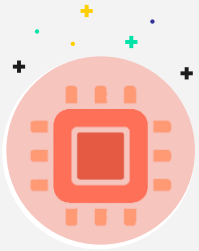
1404/03



Microcontrollers

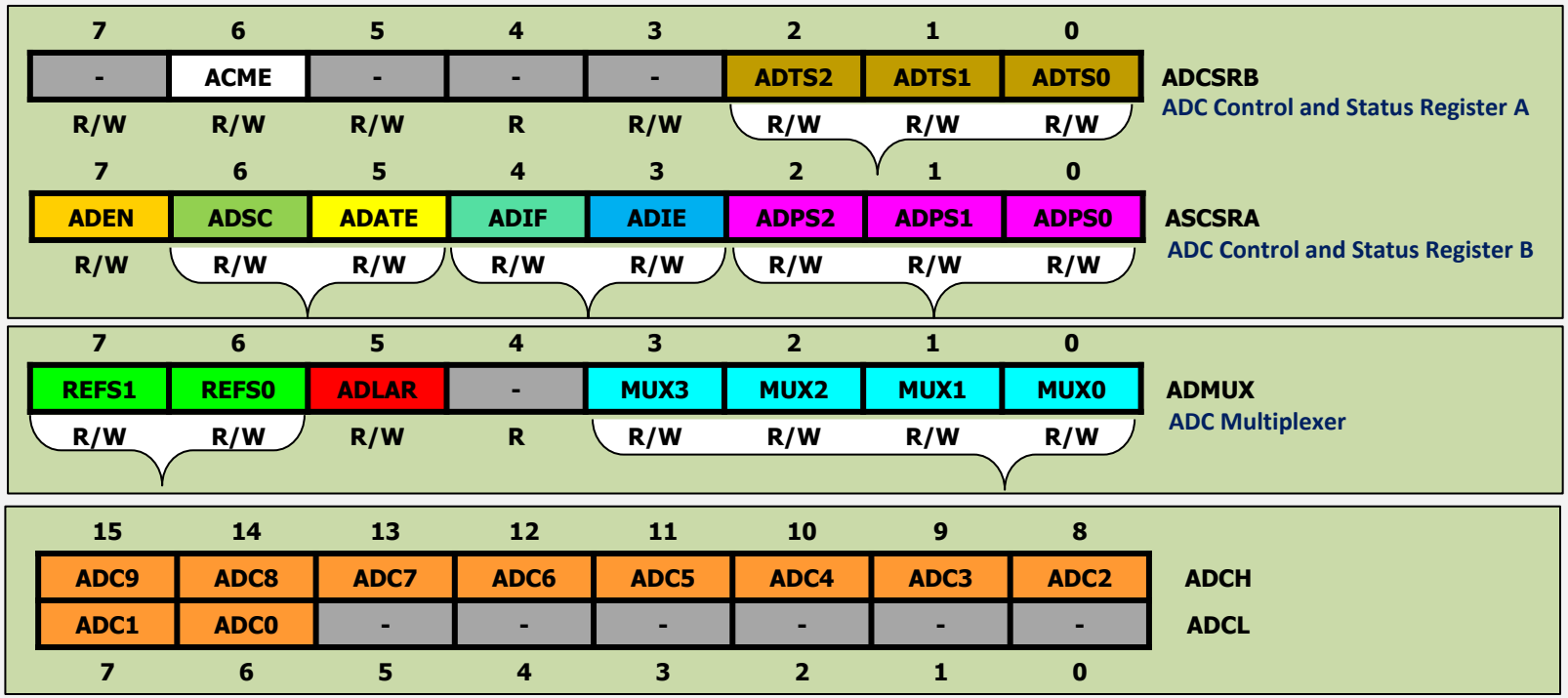
Lecture 10:

Examples

By: M.Razeghizadeh

Start now!

ADC Registers



ADC Tables

Table 24-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Table 24-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

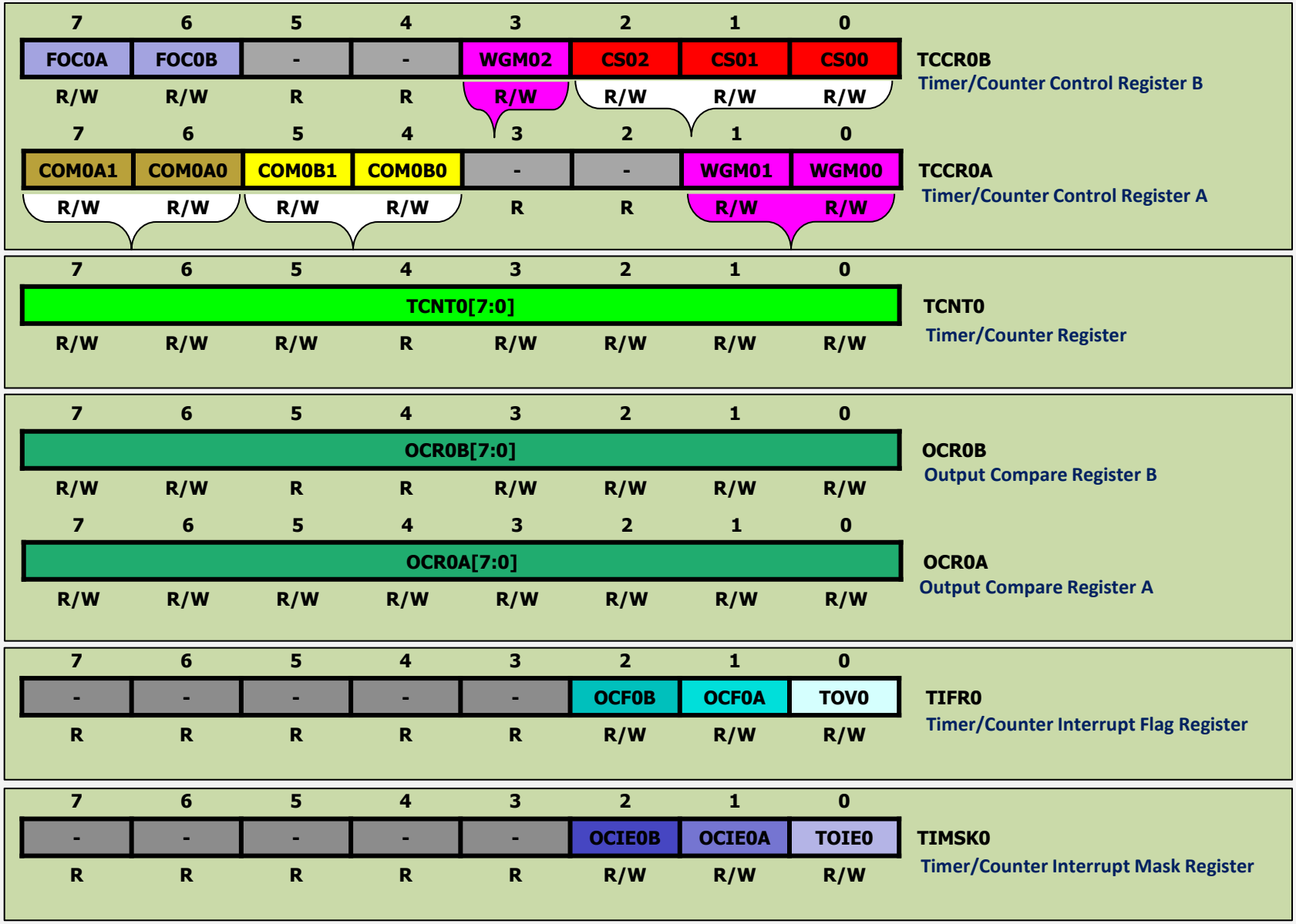
Table 24-4. Input Channel Selections

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Table 24-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Timer Registers



Timer Tables

Table 15-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[1][2]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Table 15-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Table 15-6. Compare Output Mode, Fast PWM Mode[1]

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at BOTTOM, (non-inverting mode)
1	1	Set OC0B on Compare Match, clear OC0B at BOTTOM, (inverting mode).

Table 15-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	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

