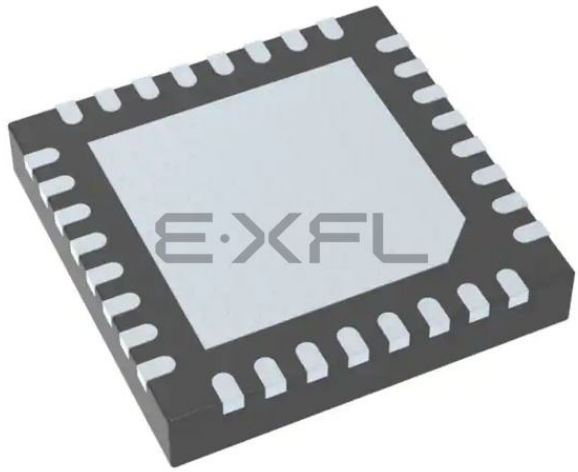


Welcome to E-XFL.COM

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 (4K 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 ~ 125°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/attiny87-a15mz

- Special microcontroller features
  - Dynamic clock switching (external/Internal RC/watchdog clock) for power control, EMC reduction
  - Debug WIRE on-chip debug (OCD) system
  - Hardware in-system programmable (ISP) via SPI port
  - External and internal interrupt sources
  - Interrupt and wake-up on pin change
  - Low power idle, ADC Noise reduction, and power-down modes
  - Enhanced power-on reset circuit
  - Programmable brown-out detection circuit
  - Internal calibrated RC oscillator 8MHz
  - 4-16MHz and 32KHz crystal/ceramic resonator oscillators
- I/O and packages
  - 16 programmable I/O lines
  - 20-pin SOIC, 32-pad QFN and 20-pin TSSOP
- Operating voltage:
  - 2.7 - 5.5V for ATtiny87/167
- Speed grade:
  - 0 - 8MHz at 2.7 - 5.5V (automotive temp. range: –40°C to +125°C)
  - 0 - 16MHz at 4.5 - 5.5V (automotive temp. range: –40°C to +125°C)

3.5.2 EEDR – EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
	EEDR7	EEDR6	EEDR5	EEDR4	EEDR3	EEDR2	EEDR1	EEDR0	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:0 – EEDR7:0: EEPROM Data**

For the EEPROM write operation the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

3.5.3 EECR – EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

- **Bit 7,6 – Res: Reserved Bits**

These bits are reserved for future use and will always read as 0 in ATtiny87/167. After reading, mask out these bits. For compatibility with future AVR® devices, always write these bits to zero.

- **Bits 5, 4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits**

The EEPROM programming mode bits setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the erase and write operations in two different operations. The programming times for the different modes are shown in Table 3-2. While EEPE is set, any write to EEPMn will be ignored. During reset, the EEPMn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 3-2. EEPROM Mode Bits

EEPM1	EEPM0	Typical Programming Time	Operation
0	0	3.4ms	Erase and write in one operation (atomic operation)
0	1	1.8ms	Erase only
1	0	1.8ms	Write only
1	1	–	Reserved for future use

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM ready interrupt if the I-bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM ready interrupt generates a constant interrupt when Non-volatile memory is ready for programming.

- **Bit 2 – EEMPE: EEPROM Master Program Enable**

The EEMPE bit determines whether writing EEPE to one will have effect or not.

When EEMPE is set, setting EEPE within four clock cycles will program the EEPROM at the selected address. If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles.

7.2 Program Setup in ATtiny87

The most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATtiny87 is (2-byte step - using “rjmp” instruction):

```
Address(1) Label Code                 Comments
0x0000           rjmp   RESET          ; Reset Handler
0x0001           rjmp   INT0addr       ; IRQ0 Handler
0x0002           rjmp   INT1addr       ; IRQ1 Handler
0x0003           rjmp   PCINT0addr     ; PCINT0 Handler
0x0004           rjmp   PCINT1addr     ; PCINT1 Handler
0x0005           rjmp   WDTaddr        ; Watchdog Timer Handler
0x0006           rjmp   ICP1addr       ; Timer1 Capture Handler
0x0007           rjmp   OC1Aaddr       ; Timer1 Compare A Handler
0x0008           rjmp   OC1Baddr       ; Timer1 Compare B Handler
0x0009           rjmp   OVF1addr       ; Timer1 Overflow Handler
0x000A           rjmp   OC0Aaddr       ; Timer0 Compare A Handler
0x000B           rjmp   OVF0addr       ; Timer0 Overflow Handler
0x000C           rjmp   LINTCaddr      ; LIN Transfer Complete Handler
0x000D           rjmp   LINERRaddr     ; LIN Error Handler
0x000E           rjmp   SPIaddr        ; SPI Transfer Complete Handler
0x000F           rjmp   ADCCaddr       ; ADC Conversion Complete Handler
0x0010           rjmp   ERDYaddr       ; EEPROM Ready Handler
0x0011           rjmp   ACIaddr        ; Analog Comparator Handler
0x0012           rjmp   USISTARTaddr   ; USI Start Condition Handler
0x0013           rjmp   USIOVFaddr     ; USI Overflow Handler

0x0014 RESET:    ldi    r16, high(RAMEND); Main program start
0x0015           out    SPH,r16          ; Set Stack Pointer to top of RAM
0x0016           ldi    r16, low(RAMEND)
0x0017           out    SPL,r16
0x0018           sei                     ; Enable interrupts
0x0019           <instr>  xxx
   ...     ...    ...    ...
```

