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	TOP	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	TOP	TOP
8	1	0	0	0	PWM, phase and frequency correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, phase and frequency correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, phase correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, phase correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	TOP	TOP
15	1	1	1	1	Fast PWM	OCR1A	TOP	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

The three-wire mode timing is shown in Figure 4-64 on page 156 At the top of the figure is a USCK cycle reference. One bit is shifted into the USI data register (USIDR) for each of these cycles. The USCK timing is shown for both external clock modes. In external clock mode 0 (USICS0 = 0), DI is sampled at positive edges, and DO is changed (data register is shifted by one) at negative edges. External clock mode 1 (USICS0 = 1) uses the opposite edges versus mode 0, i.e., samples data at negative and changes the output at positive edges. The USI clock modes corresponds to the SPI data mode 0 and 1.

Referring to the timing diagram (Figure 4-64 on page 156), a bus transfer involves the following steps:

1. The slave device and master device sets up its data output and, depending on the protocol used, enables its output driver (mark A and B). The output is set up by writing the data to be transmitted to the USI data register. Enabling of the output is done by setting the corresponding bit in the port data direction register. Note that point A and B does not have any specific order, but both must be at least one half USCK cycle before point C where the data is sampled. This must be done to ensure that the data setup requirement is satisfied. The 4-bit counter is reset to zero.
2. The master generates a clock pulse by software toggling the USCK line twice (C and D). The bit value on the slave and master's data input (DI) pin is sampled by the USI on the first edge (C), and the data output is changed on the opposite edge (D). The 4-bit counter will count both edges.
3. Step 2. is repeated eight times for a complete register (byte) transfer.
4. After eight clock pulses (i.e., 16 clock edges) the counter will overflow and indicate that the transfer is completed. The data bytes transferred must now be processed before a new transfer can be initiated. The overflow interrupt will wake up the processor if it is set to Idle mode. Depending of the protocol used the slave device can now set its output to high impedance.

4.15.3.2 SPI Master Operation Example

The following code demonstrates how to use the USI module as a SPI Master:

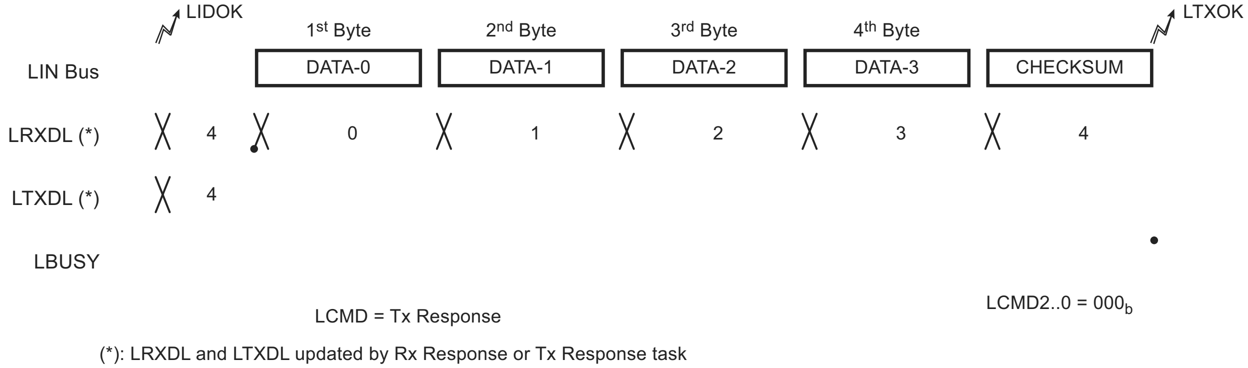
```
SPITransfer:
sts          USIDR,r16
ldi          r16,(1<<USIOIF)
sts          USISR,r16
ldi          r16,(1<<USIWM0)|(1<<USICS1)|(1<<USICLK)|(1<<USITC)
SPITransfer_loop:
sts          USICR,r16
lds          r16, USISR
sbrs         r16, USIOIF
rjmp         SPITransfer_loop
lds          r16,USIDR
ret
```

The code is size optimized using only eight instructions (+ ret). The code example assumes that the DO and USCK pins are enabled as output in the DDRA or DDRB register. The value stored in register r16 prior to the function is called is transferred to the slave device, and when the transfer is completed the data received from the slave is stored back into the r16 register.

The second and third instructions clears the USI counter overflow flag and the USI counter value. The fourth and fifth instruction set three-wire mode, positive edge shift register clock, count at USITC strobe, and toggle USCK. The loop is repeated 16 times.