External Interrupt Registers

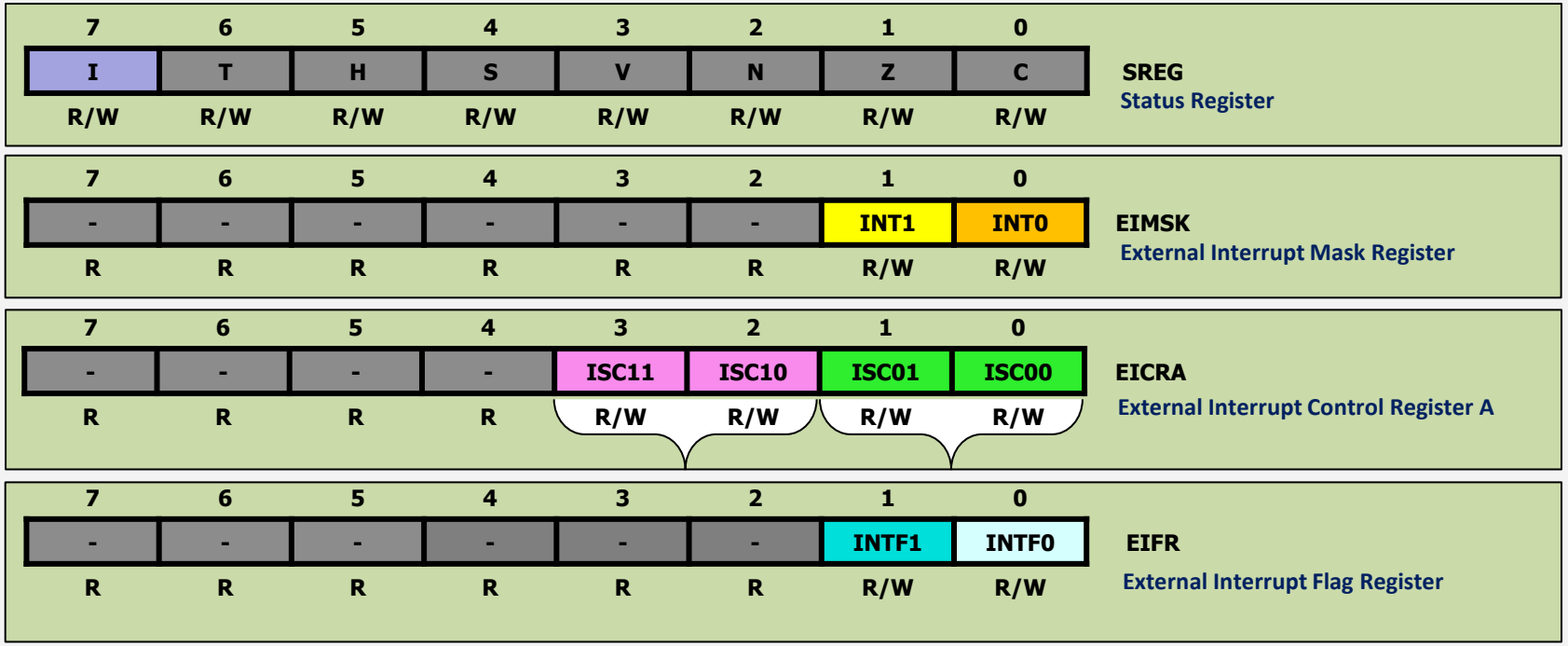


Table 12-6. Reset and Interrupt Vectors in ATmega328 and ATmega328P

VectorNo.	Program Address[2]	Source	Interrupt Definition
1	0x0000[1]	RESET	External Pin, Power-on Reset, Brown-out R
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow

Table 13-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Pin Change Interrupt Registers

7	6	5	4	3	2	1	0
I	T	H	S	V	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SREG
Status Register

7	6	5	4	3	2	1	0
-	-	-	-	-	PCIE2	PCIE1	PCIE0
R	R	R	R	R	R/W	R/W	R/W

PCICR
Pin Change Interrupt Control Register

PCIE0: Pin Change Interrupt Enable bit for PORTB
PCIE1: Pin Change Interrupt Enable bit for PORTC (0: disabled, 1: enabled)
PCIE2: Pin Change Interrupt Enable bit for PORTD (0: disabled, 1: enabled)

7	6	5	4	3	2	1	0
PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCMSK0 (for PORT B)
Pin Change Mask Register 0

7	6	5	4	3	2	1	0
PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

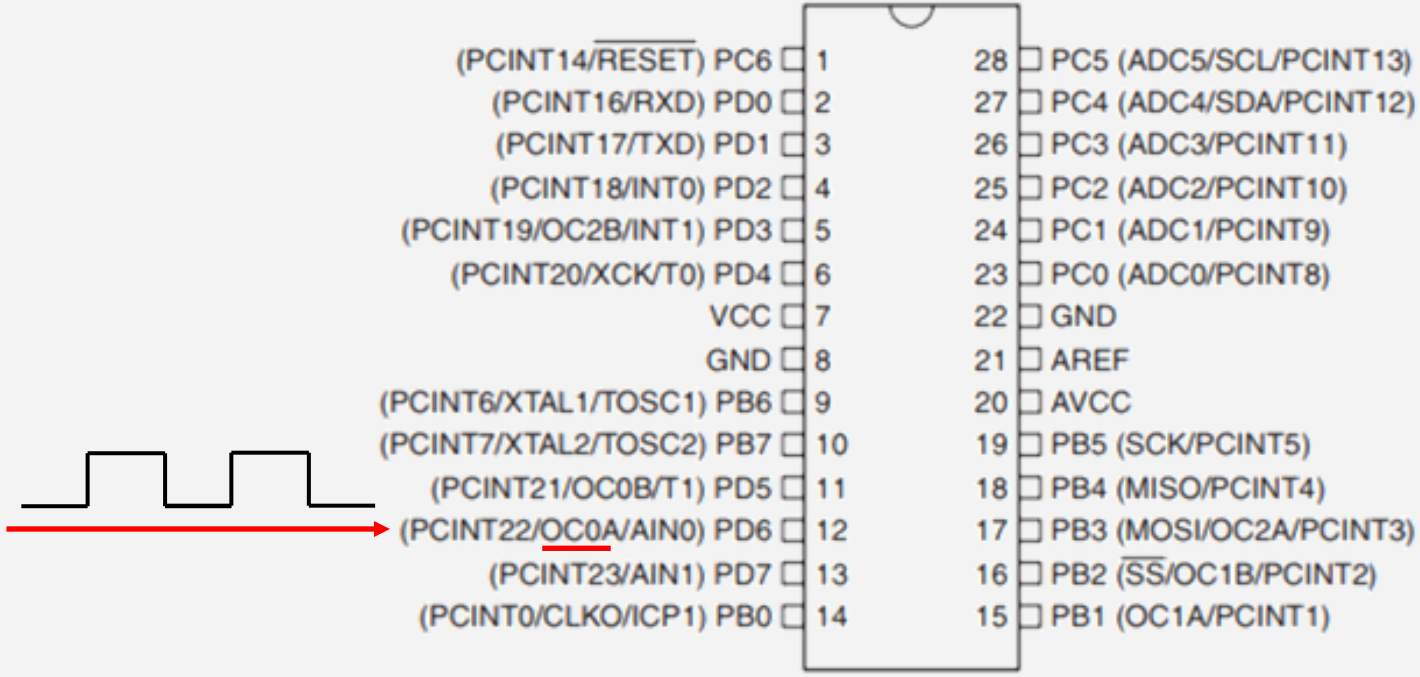
PCMSK1 (for PORT C)
Pin Change Mask Register 1

7	6	5	4	3	2	1	0
PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCMSK2 (for PORT D)
Pin Change Mask Register 1

Example 1

یک میکروکنترلر ATmega328 با فرکانس کلاک 16 مگاهرتز دارید. با استفاده از **تایمر 0** در **مد CTC** یک سیگنال مربعی با فرکانس 2 کیلوهرتز را روی پایه OC0A تولید کنید.



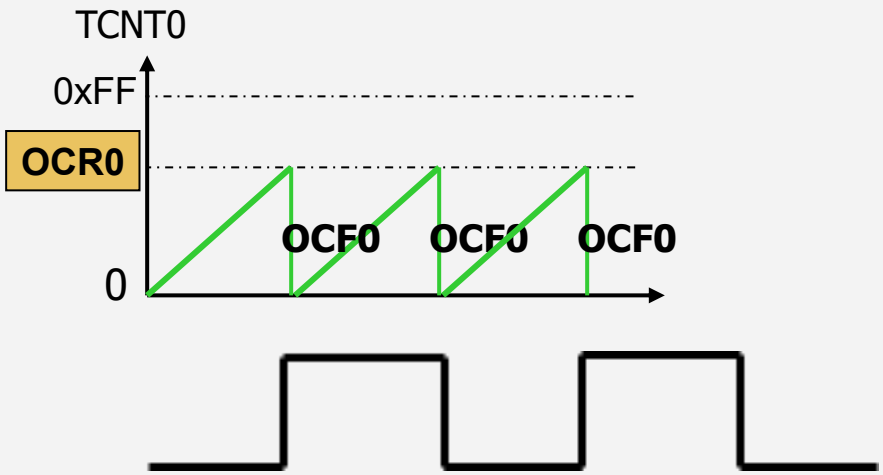
Example 1

$f_{clk} = 16\ MHz$

$f_{pulse} = 2\ kHz, Squre\ wave$

Timer 0 in CTC Mode > OCR0A=?

Output on OC0A > PD6:output



$$f_{Timer} = \frac{16MHz}{64(prescaler)}$$

$$f_{OC0A} = \frac{f_{Timer}}{2 \times (OCR0A + 1)}$$

:

$$1000 = \frac{16 \times 10^6}{2 \times 64 \times (OCR0A + 1)}$$

$$OCR0A + 1 = \frac{16 \times 10^6}{2 \times 64 \times 1000} = \frac{16 \times 10^6}{128000} = 125$$