Note: 1. 16-bit address

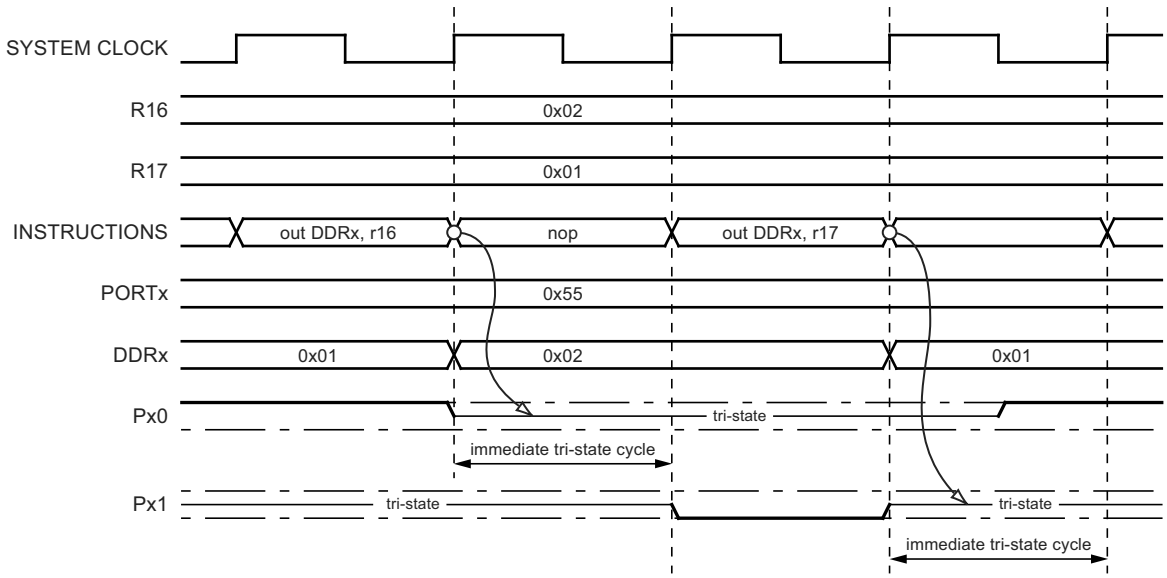
9.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI assembler instruction can be used to toggle one single bit in a port.

9.2.3 Break-before-make Switching

In the break-before-make mode when switching the DDRxn bit from input to output an immediate tri-state period lasting one system clock cycle is introduced as indicated in Figure 9-3. For example, if the system clock is 4MHz and the DDRxn is written to make an output, the immediate tri-state period of 250ns is introduced, before the value of PORTxn is seen on the port pin. To avoid glitches it is recommended that the maximum DDRxn toggle frequency is two system clock cycles. The break-before-make is a port-wise mode and it is activated by the port-wise BBMx enable bits. For further information about the BBMx bits, see Section 9.3.2 “Port Control Register – PORTCR” on page 72. When switching the DDRxn bit from output to input there is no immediate tri-state period introduced.

Figure 9-3. Break Before Make, Switching Between Input and Output



9.2.4 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0, 0) and output high ({DDxn, PORTxn} = 1, 1), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0, 1) or output low ({DDxn, PORTxn} = 1, 0) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impudent environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register or the PUDx bit in PORTCR register can be set to disable all pull-ups in the port.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0, 0) or the output high state ({DDxn, PORTxn} = 1, 1) as an intermediate step.

9.2.6 Digital Input Enable and Sleep Modes

As shown in Figure 9-2, the digital input signal can be clamped to ground at the input of the Schmitt Trigger. The signal denoted SLEEP in the figure, is set by the MCU sleep controller in power-down or power-save mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to Vcc/2.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in Section 9.3 “Alternate Port Functions” on page 70.

If a logic high level (“one”) is present on an asynchronous external interrupt pin configured as “interrupt on rising edge, falling edge, or any logic change on pin” while the external interrupt is **not** enabled, the corresponding external Interrupt flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

9.2.7 Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (reset, active mode and idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to Vcc or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

Table 9-4 and Table 9-5 on page 77 relate the alternate functions of Port A to the overriding signals shown in Figure 9-6 on page 70.

