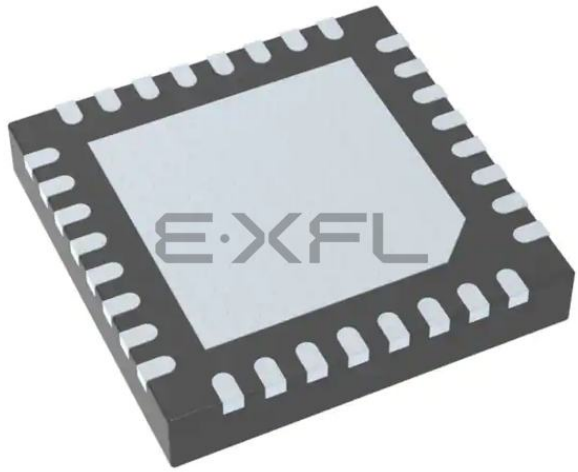


Welcome to E-XFL.COM

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	Obsolete
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	16KB (8K x 16)
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 ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny167-15md

1.7 Pin Description

1.7.1 Vcc

Supply voltage.

1.7.2 GND

Ground.

1.7.3 AVcc

Analog supply voltage.

1.7.4 AGND

Analog ground.

1.7.5 Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the Atmel® ATtiny87/167 as listed on Section 9.3.3 “Alternate Functions of Port A” on page 73.

1.7.6 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATtiny87/167 as listed on Section 9.3.4 “Alternate Functions of Port B” on page 78.

1.8 Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

1.9 About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

5.8 Minimizing Power Consumption

There are several possibilities to consider when trying to minimize the power consumption in an AVR® controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

5.8.1 Analog to Digital Converter

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to Section 17. "ADC – Analog to Digital Converter" on page 176 for details on ADC operation.

5.8.2 Analog Comparator

When entering idle mode, the analog comparator should be disabled if not used. When entering ADC noise reduction mode, the analog comparator should be disabled. In other sleep modes, the analog comparator is automatically disabled. However, if the analog comparator is set up to use the internal voltage reference as input, the analog comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to Section 18. "AnaComp - Analog Comparator" on page 194 for details on how to configure the Analog Comparator.

5.8.3 Brown-out Detector

If the brown-out detector is not needed by the application, this module should be turned off. If the brown-out detector is enabled by the BODLEVEL fuses, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to Section 6.1.5 "Brown-out Detection" on page 49 for details on how to configure the brown-out detector.

5.8.4 Internal Voltage Reference

The internal voltage reference will be enabled when needed by the brown-out detection, the analog comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to Section 6.2 "Internal Voltage Reference" on page 51 for details on the start-up time.

Output the internal voltage reference is not needed in the deeper sleep modes. This module should be turned off to reduce significantly to the total current consumption. Refer to Section 16.3.1 "AMISCR – Analog Miscellaneous Control Register" on page 175 for details on how to disable the internal voltage reference output.

5.8.5 Internal Current Source

The internal current source is not needed in the deeper sleep modes. This module should be turned off to reduce significantly to the total current consumption. Refer to Section 16.3.1 "AMISCR – Analog Miscellaneous Control Register" on page 175 for details on how to disable the internal current source.

5.8.6 Watchdog Timer

If the watchdog timer is not needed in the application, the module should be turned off. If the watchdog timer is enabled, it will be enabled in all sleep modes and hence always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to Section 6.3 "Watchdog Timer" on page 51 for details on how to configure the watchdog timer.

5.8.7 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section Section 9.2.6 "Digital Input Enable and Sleep Modes" on page 69 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to Vcc/2, the input buffer will use excessive power.

- **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.

5.9.3 PRR – Power Reduction Register

Bit	7	6	5	4	3	2	1	0	
	–	–	PRLIN	PRSPI	PRTIM1	PRTIM0	PRUSI	PRADC	PRR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 - Res: Reserved bit**

This bit is reserved in Atmel® ATtiny87/167 and will always read as zero.

- **Bit 6 - Res: Reserved bit**

This bit is reserved in Atmel ATtiny87/167 and will always read as zero.

- **Bit5 - PRLIN: Power Reduction LIN / UART controller**

Writing a logic one to this bit shuts down the LIN by stopping the clock to the module. When waking up the LIN again, the LIN should be re initialized to ensure proper operation.

- **Bit 4 - PRSPI: Power Reduction Serial Peripheral Interface**

If using debugWIRE on-chip debug system, this bit should not be written to one.

Writing a logic one to this bit shuts down the serial peripheral interface by stopping the clock to the module. When waking up the SPI again, the SPI should be re initialized to ensure proper operation.

- **Bit 3 - PRTIM1: Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the timer/counter1 module. When the timer/counter1 is enabled, operation will continue like before the shutdown.

- **Bit 2 - PRTIM0: Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the timer/counter0 module in synchronous mode (AS0 is 0). When the timer/counter0 is enabled, operation will continue like before the shutdown.

- **Bit 1 - PRUSI: Power Reduction USI**

Writing a logic one to this bit shuts down the USI by stopping the clock to the module. When waking up the USI again, the USI should be re-initialized to ensure proper operation.

