Microchip Technology - ATTINY24A-SSU Datasheet

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	20MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	12
Program Memory Size	2KB (1K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny24a-ssu

Legend / Abbreviations for *Table 2*:

Type: I = input, O = output, S = supply

In/Output level: C_T = CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger

Output level: HS = 20 mA high sink (on N-buffer only)

Port and control configuration:

- Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog
- Output: OD = open drain, PP = push-pull

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

Table 2. Device pin description (32-pin packages)

Pin number		Pin name	Type	Level		Port/control						Main function (after reset)	Alternate function
						Input				Output			
LQFP32	SDIP32			Input	Output	float	wpu	int	ana	OD[1]	PP		
1	5	PA3(HS)/ATPWM1	I/O	C_T	HS	**x**	ei0			x	x	**Port A3 (HS)**	ATPWM1
2	6	PA4(HS)/ ATPWM2/MCO	I/O	C_T	HS	**x**				x	x	**Port A4 (HS)**	ATPWM2/ MCO
3	7	PA5 (HS)ATPWM3	I/O	C_T	HS	**x**				x	x	**Port A5 (HS)**	ATPWM3
4	8	PA6(HS)/ I2CDATA/SCK[2]	I/O	C_T	HS	**x**		ei0		T		**Port A6 (HS)**	I2CDATA/SPI serial clock
5	9	PA7(HS)/I2CCLK/$\overline{SS}$ [2]	I/O	C_T	HS	**x**				T		**Port A7 (HS)**	I2CCLK/SPI slave select (active low)
6	10	RESET					**x**			x		Reset	
8	12	V_{DD}[3]	S									Digital Supply Voltage	
9	13	V_{SS}[3]	S									Digital Ground Voltage	
10	14	OSC1/CLKIN	I									Resonator oscillator inverter input or External clock input	
11	15	OSC2	O									Resonator oscillator output	
12	16	V_{SSA}[3]	S									Analog Ground Voltage	
13	17	V_{DDA}[3]	S									Analog Supply Voltage	

5 Central processing unit

5.1 Introduction

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

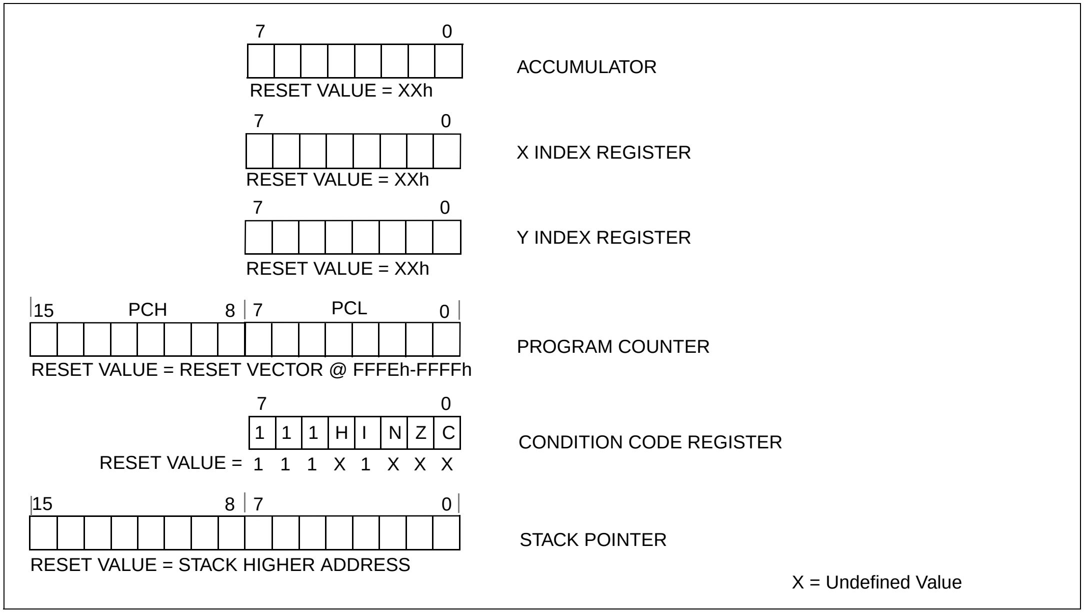
5.2 Main features

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

5.3 CPU registers

The six CPU registers shown in *Figure 7*. They are not present in the memory mapping and are accessed by specific instructions.

Figure 7. CPU registers



5.3.1 Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

5.3.2 Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

5.3.3 Program Counter (PC)

The Program Counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter low which is the LSB) and PCH (Program Counter high which is the MSB).

5.3.4 Condition Code register (CC)

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

Reset value: 111x 1xxx

7							0
1	1	I1	H	I0	N	Z	C
Read/write							

These bits can be individually tested and/or controlled by specific instructions.

Arithmetic management bits

Bit 4 = **H** *Half carry bit*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

These bytes are systematically programmed by ST.

6.1.2 Customized RC calibration

If the application requires a higher frequency accuracy or if the voltage or temperature conditions change in the application, the frequency may need to be recalibrated. Two non-volatile bytes (RCCRH_USER and RCCRL_USER) are reserved for storing these new values. These two-byte area is Electrically Erasable Programmable Read Only Memory.

Note: Refer to application note AN1324 for information on how to calibrate the RC frequency using an external reference signal.