Table 9-4. Overriding Signals for Alternate Functions in PA7..PA4

Signal Name	PA7/PCINT7/ ADC7/AIN1 /XREF/AREF	PA6/PCINT6/ ADC6/AIN0/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

- **PCINT10/OC1AV/USCK/SCL – Port B, Bit 2**

PCINT10: pin change interrupt, source 10.

OC1AV: output compare and PWM Output A-V for timer/counter1. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1AV pin is also the output pin for the PWM mode timer function (c.f. OC1AV bit of TCCR1D register).

USCK: three-wire mode USI clock input.

SCL: two-wire mode USI clock input.

- **PCINT9/OC1BU/DO – Port B, Bit 1**

PCINT9: pin change interrupt, source 9.

OC1BU: output compare and PWM output B-U for timer/counter1. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1BU pin is also the output pin for the PWM mode timer function (c.f. OC1BU bit of TCCR1D register).

DO: three-wire mode USI data output. Three-wire mode data output overrides PORTB1 and it is driven to the port when the data direction bit DDB1 is set. PORTB1 still enables the pull-up, if the direction is input and PORTB1 is set (one).

- **PCINT8/OC1AU/DI/SDA – Port B, Bit 0**

IPCINT8: pin change interrupt, source 8.

OC1AU: output compare and PWM output A-U for timer/counter1. The PB0 pin has to be configured as an output (DDB0 set (one)) to serve this function. The OC1AU pin is also the output pin for the PWM mode timer function (c.f. OC1AU bit of TCCR1D register).

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.

Table 9-7 and Table 9-8 on page 81 relate the alternate functions of Port B to the overriding signals shown in Figure 9-6 on page 70.

Table 9-7. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/PCINT15/ADC10/ OC1BX/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

9.4 Register Description for I/O Ports

9.4.1 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	

9.4.2 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	

9.4.3 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	

9.4.4 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	

9.4.5 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	

9.4.6 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	

10.2.1 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.

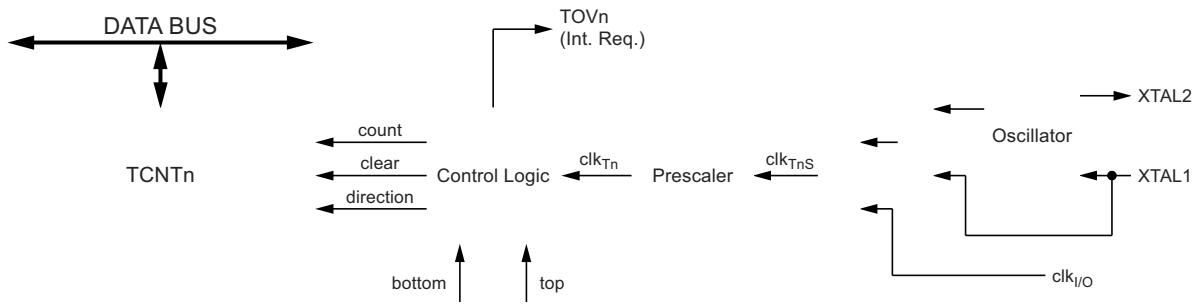
10.3 Timer/Counter Clock Sources

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 10.11.4 “Asynchronous Status Register – ASSR” on page 98. For details on clock sources and prescaler, see Section 10.10 “Timer/Counter0 Prescaler” on page 95.

10.4 Counter Unit

The main part of the 8-bit timer/counter is the programmable bi-directional counter unit. Figure 10-2 shows a block diagram of the counter and its surrounding environment.

Figure 10-2. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
direction	Selects between increment and decrement.
clear	Clear TCNT0 (set all bits to zero).
clk_T0	Timer/Counter0 clock.
top	Signalizes that TCNT0 has reached maximum value.
bottom	Signalizes that TCNT0 has reached minimum value (zero).

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_T0). clk_T0 can be generated from an external or internal clock source, selected by the clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_T0 is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the timer/counter control register (TCCR0A). There are close connections between how the counter behaves (counts) and how waveforms are generated on the output compare output OC0A. For more details about advanced counting sequences and waveform generation, see Section 10.7 “Modes of Operation” on page 88.