Data Length in Tx Response

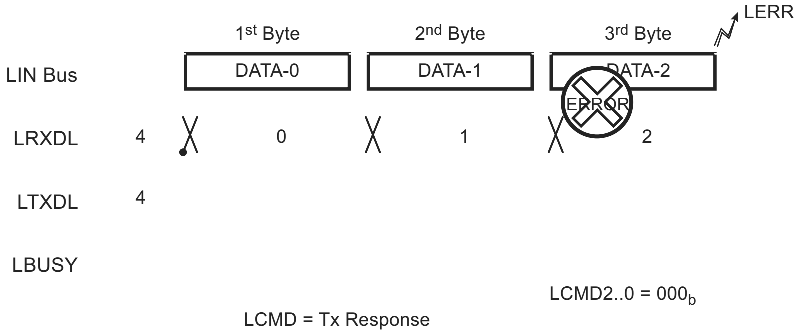
Figure 4-77. LIN1.3 - Tx Response - No Error



- The user initializes LTXDL field before setting the Tx response command,
- After setting the Tx response command, LRXDL is reset by hardware,
- LTXDL will remain unchanged during Tx (during busy signal),
- LRXDL will count the number of transmitted bytes (during busy signal),
- If an error occurs, Tx stops, the corresponding error flag is set and LRXDL will give the number of transmitted bytes without error,
- If no error occurs, LTXOK is set after the transmission of the CHECKSUM, LTXDL will be unchanged (and LRXDL = LTXDL).

Data Length after Error

Figure 4-78. Tx Response - Error



Note: Information on response (ex: error on byte) is only available at the end of the serialization/de-serialization of the byte.

Data Length in UART Mode

- The UART mode forces LRXDL and LTXDL to 0 and disables the writing in LINDLR register,
- Note that after reset, LRXDL and LTXDL are also forced to 0.

4.22.5 Page Size

Table 4-73. Number of Words in a Page and No. of Pages in the Flash

Device	Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
ATtiny87	4K words	64 words	PC[5:0]	64	PC[11:6]	11
ATtiny167	8K words	64 words	PC[5:0]	128	PC[12:6]	12

Table 4-74. Number of Words in a Page and No. of Pages in the EEPROM

Device	EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
ATtiny87 ATtiny167	512bytes	4bytes	EEA[1:0]	128	EEA[8:2]	8

4.22.6 Parallel Programming Parameters, Pin Mapping, and Commands

This section describes how to parallel program and verify flash program memory, EEPROM data memory, memory lock bits, and fuse bits in the Atmel® ATtiny87/167. Pulses are assumed to be at least 250ns unless otherwise noted.

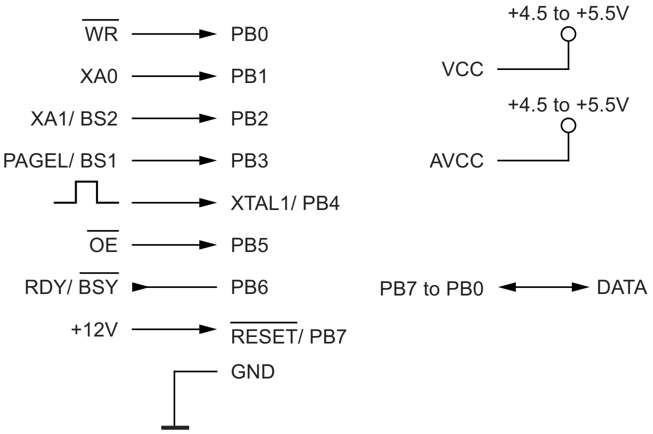
4.22.6.1 Signal Names

In this section, some pins of the Atmel ATtiny87/167 are referenced by signal names describing their functionality during parallel programming, see Figure 4-97 and Figure 4-98. Pins not described in the following table are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in Figure 4-76.

When pulsing $\overline{\text{WR}}$ or $\overline{\text{OE}}$, the command loaded determines the action executed. The different commands are shown in Figure 4-77.

Figure 4-97. Parallel Programming



Note: V_{CC} – 0.3V < AV_{CC} < V_{CC} + 0.3V, however, AV_{CC} should always be within 4.5 to 5.5V

4.22.7 Parallel Programming

4.22.7.1 Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5V between V_{CC} and GND.
2. Set RESET to “0” and toggle XTAL1 at least six times.
3. Set the Prog_enable pins listed in Table 4-75 on page 226 to “0000_b” and wait at least 100ns.
4. Apply 11.5 - 12.5V to RESET. Any activity on Prog_enable pins within 100ns after +12V has been applied to RESET, will cause the device to fail entering programming mode.
5. Wait at least 50µs before sending a new command.

