

# Timer/counter

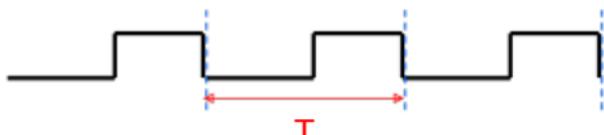
The AVR microcontroller  
and embedded  
systems  
using assembly and c



# Wave characteristics

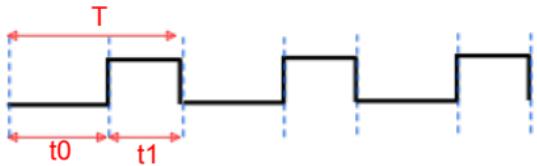
- Period
- Frequency

$$f = \frac{1}{T}$$



- Duty cycle

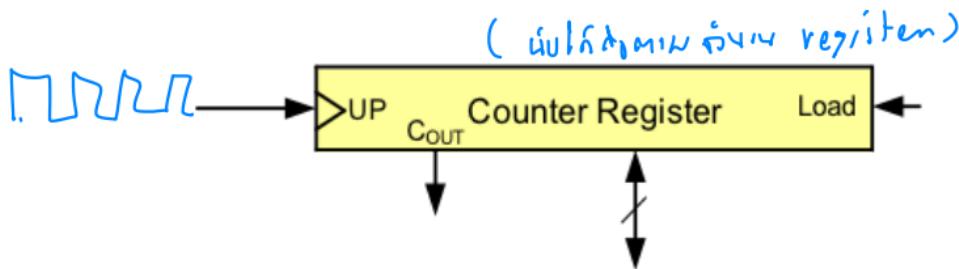
$$\text{duty cycle} = \frac{t_1}{T} \times 100 = \frac{t_1}{t_0 + t_1} \times 100$$



- Amplitude

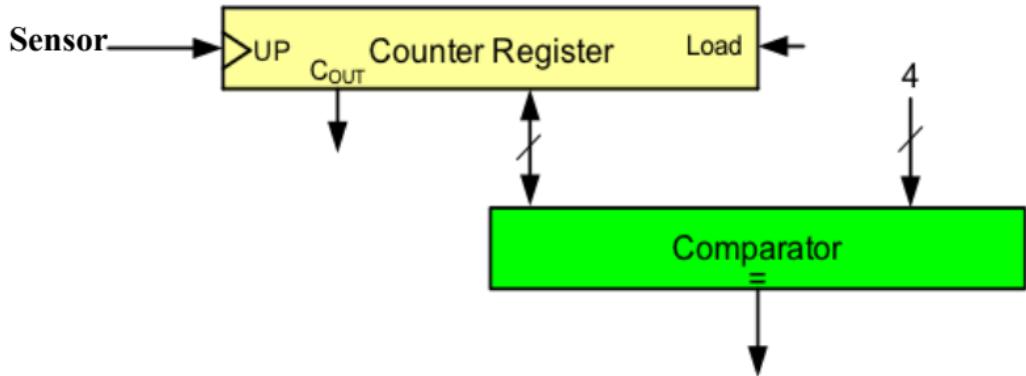


# A counter register



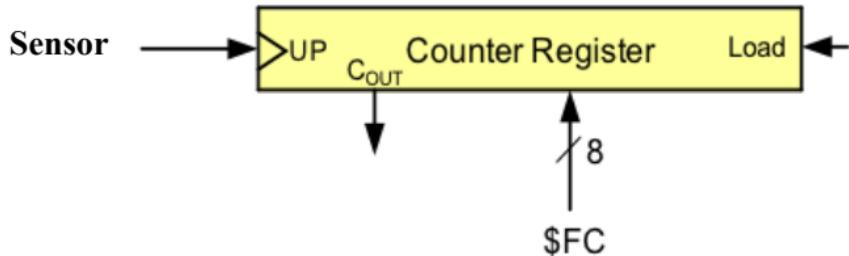
# A simple design (counting people)

## First design

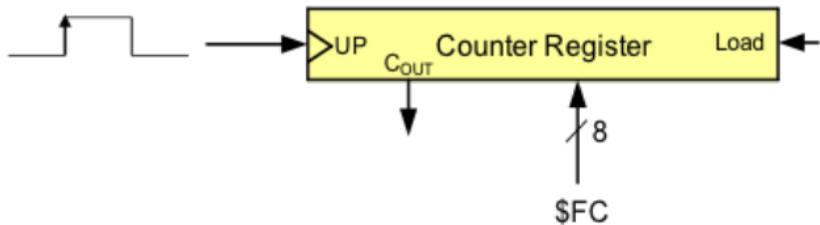


# A simple design (counting people)

## Second design

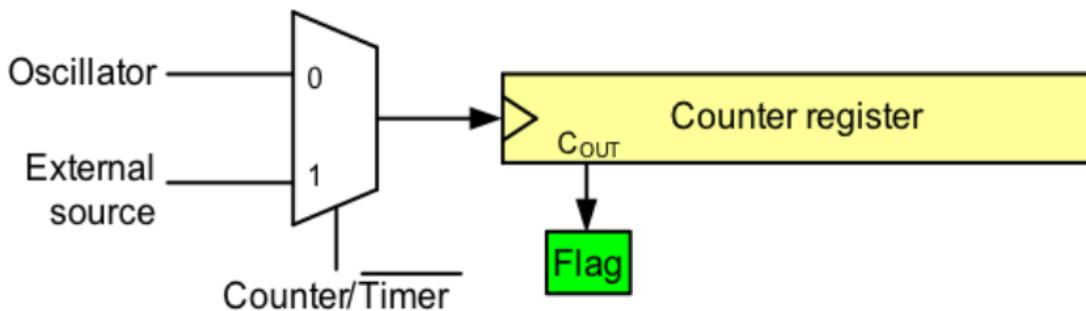


# A simple design (making delay)



# A generic timer/counter

- Delay generating
- Counting
- Wave-form generating
- Capturing



# Timers in AVR

- 1 to 6 timers
- 3 timers in ATmega32
- 8-bit and 16-bit timers
- two 8-bit timers and one 16-bit timer in ATmega32

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

© 2018 Microchip Technology Inc.