$$OCR0A = 124$$

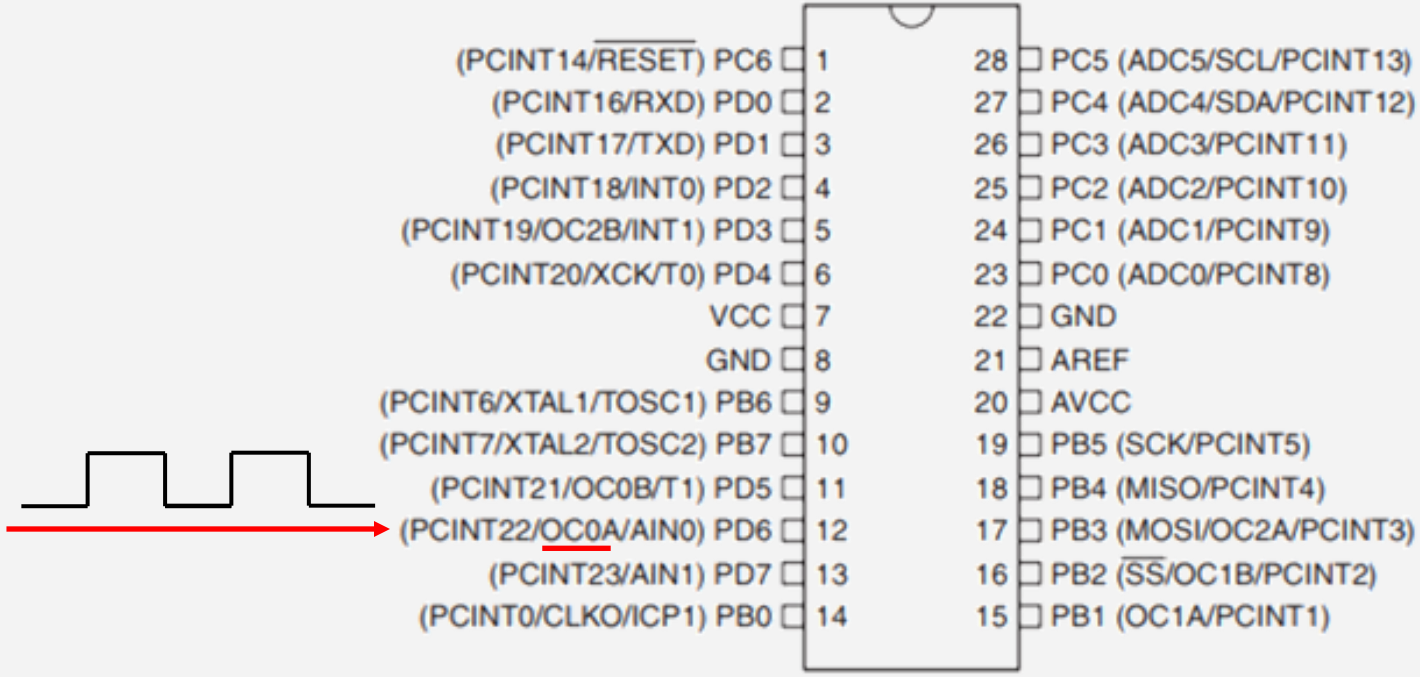
Example 1

```
void setup() {
  DDRD |= (1 << 6);
  OCR0A = 124;
  TCCR0A |= (1 << COM0A0);   // Toggle mode
  TCCR0A |= (1 << WGM01);    // CTC mode
  TCCR0B |= (1 << CS00) | (1 << CS01);
}

void loop() {}
```

```
void setup() {
  DDRD = 0x40;              // PD6 = خروجی
  TCCR0A = 0x42;            // CTC + Toggle mode
  OCR0A = 0x7C;             // 124
  TCCR0B = 0x03;            // Prescaler = 64
}

void loop() {}
```

Example 2

یک میکروکنترلر ATmega328 با فرکانس کلاک 16 مگاهرتز دارید. با استفاده از **تایمر 0** در **مد CTC** یک سیگنال مربعی با فرکانس 2 کیلوهرتز را روی پایه OC0A تولید کنید.



Example 2

به کمک واحد تایمر/ شمارنده 0 میکروکنترلر ATMEGA328، برنامه ای بنویسید که یک شکل موج مربعی با دوره تناوب 16.6 میکروثانیه بر روی پایه شماره 5 پورت B در مدهای زیر ایجاد شود. (فرضیات: XTAL = 10MHZ)

XTAL = 10 MHz

$$T_{clk} = \frac{1}{10\ MHz} = 0.1\ \mu s$$

$$T_{wave} = 16.6\ \mu s$$

$$\frac{T_{wave}}{2} = \frac{16.6\ \mu s}{2} = 8.3\ \mu s$$

$$\frac{T_{wave}}{2} = (256 - TCNT0_{initial}) \times T_{timer}$$

$$\frac{T_{wave}}{2} = (256 - TCNT0_{initial}) \times \frac{1}{f_{timer}}$$

$$\frac{T_{wave}}{2} = (256 - TCNT0_{initial}) \times \frac{Prescaler}{f_{clock}}$$

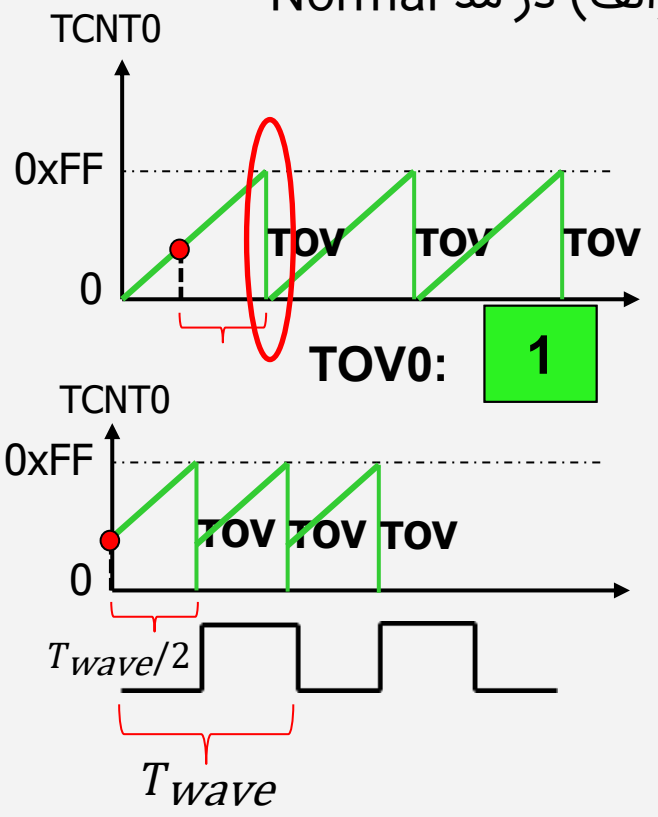
$$\frac{T_{wave}}{2} = (256 - TCNT0_{initial}) \times T_{clk} \times Prescaler$$

$$8.3\ \mu s = (256 - TCNT0_{initial}) \times 0.1\ \mu s \times Prescaler$$

$$TCNT0_{initial} = 256 - 83/\text{Prescaler}$$

$TCNT0_{initial} = 173$, Prescaler = 1

(الف) در مد Normal



```
#include <avr/io.h>
#include <avr/interrupt.h>

int main(void)
{
   DDRB |= (1 << PB5);
   PORTB &= ~(1 << PB5);

   TCCR0A = 0x00;
   TCNT0 = 173;
   TCCR0B = (1 << CS00);

   TIFR0 |= (1 << TOV0);

   while(1)
   {
      if (TIFR0 & (1 << TOV0))
      {
         TCNT0 = 173;
         PORTB ^= (1 << PB5);
         TIFR0 |= (1 << TOV0);
      }
   }
}
```

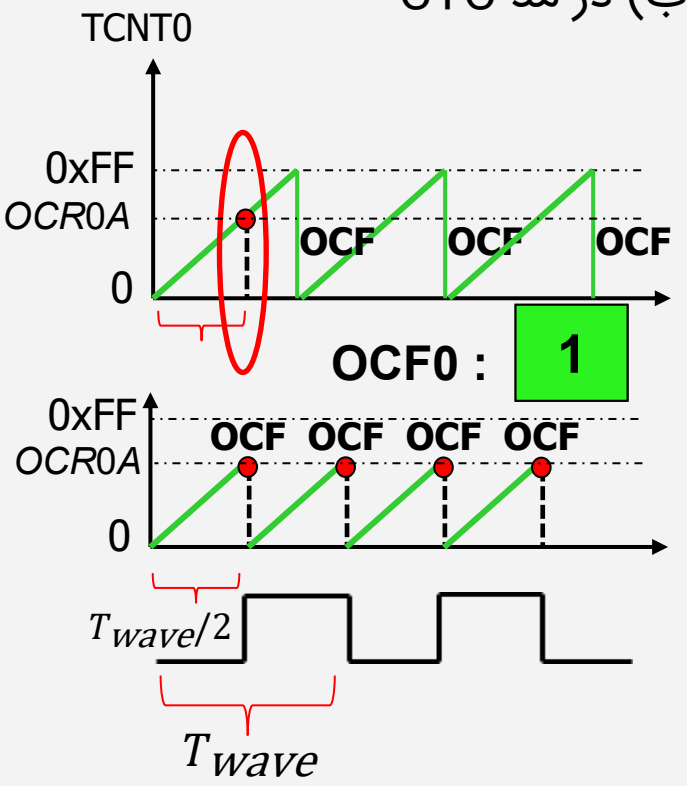
Example 2

```
#include <avr/io.h>
#include <avr/interrupt.h>

int main(void)
{
DDRB |= (1 << PB5);
PORTB &= ~(1 << PB5);

TCCR0A = 0x00;
TCNT0 = 173;
TIFR0 |= (1 << TOV0);
TIMSK0 |= (1 << TOIE0);

sei();

TCCR0B |= (1 << CS00);

   while(1){}
}


ISR(TIMER0_OVF_vect)
{
  TCNT0 = 173;
  PORTB ^= (1 << PB5);
}
```

Example 2

به کمک واحد تایمر/ شمارنده 0 میکروکنترلر ATMEGA328، برنامه ای بنویسید که یک شکل موج مربعی با دوره تناوب 16.6 میکروثانیه بر روی پایه شماره 5 پورت B در مدهای زیر ایجاد شود. (فرضیات: XTAL = 10MHZ)