The timer/counter overflow flag (TOV0) is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing the TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC0A output can be set to toggle its logical level on each compare match by setting the compare output mode bits to toggle mode (COM0A1:0 = 1). The OC0A value will not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC}0_A = f_{clk\_I/O}/2$ when OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

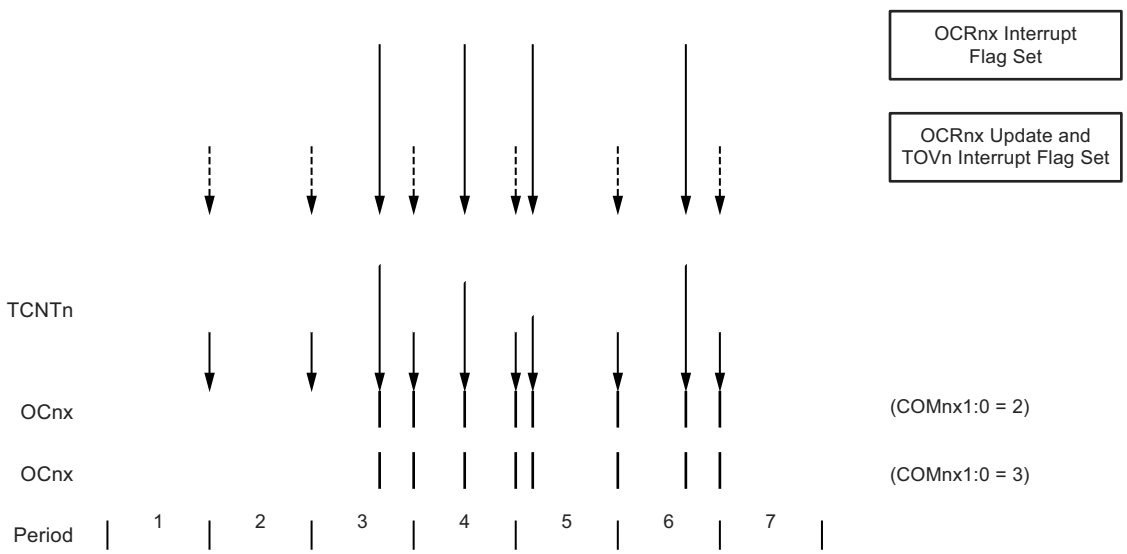
As for the normal mode of operation, the TOV0 flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

10.7.3 Fast PWM Mode

The fast pulse width modulation or fast PWM mode (WGM01:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM. In non-inverting compare output mode, the output compare (OC0A) is cleared on the compare match between TCNT0 and OCR0A, and set at BOTTOM. In inverting compare output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the MAX value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 10-6. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0A and TCNT0.

Figure 10-6. Fast PWM Mode, Timing Diagram



The timer/counter overflow flag (TOV0) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

Table 10-1. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match.
1	1	Set OC0A on Compare Match.

Table 10-2 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 10-2. Compare Output Mode, Fast PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	
1	0	Clear OC0A on compare match. Set OC0A at BOTTOM (non-inverting mode).
1	1	Set OC0A on compare match. Clear OC0A at BOTTOM (inverting mode).

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 10.7.3 “Fast PWM Mode” on page 89 for more details.

Table 10-3 shows the COM01:0 bit functionality when the WGM01:0 bits are set to phase correct PWM mode

.

Table 10-3. Compare Output Mode, Phase Correct PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	
1	0	Clear OC0A on compare match when up-counting. Set OC0A on compare match when down-counting.
1	1	Set OC0A on compare match when up-counting. Clear OC0A on compare match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See Section 10.7.4 “Phase Correct PWM Mode” on page 90 for more details.

- **Bit 5:2 – Res: Reserved Bits**

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

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:

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

- **Bit 4 - LFERR: Framing Error Flag**
  - 0 = No error,
  - 1 = Framing error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 3 - LSERR: Synchronization Error Flag**
  - 0 = No error,
  - 1 = Synchronization error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 2 - LPERR: Parity Error Flag**
  - 0 = No error,
  - 1 = Parity error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 1 - LCERR: Checksum Error Flag**
  - 0 = No error,
  - 1 = Checksum error.

This bit is cleared when LERR bit in LINSIR is cleared.

- **Bit 0 - LBERR: Bit Error Flag**
  - 0 = no error,
  - 1 = Bit error.

This bit is cleared when LERR bit in LINSIR is cleared.

15.6.5 LIN Bit Timing Register - LINBTR

Bit	7	6	5	4	3	2	1	0	
	LDISR	**-**	**LBT5**	**LBT4**	**LBT3**	**LBT2**	**LBT1**	**LBT0**	**LINBTR**
Read/Write	R/W	R	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 - LDISR: Disable Bit Timing Re synchronization**
  - 0 = Bit timing re-synchronization enabled (default),
  - 1 = Bit timing re-synchronization disabled.