Data Sheet Complete

DS40002061A-page 1

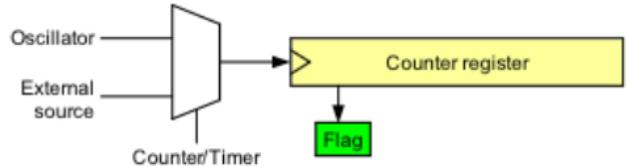
## **ATmega48A/PA/88A/PA/168A/PA/328/P**

- Real Time Counter with Separate Oscillator
- Six PWM Channels

Source: ATmega48/88/168/328P datasheet

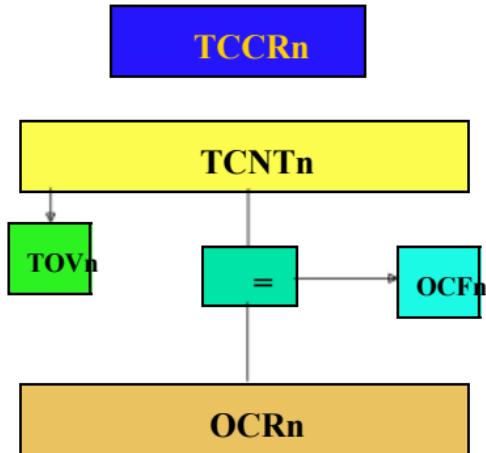
# Timer in AVR

- **TCNT<sub>n</sub>** (Timer/Counter register)
- **TOV<sub>n</sub>** (Timer Overflow flag)
- **TCCR<sub>n</sub>** (Timer Counter control register)
- **OCR<sub>n</sub>** (output compare register)
- **OCF<sub>n</sub>** (output compare match flag)



Comment:

All of the timer registers are byte-addressable I/O registers

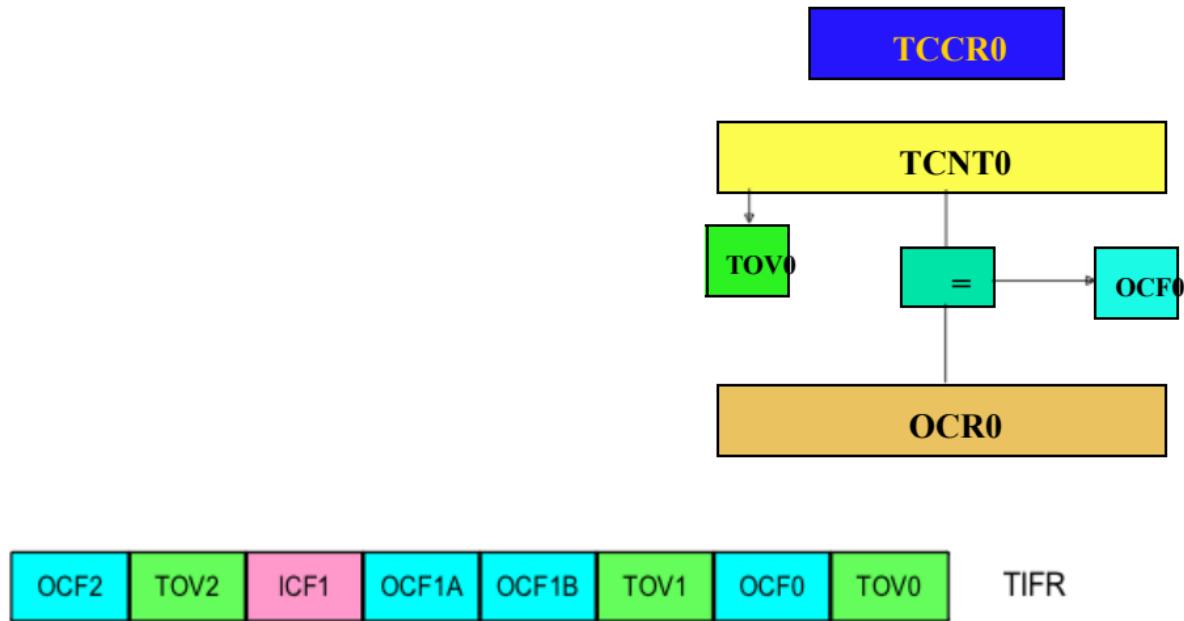


# Timer 0 (an 8-bit timer)

The AVR microcontroller  
and embedded  
systems  
using assembly and C



# Timer 0

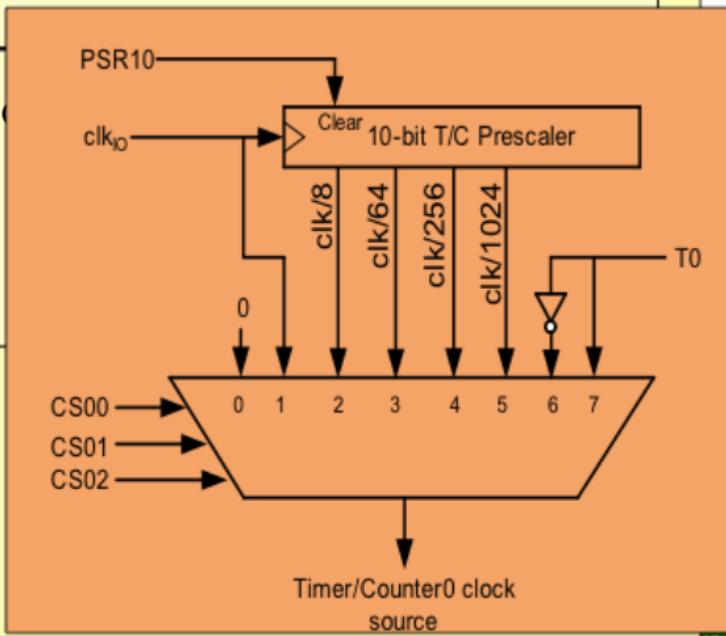




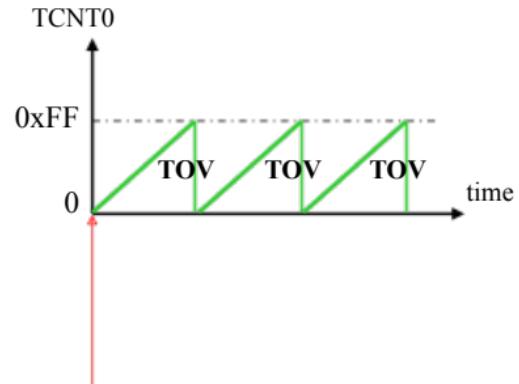
### WGM00 WGM01 | Comment

0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

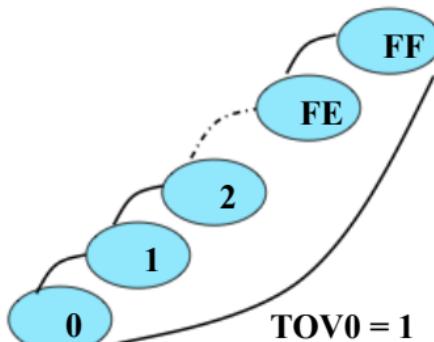
CS02	CS01	CS00	Comment
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk (No Prescaling)
0	1	0	clk / 8
0	1	1	clk / 64
1	0	0	clk / 256
1	0	1	clk / 1024
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge