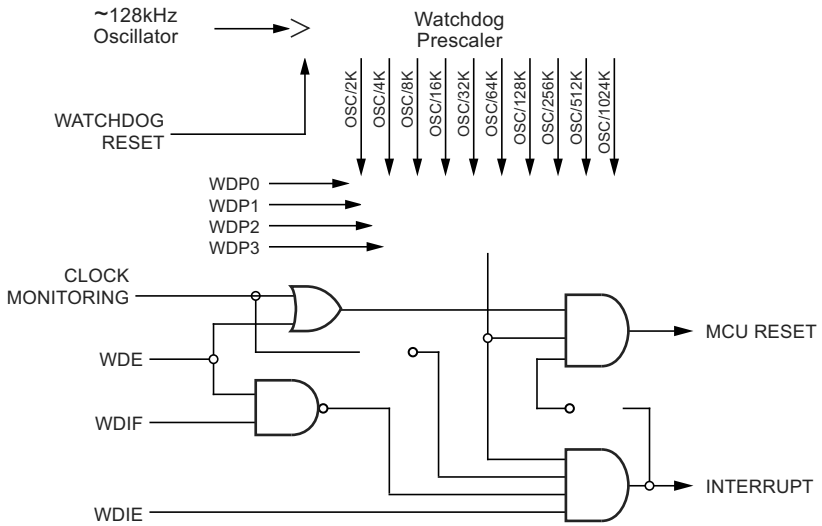
- **Bit 0 - PRADC: Power Reduction ADC**

Writing a logic one to this bit shuts down the ADC. The ADC must be disabled before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

6.3.1 Watchdog Timer Behavior

The watchdog timer (WDT) is a timer counting cycles of a separate on-chip 128KHz oscillator.

Figure 6-7. Watchdog Timer



The WDT gives an interrupt or a system reset when the counter reaches a given time-out value. In normal operation mode, it is required that the system uses the WDR - watchdog timer reset - instruction to restart the counter before the time-out value is reached. If the system doesn't restart the counter, an interrupt or system reset will be issued.

In interrupt mode, the WDT gives an interrupt when the timer expires. This interrupt can be used to wake the device from sleep-modes, and also as a general system timer. One example is to limit the maximum time allowed for certain operations, giving an interrupt when the operation has run longer than expected. In system reset mode, the WDT gives a reset when the timer expires. This is typically used to prevent system hang-up in case of runaway code. The third mode, interrupt and system reset mode, combines the other two modes by first giving an interrupt and then switch to system reset mode. This mode will for instance allow a safe shutdown by saving critical parameters before a system reset.

The watchdog always on (WDTON) fuse, if programmed, will force the watchdog timer to system reset mode. With the fuse programmed the system reset mode bit (WDE) and interrupt mode bit (WDIE) are locked to 1 and 0 respectively. To further ensure program security, alterations to the watchdog set-up must follow timed sequences. The sequence for clearing WDE and changing time-out configuration is as follows:

1. In the same operation, write a logic one to the watchdog change enable bit (WDCE) and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, write the WDE and watchdog prescaler bits (WDP) as desired, but with the WDCE bit cleared. This must be done in one operation.

7. Interrupts

This section describes the specifics of the interrupt handling as performed in Atmel® ATtiny87/167. For a general explanation of the AVR® interrupt handling, refer to “Reset and Interrupt Handling” on page 13.

7.1 Interrupt Vectors in ATtiny87/167

Table 7-1. Reset and Interrupt Vectors in ATtiny87/167

Vector Nb.	Program Address		Source	Interrupt Definition
	ATtiny87	ATtiny167		
1	0x0000	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0001	0x0002	INT0	External Interrupt Request 0
3	0x0002	0x0004	INT1	External Interrupt Request 1
4	0x0003	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0004	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x0005	0x000A	WDT	Watchdog Time-out Interrupt
7	0x0006	0x000C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	0x0007	0x000E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	0x0008	0x0010	TIMER1 COMPB	Timer/Coutner1 Compare Match B
10	0x0009	0x0012	TIMER1 OVF	Timer/Counter1 Overflow
11	0x000A	0x0014	TIMER0 COMPA	Timer/Counter0 Compare Match A
12	0x000B	0x0016	TIMER0 OVF	Timer/Counter0 Overflow
13	0x000C	0x0018	LIN TC	LIN/UART Transfer Complete
14	0x000D	0x001A	LIN ERR	LIN/UART Error
15	0x000E	0x001C	SPI, STC	SPI Serial Transfer Complete
16	0x000F	0x001E	ADC	ADC Conversion Complete
17	0x0010	0x0020	EE READY	EEPROM Ready
18	0x0011	0x0022	ANALOG COMP	Analog Comparator
19	0x0012	0x0024	USI START	USI Start Condition Detection
20	0x0013	0x0026	USI OVF	USI Counter Overflow

8. External Interrupts

8.1 Overview

The external interrupts are triggered by the INT1..0 pins or any of the PCINT15..0 pins. Observe that, if enabled, the interrupts will trigger even if the INT1..0 or PCINT15..0 pins are configured as outputs. This feature provides a way of generating a software interrupt.

The pin change interrupt PCINT1 will trigger if any enabled PCINT15..8 pin toggles. The pin change interrupt PCINT0 will trigger if any enabled PCINT7..0 pin toggles. The PCMSK1 and PCMSK0 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT15..0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