How to program RCCRH_USER and RCCRL_USER

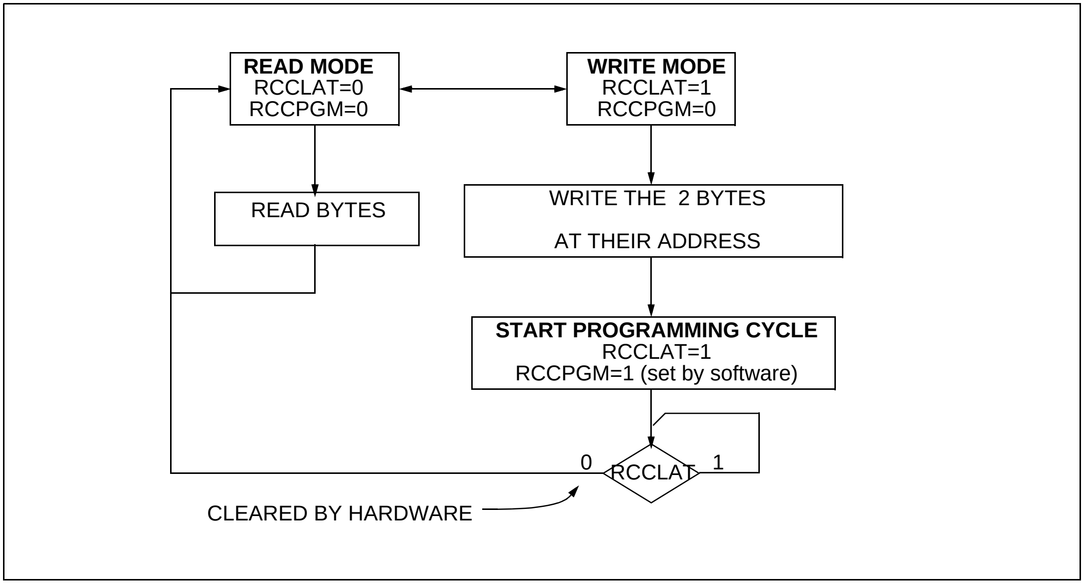
To access the write mode, the RCCLAT bit has to be set by software (the RCCPGM bit remains cleared). When a write access to this two-byte area occurs, the values are latched.

When RCCPGM bit is set by the software, the latched data are programmed in the EEPROM cells. To avoid wrong programming, the user must take care to only access these two-byte addresses.

At the end of the programming cycle, the RCCPGM and RCCLAT bits are cleared simultaneously.

Note: During the programming cycle, it is forbidden to access the latched data (see Figure 9).

Figure 9. RCCRH_USER and RCCRL_USER programming flowchart



Note: If a programming cycle is interrupted (by a reset action), the integrity of the data in memory is not guaranteed.

Access error handling

If a read access occurs while RCCLAT=1, then the data bus will not be driven.

If a write access occurs while RCCLAT=0, then the data on the bus will not be latched.

7.5 Description of interrupt registers

7.5.1 CPU CC register interrupt bits

Reset value: 111x 1010(xAh)

7							0
1	1	I1	H	I0	N	Z	C
Read/write							

Bits 5, 3 = **I1, I0** *Software Interrupt Priority bits*

These two bits indicate the current interrupt software priority (see *Table 14*).

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see *Table 16: Dedicated interrupt instruction set*).

TRAP and RESET events can interrupt a level 3 program.

Table 14. Setting the interrupt software priority

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low ↓ High	1	0
Level 1		0	1
Level 2			0
Level 3 (= interrupt disable*)		1	1

7.5.2 Interrupt software priority registers (ISPRx)

All ISPRx register bits are read/write except bit 7:4 of **ISPR3** which are read only.

Reset value: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	1	1	I1_12	I0_12

ISPRx registers contain the interrupt software priority of each interrupt vector. Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers to define its software priority. This correspondence is shown in *Table 15*.

Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

9.6 Interrupts

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and if the I bit in the CC register is cleared (RIM instruction).

Table 27. Description of interrupt events

Interrupt Event	Event flag	Enable Control bit	Exit from Wait	Exit from Halt
External interrupt on selected external event	-	DDRx ORx	Yes	Yes

See application notes AN1045 software implementation of I^2C bus master, and AN1048 - software LCD driver

9.7 Device-specific I/O port configuration

The I/O port register configurations are summarized in *Section 9.7.1: Standard ports* and *Section 9.7.2: Other ports*.

9.7.1 Standard ports

Table 28. PA5:0, PB7:0, PC7:4 and PC2:0 pins

Mode	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

9.7.2 Other ports

Table 29. PA7:6 pins

Mode	DDR	OR
floating input	0	0
interrupt input	0	1
open drain output	1	0
push-pull output	1	1

10.1.6 Register description

Control register (WDGCR)

Reset value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0
Read/Write							

Bit 7 = **WDGA** *Activation bit*

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** *7-bit timer (MSB to LSB)*

These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Table 34. Watchdog timer register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0033h	WDGCR Reset Value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

Figure 35. Single timer mode (ENCNTR2=0)