- **Bits 5:0 - LBT[5:0]: LIN Bit Timing**

Gives the number of samples of a bit.
Sample-time = (1 / $f_{clk_{i/o}}$) x (LDIV[11..0] + 1)

Default value: LBT[6:0]=32 — Min. value: LBT[6:0]=8 — Max. value: LBT[6:0]=63

Figure 17-6. ADC Timing Diagram, Auto Triggered Conversion

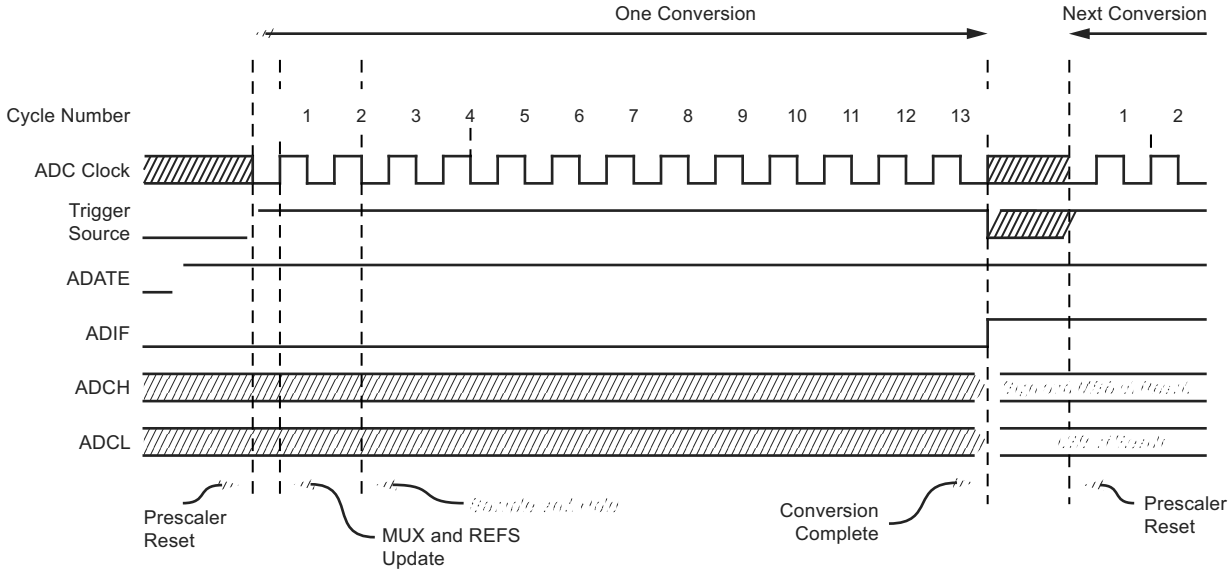


Figure 17-7. ADC Timing Diagram, Free Running Conversion

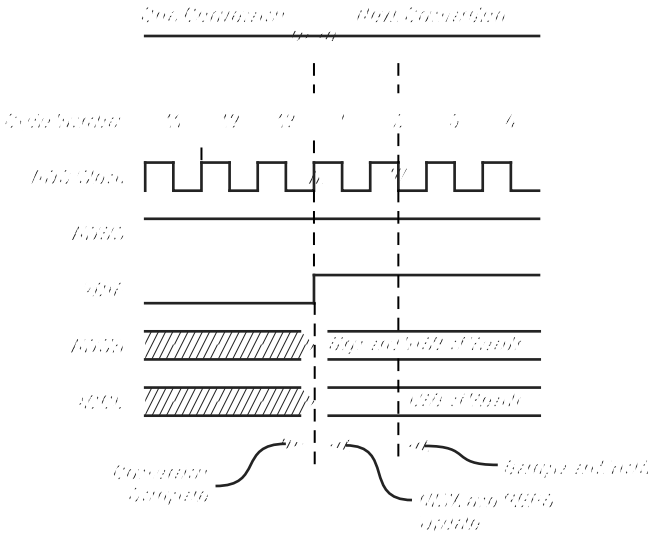


Table 17-1. ADC Conversion Time

Condition	Sample and Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5 cycles	25 cycles
Normal conversions	1.5 cycles	13 cycles
Auto Triggered conversions	2 cycles	13.5 cycles

17.9 Temperature Measurement

The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC input. MUX[4..0] bits in ADMUX register enables the temperature sensor. The internal 1.1V voltage reference must also be selected for the ADC voltage reference source in the temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

The measured voltage has a linear relationship to the temperature as described in Table 17-2 on page 187. The voltage sensitivity is approximately 1 LSB/°C and the accuracy of the temperature measurement is ±10°C using manufacturing calibration values (TS_GAIN, TS_OFFSET). The values described in Table 17-2 on page 187 are typical values. However, due to the process variation the temperature sensor output varies from one chip to another.

Table 17-2. Temperature versus Sensor Output Voltage (Typical Case): Example ADC Values

Temperature/°C	–40°C	+25°C	+85°C
	0x00F6	0x0144	0c01B8

17.9.1 Manufacturing Calibration

Calibration values determined during test are available in the signature row.