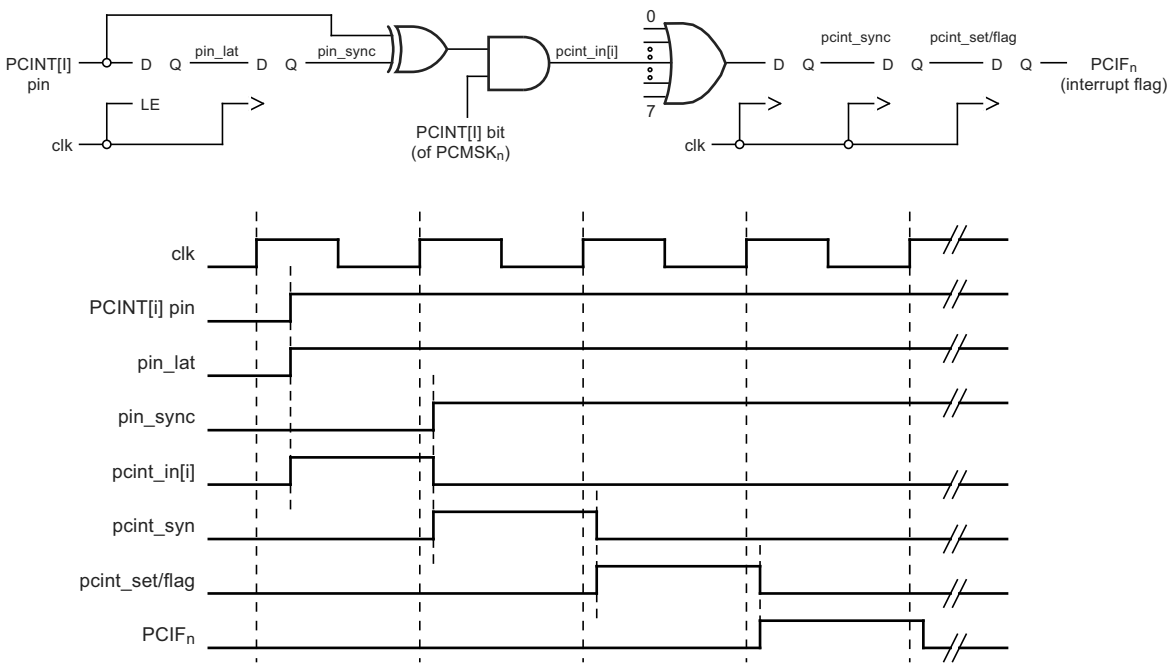
The INT1..0 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the external interrupt control register A – EICRA. When the INT1..0 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. The recognition of falling or rising edge interrupts on INT1..0 requires the presence of an I/O clock, described in Section 4.1 “Clock Systems and their Distribution” on page 25. Low level interrupts and the edge interrupt on INT1..0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except idle mode.

Note that if a level triggered interrupt is used for wake-up from power-down or power-save, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses as described in Section 4.1 “Clock Systems and their Distribution” on page 25.

8.2 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 8-1.

Figure 8-1. Timing of pin change interrupts



The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 register is not double buffered. This means that if ICR1 is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICR1 value written is lower than the current value of TCNT1. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR1A register however, is double buffered. This feature allows the OCR1A I/O location to be written anytime. When the OCR1A I/O location is written the value written will be put into the OCR1A buffer register. The OCR1A compare register will then be updated with the value in the buffer register at the next timer clock cycle the TCNT1 matches TOP. The update is done at the same timer clock cycle as the TCNT1 is cleared and the TOV1 flag is set.

Using the ICR1 register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed (by changing the TOP value), using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In fast PWM mode, the compare units allow generation of PWM waveforms on the OC1A/B pins. Setting the COM1x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1A/B1:0 to three (see Table 12-2 on page 124). The actual OC1A/B value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1A/B) and OC1A/Bi is set. The PWM waveform is generated by setting (or clearing) the OC1A/B register at the compare match between OCR1A/B and TCNT1, and clearing (or setting) the OC1A/B register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

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

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

The extreme values for the OCR1A/B register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1A/B is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR1A/B equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COM1A/B1:0 bits.)

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

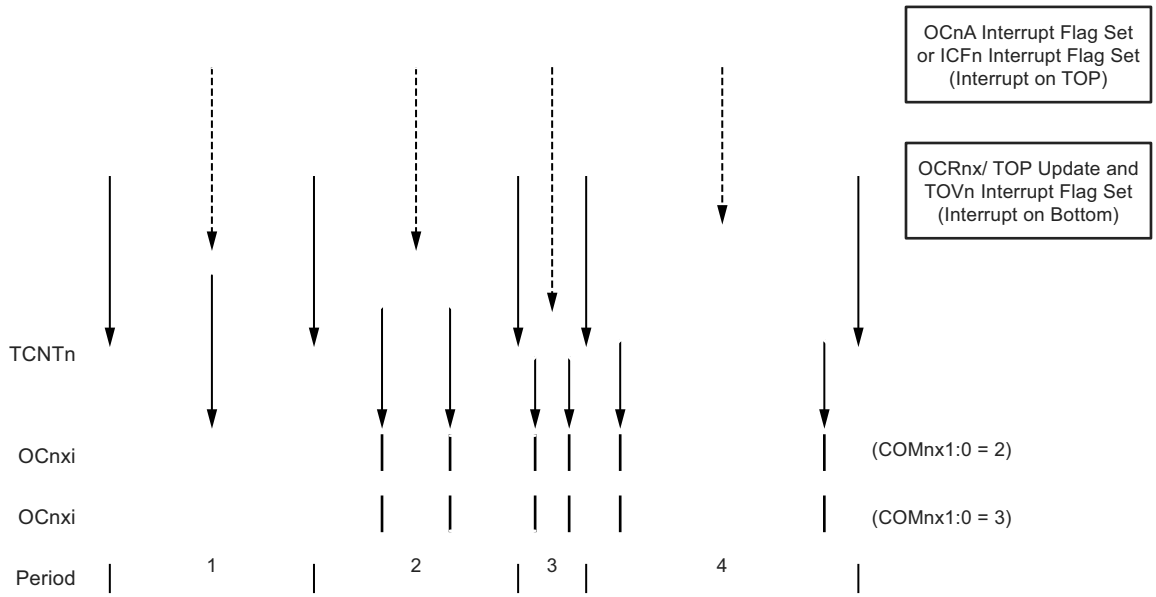
12.9.4 Phase Correct PWM Mode

The phase correct pulse width modulation or phase correct PWM mode (WGM13:0 = 1, 2, 3, 10, or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting compare output mode, the output compare (OC1A/B) is cleared on the compare match between TCNT1 and OCR1A/B while up counting, and set on the compare match while down counting. In inverting output compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