# Normal mode



**TOV0:** 1



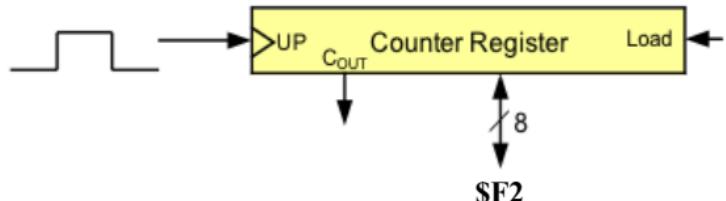
# Example 1: Write a program that waits 14 machine cycles in Normal mode.

**14 = \$0E**

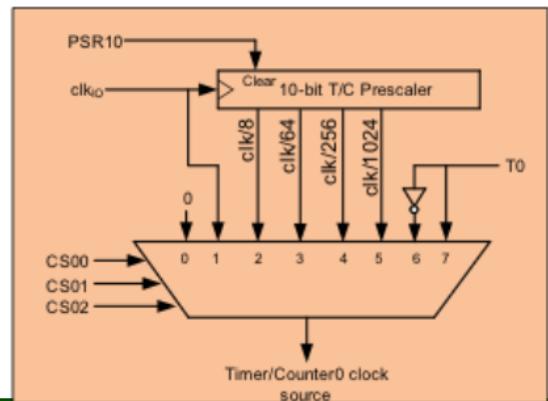
\$100

-\$0E

**\$F2**



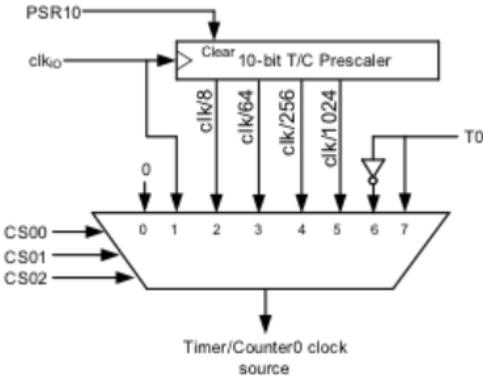
WGM00 WGM01		Comment
0	0	Normal
0	1	CTC
1	0	PWM, phase correct
1	1	Fast PWM



# Generating Large Delays

- Using loop
- Prescaler
- Bigger counters

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode



© 2018 Microchip Technology Inc.

Data Sheet Complete

DS40002061A-page 1

## ATmega48A/PA/88A/PA/168A/PA/328/P

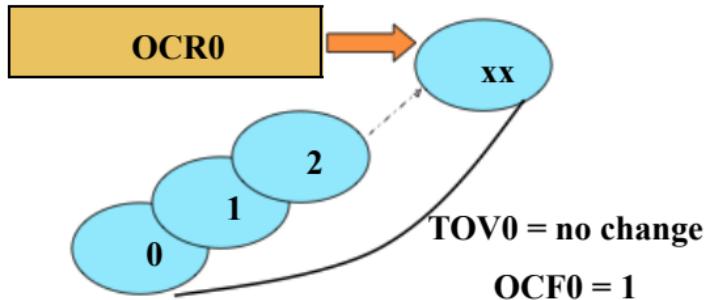
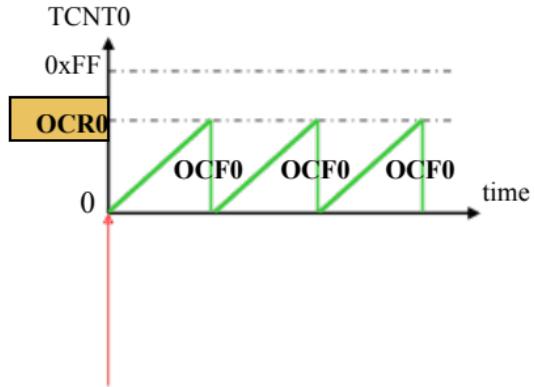
- Real Time Counter with Separate Oscillator
- Six PWM Channels

Source: ATmega48/88/168/328P datasheet

# DEMO #1

Toggle GPIO Pin with Timer Overflow Interrupt  
(Normal Mode)

# CTC (Clear Timer on Compare match) mode



TOV0: 0

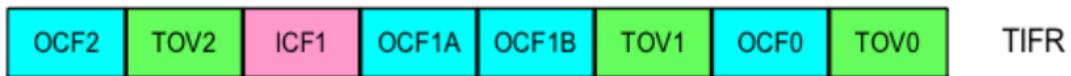
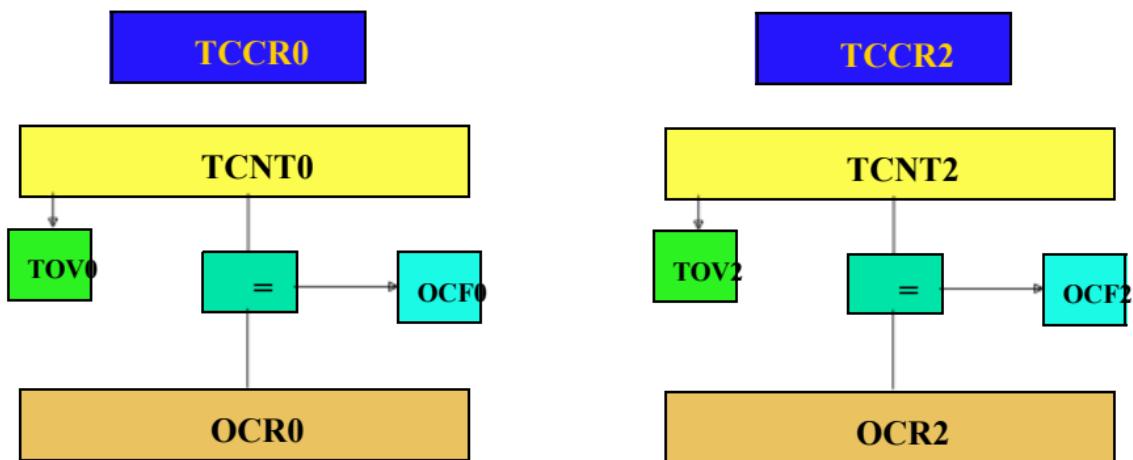
OCF0: 1