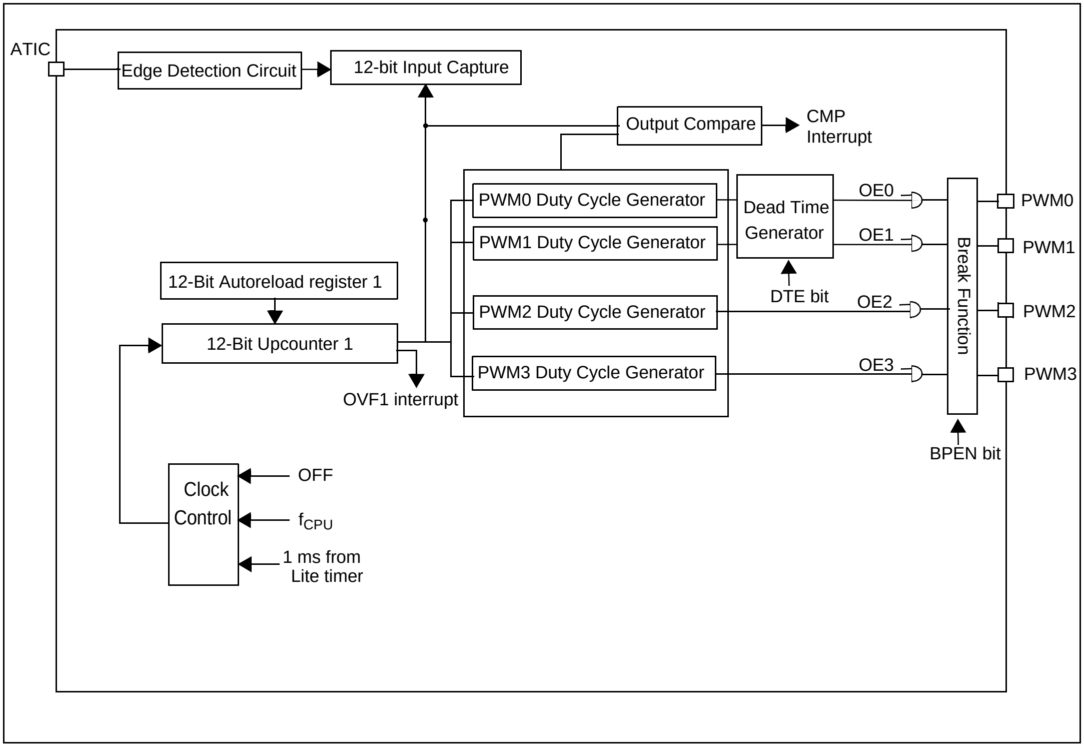
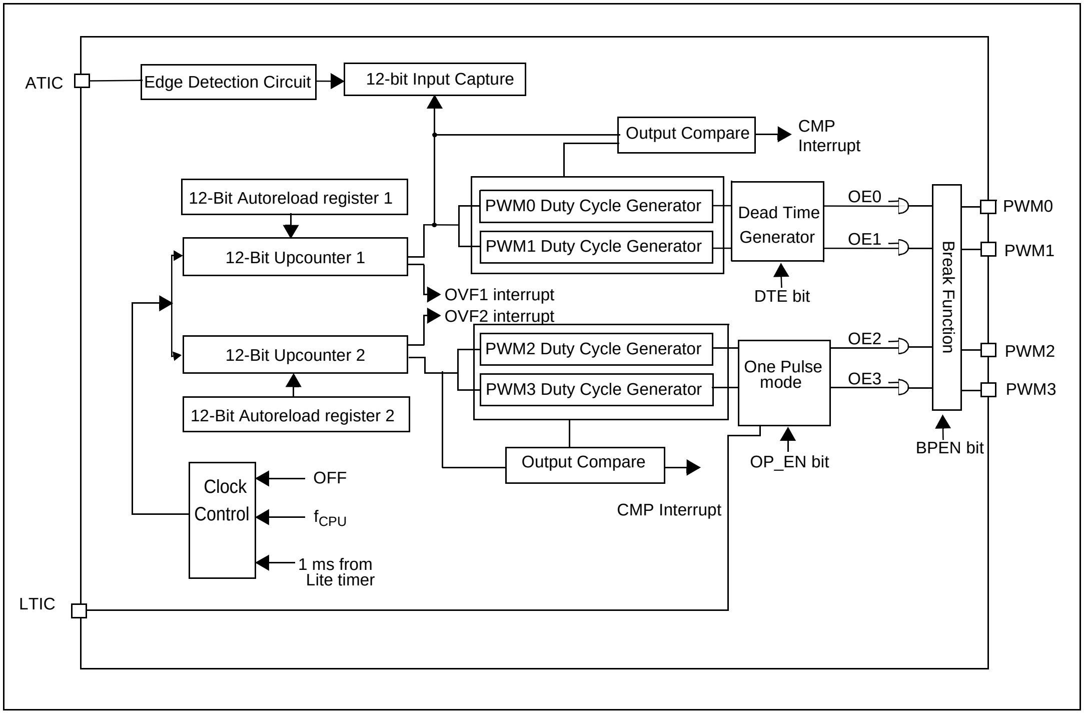


Figure 36. Dual timer mode (ENCNTR2=1)



10.2.3 Functional description

PWM mode

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins.

