E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	20-WFQFN Exposed Pad
Supplier Device Package	20-WQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny45-15mt2

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

• Bit 7 – I: Global Interrupt Enable

The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

• Bit 6 – T: Bit Copy Storage

The bit copy instructions BLD (bit LoaD) and BST (bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

• Bit 5 – H: Half Carry Flag

The half carry flag H indicates a half carry in some arithmetic operations. Half carry is useful in BCD arithmetic. See the "instruction set description" for detailed information.

Bit 4 – S: Sign Bit, S = N ⊕ V

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the "instruction set description" for detailed information.

• Bit 3 – V: Two's Complement Overflow Flag

The two's complement overflow flag V supports two's complement arithmetic. See the "instruction set description" for detailed information.

• Bit 2 – N: Negative Flag

The negative flag N indicates a negative result in an arithmetic or logic operation. See the "instruction set description" for detailed information.

• Bit 1 – Z: Zero Flag

The zero flag Z indicates a zero result in an arithmetic or logic operation. See the "instruction set description" for detailed information.

• Bit 0 – C: Carry Flag

Atmel

The carry flag C indicates a carry in an arithmetic or logic operation. See the "instruction set description" for detailed information.

3.5 General Purpose Register File

The register file is optimized for the AVR enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the register file:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

6. Power Management and Sleep Modes

The high performance and industry leading code efficiency makes the AVR[®] microcontrollers an ideal choice for low power applications.

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

To enter any of the three sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM1..0 bits in the MCUCR register select which sleep mode (idle, ADC noise reduction, or power-down) will be activated by the SLEEP instruction. See Table 6-1 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

Figure 5-1 on page 19 presents the different clock systems in the Atmel ATtiny25/45/85, and their distribution. The figure is helpful in selecting an appropriate sleep mode.

6.1 MCU Control Register – MCUCR

The MCU control register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	_
	BODS	PUD	SE	SM1	SM0	BODSE	ISC01	ISC00	MCUCR
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 – BODS: BOD Sleep

BOD disable functionality is available in some devices, only. See Section 6.5 "Limitations" on page 29.

In order to disable BOD during sleep (see Table 6-2 on page 29) the BODS bit must be written to logic one. This is controlled by a timed sequence and the enable bit, BODSE in MCUCR. First both BODS and BODSE must be set to one. Second, within four clock cycles, BODS must be set to one and BODSE must be set to zero. The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

In devices where sleeping BOD has not been implemented this bit is unused and will always read zero.

• Bit 5 – SE: Sleep Enable

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the sleep enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

• Bits 4, 3 – SM1..0: Sleep Mode Select Bits 2..0

These bits select between the three available sleep modes as shown in Table 6-1.

SM1	SM0	Sleep Mode
0	0	Idle
0	1	ADC noise reduction
1	0	Power-down
1	1	Stand-by mode

Table 6-1. Sleep Mode Select

• Bit 2 – BODSE: BOD Sleep Enable

BOD disable functionality is available in some devices, only. See Section 6.5 "Limitations" on page 29.

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

This bit is unused in devices where software BOD disable has not been implemented and will read as zero in those devices.



Figure 7-1. Reset Logic



• Bit 6 – WDIE: Watchdog Timeout Interrupt Enable

When this bit is written to one, WDE is cleared, and the I-bit in the status register is set, the watchdog time-out interrupt is enabled. In this mode the corresponding interrupt is executed instead of a reset if a time-out in the watchdog timer occurs.

If WDE is set, WDIE is automatically cleared by hardware when a time-out occurs. This is useful for keeping the watchdog reset security while using the interrupt. After the WDIE bit is cleared, the next time-out will generate a reset. To avoid the watchdog reset, WDIE must be set after each interrupt.