(ب) در مد CTC



XTAL = 10 MHz

$$T_{clk} = \frac{1}{10\ MHz} = 0.1\ \mu s$$

$$T_{wave} = 16.6\ \mu s$$

$$\frac{T_{wave}}{2} = \frac{16.6\ \mu s}{2} = 8.3\ \mu s$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times T_{timer}$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times \frac{1}{f_{timer}}$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times \frac{Prescaler}{f_{clock}}$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times T_{clk} \times Prescaler$$

$$8.3\ \mu s = (OCR0A + 1) \times 0.1\ \mu s \times Prescaler$$

$$OCR0A = \frac{83}{\text{Prescaler}} - 1$$

$OCR0A = 82$, Prescaler = 1

Example 2

```
#include <avr/io.h>

int main(void)
{
    DDRB |= (1 << PB5);
    PORTB &= ~(1 << PB5);

    OCR0A = 82;
    TCNT0 = 0;
    TCCR0A = (1 << WGM01);
    TCCR0B = (1 << CS00);


    while(1)
    {
        if (TIFR0 & (1 << OCF0A))
        {
            PORTB ^= (1 << PB5);
            TIFR0 |= (1 << OCF0A);
        }
    }
}
```

```
#include <avr/io.h>
#include <avr/interrupt.h>

int main(void)
{
   DDRB |= (1 << PB5);
   PORTB &= ~(1 << PB5);

   TCCR0A = (1 << WGM01);
   OCR0A = 82;
   TCNT0 = 0;

   TIMSK0 |= (1 << OCIE0A);
   sei();

   TCCR0B = (1 << CS00);

   while(1);
}

ISR(TIMER0_COMPA_vect)
{
   PORTB ^= (1 << PB5);
}
```

Example 2

به کمک واحد تایمر/ شمارنده 0 میکروکنترلر ATMEGA328، برنامه ای بنویسید که یک شکل موج مربعی با دوره تناوب 16.6 میکروثانیه بر روی پایه خروجی تایمر شماره 0 در مدهای زیر ایجاد شود. (فرضیات: XTAL = 10MHZ)

(ب) در مد CTC

XTAL = 10 MHz

$$T_{clk} = \frac{1}{10\ MHz} = 0.1\ \mu s$$

$$T_{wave} = 16.6\ \mu s$$

$$\frac{T_{wave}}{2} = \frac{16.6\ \mu s}{2} = 8.3\ \mu s$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times T_{timer}$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times \frac{1}{f_{timer}}$$

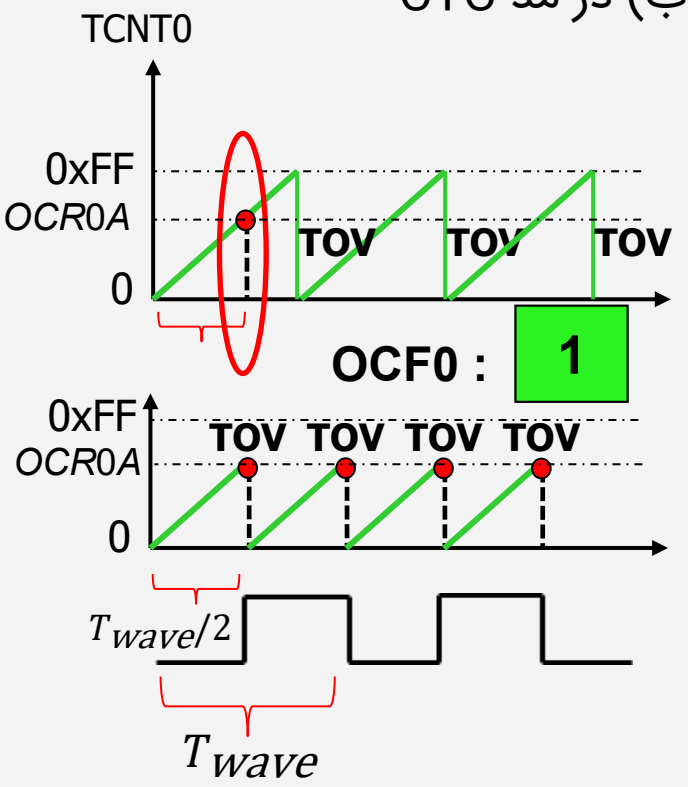
$$\frac{T_{wave}}{2} = (OCR0A + 1) \times \frac{Prescaler}{f_{clock}}$$

$$\frac{T_{wave}}{2} = (OCR0A + 1) \times T_{clk} \times Prescaler$$

$$8.3\ \mu s = (OCR0A + 1) \times 0.1\ \mu s \times Prescaler$$

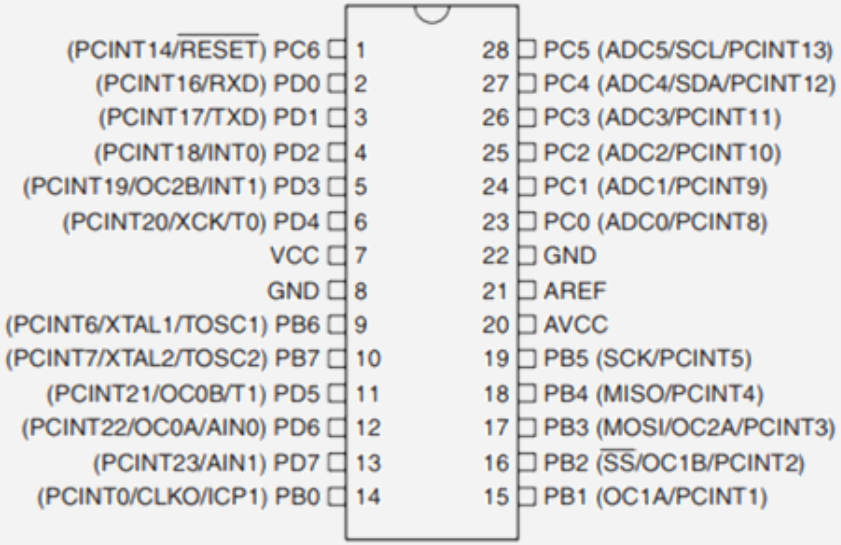
$$OCR0A = \frac{83}{\text{Prescaler}} - 1$$

$OCR0A = 82$, Prescaler = 1



Example 2

```
#include <avr/io.h>

int main(void)
{
    DDRD |= (1 << PD6);

    OCR0A = 82;
    TCNT0 = 0;

    TCCR0A = (1 << COM0A0) | (1 << WGM01);
    TCCR0B = (1 << CS00);


    while(1)
    {
    }
}
```



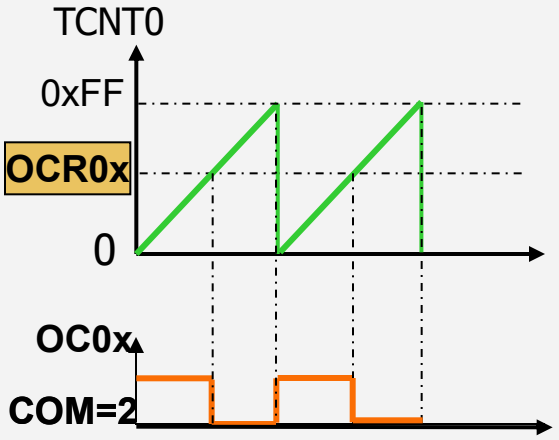
Example 3

به کمک واحد تایمر/ شمارنده میکروکنترلر ATMEGA328، برنامه ای بنویسید که در مد PWM Fast یک شکل موج پالسی با دوره تناوب 16 میکروثانیه و duty cycle = 75% بر روی پایه شماره 6 پورت D ایجاد شود. (فرضیات: XTAL = 16MHZ)

$$F_{OC0}=\frac{f_{clk}}{N(256)} \qquad \frac{1}{16\mu s}=\frac{16Mhz}{N(256)} \qquad N=1$$

$$\text{Duty Cycle} = \frac{OCR0 + 1}{256} \times 100$$

$$75 = \frac{OCR0 + 1}{256} \times 100 \rightarrow OCR0 = 191$$



$$\text{Duty Cycle} = \frac{OCR0 + 1}{256} \times 100$$

$$F_{OC0}=\frac{f_{clk}}{N(256)}$$

Example 3

به کمک واحد تایمر/ شمارنده میکروکنترلر ATMEGA328، برنامه ای بنویسید که در مد PWM Fast یک شکل موج پالسی با دوره تناوب 16 میکروثانیه و duty cycle = 75% بر روی پایه شماره 6 پورت D ایجاد شود. (فرضیات: XTAL = 16MHZ)

```
#include <avr/io.h>

int main(void)
{
   DDRD |= (1 << PD5);

   TCCR0A = (1 << WGM01) | (1 << WGM00);
   OCR0A = 191;

   TCCR0B = (1 << CS00);
      while(1){

      if (TIFR0 & (1 << OCF0A))
      {
         PORTB ^= (1 << PD6);
         TIFR0 |= (1 << OCF0A);
      }
}
```

به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 8-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 490 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت HIGH باشد.

$$f_{OC1} = \frac{f_{clk}}{N \times (2 \times TOP)} \qquad \frac{1}{490\ hz} = \frac{16Mhz}{N(510)}$$

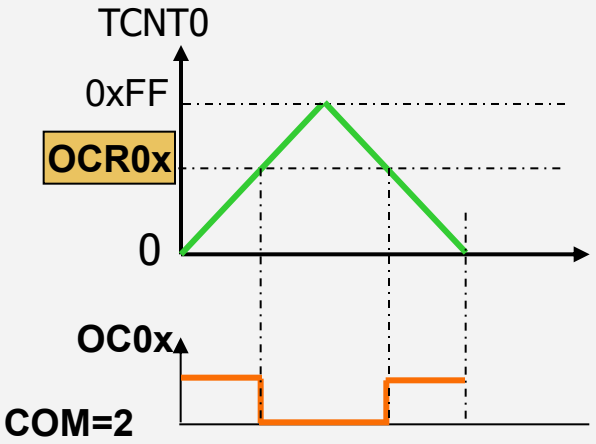
$$N=64.02 \qquad N \approx 64$$

$$f = \frac{16 \times 10^6}{2 \times 64 \times 255} = \frac{16 \times 10^6}{32640} \approx 490.2\ Hz$$

$$\text{Duty Cycle} = \left(\frac{OCR1A}{TOP}\right) \times 100\%$$

$$0.25 = \frac{OCR1A}{255}$$

$$OCR1A = 63.75 \approx 64$$



$$\text{Duty Cycle} = \frac{OCRx}{255} \times 100$$

$$F_{OC0} = \frac{f_{clk}}{N(510)}$$

Example 3

به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 8-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 490 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت HIGH باشد.

```
#include <avr/io.h>

int main(void) {

        DDRB |= (1 << PB1);

        OCR1A = 64;

        TCCR1A = (1 << WGM10) | (1 << COM1A1);

        TCCR1B |= (1 << CS11) | (1 << CS10);

        while (1){}

}
```