- **PWM frequency**

  The four PWM signals can have the same frequency (f_{PWM}) or can have two different frequencies. This is selected by the ENCNTR2 bit which enables Single Timer or Dual Timer mode (see *Figure 35* and *Figure 36*). The frequency is controlled by the counter period and the ATR register value. In Dual Timer mode, PWM2 and PWM3 can be generated with a different frequency controlled by CNTR2 and ATR2.

  $$f_{PWM} = f_{COUNTER} \diagup (4096 - ATR)$$

  Following the above formula, if $f_{COUNTER}$ equals 4 MHz, the maximum value of f_{PWM} is 2 MHz (ATR register value = 4094), and the minimum value is 1 kHz (ATR register value = 0).

  The maximum value of ATR is 4094 because it must be lower than the DC4R value which must be 4095 in this case.

- **Duty cycle**

  The duty cycle is selected by programming the DCRx registers. These are preload registers. The DCRx values are transferred in Active duty cycle registers after an overflow event if the corresponding transfer bit (TRANx bit) is set.

  The TRAN1 bit controls the PWMx outputs driven by counter 1 and the TRAN2 bit controls the PWMx outputs driven by counter 2.

  PWM generation and output compare are done by comparing these active DCRx values with the counter.

  The maximum available resolution for the PWMx duty cycle is:

  $$\text{Resolution} = 1 \diagup (4096 - ATR)$$

  where ATR is equal to 0. With this maximum resolution, 0% and 100% duty cycle can be obtained by changing the polarity.

  At reset, the counter starts counting from 0.

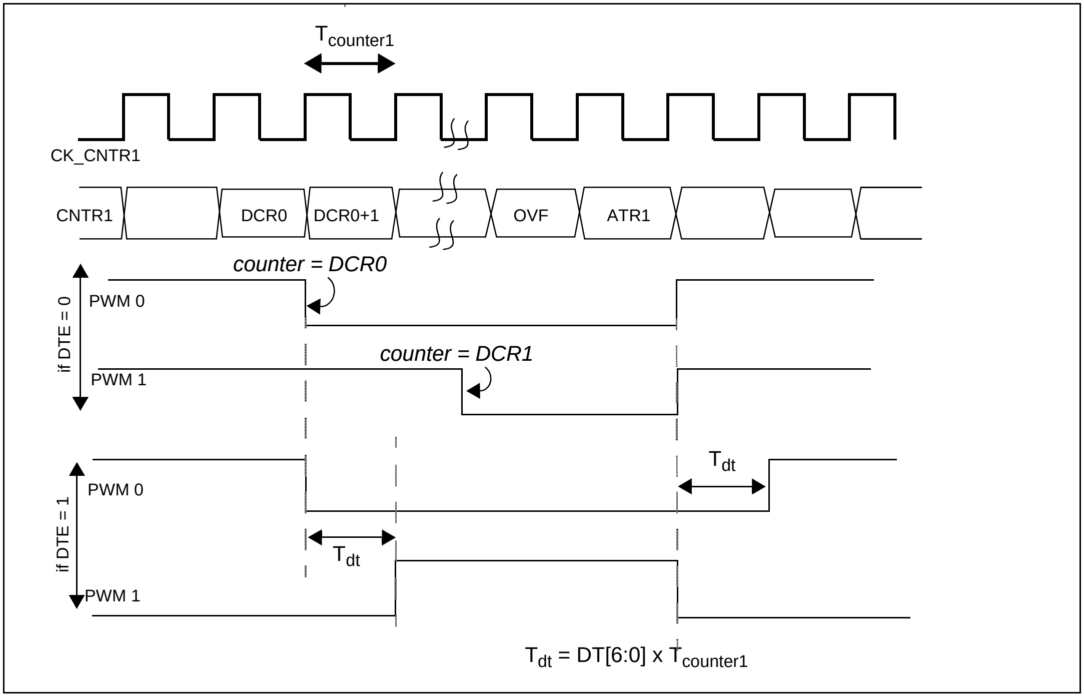
  When a upcounter overflow occurs (OVF event), the preloaded Duty cycle values are transferred to the active Duty Cycle registers and the PWMx signals are set to a high level. When the upcounter matches the active DCRx value the PWMx signals are set to a low level. To obtain a signal on a PWMx pin, the contents of the corresponding active DCRx register must be greater than the contents of the ATR register.

  The maximum value of ATR is 4094 because it must be lower than the DCR value which must be 4095 in this case.

- **Polarity inversion**

  The polarity bits can be used to invert any of the four output signals. The inversion is synchronized with the counter overflow if the corresponding transfer bit in the ATCSR2 register is set (reset value). See *Figure 37*.

Figure 40. Dead time generation



$T_{dt} = DT[6:0] \times T_{counter1}$

In the above example, when the DTE bit is set:

- PWM goes low at DCR0 match and goes high at ATR1+Tdt
- PWM1 goes high at DCR0+Tdt and goes low at ATR match.

With this programmable delay (Tdt), the PWM0 and PWM1 signals which are generated are not overlapped.

Break function

The break function can be used to perform an emergency shutdown of the application being driven by the PWM signals.

The break function is activated by the external BREAK pin. This can be selected by using the BRSEL bit in BREAKCR register. In order to use the break function it must be previously enabled by software setting the BPEN bit in the BREAKCR register.

The Break active level can be programmed by the BREDGE bit in the BREAKCR register. When an active level is detected on the BREAK pin, the BA bit is set and the break function is activated. In this case, the PWM signals are forced to BREAK value if respective OEx bit is set in PWMCR register.