WDE	WDIE	Watchdog Timer State	Action on Time-out
0	0	Stopped	None
0	1	Running	Interrupt
1	0	Running	Reset
1	1	Running	Interrupt

Table 7-6. Watchdog Timer Configuration

• Bit 4 – WDCE: Watchdog Change Enable

This bit must be set when the WDE bit is written to logic zero. Otherwise, the watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure. This bit must also be set when changing the prescaler bits. See Section 7.10 "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 41

• Bit 3 – WDE: Watchdog Enable

Atmel

When the WDE is written to logic one, the watchdog timer is enabled, and if the WDE is written to logic zero, the watchdog timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one. To disable an enabled watchdog timer, the following procedure must be followed:

- 1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
- 2. Within the next four clock cycles, write a logic 0 to WDE. This disables the watchdog.

In safety level 2, it is not possible to disable the watchdog timer, even with the algorithm described above. See Section 7.10 "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 41

In safety level 1, WDE is overridden by WDRF in MCUSR. See Section 7.7 "MCU Status Register – MCUSR" on page 37 for description of WDRF. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared before disabling the Watchdog with the procedure described above. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

Note: If the watchdog timer is not going to be used in the application, it is important to go through a watchdog disable procedure in the initialization of the device. If the watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset, which in turn will lead to a new watchdog reset. To avoid this situation, the application software should always clear the WDRF flag and the WDE control bit in the initialization routine.

• Bits 5, 2..0 – WDP3..0: Watchdog Timer Prescaler 3, 2, 1, and 0

The WDP3..0 bits determine the watchdog timer prescaling when the watchdog timer is enabled. The different prescaling values and their corresponding time-out periods are shown in Table 7-7.

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V _{cc} = 5.0V						
0	0	0	0	2Kcycles	16ms						
0	0	0	1	4Kcycles	32ms						
0	0	1	0	8Kcycles	64ms						
0	0	1	1	16Kcycles	0.125s						
0	1	0	0	32Kcycles	0.25s						
0	1	0	1	64Kcycles	0.5s						
0	1	1	0	128Kcycles	1.0s						
0	1	1	1	256Kcycles	2.0s						
1	0	0	0	512Kcycles	4.0s						
1	0	0	1	1024Kcycles	8.0s						
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1	Reserve	Reserved						
1	1	1	0								
1	1	1	1								

Table 7-7. Watchdog Timer Prescale Select

Note: 1. If selected, one of the valid settings below 0b1010 will be used.



9.2.4 Reading the Pin Value

Independent of the setting of data direction bit DDxn, the port pin can be read through the PINxn register bit. As shown in Figure 9-2 on page 44, the PINxn register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 9-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.





Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn register at the succeeding positive clock edge. As indicated by the two arrows tpd,max and tpd,min, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 9-4. The out instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is one system clock period.