Example 3

به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 10-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 122 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت HIGH باشد.

$$f_{OC1} = \frac{f_{clk}}{N \times (2 \times TOP)} \qquad \frac{1}{122\ hz} = \frac{16Mhz}{N(2*1023)}$$

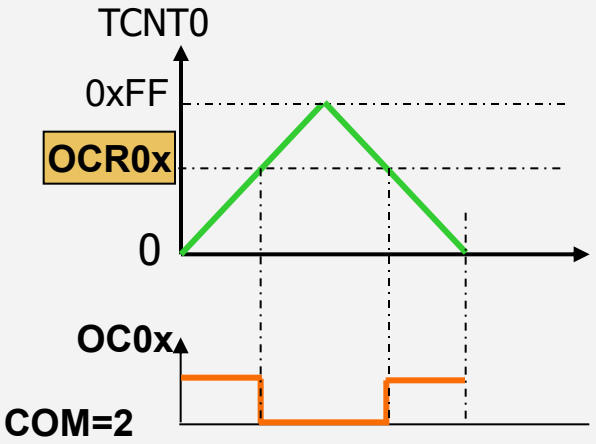
$$N=64.1 \qquad N \approx 64$$

$$f = \frac{16 \times 10^6}{2 \times 64 \times 1023} = \frac{16 \times 10^6}{130944} \approx 122.19\ Hz$$

$$\text{Duty Cycle} = \left(\frac{OCR1A}{TOP}\right) \times 100\%$$

$$0.25 = \frac{OCR1A}{1023}$$

$$OCR1A = 255.75 \approx 256$$



$$\text{Duty Cycle} = \frac{OCRx}{255} \times 100$$

$$F_{OC0} = \frac{f_{clk}}{N(510)}$$

Example 3

به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 8-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 490 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت HIGH باشد.

```
#include <avr/io.h>

int main(void) {

        DDRB |= (1 << PB1);

        OCR1A = 256;

        TCCR1A = (1 << WGM10) | (1 << WGM11) | (1 << COM1A1);

        TCCR1B |= (1 << CS11) | (1 << CS10);

        while (1){}

}
```

Example 3

به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 10-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 122 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت low باشد.

$$f_{OC1} = \frac{f_{clk}}{N \times (2 \times TOP)} \qquad \frac{1}{122\ hz} = \frac{16Mhz}{N(2*1023)}$$

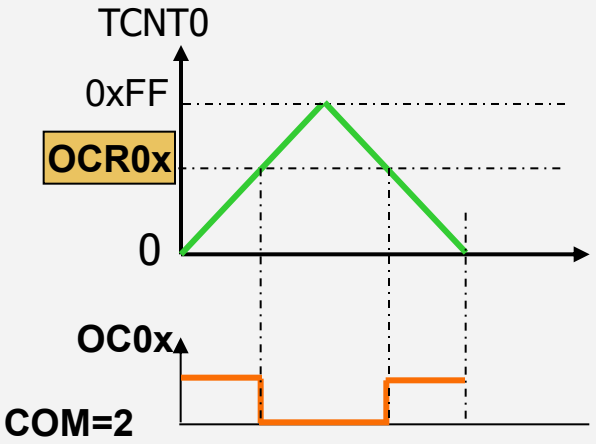
$$N = 64.1 \qquad N \approx 64$$

$$f = \frac{16 \times 10^6}{2 \times 64 \times 1023} = \frac{16 \times 10^6}{130944} \approx 122.19\ Hz$$

$$\text{Duty Cycle} = \left(\frac{TOP - OCR1A}{TOP}\right) \times 100\%$$

$$0.25 = \frac{1023 - OCR1A}{1023}$$

$$OCR1A = 767.25 \approx 767$$



$$\text{Duty Cycle} = \frac{OCRx}{255} \times 100$$

$$F_{OC0} = \frac{f_{clk}}{N(510)}$$

Example 3

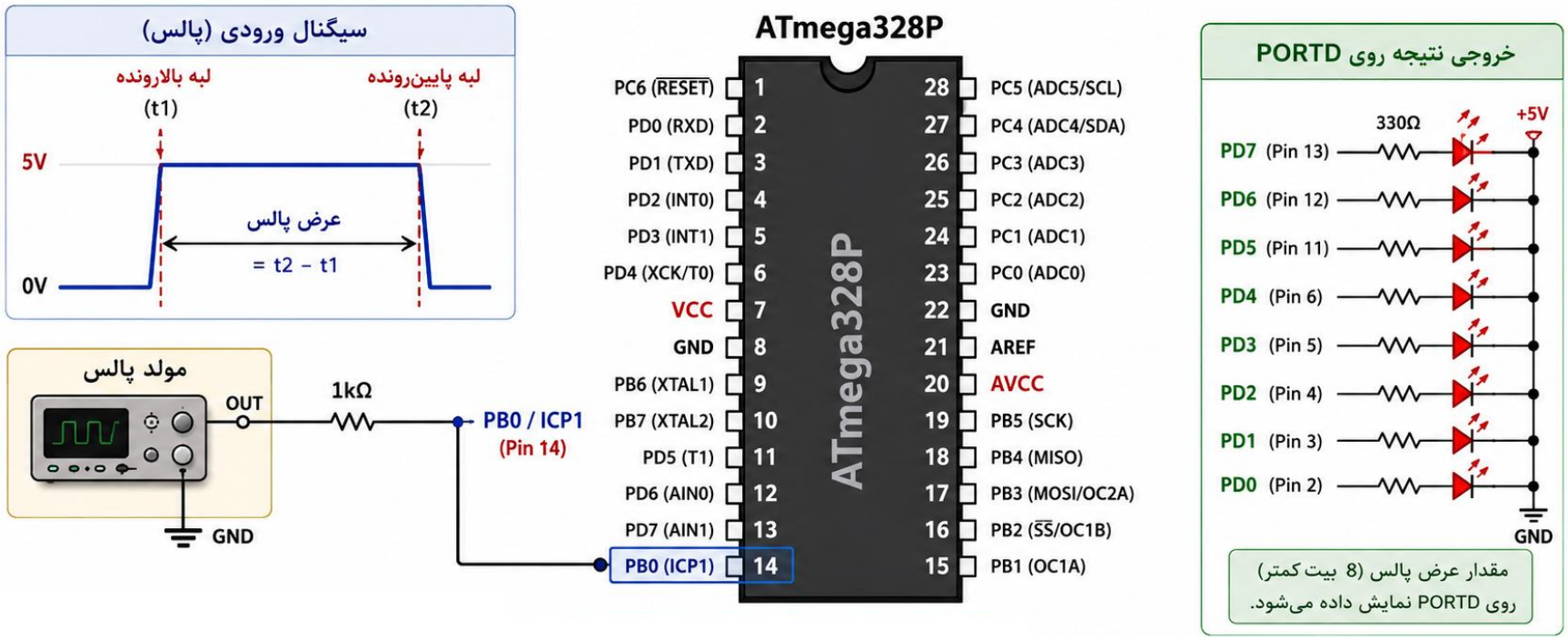
به کمک واحد تایمر/ شمارنده شماره 1 میکروکنترلر ATMEGA328، در مد Phase Correct PWM 10-bit برنامه‌ای بنویسید که یک PWM با فرکانس تقریبی 122 Hz و Duty Cycle برابر 25% روی OC1A ایجاد کند. (فرضیات: XTAL = 16MHZ)

شروع سیگنال در ابتدای هر سیکل باید در حالت low باشد.

```
#include <avr/io.h>

int main(void) {

        DDRB |= (1 << PB1);

        OCR1A = 767;

        TCCR1A = (1 << WGM10) | (1 << WGM11) | (1 << COM1A0)| (1 << COM1A1);

        TCCR1B |= (1 << CS11) | (1 << CS10);

        while (1){}

}
```

Example 4

در میکروکنترلر ATmega۳۲۸P برنامه‌ای به زبان C بنویسید که عرض یک پالس ورودی با پهنای پالس بین ۱ تا ۱۲۵ µs را با استفاده از واحد Input Capture تایمر ۱ اندازه‌گیری کند. برنامه باید ابتدا لبه بالارونده و سپس لبه پایین‌رونده را تشخیص داده، عرض پالس را محاسبه کرده و ۸ بیت کم‌ارزش نتیجه را روی پورت D نمایش دهد. از روش Polling برای تشخیص رویدادهای Input Capture استفاده شود.
(فرضیات: XTAL = 16MHZ)



Example 4

$1\mu s < Pulse\ Width < 125\mu s \rightarrow Min: 1\mu s \rightarrow T_{timer} < 1\mu s$ (1)

$f_{timer} = \frac{f_{CPU}}{Prescaler} \rightarrow \frac{1}{T_{timer}} = \frac{f_{CPU}}{Prescaler} \rightarrow T_{timer} = \frac{Prescaler}{f_{CPU}}$ (2)

(1) (2) ⇒ $T_{timer} < 1\mu s \gg \frac{Prescaler}{f_{CPU}} < 1\mu s$

$\frac{Prescaler}{16\ MHz} < 1\mu s \rightarrow Prescaler < 16$, *Avail Prescalers in Timer* 1: 8, 64, 256, 1024