Software can set the BA bit to activate the break function without using the BREAK pin. The BREN1 and BREN2 bits in the BREAKEN register are used to enable the break activation on the 2 counters respectively. In Dual Timer mode, the break for PWM2 and PWM3 is enabled by the BREN2 bit. In Single Timer mode, the BREN1 bit enables the break for all PWM channels.

How to enter one pulse mode

The steps required to enter One Pulse mode are the following:

1. Load ATR2H/ATR2L with required value.
2. Load DCR3H/DCR3L for PWM3. ATR2 value must be greater than DCR3.
3. Set OP3 in PWM3CSR if polarity change is required.
4. Select CNTR2 by setting ENCNTR2 bit in ATCSR2.
5. Set TRAN2 bit in ATCSR2 to enable transfer.
6. "Wait for Overflow" by checking the OVF2 flag in ATCSR2.
7. Select counter clock using CK<1:0> bits in ATCSR.
8. Set OP_EN bit in PWM3CSR to enable one-pulse mode.
9. Enable PWM3 by OE3 bit of PWMCR.

The "Wait for Overflow" in step 6 can be replaced by a forced update.

Follow the same procedure for PWM2 with the bits corresponding to PWM2.

Note: *When break is applied in one-pulse mode, CNTR2, DCR2/3 & ATR2 registers are reset. So, these registers have to be initialized again when break is removed.*

Figure 48. Block diagram of one pulse mode

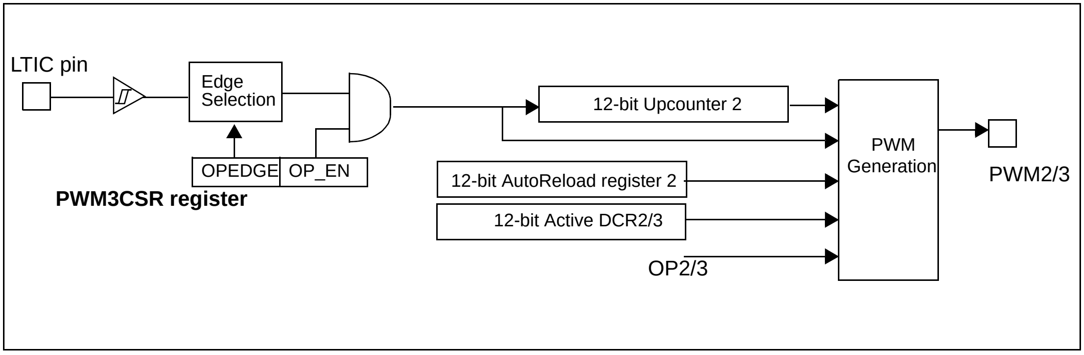
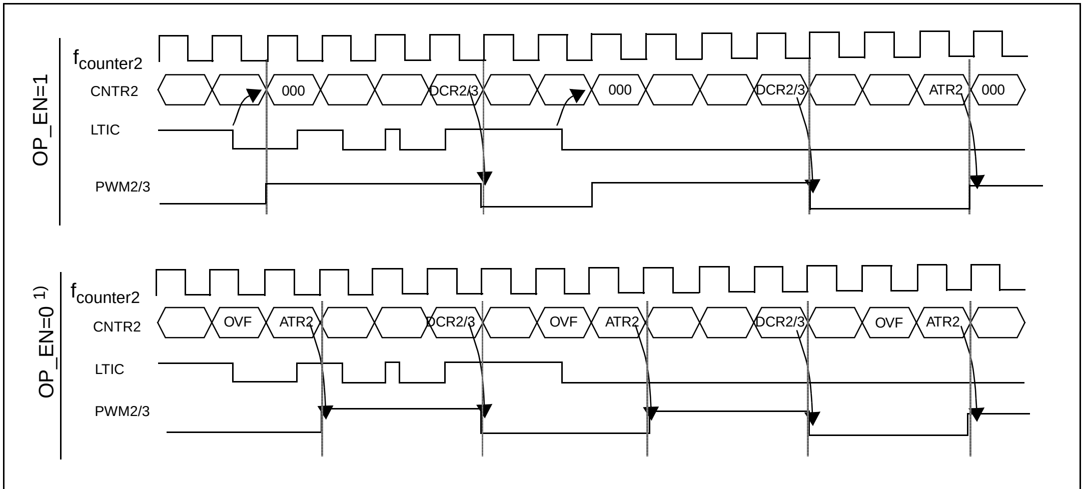


Figure 49. One pulse mode and PWM timing diagram



Note 1: When OP_EN=0, LTIC edges are not taken into account as the timer runs in PWM mode.

Bit 0 = **CMPIE** *Compare Interrupt Enable bit*

This bit is read/write by software and cleared by hardware after a reset. it can be used to mask the interrupt generated when any of the cmpfx bit is set.

0: Output Compare Interrupt Disabled.

1: Output Compare Interrupt Enabled.

Counter register 1 High (CNTR1H)

Reset value: 0000 0000 (00h)

15							8
0	0	0	0	CNTR1_ 11	CNTR1_ 10	CNTR1_9	CNTR1_8
Read only							

Counter register 1 Low (CNTR1L)

Reset value: 0000 0000 (00h)

7							0
CNTR1_7	CNTR1_6	CNTR1_5	CNTR1_4	CNTR1_3	CNTR1_2	CNTR1_1	CNTR1_0
Read only							

Bits 15:12 = Reserved

Bits 11:0 = **CNTR1[11:0]** *Counter value*

This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when f_{timer}=f_{CPU}, special care must be taken when CNTR1L values close to FFh are read.