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

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

Figure 12-10.Phase and Frequency Correct PWM Mode, Timing Diagram



The timer/counter overflow flag (TOV1) is set at the same timer clock cycle as the OCR1A/B registers are updated with the double buffer value (at BOTTOM). When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 flag set when TCNT1 has reached TOP. The interrupt flags can then be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the compare registers. If the TOP value is lower than any of the compare registers, a compare match will never occur between the TCNT1 and the OCR1A/B.

As Figure 12-10 shows the output generated is, in contrast to the phase correct mode, symmetrical in all periods. Since the OCR1A/B registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.

Using the ICR1 register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed by changing the TOP value, using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In phase and frequency correct PWM mode, the compare units allow generation of PWM waveforms on the OC1A/B pins. Setting the COM1A/B1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1A/B1:0 to three (See Table on page 125). The actual OC1A/B value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1A/B) and OC1A/Bi is set. The PWM waveform is generated by setting (or clearing) the OC1A/B register at the compare match between OCR1A/B and TCNT1 when the counter increments, and clearing (or setting) the OC1A/B register at compare match between OCR1A/B and TCNT1 when the counter decrements. The PWM frequency for the output when using phase and frequency correct PWM can be calculated by the following equation:

$$f_{OCnxPFCPWM} = \frac{f_{\text{clk\_I/O}}}{2 \cdot N \cdot TOP}$$

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

The extreme values for the OCR1A/B register represents special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1A/B is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be set to high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

12.11 16-bit Timer/Counter Register Description

12.11.1 Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**

- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the output compare pins (OC1Ai and OC1Bi respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1Ai output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1Bi output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit and OC1xi bit (TCCR1D) corresponding to the OC1Ai or OC1Bi pin must be set in order to enable the output driver.

When the OC1Ai or OC1Bi is connected to the pin, the function of the COM1A/B1:0 bits is dependent of the WGM13:0 bits setting. Table 12-1 shows the COM1A/B1:0 bit functionality when the WGM13:0 bits are set to a Normal or a CTC mode (non-PWM).

Table 12-1. Compare Output Mode, non-PWM

OC1Ai OC1Bi	COM1A1 COM1B1	COM1A0 COM1B0	Description
0	x	x	Normal port operation, OC1A/OC1B disconnected.
1	0	0	
	0	1	Toggle OC1A/OC1B on compare match.
	1	0	Clear OC1A/OC1B on compare match (set output to low level).
	1	1	Set OC1A/OC1B on compare match (set output to high level).

Table 12-2 shows the COM1A/B1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 12-2. Compare Output Mode, Fast 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 Set OC1A/OC1B at TOP
1	1	1	Set OC1A/OC1B on compare match Clear OC1A/OC1B at TOP

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at TOP. See Section 12.9.3 “Fast PWM Mode” on page 117 for more details.

12.11.3 Timer/Counter1 Control Register C – TCCR1C

Bit	7	6	5	4	3	2	1	0	
	FOC1A	FOC1B	–	–	–	–	–	–	TCCR1C
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC1A: Force Output Compare for Channel A**

- **Bit 6 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. However, for ensuring compatibility with future devices, these bits must be set to zero when TCCR1A is written when operating in a PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the waveform generation unit. The OC1nx output is changed according to its COM1A/B1:0 and OC1nx bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1A/B1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in clear timer on compare match (CTC) mode using OCR1A as TOP.

The FOC1A/FOC1B bits are always read as zero.

12.11.4 Timer/Counter1 Control Register D – TCCR1D

Bit	7	6	5	4	3	2	1	0	
	OC1BX	OC1BW	OC1BV	OC1BU	OC1AX	OC1AW	OC1AV	OC1AU	TCCR1D
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 7:4 – OC1Bi: Output Compare Pin Enable for Channel B**

The OC1Bi bits enable the output compare pins of channel B as shown in Figure 12-6 on page 115.

- **Bit 3:0 – OC1Ai: Output Compare Pin Enable for Channel A**

The OC1Ai bits enable the output compare pins of channel A as shown in Figure 12-6 on page 115.

12.11.5 Timer/Counter1 – TCNT1H and TCNT1L

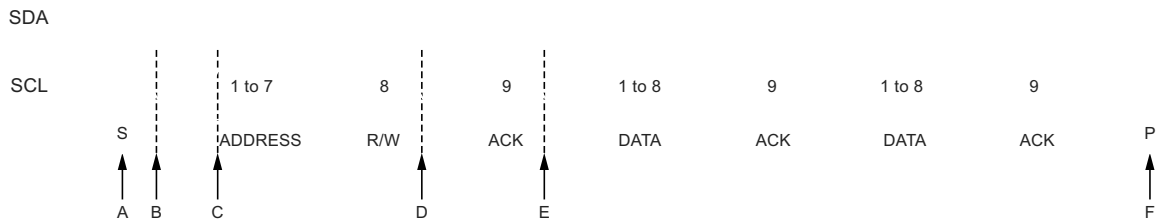
Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The two timer/counter I/O locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the timer/counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers, see Section 12.3 “Accessing 16-bit Registers” on page 105.