$Prescaler = 8 \rightarrow T_{timer} = \frac{8}{16\ MHz} = 0.5\ \mu s$

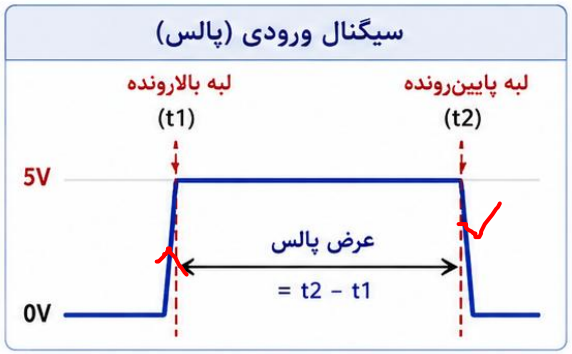
Example 4

$\frac{1\mu s}{0.5\ \mu s} = 2$

$\frac{125\mu s}{0.5\ \mu s} = 250$

$\longrightarrow \quad 2 < CNT_{T_2} - CNT_{T_1} < 250$

$OUTPUT: 8bit\ Register$



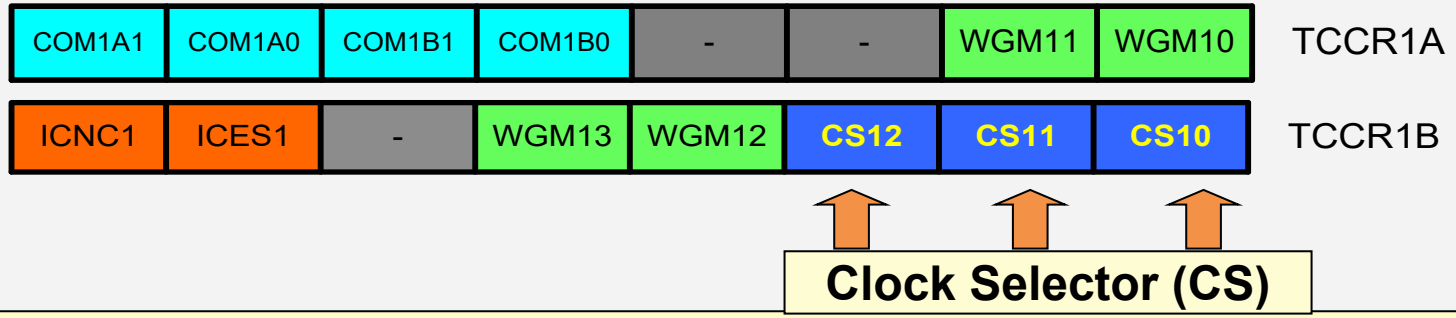
$Mode: Normal$ ✓

$Prescaler: 8$ ✓

$T1:\ Rising\ Edge$ ✓

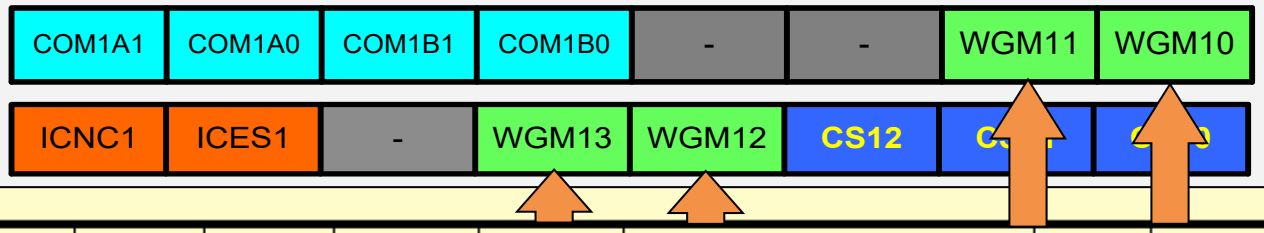
$T2: Falling\ Edge$ ✓

Example 4



CS12	CS11	CS10	Comment
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk (No Prescaling)
0	1	0	clk / 8
0	1	1	clk / 64
1	0	0	clk / 256
1	0	1	clk / 1024
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

Example 4



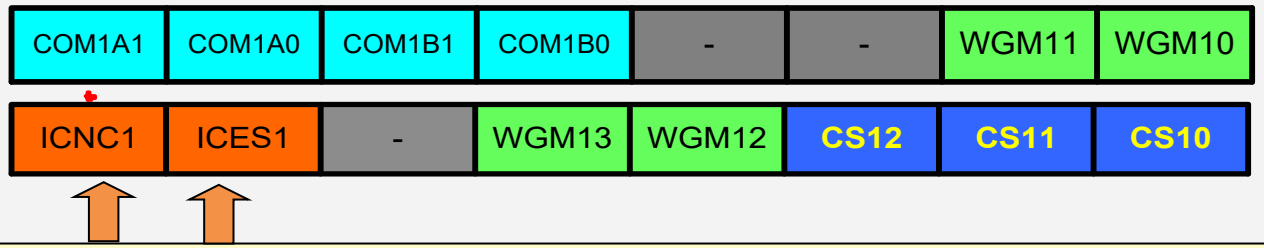
Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	TOP	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	TOP	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	TOP	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	–	–	–
14	1	1	1	0	Fast PWM	ICR1	TOP	TOP
15	1	1	1	1	Fast PWM	OCR1A	TOP	TOP

Example 4

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

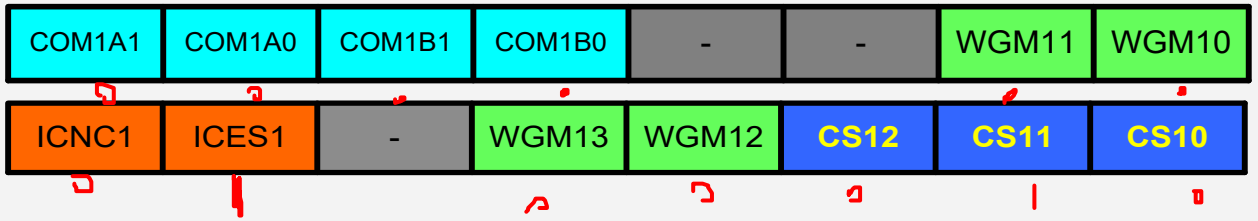
Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Example 4

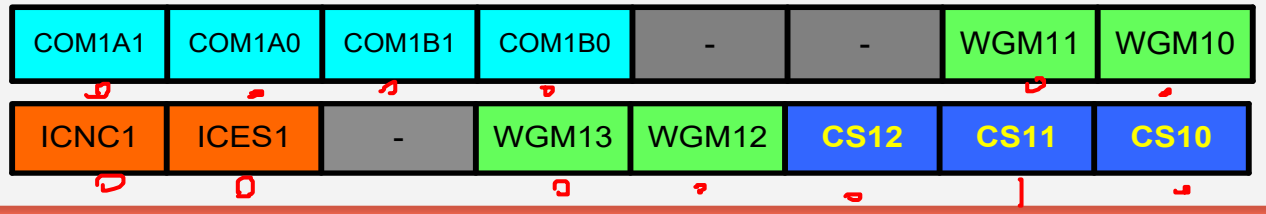


ICNC1: Input Capture Noise Canceller
0:disabled
1:Enabled (captures after 4 successive equal valued samples)
ICSES1: Input Capture Edge Select
0: Falling edge
1: Rising edge