4.22.7.2 Considerations for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF, that is the contents of the entire EEPROM (unless the EESAVE fuse is programmed) and flash after a chip erase.
- Address high byte needs only be loaded before programming or reading a new 256 word window in flash or 256 byte EEPROM. This consideration also applies to signature bytes reading.

4.22.7.3 Chip Erase

The chip erase will erase the flash and EEPROM(1) memories plus lock bits. The lock bits are not reset until the program memory has been completely erased. The fuse bits are not changed. A chip erase must be performed before the flash and/or EEPROM are reprogrammed.

Note: 1. The EEPRPOM memory is preserved during chip erase if the EESAVE fuse is programmed.

Load Command “Chip Erase”

1. Set XA1, XA0 to “1,0 ”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “1000 0000 $_b$”. This is the command for chip erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{\text{WR}}$ a negative pulse. This starts the chip erase. RDY/$\overline{\text{BSY}}$ goes low.
6. Wait until RDY/$\overline{\text{BSY}}$ goes high before loading a new command.

4.22.7.4 Programming the Flash

The flash is organized in pages, see Table 4-73 on page 225. When programming the flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

A. Load Command “Write Flash”

1. Set XA1, XA0 to “1,0”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “0001 0000 $_b$”. This is the command for write flash.

Give XTAL1 a positive pulse. This loads the command.

B. Load Address Low byte

1. Set XA1, XA0 to “00”. This enables address loading.
2. Set BS1 to “0”. This selects low address.
3. Set DATA = Address low byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address low byte.

4.23 Electrical Characteristics

Note: All characteristics contained in this data sheet are based on simulation and characterization of Atmel® ATtiny87/167 AVR® microcontrollers manufactured in a typical process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual Automotive silicon.

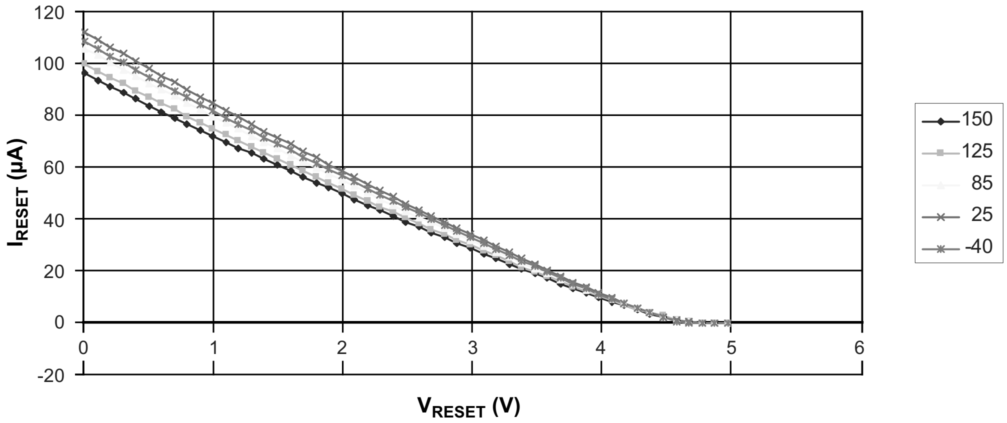
4.23.1 DC Characteristics

T_A = –40°C to +125°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Parameters	Test Conditions	Symbol	Min.	Typ.[1]	Max.	Unit
Input low voltage	Except XTAL1 and $\overline{\text{RESET}}$ pins	V_{IL}	–0.5		0.2 V_{CC}[2]	V
	XTAL1 pin - External clock selected	V_{IL1}	–0.5		0.1 V_{CC}[2]	V
	$\overline{\text{RESET}}$ pin	V_{IL2}	–0.5		0.2 V_{CC}[2]	V
	$\overline{\text{RESET}}$ pin as I/O	V_{IL3}	–0.5		0.2 V_{CC}[2]	V
Input high voltage	Except XTAL1 and $\overline{\text{RESET}}$ pins	V_{IH}	0.7 V_{CC}[3]		V_{CC} + 0.5	V
	XTAL1 pin - External clock selected	V_{IH1}	0.8 V_{CC}[3]		V_{CC} + 0.5	V
	$\overline{\text{RESET}}$ pin	V_{IH2}	0.9 V_{CC}[3]		V_{CC} + 0.5	V
	$\overline{\text{RESET}}$ pin as I/O	V_{IH3}	0.7 V_{CC}[3]		V_{CC} + 0.5	V
Output low voltage[4] (Ports A, B,)	I_{OL} = 10mA, V_{CC} = 5V I_{OL} = 5mA, V_{CC} = 3V	V_{OL}			0.6 0.5	V
Output high voltage[5] (Ports A, B)	I_{OH} = –10mA, V_{CC} = 5V I_{OH} = –5mA, V_{CC} = 3V	V_{OH}	4.3 2.5			V
Input leakage current I/O pin	V_{CC} = 5.5V, pin low (absolute value)	I_{IL}		< 0.05	1	µA
Input leakage current I/O pin	V_{CC} = 5.5V, pin high (absolute value)	I_{IH}		< 0.05	1	µA
Reset pull-up resistor		R_{RST}	30		60	kΩ
I/O pin pull-up resistor		R_{pu}	20		50	kΩ