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1A/B registers.

Writing to the TCNT1 register blocks (removes) the compare match on the following timer clock for all compare units.

Figure 14-5. Two-wire Mode, Typical Timing Diagram

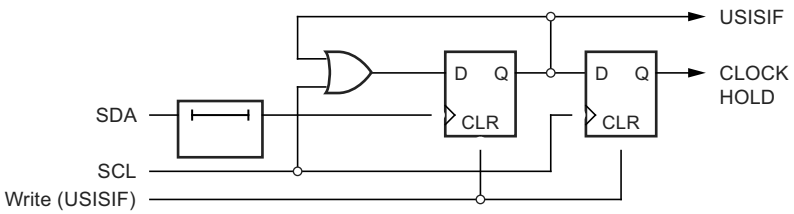


Referring to the timing diagram (Figure 14-5 on page 143), a bus transfer involves the following steps:

1. The a start condition is generated by the master by forcing the SDA low line while the SCL line is high (A). SDA can be forced low either by writing a zero to bit 7 of the shift register, or by setting the corresponding bit in the PORT register to zero. Note that the USI data register bit must be set to one for the output to be enabled. The slave device's start detector logic (Figure 14-6.) detects the start condition and sets the USISIF flag. The flag can generate an interrupt if necessary.
2. In addition, the start detector will hold the SCL line low after the master has forced an negative edge on this line (B). This allows the slave to wake up from sleep or complete its other tasks before setting up the USI data register to receive the address. This is done by clearing the start condition flag and reset the counter.
3. The master set the first bit to be transferred and releases the SCL line (C). The slave samples the data and shift it into the USI data register at the positive edge of the SCL clock.
4. After eight bits are transferred containing slave address and data direction (read or write), the slave counter overflows and the SCL line is forced low (D). If the slave is not the one the master has addressed, it releases the SCL line and waits for a new start condition.
5. If the slave is addressed it holds the SDA line low during the acknowledgment cycle before holding the SCL line low again (i.e., the counter register must be set to 14 before releasing SCL at (D)). Depending of the R/W bit the master or slave enables its output. If the bit is set, a master read operation is in progress (i.e., the slave drives the SDA line) The slave can hold the SCL line low after the acknowledge (E).
6. Multiple bytes can now be transmitted, all in same direction, until a stop condition is given by the master (F). Or a new start condition is given.

If the slave is not able to receive more data it does not acknowledge the data byte it has last received. When the master does a read operation it must terminate the operation by force the acknowledge bit low after the last byte transmitted.

Figure 14-6. Start Condition Detector, Logic Diagram



14.3.5 Start Condition Detector

The start condition detector is shown in Figure 14-6. The SDA line is delayed (in the range of 50 to 300 ns) to ensure valid sampling of the SCL line. The start condition detector is only enabled in two-wire mode.

The start condition detector is working asynchronously and can therefore wake up the processor from the power-down sleep mode. However, the protocol used might have restrictions on the SCL hold time. Therefore, when using this feature in this case the oscillator start-up time set by the CKSEL fuses (see Section 4.1 "Clock Systems and their Distribution" on page 25) must also be taken into the consideration. Refer to the USISIF bit description on page 145 for further details.

When the busy signal is set, some registers are locked, user writing is not allowed:

- “LIN control register” - LINCR - except LCMD[2..0], LENA and LSWRES,
- “LIN baud rate registers” - LINBRRL and LINBRRH,
- “LIN data length register” - LINDLR,
- “LIN identifier register” - LINIDR,
- “LIN data register” - LINDAT.

If the busy signal is set, the only available commands are:

- LCMD[1..0] = 00_b, the abort command is taken into account at the end of the byte,
- LENA = 0 and/or LCMD[2] = 0, the kill command is taken into account immediately,
- LSWRES = 1, the reset command is taken into account immediately.

Note that, if another command is entered during busy signal, the new command is not validated and the LOVRERR bit flag of the LINERR register is set. The on-going transfer is not interrupted.

15.5.5.2 Busy Signal in UART Mode

During the byte transmission, the busy signal is set. This locks some registers from being written:

- “LIN control register” - LINCR - except LCMD[2..0], LENA and LSWRES,
- “LIN data register” - LINDAT.

The busy signal is not generated during a byte reception.

15.5.6 Bit Timing

15.5.6.1 Baud rate Generator

The baud rate is defined to be the transfer rate in bits per second (bps):

- BAUD: Baud rate (in bps),
- $fclk_{i/o}$: System I/O clock frequency,
- LDIV[11..0]: Contents of LINBRRH & LINBRRL registers - (0-4095), the pre-scaler receives $clk_{i/o}$ as input clock.
- LBT[5..0]: Least significant bits of - LINBTR register- (0-63) is the number of samplings in a LIN or UART bit (default value 32).

Equation for calculating baud rate:

$$BAUD = fclk_{i/o} / LBT[5..0] \times (LDIV[11..0] + 1)$$

Equation for setting LINDIV value:

$$LDIV[11..0] = (fclk_{i/o} / LBT[5..0] \times BAUD) - 1$$

Note that in reception a majority vote on three samplings is made.