Figure 9-4. Synchronization when Reading a Software Assigned Pin Value





The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 5 as input with a pull-up assigned to port pin 4. The resulting pin values are read back again, but as previously discussed, a *nop* instruction is included to be able to read back the value recently assigned to some of the pins.

```
Assembly Code Example<sup>(1)</sup>
```



```
unsigned char i;
```

```
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB4) | (1<<PB1) | (1<<PB0);
DDRB = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
/* Insert nop for synchronization*/
_NOP();
/* Read port pins */
i = PINB;
....</pre>
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1 and 4, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

9.2.5 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 mode, power-save mode, and standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

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 48.

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.

9.2.6 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 pulldown. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.



11.1.1 Registers

The Timer/Counter (TCNT0) and output compare registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the timer interrupt flag register (TIFR). All interrupts are individually masked with the timer interrupt mask register (TIMSK). TIFR and TIMSK are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The clock select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk_{T0}).

The double buffered output compare registers (OCR0A and OCR0B) is compared with the Timer/Counter value at all times. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the output compare pins (OC0A and OC0B). See Section 11.4 "Output Compare Unit" on page 58 for details. The compare match event will also set the compare flag (OCF0A or OCF0B) which can be used to generate an output compare interrupt request.

11.1.2 Definitions

Many register and bit references in this section are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 0. A lower case "x" replaces the output compare unit, in this case compare unit A or compare unit B. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions below are also used extensively throughout the document.

Parameter	Definition
BOTTOM	The counter reaches the BOTTOM when it becomes 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.

Table 11-1. Definitions

11.2 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external 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 (TCCR0B). For details on clock sources and prescaler, see Section 12. "Timer/Counter Prescaler" on page 72.

11.3 Counter Unit

Atmel

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 11-2 shows a block diagram of the counter and its surroundings.

Figure 11-2. Counter Unit Block Diagram



Table 11-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

COM01	COM00	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A disconnected. WGM02 = 1: Toggle OC0A on compare match.
1	0	Clear OC0A on compare match, set OC0A at TOP
1	1	Set OC0A on compare match, clear OC0A at TOP

Table 11-3. Compare Output Mode, Fast PWM Mode⁽¹⁾

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 11.6.3 "Fast PWM Mode" on page 62 for more details.

Table 11-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 11-4.	Compare Out	out Mode, Phase	e Correct PWM	Mode ⁽¹⁾
-------------	-------------	-----------------	---------------	---------------------

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal port operation, OC0A disconnected. WGM02 = 1: Toggle OC0A on compare match.
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 11.6.4 "Phase Correct PWM Mode" on page 63 for more details.

• Bits 5:4 – COM0B1:0: Compare Match Output B Mode

These bits control the output compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 11-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 11-5. Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on compare match
1	0	Clear OC0B on compare match
1	1	Set OC0B on compare match

11.8.3 Timer/Counter Register – TCNT0



The Timer/Counter register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. writing to the TCNT0 register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a compare match between TCNT0 and the OCR0x registers.

11.8.4 Output Compare Register A – OCR0A



The output compare register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0A pin.

11.8.5 Output Compare Register B – OCR0B



The output compare register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0B pin.

11.8.6 Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	-	OCIE1A	OCIE1B	OCIE0A	OCIE0B	TOIE1	TOIE0	-	TIMSK
Read/Write	R	R	R	R	R/W	R/W	R/W	R	-
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7..4, 0 - Res: Reserved Bits

These bits are reserved bits in the Atmel® ATtiny25/45/85 and will always read as zero.

• Bit 3 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable

When the OCIE0B bit is written to one, and the I-bit in the status register is set, the Timer/Counter compare match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter interrupt flag register – TIFR0.

• Bit 2 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable

When the OCIE0A bit is written to one, and the I-bit in the status register is set, the Timer/Counter0 compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 interrupt flag register – TIFR0.

• Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable

When the TOIE0 bit is written to one, and the I-bit in the status register is set, the Timer/Counter0 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 interrupt flag register – TIFR0.



13.1.6 Timer/Counter1 Output Compare RegisterC - OCR1C



The output compare register C is an 8-bit read/write register.

The Timer/Counter output compare register C contains data to be continuously compared with Timer/Counter1. A compare match does only occur if Timer/Counter1 counts to the OCR1C value. A software write that sets TCNT1 and OCR1C to the same value does not generate a compare match. If the CTC1 bit in TCCR1 is set, a compare match will clear TCNT1.

This register has the same function in normal mode and PWM mode.

13.1.7 Timer/Counter Interrupt Mask Register - TIMSK

Bit	7	6	5	4	3	2	1	0	
\$39 (\$59)	-	OCIE1A	OCIE1B	OCIE0A	OCIE0B	TOIE1	TOIE0	-	TIMSK
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R	-
Initial value	0	0	0	0	0	0	0	0	

• Bit 7 - Res: Reserved Bit

This bit is a reserved bit in the Atmel® ATtiny25/45/85 and always reads as zero.

• Bit 6 - OCIE1A: Timer/Counter1 Output Compare Interrupt Enable

When the OCIE1A bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 compare match A, interrupt is enabled. The corresponding interrupt at vector \$003 is executed if a compare match A occurs. The compare flag in Timer/Counter1 is set (one) in the Timer/Counter interrupt flag register.

• Bit 5 - OCIE1B: Timer/Counter1 Output Compare Interrupt Enable

When the OCIE1B bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 compare match B, interrupt is enabled. The corresponding interrupt at vector \$009 is executed if a compare match B occurs. The compare flag in Timer/Counter1 is set (one) in the Timer/Counter interrupt flag register.

• Bit 4– OCIE0A: Timer/Counter Output Compare Match A Interrupt Enable

When the OCIE0A bit is written to one, and the I-bit in the status register is set, the Timer/Counter compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter occurs, i.e., when the OCF0A bit is set in the Timer/Counter interrupt flag register – TIFR0.

• Bit 3 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable

When the OCIE0B bit is written to one, and the I-bit in the status register is set, the Timer/Counter compare match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter interrupt flag register – TIFR0.

• Bit 2 - TOIE1: Timer/Counter1 Overflow Interrupt Enable

When the TOIE1 bit is set (one) and the I-bit in the status register is set (one), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt (at vector \$004) is executed if an overflow in Timer/Counter1 occurs. The overflow flag (timer1) is set (one) in the Timer/Counter interrupt flag register - TIFR.

• Bit 0 - Res: Reserved Bit

This bit is a reserved bit in the Atmel ATtiny25/45/85 and always reads as zero.



Bit 6 – ACBG: Analog Comparator Bandgap Select

When this bit is set an internal 1.1V/2.56V reference voltage replaces the positive input to the analog comparator. The selection of the internal voltage reference is done by writing the REFS2..0 bits in ADMUX register. When this bit is cleared, AIN0 is applied to the positive input of 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 - Res: Reserved Bit

This bit is a reserved bit in the Atmel® ATtiny25/45/85 and will always read as zero.

• 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 16-1.

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.

Table 16-1. ACIS1/ACIS0 Settings

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.

16.3 Analog Comparator Multiplexed Input

It is possible to select any of the ADC3..0 pins to replace the negative input to the analog comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the analog comparator multiplexer enable bit (ACME in ADCSRB) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX1..0 in ADMUX select the input pin to replace the negative input to the analog comparator, as shown in Table 16-2. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the analog comparator.

ACME	ADEN	MUX10	Analog Comparator Negative Input
0	x	ХХ	AIN1
1	1	XX	AIN1
1	0	00	ADC0
1	0	01	ADC1
1	0	10	ADC2
1	0	11	ADC3

Table 16-2. Analog Comparator Multiplexed Input

16.3.1 Digital Input Disable Register 0 – DIDR0



• Bits 1, 0 - AIN1D, AIN0D: AIN1, AIN0 Digital Input Disable

When this bit is written logic one, the digital input buffer on the AIN1/0 pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the AIN1/0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.