When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

Figure 57. Counter timing diagram, internal clock divided by 2

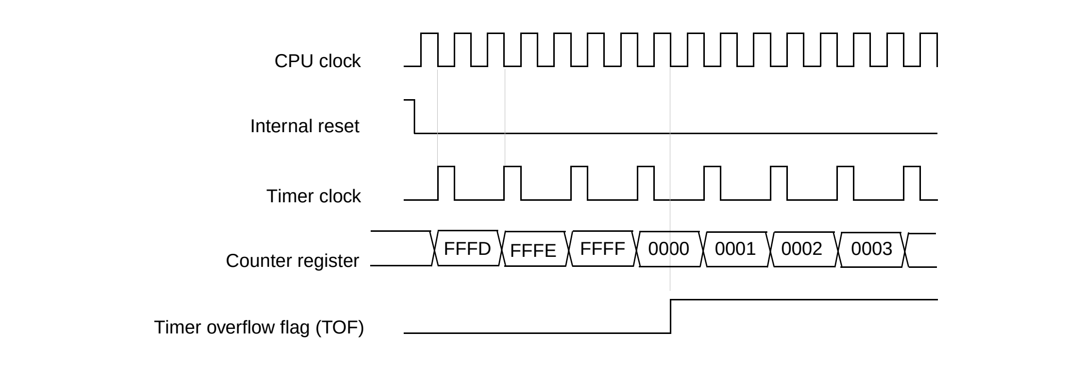


Figure 58. Counter timing diagram, internal clock divided by 4

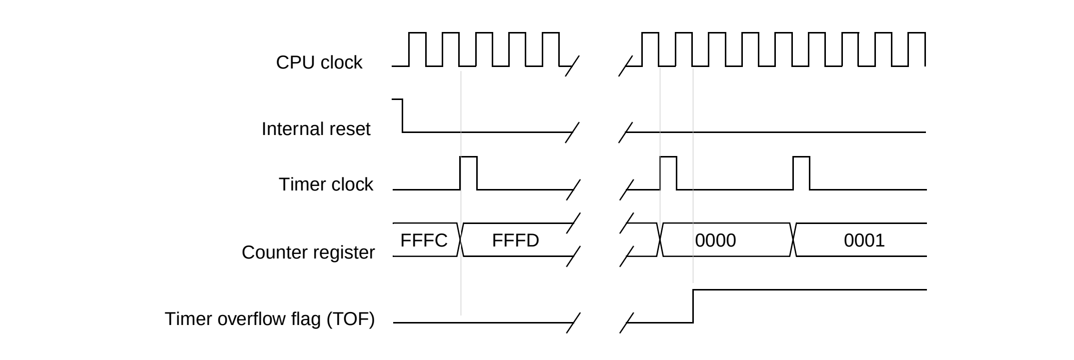
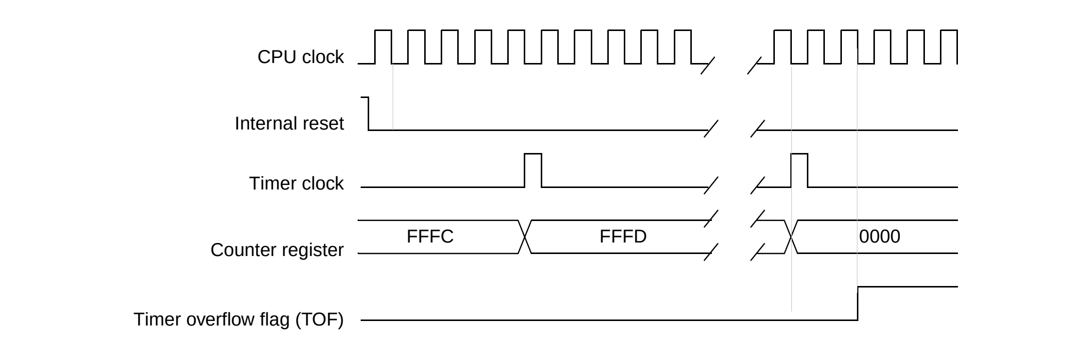
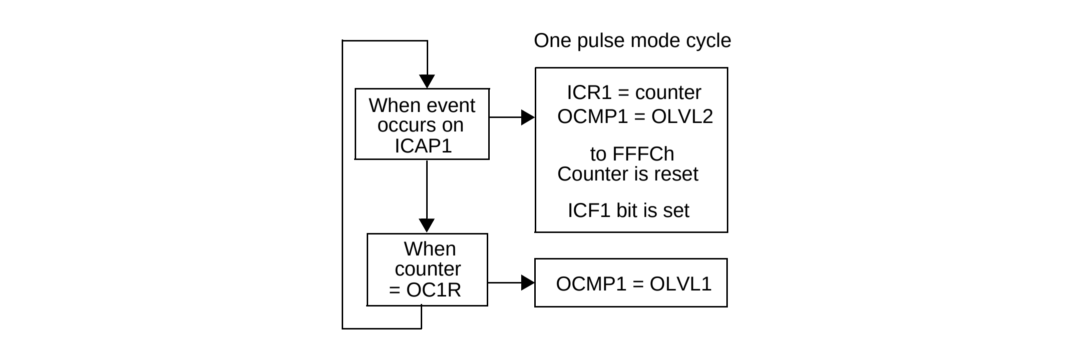


Figure 59. Counter timing diagram, internal clock divided by 8



Note: *The device is in reset state when the internal reset signal is high, when it is low the device is running.*

Figure 65. One pulse mode sequence



When a valid event occurs on the ICAP1 pin, the counter value is loaded in the ICR1 register. The counter is then initialized to FFFCh, the OLVL2 bit is output on the OCMP1 pin and the ICF1 bit is set.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the input capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. Accessing (reading or writing) the IC*i*LR register.