# DEMO #2

Toggle GPIO Pin with Compare Match Interrupt  
(CTC mode)

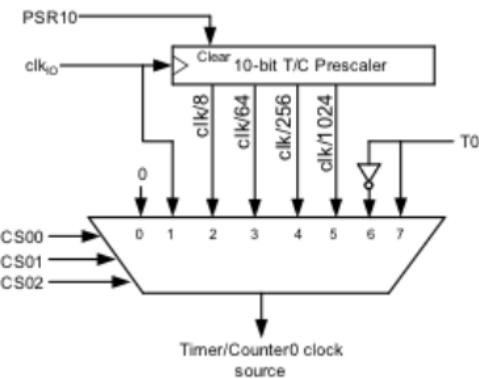
# Timer2

- Timer0
- Timer2

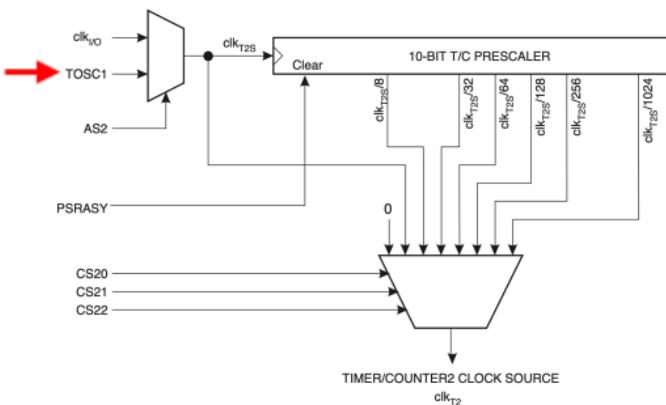


# The difference between Timer0 and Timer2

- Timer0



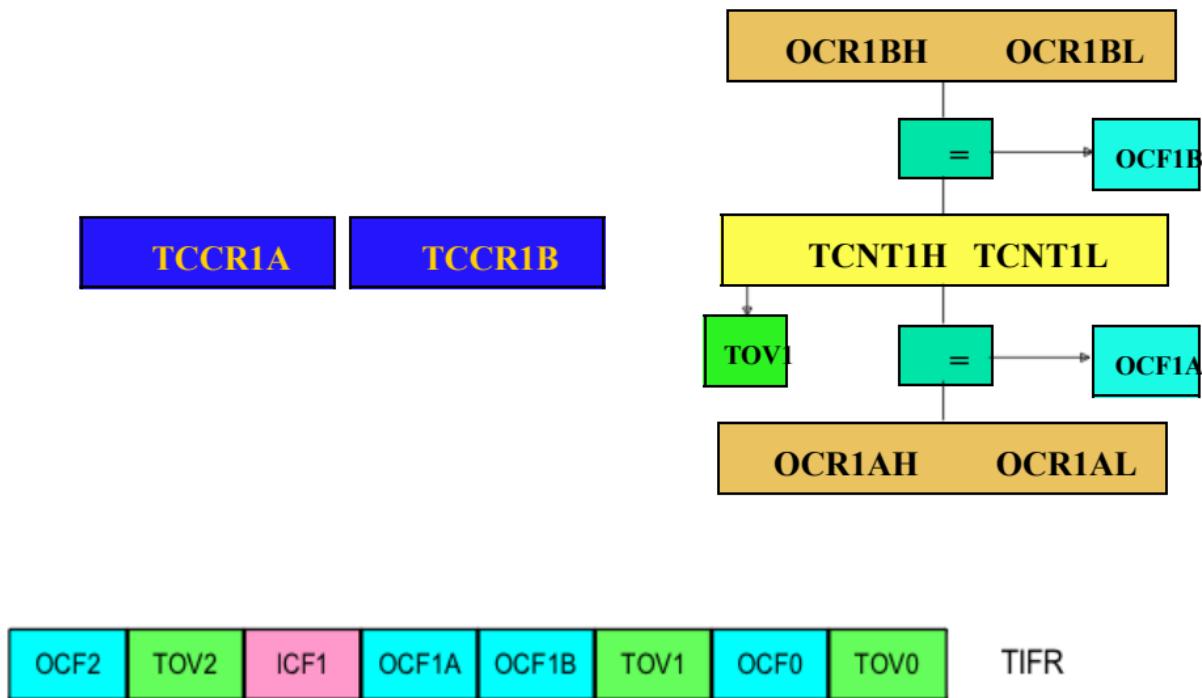
- Timer2

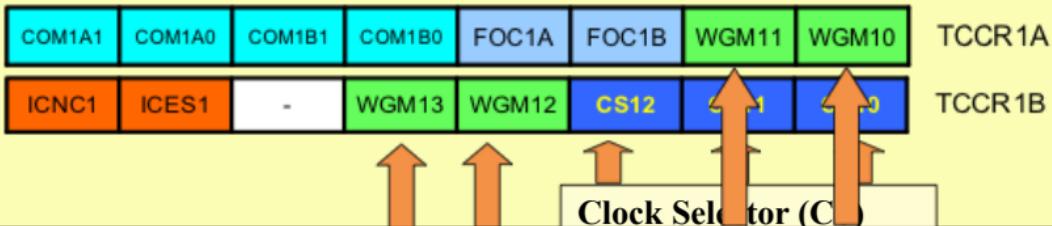


CS02	CS01	CS00	Comment
0	0	0	Timer/Counter stopped
0	0	1	clk (No Prescaling)
0	1	0	clk / 8
0	1	1	clk / 64
1	0	0	clk / 256
1	0	1	clk / 1024
1	1	0	External clock (falling edge)
1	1	1	External clock (rising edge)