15.5.6.2 Re-synchronization in LIN Mode

When waiting for Rx header, LBT[5..0] = 32 in LINBTR register. The re-synchronization begins when the BREAK is detected. If the BREAK size is not in the range (10.5 bits min., 28 bits max. — 13 bits nominal), the BREAK is refused. The re-synchronization is done by adjusting LBT[5..0] value to the SYNCH field of the received header (0x55). Then the PROTECTED IDENTIFIER is sampled using the new value of LBT[5..0].

The re-synchronization implemented in the controller tolerates a clock deviation of ±20% and adjusts the baud rate in a ±2% range.

The new LBT[5..0] value will be used up to the end of the response. Then, the LBT[5..0] will be reset to 32 for the next header.

The LINBTR register can be used to (software) re-calibrate the clock oscillator.

The re-synchronization is not performed if the LIN node is enabled as a master.

- **Bits 5:4 - LDL[1:0]: LIN 1.3 Data Length**

In LIN 1.3 mode:

- 00 = 2-byte response,
- 01 = 2-byte response,
- 10 = 4-byte response,
- 11 = 8-byte response.

In UART mode this field is unused.

- **Bits 3:0 - LID[3:0]: LIN 1.3 Identifier**

In LIN 1.3 mode: 4-bit identifier.

In UART mode this field is unused.

- **Bits 5:0 - LID[5:0]: LIN 2.1 Identifier**

In LIN 2.1 mode: 6-bit identifier (no length transported).

In UART mode this field is unused.

15.6.9 LIN Data Buffer Selection Register - LINSEL

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	LAINC	LINDX2	LINDX1	LINDX0	LINSEL
Read/Write	-	-	-	-	R/W	R/W	R/W	R/W	
Initial Value	-	-	-	-	0	0	0	0	

- **Bits 7:4 - Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, they must be written to zero when LINSEL is written.

- **Bit 3 - $\overline{\text{LAINC}}$: $\overline{\text{Auto Increment}}$ of Data Buffer Index**

In LIN mode:

- 0 = Auto incrementation of FIFO data buffer index (default),
- 1 = No auto incrementation.

In UART mode this field is unused.

- **Bits 2:0 - LINDX 2:0: FIFO LIN Data Buffer Index**

In LIN mode: location (index) of the LIN response data byte into the FIFO data buffer. The FIFO data buffer is accessed through LINDAT.

In UART mode this field is unused.

17.11.3 ADCL and ADCH – The ADC Data Register

17.11.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

17.11.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers.

When ADCL is read, the ADC data register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in Section 17.8 “ADC Conversion Result” on page 186.

17.11.4 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7– BIN: Bipolar Input Mode**

The gain stage is working in the unipolar mode as default, but the bipolar mode can be selected by writing the BIN bit in the ADCSRB register. In the unipolar mode only one-sided conversions are supported and the voltage on the positive input must always be larger than the voltage on the negative input. Otherwise the result is saturated to the voltage reference. In the bipolar mode two-sided conversions are supported and the result is represented in the two’s complement form. In the unipolar mode the resolution is 10 bits and the bipolar mode the resolution is 9 bits + 1 sign bit.

- **Bit 3 – Res: Reserved Bit**

This bit is reserved for future use. For compatibility with future devices it must be written to zero when ADCSRB register is written.

- **Bits 5, 4 – ACIR1, ACIR0: Analog Comparator Internal Voltage Reference Select**

When ACIRS bit is set in ADCSRA register, these bits select a voltage reference for the negative input to the analog comparator, see Table 18-3 on page 197.

18.1.2 ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
	ACD	**ACIRS**	**ACO**	**ACI**	**ACIE**	**ACIC**	**ACIS1**	**ACIS0**	**ACSR**
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is written logic one, the power to the analog comparator is switched off. This bit can be set at any time to turn off the analog comparator. This will reduce power consumption in active and idle mode. When changing the ACD bit, the analog comparator interrupt must be disabled by clearing the ACIE bit of ACSR register. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACIRS: Analog Comparator Internal Reference Select**

When this bit is set an internal reference voltage replaces the negative input to the analog comparator (c.f. Table 18-3 on page 197). If ACIRS is cleared, AIN0 is applied to the negative input to the analog comparator.

- **Bit 5 – ACO: Analog Comparator Output**

The output of the analog comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The analog comparator interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is written logic one and the I-bit in the status register is set, the analog comparator interrupt is activated. When written logic zero, the interrupt is disabled.

- **Bit 2 – ACIC: Analog Comparator Input Capture Enable**

When written logic one, this bit enables the input capture function in Timer/counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the input capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/counter1 Input capture interrupt. When written logic zero, no connection between the analog comparator and the input capture function exists. To make the comparator trigger the Timer/counter1 input capture interrupt, the ICIE1 bit in the timer interrupt mask register (TIMSK1) must be set.

- **Bits 1, 0 – ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the analog comparator interrupt. The different settings are shown in Table 18-1.

Table 18-1. ACIS1 / ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator interrupt on output toggle.
0	1	Reserved
1	0	Comparator interrupt on falling output edge.
1	1	Comparator interrupt on rising output edge.

Note: When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its interrupt enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

19.4 Software Break Points

DebugWIRE supports program memory break points by the AVR® BREAK instruction. Setting a break point in Atmel® AVR Studio® will insert a BREAK instruction in the program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The flash must be re-programmed each time a break point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of break points will therefore reduce the flash data retention. Devices used for debugging purposes should not be shipped to end customers.

19.5 Limitations of DebugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as external reset (RESET). An external reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e., when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O registers via the debugger (AVR Studio).

A programmed DWEN fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

19.6 DebugWIRE Related Register in I/O Memory

The following section describes the registers used with the debugWire.