The OC1R register value required for a specific timing application can be calculated using the following formula:

Equation 3

$$\text{OC}i\text{R value} = \frac{t * f_{CPU}}{\text{PRESC}} - 5$$

Where:

t = pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits, see *: Timer A Control register 2 (TACR2) on page 134*)

If the timer clock is an external clock the formula is:

Equation 4

$$\text{OC}i\text{R} = t * f_{EXT} - 5$$

Where:

t = pulse period (in seconds)

f_{EXT} = external timer clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (see *Figure 66*).

Timer A Output compare 1 high register (TAOC1HR)

Reset value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB
Read / Write							

Timer A Output compare 1 low register (TAOC1LR)

Reset value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB
Read / Write							

Output compare 2 high register (OC2HR)

Reset value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB
Read / Write							

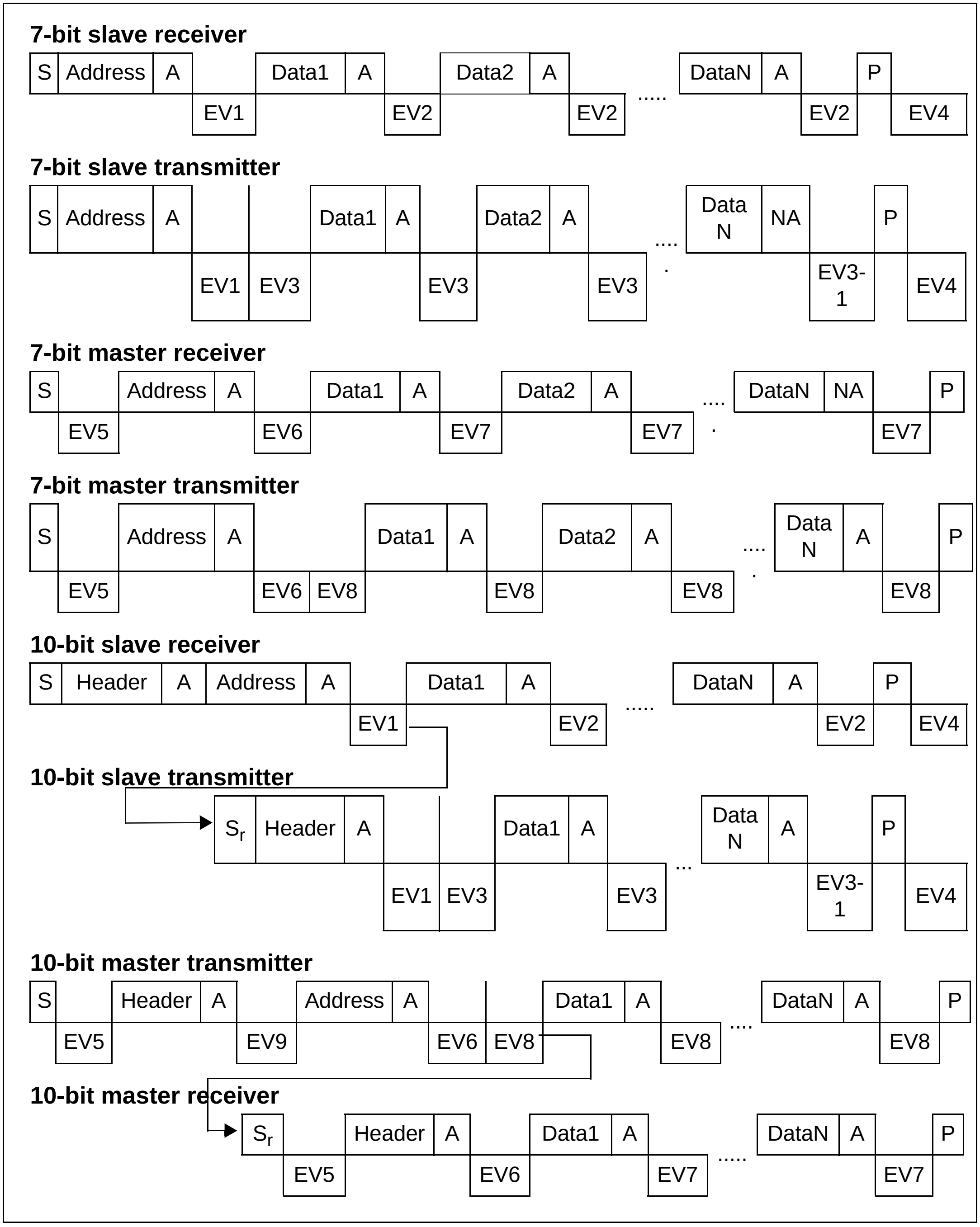
Output compare 2 low register (OC2LR)