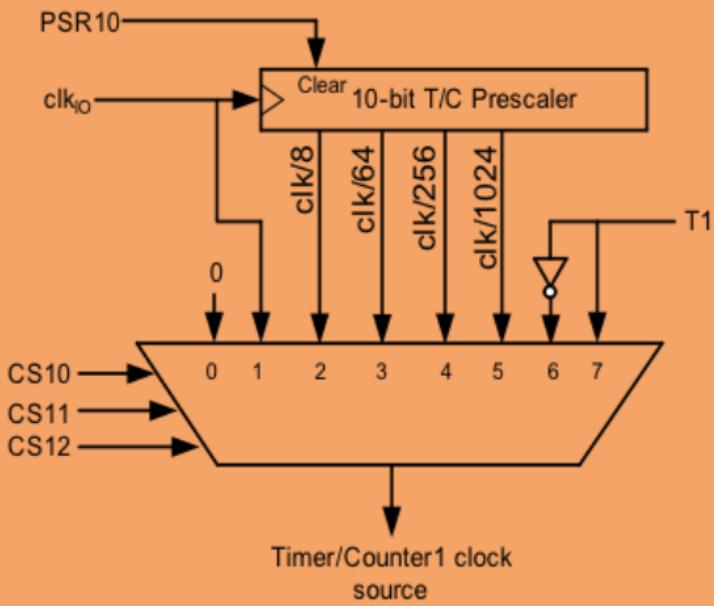
CS22	CS21	CS20	Comment
0	0	0	Timer/Counter stopped
0	0	1	clk (No Prescaling)
0	1	0	clk / 8
0	1	1	clk / 32
1	0	0	clk / 64
1	0	1	clk / 128
1	1	0	clk / 256
1	1	1	clk / 1024

# Timer 1





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					
1	0	0	0					
2	0	0	1					
3	0	0	1					
4	0	1	0					
5	0	1	0					
6	0	1	1					
7	0	1	1					
8	1	0	0					
9	1	0	0					
10	1	0	1					
11	1	0	1					
12	1	1	0					
13	1	1	0					
14	1	1	1					
15	1	1	1					



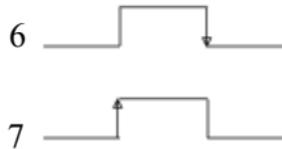
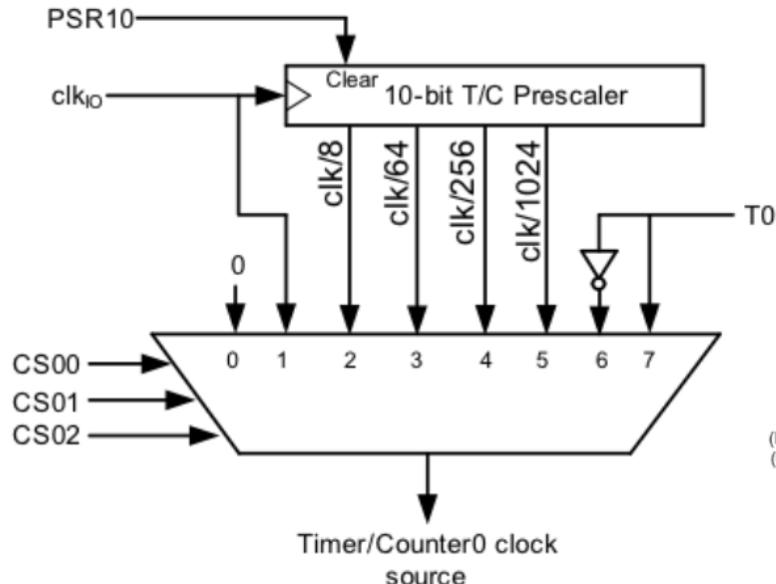
# Counting

The AVR microcontroller  
and embedded  
systems  
using assembly and C



MUHAMMAD ALI MAZIDI  
SARMAD NAIMI  
SEPEHR NAIMI

# Counting



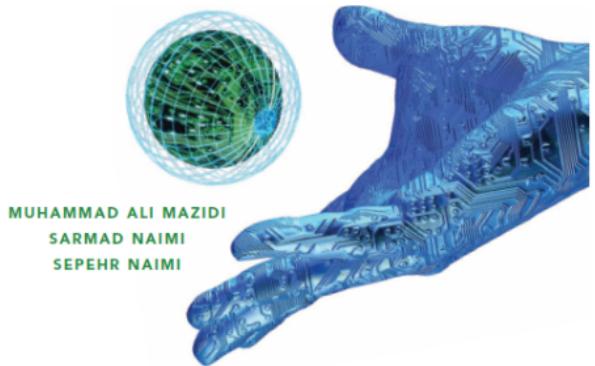
(XCK) <b>T0</b>	PB0	1	40	PA0 (ADC0)
(T1)	PB1	2	39	PA1 (ADC1)
(INT2/AIN0)	PB2	3	38	PA2 (ADC2)
(OC0/AIN1)	PB3	4	37	PA3 (ADC3)
(SS)	PB4	5	36	PA4 (ADC4)
(MOSI)	PB5	6	35	PA5 (ADC5)
(MISO)	PB6	7	34	PA6 (ADC6)
(SCK)	PB7	8	33	PA7 (ADC7)
RESET		9	32	AREF
VCC		10	31	GND
GND		11	30	AVCC
XTAL2		12	29	PC7 (TOSC2)
XTAL1		13	28	PC6 (TOSC1)
(RXD)	PD0	14	27	PC5 (TDI)
(TXD)	PD1	15	26	PC4 (TDO)
(INT0)	PD2	16	25	PC3 (TMS)
(INT1)	PD3	17	24	PC2 (TCK)
(OC1B)	PD4	18	23	PC1 (SDA)
(OC1A)	PD5	19	22	PC0 (SCL)
(ICP)	PD6	20	21	PD7 (OC2)