19.6.1 DebugWIRE Data Register – DWDR

Bit	7	6	5	4	3	2	1	0	
	DWDR[7:0]								DWDR
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 DWDR register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

20.2.1 Store Program Memory Control and Status Register – SPMCSR

The store program memory control and status register contains the control bits needed to control the boot loader operations.

Bit	7	6	5	4	3	2	1	0	
	–	**RWWSB**	SIGRD	CTPB	**RFLB**	**PGWRT**	**PGERS**	**SPMEN**	**SPMCSR**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the Atmel® ATtiny87/167 and will always read as zero.

- **Bit 6 – RWWSB: Read-While-Write Section Busy**

This bit is for compatibility with devices supporting read-while-write. It will always read as zero in Atmel ATtiny87/167.

- **Bit 5 – SIGRD: Signature Row Read**

If this bit is written to one at the same time as SPMEN, the next LPM instruction within three clock cycles will read a byte from the signature row into the destination register. See Section 20.2.4 “Reading the Signature Row from Software” on page 204 for details. An SPM instruction within four cycles after SIGRD and SPMEN are set will have no effect.

- **Bit 4 – CTPB: Clear Temporary Page Buffer**

If the CTPB bit is written while filling the temporary page buffer, the temporary page buffer will be cleared and the data will be lost.

- **Bit 3 – RFLB: Read Fuse and Lock Bits**

An LPM instruction within three cycles after RFLB and SPMEN are set in the SPMCSR register, will read either the lock bits or the fuse bits (depending on Z0 in the Z-pointer) into the destination register. See Section 20.2.3 “Reading the Fuse and Lock Bits from Software” on page 203 for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes page write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGWRT bit will auto-clear upon completion of a page write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes page erase. The page address is taken from the high part of the Zpointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a page erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation.

- **Bit 0 – SPMEN: Self Programming Enable**

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either SIGRD, CTPB, RFLB, PGWRT, or PGERS, the following SPM instruction will have a special meaning, see description above. If only SPMEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SPMEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During page erase and page write, the SPMEN bit remains high until the operation is completed.

Writing any other combination than “$10\ 0001_b$”, “$01\ 0001_b$”, “$00\ 1001_b$”, “$00\ 0101_b$”, “$00\ 0011_b$” or “$00\ 0001_b$” in the lower six bits will have no effect.

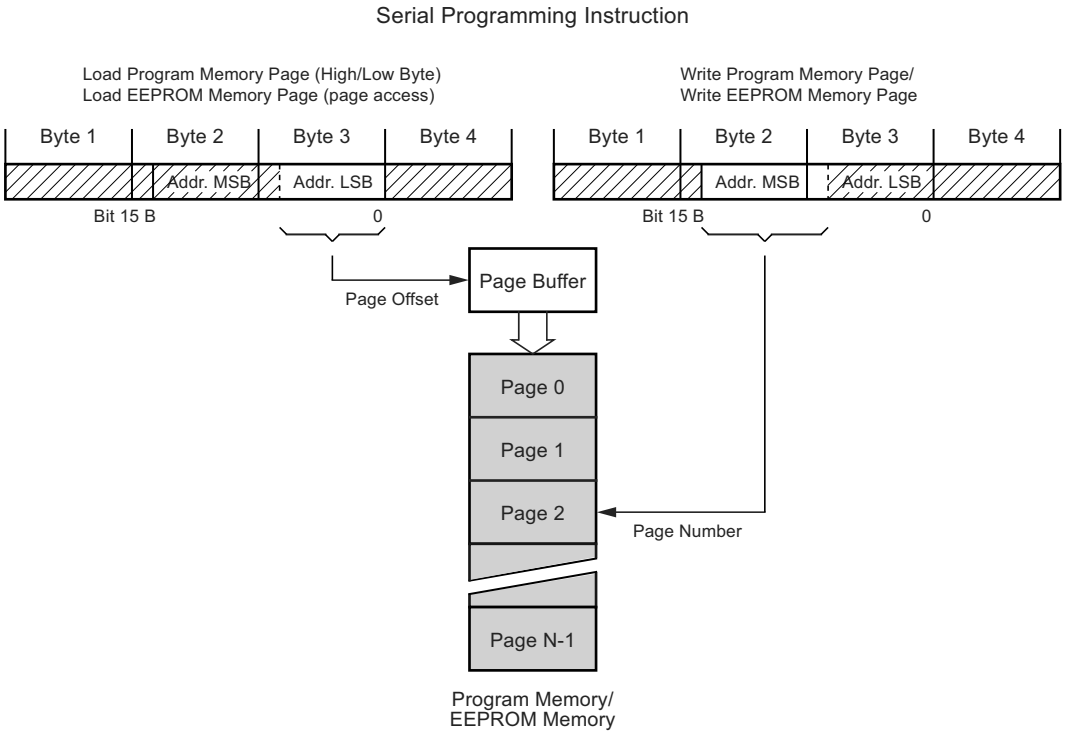
Note: Only one SPM instruction should be active at any time.

If the LSB in RDY/BSY data byte out is ‘1’, a programming operation is still pending. Wait until this bit returns ‘0’ before the next instruction is carried out.

Within the same page, the low data byte must be loaded prior to the high data byte.