18.4 Software Break Points

debugWIRE supports program memory break points by the AVR[®] break instruction. Setting a break point in 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.

18.5 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.

18.6 debugWIRE Related Register in I/O Memory

The following section describes the registers used with the debugWire.

18.6.1 debugWire Data Register – DWDR



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.



19.4.1 Store Program Memory Control and Status Register – SPMCSR

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



• Bits 7..5 - Res: Reserved Bits

These bits are reserved bits in the Atmel[®] ATtiny25/45/85 and always read as zero.

• 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 19.4.2 "EEPROM Write Prevents Writing to SPMCSR" on page 121 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 Z-pointer. 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: Store Program Memory Enable

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either 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 "10001", "01001", "00101", "00011" or "00001" in the lower five bits will have no effect.

19.4.2 EEPROM Write Prevents Writing to SPMCSR

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

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the chip erase instruction. The chip erase operation turns the content of every memory location in both the program and EEPROM arrays into 0xFF.

Depending on CKSEL fuses, a valid clock must be present. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2CPU clock cycles for $f_{ck} < 12MHz$, 3CPU clock cycles for $f_{ck} \ge 12MHz$ High: > 2CPU clock cycles for $f_{ck} < 12MHz$, 3CPU clock cycles for $f_{ck} \ge 12MHz$

20.6.1 Serial Programming Algorithm

When writing serial data to the Atmel® ATtiny25/45/85, data is clocked on the rising edge of SCK.

When reading data from the Atmel ATtiny25/45/85, data is clocked on the falling edge of SCK. See Figure 20-2 on page 128 and Figure 20-3 on page 129 for timing details.

To program and verify the ATtiny25/45/85 in the serial programming mode, the following sequence is recommended (see four byte instruction formats in Table 20-10 on page 128):

1. Power-up sequence:

Apply power between V_{CC} and GND while RESET and SCK are set to "0". In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case, RESET must be given a positive pulse of at least two CPU clock cycles duration after SCK has been set to "0".

- 2. Wait for at least 20ms and enable serial programming by sending the programming enable serial instruction to pin MOSI.
- 3. The serial programming instructions will not work if the communication is out of synchronization. When in sync. the second byte (0x53), will echo back when issuing the third byte of the programming enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the 0x53 did not echo back, give RESET a positive pulse and issue a new programming enable command.
- 4. The flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 5 LSB of the address and data together with the load program memory page instruction. To ensure correct loading of the page, the data low byte must be loaded before data high byte is applied for a given address. The program memory page is stored by loading the write program memory page instruction with the 6 MSB of the address. If polling (RDY/BSY) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page. (See Table 20-9 on page 128.) Accessing the serial programming interface before the flash write operation completes can result in incorrect programming.
- 5. A: The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling (RDY/BSY) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte. (See Table 20-9 on page 128.) In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
 B: The EEPROM array is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 2 LSB of the address and data together with the Load EEPROM memory page instruction. The EEPROM memory page is stored by loading the write EEPROM memory page Instruction with the 6 MSB of the address. When using EEPROM page access only byte locations loaded with the Load EEPROM memory page instruction is altered. The remaining locations remain unchanged. If polling (RDY/BSY) is not used, the used must wait at least t_{WD_EEPROM} before issuing the next page (See Table 20-7 on page 126). In a chip erased device, no 0xFF in the data file(s) need to be programmed.
- 6. Any memory location can be verified by using the read instruction which returns the content at the selected address at serial output MISO.
- 7. At the end of the programming session, RESET can be set high to commence normal operation.
- Power-off sequence (if needed): Set RESET to "1". Turn V_{CC} power off.

Atmel

22.2.1.1Example 1