# DEMO #3

Count the number of times a button is pressed and display on LEDs with Timer 1

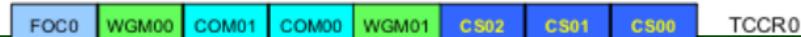
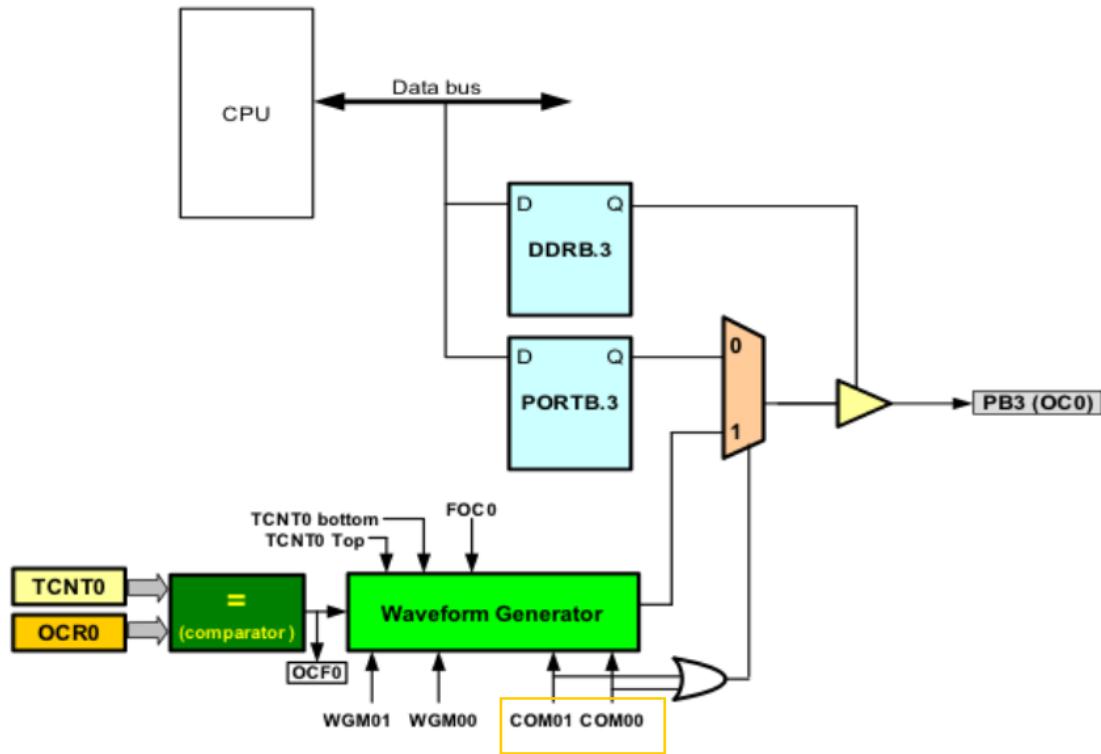
# Wave generating and Capturing

The AVR microcontroller  
and embedded  
systems  
using assembly and c

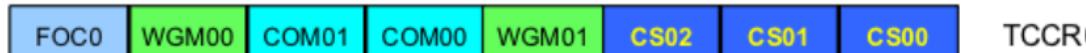
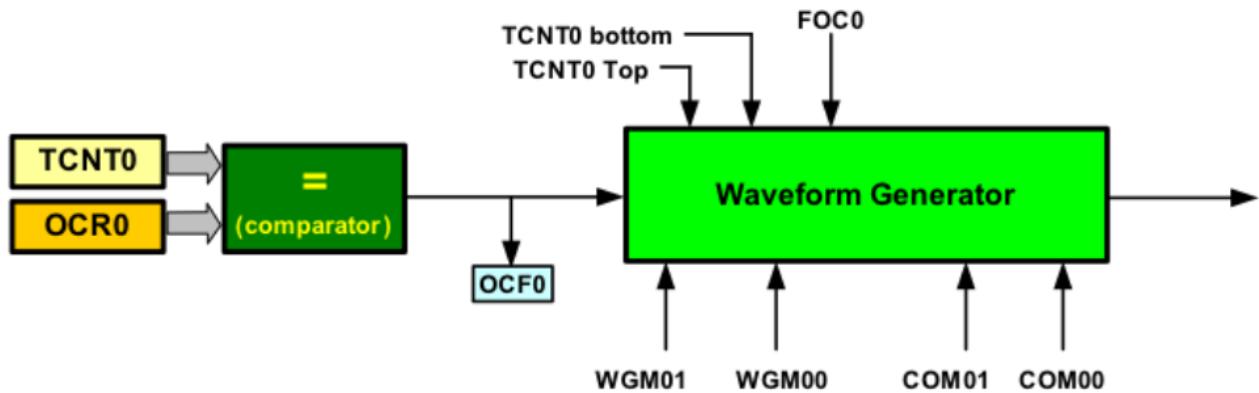


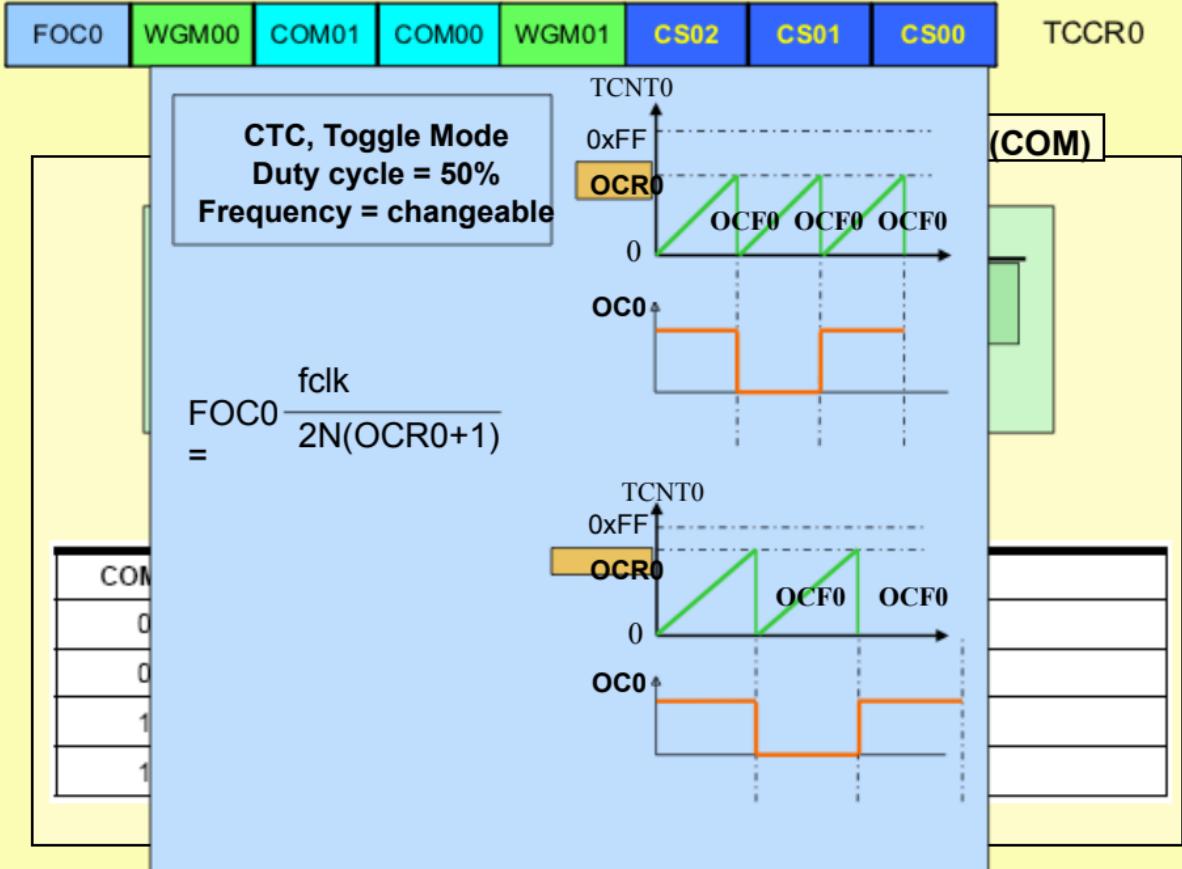
MUHAMMAD ALI MAZIDI  
SARMAD NAIMI  
SEPEHR NAIMI