T_1

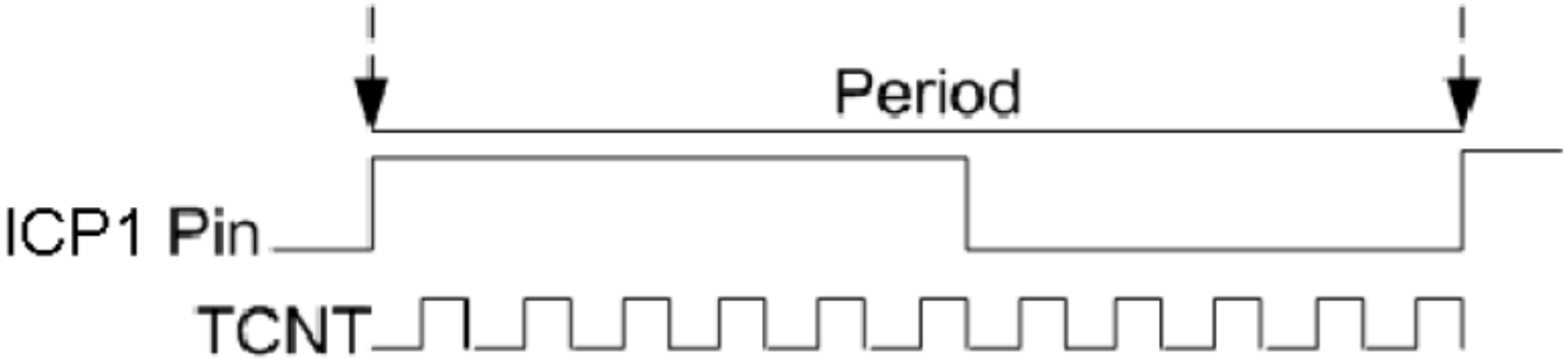


T_2

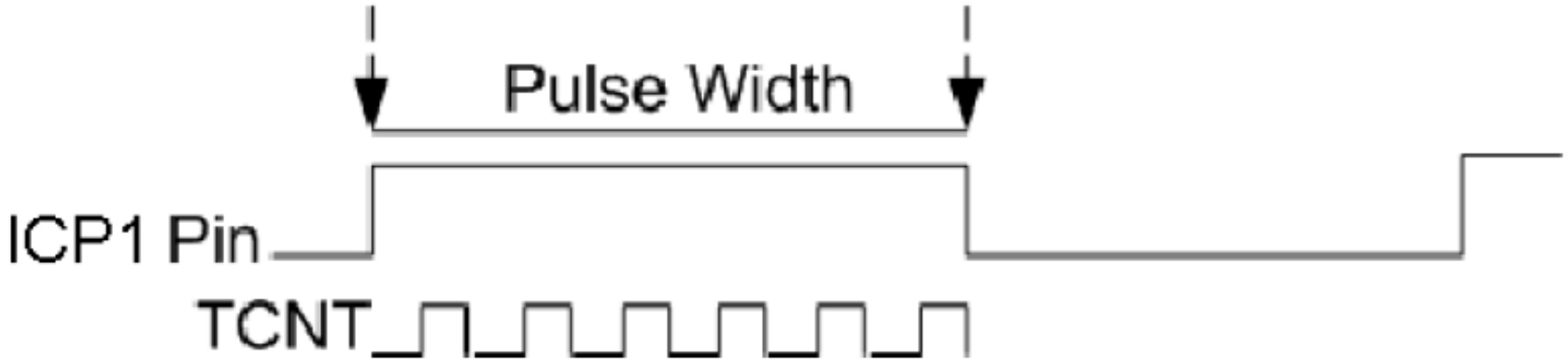


Example 4

```
#include <avr/io.h>

int main ( )
{
unsigned char t1;

DDRD = 0xFF; //Port D as output
DDRB &= ~(1 << PB0);

TCCR1A = 0; //Timer Mode = Normal
TCCR1B = 0x42; //rising edge, prescaler = 8, no noise canc.

while ((TIFR1&(1<<ICF1)) == 0);

t1 = ICR1L;
TIFR1 = (1<<ICF1); //clear ICF1 flag
TCCR1B = 0x02; //falling edge

while ((TIFR1&(1<<ICF1)) == 0);

PORTD = ICR1L - t1; //pulse width = falling - rising
TIFR1 = (1<<ICF1); //clear ICF1 flag

while (1); //wait forever
}
```



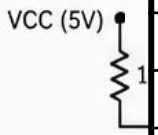
Measuring Period in Terms of the Number of Clocks Counted By TCNT



Measuring Pulse Width in Terms of the Number of Clocks Counted By TCNT

بنویسید که هرگاه پایه INT0 در لبه ... به ک
... تغییر وضعیت دهد. B پایه

```
#include <avr/io.h>

int main(void)
{
DDRB |= (1 << PB4);

DDRD &= ~(1 << PD2);
EICRA |= (1 << ISC01);

EIMSK |= (1 << INT0);

sei();

   while(1){}
}

ISR(INT0_vect)
{
   PORTB ^= (1 << PB4);
}
```



Interrupt	Vector Name
External Interrupt Request 0	INT0_vect
External Interrupt Request 1	INT1_vect
Pin Change Interrupt Request 0	PCINT0_vect
Pin Change Interrupt Request 1	PCINT1_vect
Pin Change Interrupt Request 2	PCINT2_vect
Watchdog Time-out Interrupt	WDT_vect
Timer/Counter2 Compare Match A	TIMER2_COMPA_vect
Timer/Counter2 Compare Match B	TIMER2_COMPB_vect
Timer/Counter2 Overflow	TIMER2_OVF_vect
Timer/Counter1 Capture Event	TIMER1_CAPT_vect
Timer/Counter1 Compare Match A	TIMER1_COMPA_vect
Timer/Counter1 Compare Match B	TIMER1_COMPB_vect
Timer/Counter1 Overflow	TIMER1_OVF_vect
Timer/Counter0 Compare Match A	TIMER0_COMPA_vect
Timer/Counter0 Compare Match B	TIMER0_COMPB_vect
Timer/Counter0 Overflow	TIMER0_OVF_vect
SPI Serial Transfer Complete	SPI_STC_vect
USART Rx Complete	USART_RX_vect
USART Data Register Empty	USART_UDRE_vect
USART Tx Complete	USART_TX_vect

External Interrupt Registers

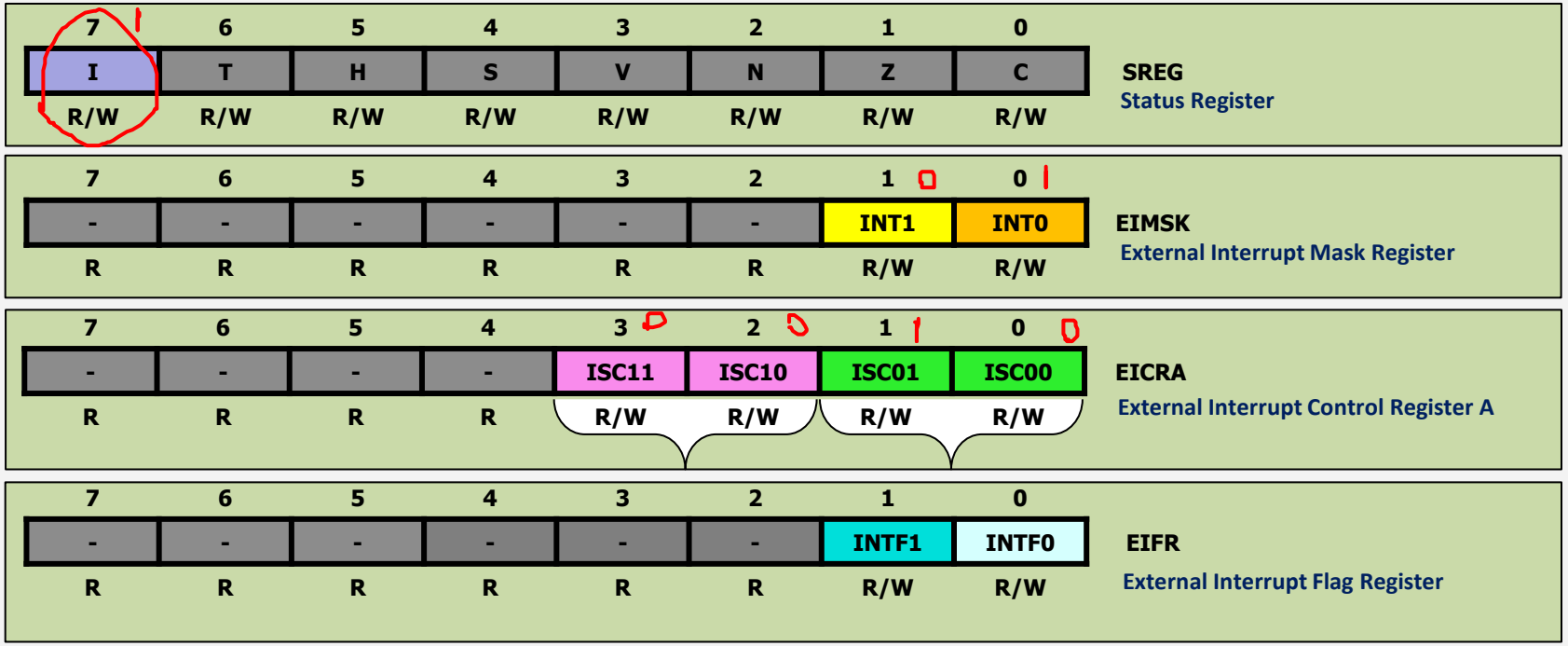


Table 12-6. Reset and Interrupt Vectors in ATmega328 and ATmega328P

VectorNo.	Program Address[2]	Source	Interrupt Definition
1	0x0000[1]	RESET	External Pin, Power-on Reset, Brown-out R
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow

Table 13-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Example 5

در یک دستگاه دزدگیر ساده خانگی، سه درب به ترتیب به پایه‌های PB۰، PB۱ و PB۴ یک میکروکنترلر ATmega۳۲۸ متصل شده‌اند. هر درب دارای یک سوئیچ مغناطیسی (رید سوئیچ) است که در حالت بسته بودن در، پایه مربوطه را به GND متصل می‌کند و در حالت باز بودن در، پایه به VCC می‌رود (با استفاده از پول‌آپ داخلی). یک آژیر نیز به پایه PB۵ متصل است.

برنامه‌ای بنویسید که:

- هر زمان هر کدام از درب‌ها باز شود (وضعیت پایه تغییر کند)، آژیر فوراً روشن شود.
- آژیر تا زمانی که هر سه درب بسته نشده‌اند، روشن بماند.
- به محض بسته شدن همه درب‌ها، آژیر خاموش شود.
- از Pin Change Interrupt برای تشخیص لحظه‌ای تغییر وضعیت درب‌ها استفاده شود تا آژیر بدون تأخیر فعال گردد.