The temperature in degrees celsius can be calculated using the formula:

$$T = \frac{([(ADCH \ll 8) | ADCL] - (273 + 25 - TS\_OFFSET)) \times 128}{TS\_GAIN} + 25$$

Where:

a. ADCH and ADCL are the ADC data registers,

b. is the temperature sensor gain

c. TSOFFSET is the temperature sensor offset correction term
TS_GAIN is the unsigned fixed point 8-bit temperature sensor gain factor in 1/128th units stored in the signature row
TS_OFFSET is the signed twos complement temperature sensor offset reading stored in the signature row. See Table 20-1 on page 204 for signature row parameter address.

The following code example allows to read signature row data:

```
.equ TS_GAIN = 0x0007
.equ TS_OFFSET = 0x0005
LDI R30,LOW(TS_GAIN)
LDI R31,HIGH (TS_GAIN)
RCALL Read_signature_row
MOV R17,R16; Save R16 result
LDI R30,LOW(TS_OFFSET)
LDI R31,HIGH (TS_OFFSET)
RCALL Read_signature_row
; R16 holds TS_OFFSET and R17 holds TS_GAIN
Read_signature_row:
IN R16,SPMCSR ; Wait for SPMEN ready
SBRC R16,SPMEN ; Exit loop here when SPMCSR is free
RJMP Read_signature_row
LDI R16,((1<<SIGRD)|(1<<SPMEN)); We need to set SIGRD and SPMEN
together
OUT SPMCSR,R16 ; and execute the LPM within 3 cycles
LPM R16,Z
RET
```

Table 21-9. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
$\overline{WR}$	PB0	I	Write pulse (active low).
XA0	PB1	I	XTAL1 action bit 0
XA1 / BS2	PB2	I	- XTAL1 action bit 1 - Byte select 2 ("0" selects low byte, "1" selects 2'nd high byte)
PAGEL / BS1	PB3	I	- Program memory and EEPROM data page load - Byte select 1 ("0" selects low byte, "1" selects high byte)
	PB4	I	XTAL1 (clock input)
$\overline{OE}$	PB5	I	Output enable (active low).
RDY / $\overline{BSY}$	PB6	O	0: Device is busy programming, 1: Device is ready for new command.
+12V	PB7	I	- Reset (active low) - Parallel programming mode (+12V).
DATA	PA7-PA0	I/O	Bi-directional data bus (output when $\overline{OE}$ is low).

Table 21-10. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL / BS1	Prog_enable[3]	0
XA1 / BS2	Prog_enable[2]	0
XA0	Prog_enable[1]	0
$\overline{WR}$	Prog_enable[0]	0

Table 21-11. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load flash or EEPROM address (high or low address byte determined by BS1).
0	1	Load data (high or low data byte for flash determined by BS1).
1	0	Load command
1	1	No action, idle

Table 21-12. Command Byte Bit Coding

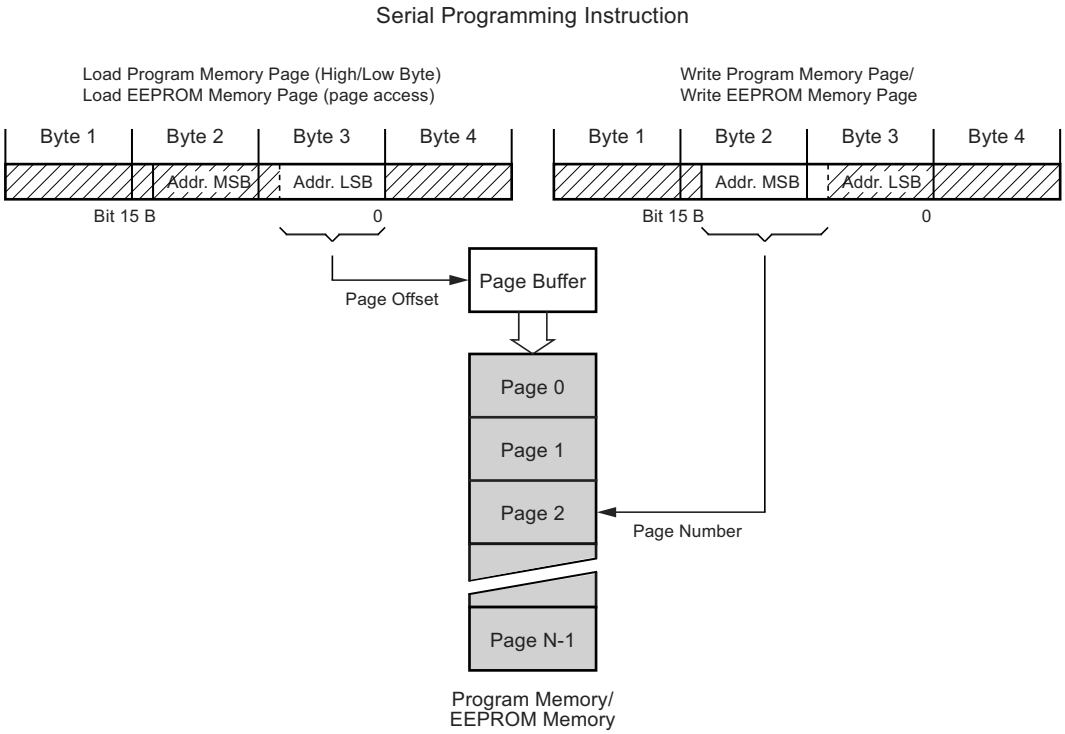
Command Byte	Command Executed
$1000\ 0000_b$	Chip erase
$0100\ 0000_b$	Write fuse bits
$0010\ 0000_b$	Write lock bits
$0001\ 0000_b$	Write flash
$0001\ 0001_b$	Write EEPROM
$0000\ 1000_b$	Read signature bytes and calibration byte
$0000\ 0100_b$	Read fuse and lock bits
$0000\ 0010_b$	Read flash
$0000\ 0011_b$	Read EEPROM