# Waveform Generator



# Waveform Generator





Assuming XTAL = 8 MHz, make a pulse with duty cycle = 50% and frequency = 500KHz

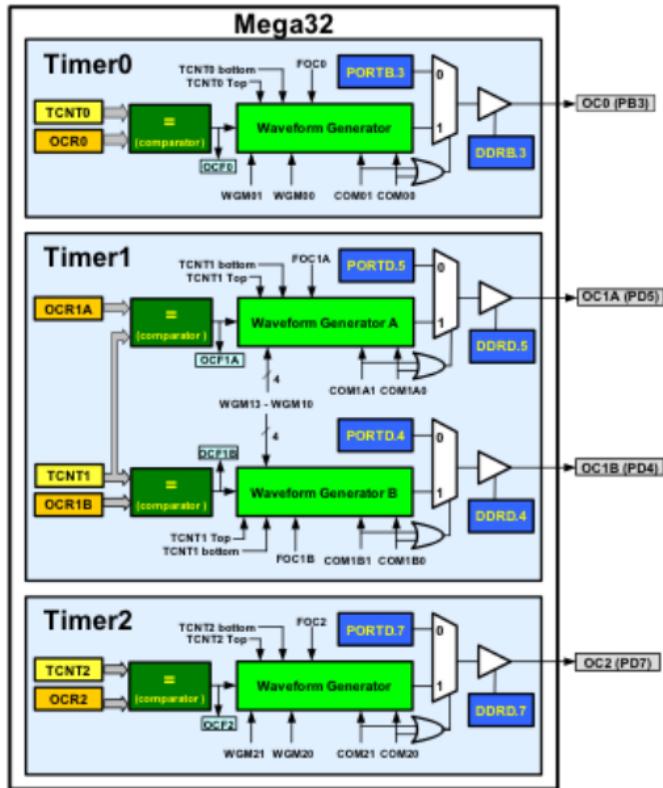
$$FOC0 = \frac{fclk}{2N(OCR0+1)} \quad \boxed{500\text{KHz}} = \frac{8\text{MHz}}{2N(OCR0+1)} \quad \boxed{N(OCR0+1)} = \frac{8\text{MHz}}{1\text{MHz}}$$

$\boxed{N(OCR0+1) = 8}$     $\boxed{\begin{array}{|c|} \hline N = 1 \text{ and } OCR0 = 7 \\ \hline \end{array}}$

$\boxed{\begin{array}{|c|} \hline N = 8 \text{ and } OCR0 = 0 \\ \hline \end{array}}$

<pre> LDI R20,7 OUT OCR0,R20 LDI R20,0x19 OUT TCCR0,R20 </pre>	<pre> OCR0 = 7; TCCR0 = 0x19; //prescaler = 1 </pre>
<pre> LDI R20,0 OUT OCR0,R20 LDI R20,0x1A OUT TCCR0,R20 </pre>	<pre> OCR0 = 0; TCCR0 = 0x1A; //prescaler = 8 </pre>

# Waveform generators in ATmega32



# DEMO #4

Toggle GPIO Pin with Timer (Waveform Generator)

# Capturing in Timer/counter 1

The AVR microcontroller  
and embedded  
systems  
using assembly and C



# Capturing

- Usages
- Measuring duty cycle
- Measuring period

## ■ Period

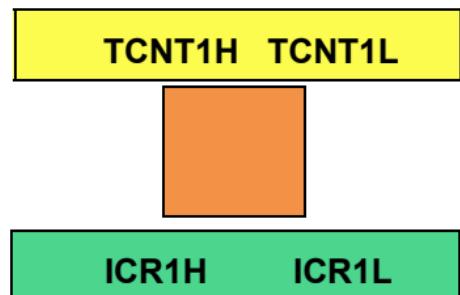
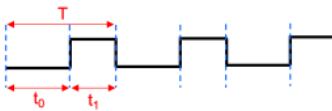
### ■ Frequency