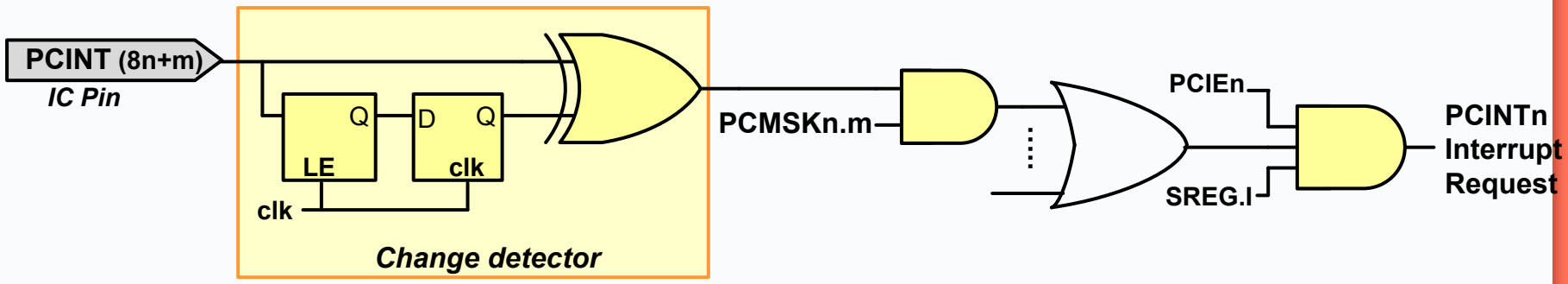
```
#include <avr/io.h>

int main(void) {
    DDRB |= (1 << PB5);

    PORTB |= (1 << PB0) | (1 << PB1) | (1 << PB4);
    DDRB  &= ~((1 << PB0) | (1 << PB1) | (1 << PB4));

    updateBuzzer();

    PCICR |= (1 << PCIE0);
    PCMSK0 |= (1 << PCINT0) | (1 << PCINT1) | (1 << PCINT4);

    sei();

    while (1) {}
}


ISR(PCINT0_vect) {
    updateBuzzer();
}
```

```
void updateBuzzer(void) {
    unsigned char currentState = 0;

    if (PINB & (1 << PB0)) currentState |= (1 << 0);
    if (PINB & (1 << PB1)) currentState |= (1 << 1);
    if (PINB & (1 << PB4)) currentState |= (1 << 2);

    if (currentState != 0) {
        PORTB |= (1 << PB5);
    } else {
        PORTB &= ~(1 << PB5);
    }
}
```

```
#include <avr/io.h>

int main(void) {
   DDRB |= (1 << PB5);

   PORTB |= (1 << PB0) | (1 <<
   DDRB  &= ~((1 << PB0) | (1

   updateBuzzer();

   PCICR |= (1 << PCIE0);
   PCMSK0 |= (1 << PCINT0) |

   sei();

   while (1) {}
}


ISR(PCINT0_vect) {
   updateBuzzer();
}
```

Interrupt	Vector Name
External Interrupt Request 0	INT0_vect
External Interrupt Request 1	INT1_vect
Pin Change Interrupt Request 0	PCINT0_vect
Pin Change Interrupt Request 1	PCINT1_vect
Pin Change Interrupt Request 2	PCINT2_vect
Watchdog Time-out Interrupt	WDT_vect
Timer/Counter2 Compare Match A	TIMER2_COMPA_vect
Timer/Counter2 Compare Match B	TIMER2_COMPB_vect
Timer/Counter2 Overflow	TIMER2_OVF_vect
Timer/Counter1 Capture Event	TIMER1_CAPT_vect
Timer/Counter1 Compare Match A	TIMER1_COMPA_vect
Timer/Counter1 Compare Match B	TIMER1_COMPB_vect
Timer/Counter1 Overflow	TIMER1_OVF_vect
Timer/Counter0 Compare Match A	TIMER0_COMPA_vect
Timer/Counter0 Compare Match B	TIMER0_COMPB_vect
Timer/Counter0 Overflow	TIMER0_OVF_vect
SPI Serial Transfer Complete	SPI_STC_vect
USART Rx Complete	USART_RX_vect
USART Data Register Empty	USART_UDRE_vect
USART Tx Complete	USART_TX_vect

Example 5



Bit	D7							D0	
PCICR:	-	-	-	-	-	PCIE2	PCIE1	PCIE0	0x68

PCIE0: Pin Change Interrupt Enable bit for PORTB
PCIE1: Pin Change Interrupt Enable bit for PORTC (0: disabled, 1: enabled)
PCIE2: Pin Change Interrupt Enable bit for PORTD (0: disabled, 1: enabled)

Bit	D7							D0	
PCMSK0: (for PORTB)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	0x6B
PCMSK1: (for PORTC)	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	0x6C
PCMSK2: (for PORTD)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	0x6D

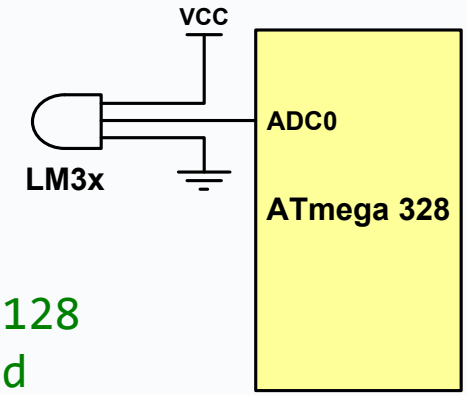
Example 6

برنامه‌ای به زبان C برای میکروکنترلر ATMega۳۲۸P بنویسید که با استفاده از مبدل آنالوگ به دیجیتال، دمای محیط را از سنسور LM۳۵ که خروجی آن به کانال صفر (ADC۰) متصل است، خوانده و مقدار دما را بر حسب درجه سانتی‌گراد به صورت یک عدد صحیح روی تمام پایه‌های PORTD نمایش دهد. فرض کنید:

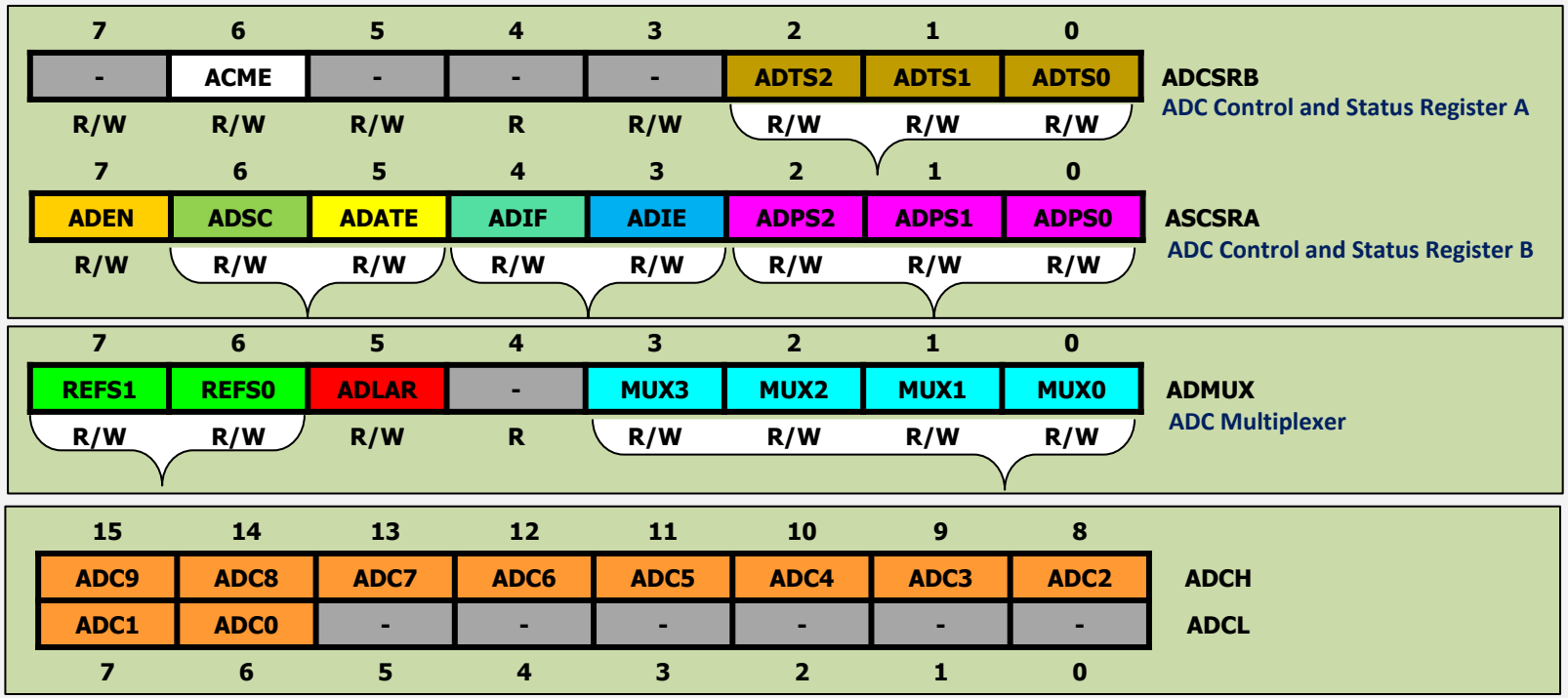
- فرکانس کریستال میکروکنترلر ۱۶ مگاهرتز است.
- از رفرنس داخلی ۱.۱ ولت برای ADC استفاده می‌کند.
- مقدار Prescaler را چنان انتخاب کنید که فرکانس کلاک ADC در محدوده ۵۰ تا ۲۰۰ کیلوهرتز قرار گیرد.
- ضریب تبدیل سنسور LM۳۵ برابر ۱۰ میلی‌ولت بر درجه سانتی‌گراد است.

Example 6

```
#include <avr/io.h> //standard AVR header
int main (void)
{
 DDRD = 0xFF;
 DDRC &= ~(1<<0);
 ADCSRA = 0x87;//make ADC enable and select ck/128
 ADMUX = 0xC0;//1.1V Vref, ADC0, right-justified

 while (1){
  ADCSRA |= (1<<ADSC);//start conversion
  while((ADCSRA&(1<<ADIF))==0); //wait for end of conversion
  ADCSRA |= (1<<ADIF); //clear the ADIF flag
  PORTB = (ADCL|(ADCH<<8))*10/93;//PORTB = adc value/9.3
 }
}
```



ADC Registers



ADC Tables

Table 24-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Table 24-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Table 24-4. Input Channel Selections

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Table 24-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Thanks!

Do you have any questions?

razeghizade@gmail.com

www.pudica.ir

CREADITS: This presentation was created by M.Razeghizadeh
Please keep this slide for attribution