Notes:
1. "Typ.", typical values at 25°C. Maximum values are characterized values and not test limits in production.
2. "Max." means the highest value where the pin is guaranteed to be read as low.
3. "Min." means the lowest value where the pin is guaranteed to be read as high.
4. Although each I/O port can sink more than the test conditions (10mA at V_{CC} = 5V, 5mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   The sum of all IOL, for all ports, should not exceed 120mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
5. Although each I/O port can source more than the test conditions (10mA at V_{CC} = 5V, 5mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   The sum of all IOH, for all ports, should not exceed 120mA.
   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.
6. Values using methods described in Section 4.6.8 "Minimizing Power Consumption" on page 64. Power reduction is enabled (PRR = 0xFF) and there is no I/O drive.
7. BOD disabled.

Figure 4-127. Reset Pull-up Resistor Current versus Reset Pin Voltage (V_{CC} = 5V)



4.25.6 Pin Driver Strength

Figure 4-128. I/O Pin Output Voltage versus Sink Current (V_{CC} = 3V)

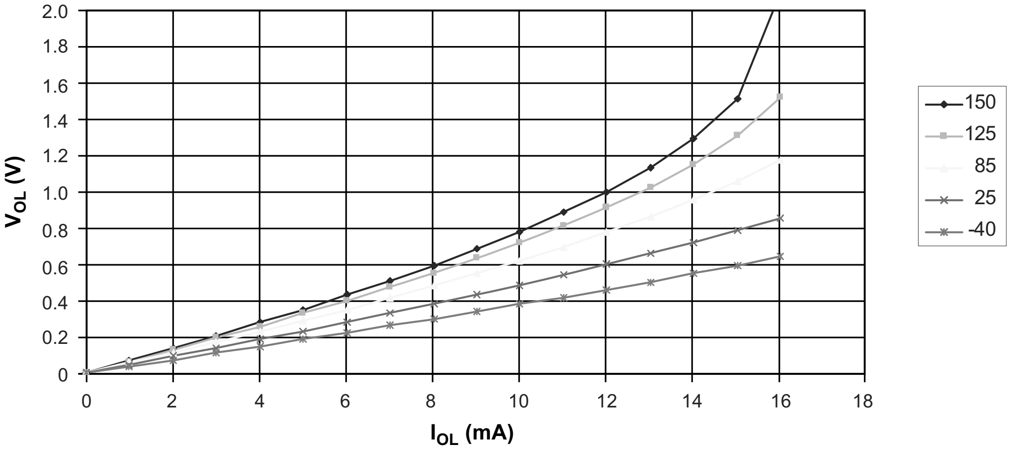
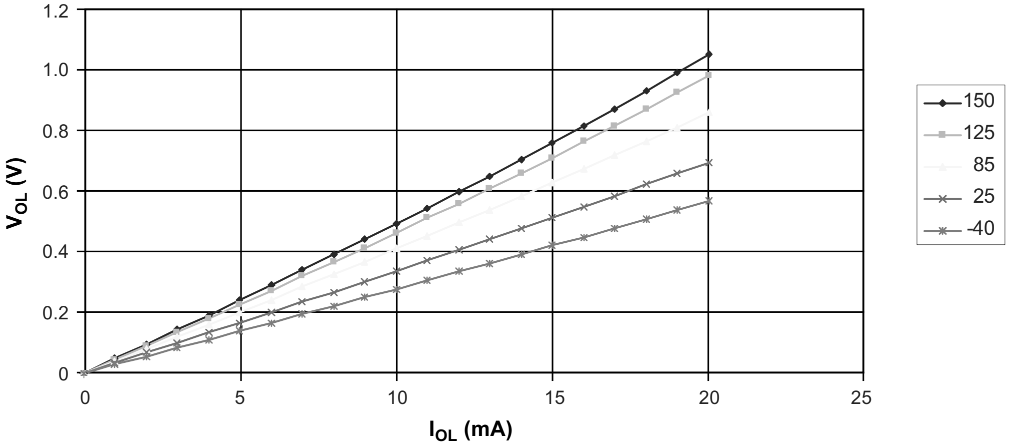


Figure 4-129. I/O Pin Output Voltage versus Sink Current (V_{CC} = 5V)



Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.