$$f = \frac{1}{T}$$

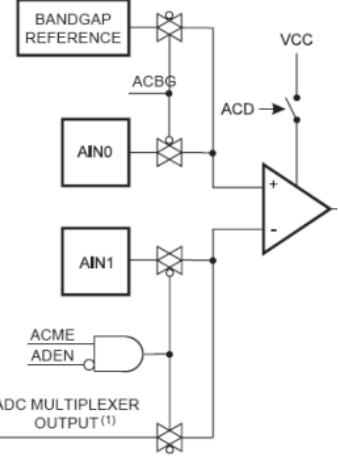
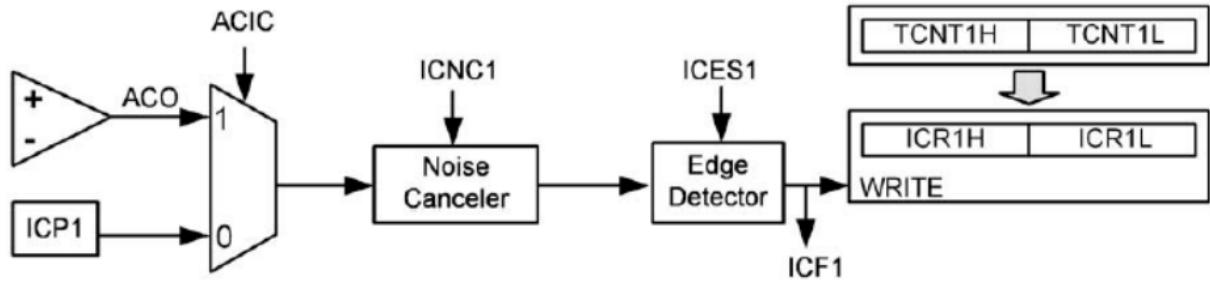


## ■ Duty cycle

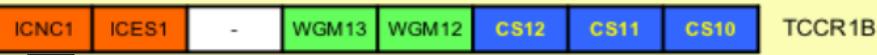
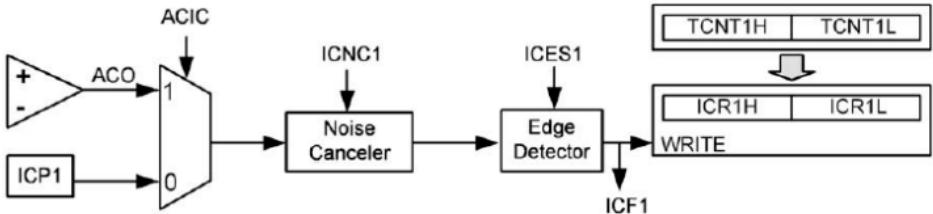
$$\text{duty cycle} = \frac{t_0}{T} \times 100 = \frac{t_0}{t_0 + t_1} \times 100$$



# Capturing & Comparator



(XCK/T0)	PB0	1	40	PA0 (ADC0)
(T1)	PB1	2	39	PA1 (ADC1)
(INT2/AIN0)	PB2	3	38	PA2 (ADC2)
(OC0/AIN1)	PB3	4	37	PA3 (ADC3)
(SS)	PB4	5	36	PA4 (ADC4)
(MOSI)	PB5	6	35	PA5 (ADC5)
(MISO)	PB6	7	34	PA6 (ADC6)
(SCK)	PB7	8	33	PA7 (ADC7)
RESET		9	32	AREF
VCC		10	31	GND
GND		11	30	AVCC
XTAL2		12	29	PC7 (TOSC2)
XTAL1		13	28	PC6 (TOSC1)
(RXD)	PD0	14	27	PC5 (TDI)
(TXD)	PD1	15	26	PC4 (TDO)
(INT0)	PD2	16	25	PC3 (TMS)
(INT1)	PD3	17	24	PC2 (TCK)
(OC1B)	PD4	18	23	PC1 (SDA)
(OC1A)	PD5	19	22	PC0 (SCL)
(ICP)	PD6	20	21	PD7 (OC2)



### ICNC1: Input Capture Noise Canceller

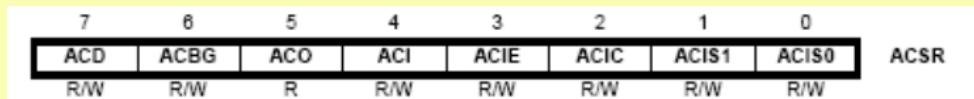
0:disabled

1:Enabled (captures after 4 successive equal valued samples)

### ICES1: Input Capture Edge Select

0: Falling edge

1: Rising edge

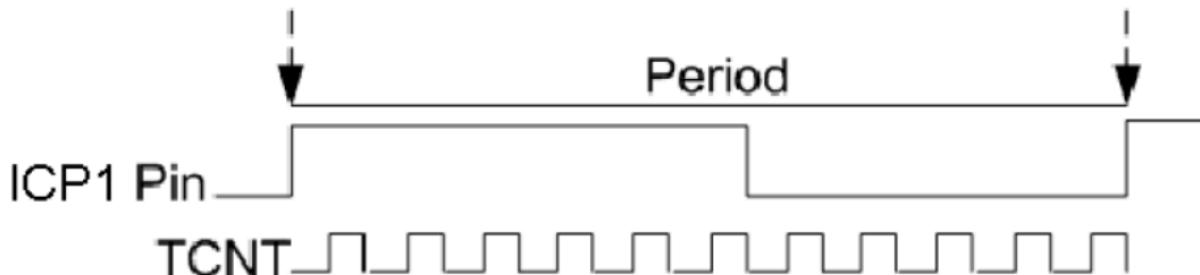


### ACIC: Analog Comparator Input Capture Enable

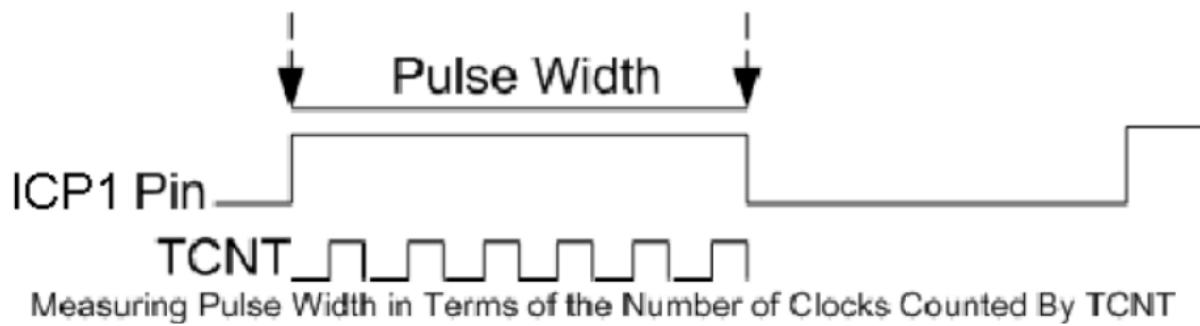
0: ICP1 provides the capture signal

1: analog comparator is connected to the capturer

# Measuring duty cycle and period



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



# DEMO #5

Measure the duration of button press (Input Capture Pin)