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.

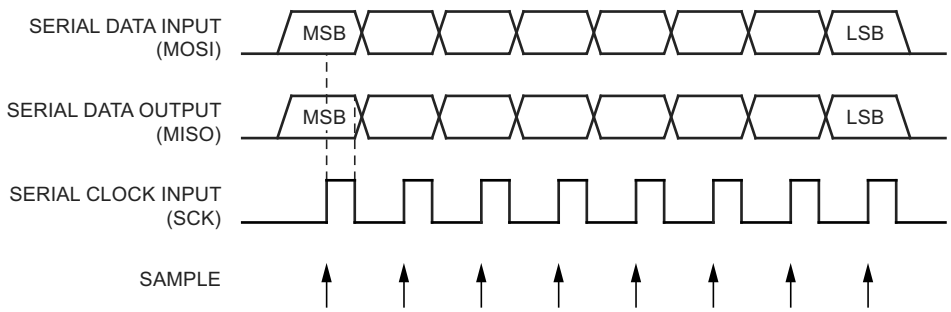
After data is loaded to the page buffer, program the EEPROM page, see Figure 21-8.

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

24.2 Idle Supply Current

Figure 24-5. Idle Supply Current versus Frequency (≥ 1MHz)

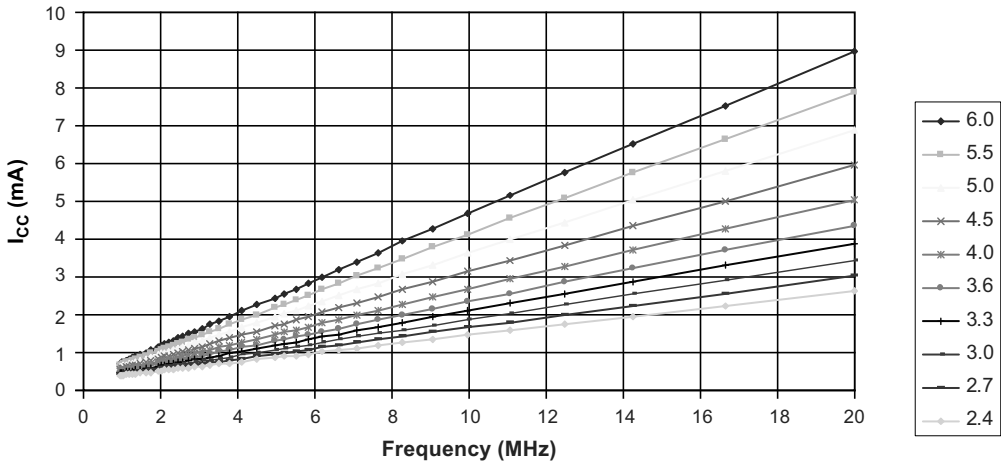


Figure 24-6. Idle Supply Current versus Vcc (Internal RC Oscillator, 8MHz)