Reset value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB
Read / Write							

Figure 71. Transfer sequencing


7-bit slave receiver
S Address A Data1 A Data2 A ..... DataN A P
EV1 EV2 EV2 EV2 EV4
7-bit slave transmitter
S Address A Data1 A Data2 A ..... Data N NA P
EV1 EV3 EV3 EV3 EV3-1 EV4
7-bit master receiver
S Address A Data1 A Data2 A ..... DataN NA P
EV5 EV6 EV7 EV7 EV7
7-bit master transmitter
S Address A Data1 A Data2 A ..... Data N A P
EV5 EV6 EV8 EV8 EV8 EV8
10-bit slave receiver
S Header A Address A Data1 A ..... DataN A P
EV1 EV2 EV2 EV4
10-bit slave transmitter
Sr Header A Data1 A ... Data N A P
EV1 EV3 EV3 EV3-1 EV4
10-bit master transmitter
S Header A Address A Data1 A .... DataN A P
EV5 EV9 EV6 EV8 EV8 EV8
10-bit master receiver
Sr Header A Data1 A ..... DataN A P
EV5 EV6 EV7 EV7

1. S=Start, S_r = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1).
2. **EV1:** EVF=1, ADSL=1, cleared by reading SR1 register.
3. **EV2:** EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.
4. **EV3:** EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.
5. **EV3-1:** EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). If lines are released by STOP=1, STOP=0, the

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see *Section : Overrun condition (OVR)*).

10.6.5 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See *Figure 77*).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

The combination of the CPOL clock polarity and CPHA (clock phase) bits selects the data capture clock edge.

Figure 77 shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin and the MOSI pin are directly connected between the master and the slave device.

Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.

Bit 2 = **CPHA** *Clock phase.*

This bit is set and cleared by software.

0: The first clock transition is the first data capture edge.

1: The second clock transition is the first capture edge.

The slave must have the same CPOL and CPHA settings as the master.

Bits 1:0 = **SPR[1:0]** *Serial clock frequency.*

These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode.

These 2 bits have no effect in slave mode.

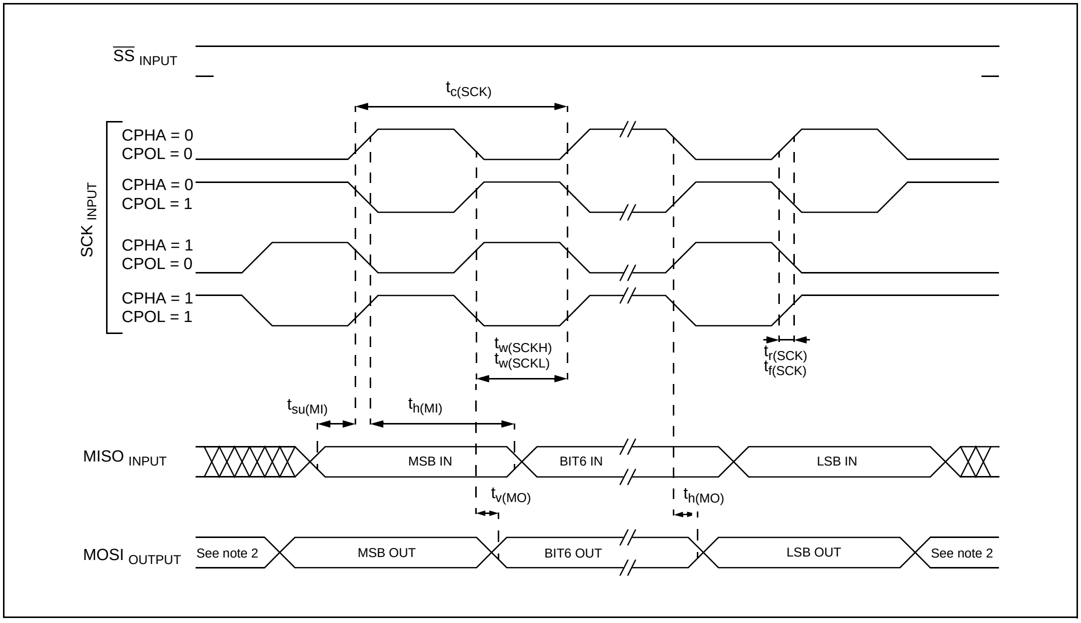
Table 52. SPI master mode SCK frequency

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

Table 53. SPI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
70	SPIDR Reset Value	MSB x	x	x	x	x	x	x	LSB x
71	SPICR Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
72	SPICSR Reset Value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

Figure 85. SPI master timing diagram



1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

12.6 Clock and timing characteristics

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A.

Table 77. General timings

Symbol	Parameter[1]	Conditions	Min	Typ[2]	Max	Unit
$t_{c(INST)}$	Instruction cycle time	f_{CPU} = 8 MHz	2	3	12	t_{CPU}
			250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time[3] $t_{v(IT)} = \Delta t_{c(INST)} + 10$	f_{CPU} = 8 MHz	10		22	t_{CPU}
			1.25		2.75	µs

1. Guaranteed by Design. Not tested in production.
2. Data based on typical application software.
3. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.