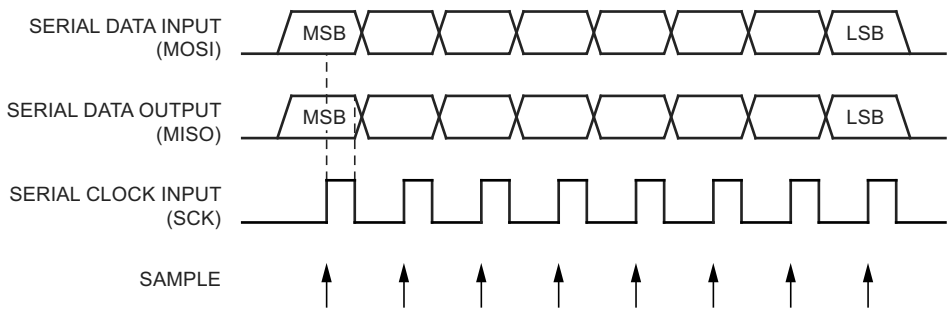
After data is loaded to the page buffer, program the EEPROM page, see Figure 21-8.

Figure 21-8. Serial programming Instruction Example



21.9 Serial Programming Characteristics

Figure 21-9. Serial Programming Waveforms



For characteristics of the SPI module, see Section 22.10 “SPI Timing Characteristics” on page 231

Table 22-5. BODLEVEL Fuse Coding

BODLEVEL 2:0 Fuses	Min. V_{BOT} (1)	Typ. V_{BOT}	Max. V_{BOT}	Units
$1\ 1\ 1_b$	BOD Disabled			
$1\ 1\ 0_b$	1.7	1.8	2.0	V
$1\ 0\ 1_b$	2.5	2.7	2.9	
$1\ 0\ 0_b$	4.1	4.3	4.5	
$0\ 1\ 1_b$	Reserved			
$0\ 1\ 0_b$				
$0\ 0\ 1_b$				
$0\ 0\ 0_b$				

Notes: 1. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to Vcc = V_{BOT} during the production test. This guarantees that a brown-out reset will occur before Vcc drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 101 for low operating voltage and BODLEVEL = 100 for high operating voltage.

Table 22-6. Brown-out Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Units
Brown-out detector hysteresis	V_{HYST}		80		mV
Min pulse width on brown-out reset	t_{BOD}		2		µs

22.6 Internal Voltage Characteristics

Table 22-7. Internal Voltage Reference Characteristics

Parameter	Condition	Symbol	Min.	Typ.	Max.	Units
Bandgap reference voltage	Vcc = 4.5 T_A = 25°C	V_{BG}	1.0	1.1	1.2	V
Bandgap reference start-up time	Vcc = 4.5 T_A = 25°C	t_{BG}		40	70	µs
Bandgap reference current consumption	Vcc = 4.5 T_A = 25°C	I_{BG}		10		µA

22.7 Current Source Characteristics

Table 22-8. Current Source Characteristics

Parameter	Condition	Symbol	Min.	Typ.	Max.	Units
Current	Vcc = 2.7 V/5.5 V T = –40°C/+125°C	I_{ISRC}	94		106	µA
Current Source start-up time	Vcc = 4.5 T_A = 25°C	t_{ISRC}		60		µs

25. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xDB)	Reserved									
(0xDA)	Reserved									
(0xD9)	Reserved									
(0xD8)	Reserved									
(0xD7)	Reserved									
(0xD6)	Reserved									
(0xD5)	Reserved									
(0xD4)	Reserved									
(0xD3)	Reserved									
(0xD2)	LINDAT	LDATA7	LDATA6	LDATA5	LDATA4	LDATA3	LDATA2	LDATA1	LDATA0	171
(0xD1)	LINSEL	–	–	–	–	/LAINC	LINDX2	LINDX1	LINDX0	171
(0xD0)	LINIDR	LP1	LP0	LID5 / LDL1	LID4 / LDL0	LID3	LID2	LID1	LID0	170
(0xCF)	LINDLR	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	170
(0xCE)	LINBRRH	–	–	–	–	LDIV11	LDIV10	LDIV9	LDIV8	170
(0xCD)	LINBRRL	LDIV7	LDIV6	LDIV5	LDIV4	LDIV3	LDIV2	LDIV1	LDIV0	170
(0xCC)	LINBTR	LDISR	–	LBT5	LBT4	LBT3	LBT2	LBT1	LBT0	169
(0xCB)	LINERR	LABORT	LTOERR	LOVERR	LFERR	LSERR	LPERR	LCERR	LBERR	168
(0xCA)	LINENIR	–	–	–	–	LENERR	LENIDOK	LENTXOK	LENRXOK	168
(0xC9)	LINSIR	LIDST2	LIDST1	LIDST0	LBUSY	LERR	LIDOK	LTXOK	LRXOK	167
(0xC8)	LINCR	LSWRES	LIN13	LCONF1	LCONF0	LENA	LCMD2	LCMD1	LCMD0	166
(0xC7)	Reserved									
(0xC6)	Reserved									
(0xC5)	Reserved									
(0xC4)	Reserved									
(0xC3)	Reserved									
(0xC2)	Reserved									
(0xC1)	Reserved									
(0xC0)	Reserved									
(0xBF)	Reserved									
(0xBE)	Reserved									
(0xBD)	Reserved									
(0xBC)	USIPP								USIPOS	148
(0xBB)	USIBR	USIB7	USIB6	USIB5	USIB4	USIB3	USIB2	USIB1	USIB0	145
(0xBA)	USIDR	USID7	USID6	USID5	USID4	USID3	USID2	USID1	USID0	144
(0xB9)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	145
(0xB8)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	146

Notes:
1. Address bits exceeding EEAMSB (Table 21-8 on page 210) are don't care.
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
4. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
5. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The Atmel® ATtiny87/167 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in opcode for the IN and OUT instructions. For the extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.