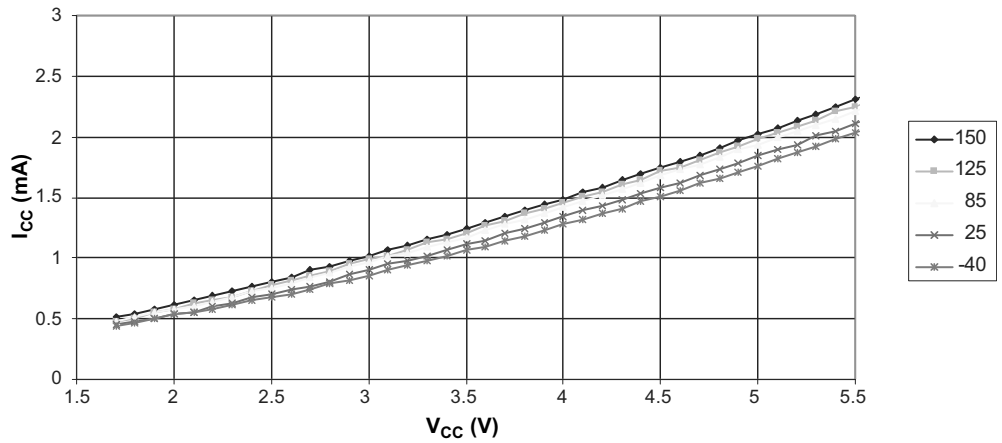
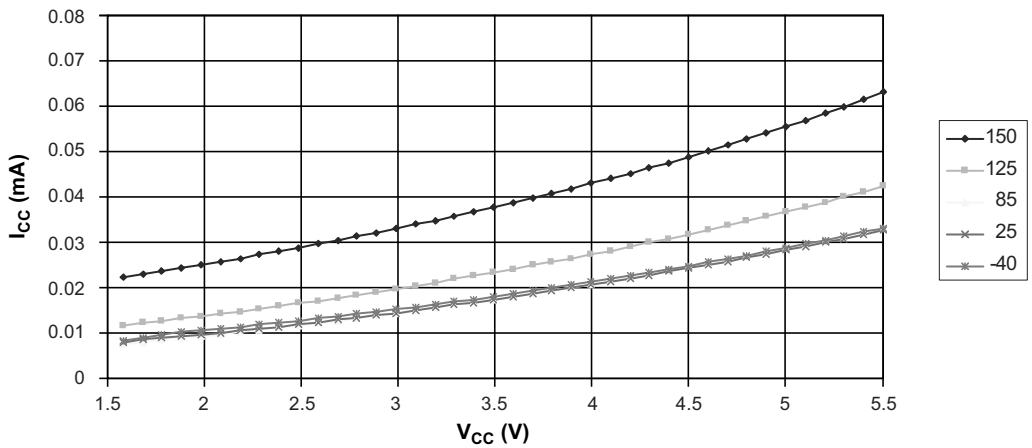


Figure 24-7. Idle Supply Current versus Vcc (Internal RC Oscillator, 128kHz)



25. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xB7)	Reserved									
(0xB6)	ASSR	–	EXCLK	AS0	TCN0UB	OCR0AUB	–	TCR0AUB	TCR0BUB	98
(0xB5)	Reserved									
(0xB4)	Reserved									
(0xB3)	Reserved									
(0xB2)	Reserved									
(0xB1)	Reserved									
(0xB0)	Reserved									
(0xAF)	Reserved									
(0xAE)	Reserved									
(0xAD)	Reserved									
(0xAC)	Reserved									
(0xAB)	Reserved									
(0xAA)	Reserved									
(0xA9)	Reserved									
(0xA8)	Reserved									
(0xA7)	Reserved									
(0xA6)	Reserved									
(0xA5)	Reserved									
(0xA4)	Reserved									
(0xA3)	Reserved									
(0xA2)	Reserved									
(0xA1)	Reserved									
(0xA0)	Reserved									
(0x9F)	Reserved									
(0x9E)	Reserved									
(0x9D)	Reserved									
(0x9C)	Reserved									
(0x9B)	Reserved									
(0x9A)	Reserved									
(0x99)	Reserved									
(0x98)	Reserved									
(0x97)	Reserved									
(0x96)	Reserved									
(0x95)	Reserved									
(0x94)	Reserved									

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.

30. Revision History

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

Revision No.	History
7728H-AVR-03/14	• Took datasheet into the latest template
7728G-AVR-06/10	• Power on reset values updated
7728F-AVR-05/10	• Clock characteristics updated
7728E-AVR-04/10	• Ordering information with new part numbers for silicon revision D updated • Errata updated • Revision history updated
7728D-AVR-07/09	• ISRC updated • Brown-out updated • Analog comparator updated • Temperature sensor updated
7728C-AVR-05/09	• Atmel® ATtiny87 devices added. • Updated Section 22.8 “ADC Characteristics” on page 227. • Updated ADC parameter.
7728B-AVR-04/09	• Added Atmel ATtiny87 specification (Table 1-1 on page 3, Table 3-1 on page 16, Table 7-1 on page 57, Table 21-6 on page 209, Table 21-7 on page 210, Table 21-8 on page 210 and Section 27. “Ordering Information” on page 253). • Updated Figure 18-1 on page 194 and Table 18-3 on page 197 in analog comparator chapter. • Updated DIDR1 register on page 192 and in register summary paragraph. • Updated Section 29. “Errata” on page 257.
7728A-AVR-07/08	• Document Creation