Calculate the expected current consumption in idle mode with USI, TIMER0, and ADC enabled at V_{CC} = 2.0V and F = 1MHz. From Table 22-2 on page 145, third column, we see that we need to add 6.4% for the USI, 7.3% for the TIMER0 module, and 21.4% for the ADC module. Reading from Figure 22-9, we find that the idle current consumption is ~0.25mA at V_{CC} = 3.0V and F = 1MHz. The total current consumption in idle mode with USI, TIMER0, and ADC enabled, gives:

 $I_{CCtotal} \approx (0.25) \text{ mA} \times (1 + 0.064 + 0.073 + 0.214) \approx 0.337 \text{ mA}$

22.3 Power-Down Supply Current



Figure 22-11. Power-Down Supply Current versus V_{cc} (Watchdog Timer Disabled)

Figure 22-12. Power-Down Supply Current versus V_{CC} (Watchdog Timer Enabled)





24. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
Arithmetic an	d Logic Instr	ructions			
ADD	Rd, Rr	Add two registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with carry two registers	$Rd \gets Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl, K	Add immediate to word	$Rdh:RdI \leftarrow Rdh:RdI + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract constant from register	$Rd \gets Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with carry two registers	$Rd \gets Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with carry constant from Reg.	$Rd \gets Rd - K - C$	Z,C,N,V,H	1
SBIW	Rdl, K	Subtract immediate from word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND registers	$Rd \leftarrow Rd imes Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND register and constant	$Rd \gets Rd \times K$	Z,N,V	1
OR	Rd, Rr	Logical OR registers	$Rd \leftarrow Rd \lor Rr$	Z,N,V	1
ORI	Rd, K	Logical OR register and constant	$Rd \gets Rd \lor K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR registers	$Rd \gets Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's complement	$Rd \gets 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd, K	Set Bit(s) in register	$Rd \gets Rd \lor K$	Z,N,V	1
CBR	Rd, K	Clear Bit(s) in register	$Rd \gets Rd \times (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for zero or minus	$Rd \gets Rd \times Rd$	Z,N,V	1
CLR	Rd	Clear register	$Rd \gets Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set register	$Rd \leftarrow 0xFF$	None	1
Branch Instru	ictions				
RJMP	k	Relative jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect jump to (Z)	$PC \leftarrow Z$	None	2
RCALL	k	Relative subroutine call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect call to (Z)	$PC \leftarrow Z$	None	3
RET		Subroutine return	$PC \gets STACK$	None	4
RETI		Interrupt return	$PC \leftarrow STACK$	I	4
CPSE	Rd, Rr	Compare, skip if equal	if (Rd = Rr) PC \leftarrow PC + 2 or 3	None	1/2/3
CP	Rd, Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd, Rr	Compare with carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd, K	Compare register with immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if bit in register cleared	if (Rr(b)=0) PC \leftarrow PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if bit in register is set	if (Rr(b)=1) PC \leftarrow PC + 2 or 3	None	1/2/3
SBIC	P, b	Skip if bit in I/O register cleared	if (P(b)=0) PC \leftarrow PC + 2 or 3	None	1/2/3
SBIS	P, b	Skip if bit in I/O register is set	if (P(b)=1) PC \leftarrow PC + 2 or 3	None	1/2/3
BRBS	s, k	Branch if status flag set	if (SREG (s) = 1) then PC \leftarrow PC + k + 1	None	1/2
BRBC	s, k	Branch if status flag cleared	if (SREG (s) = 0) then PC \leftarrow PC + k + 1	None	1/2
BREQ	k	Branch if equal	if (Z = 1) then PC \leftarrow PC + k + 1	None	1/2
BRNE	k	Branch if not equal	if (Z = 0) then PC \leftarrow PC + k + 1	None	1/2
BRCS	k	Branch if carry set	if (C = 1) then PC \leftarrow PC + k + 1	None	1/2



26.2 PC



Atmel

	7.8 7.9 7.10	Internal Voltage Reference Watchdog Timer Timed Sequences for Changing the Configuration of the Watchdog Timer.	37 38 41
8.	Inte 8.1	errupts	42 42
9.	I/O 9.1 9.2 9.3 9.4	Ports Introduction Ports as General Digital I/O Alternate Port Functions Register Description for I/O-Ports	43 43 44 48 53
10.	Ext 10.1 10.2 10.3 10.4	ernal Interrupts . MCU Control Register – MCUCR General Interrupt Mask Register – GIMSK General Interrupt Flag Register – GIFR. Pin Change Mask Register – PCMSK	54 54 54 55 55
11.	8-b 11.1 11.2 11.3 11.4 11.5 11.6 11.7 11.8	it Timer/Counter0 with PWM Overview	56 57 57 58 60 61 65 66
12.	Tin 12.1 12.2	ner/Counter Prescaler	72 72 72
13.	Co 13.1	unter and Compare Units	74 74
14.	De 14.1 14.2 14.3	ad Time Generator Timer/Counter1 Dead Time Prescaler register 1 - DTPS1 Timer/Counter1 Dead Time A - DT1A Timer/Counter1 Dead Time B - DT1B	85 86 87 87
15.	Un 15.1 15.2 15.3 15.4	iversal Serial Interface – USI Overview Functional Descriptions Alternative USI Usage USI Register Descriptions	88 88 89 93 94
16.	Ana 16.1 16.2 16.3	alog Comparator ADC Control and Status Register B – ADCSRB Analog Comparator Control and Status Register – ACSR Analog Comparator Multiplexed Input	98 98 98 100
17.	Ana 17.1 17.2 17.3 17.4 17.5	alog to Digital Converter 1 Features 2 Operation 2 Starting a Conversion 2 Prescaling and Conversion Timing 2 Changing Channel or Reference Selection 2	101 102 103 104 107

