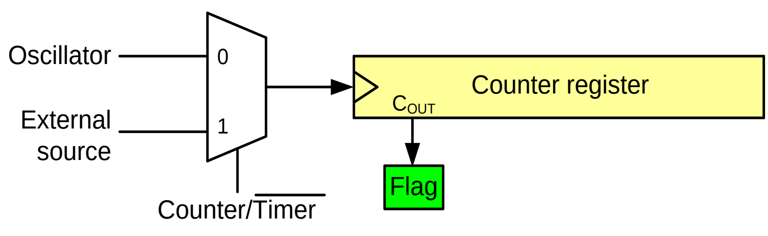
Mechatronics Engineering

Timers

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Materials Science
German University in Cairo

A generic timer/counter

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

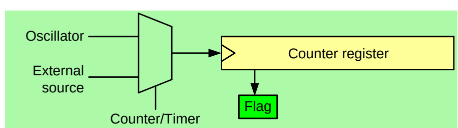


Timers in AVR

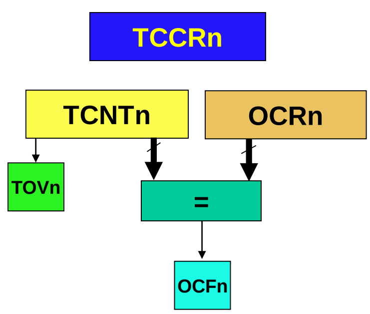
- 1 to 6 timers
  - 3 timers in ATmega328
- 8-bit and 16-bit timers
  - two 8-bit timers and one 16-bit timer in ATmega328

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Timer in AVR



- TCNTn (Timer/Counter register)
- TOVn (Timer Overflow flag)
- TCCRn (Timer Counter control register)
- OCRn (output compare register)
- OCFn (output compare match flag)

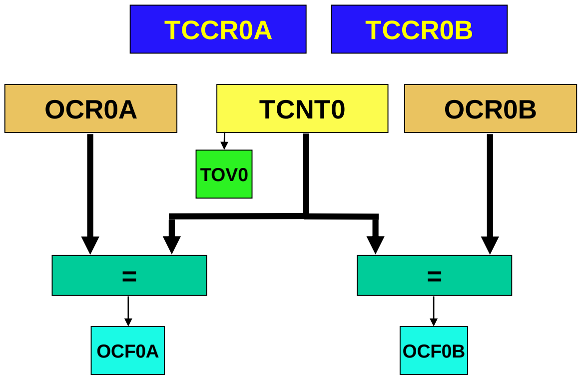


Comment:

All of the timer registers are byte-addressable registers

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Timer 0



-	-	-	-	-	OCF0B	OCF0A	TOV0	TIFR0

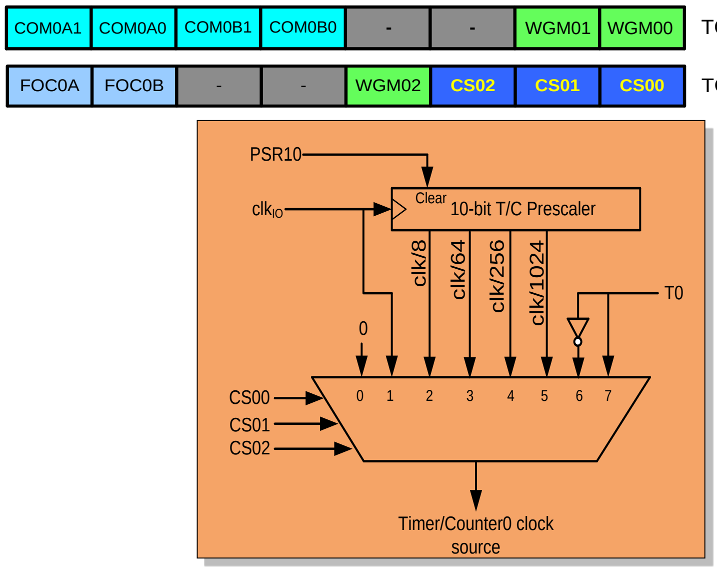
GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	TCCR0A
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B

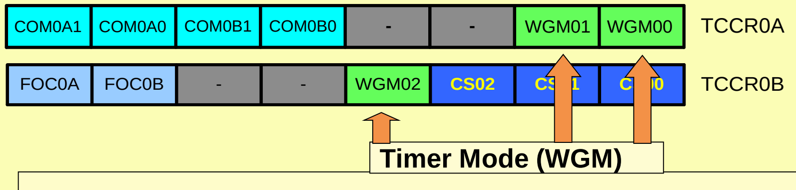
Clock Selector (CS)

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

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science



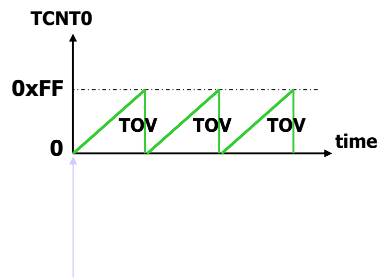
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

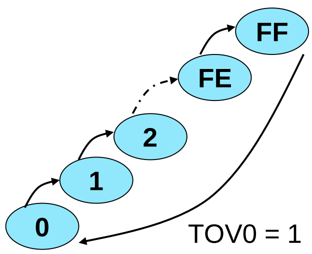


WGM02	WGM01	WGM00	Comment
0	0	0	Normal
0	0	1	Phase correct PWM
0	1	0	CTC (Clear Timer on Compare Match)
0	1	1	Fast PWM
1	0	0	Reserved
1	0	1	Phase correct PWM
1	1	0	Reserved
1	1	1	Fast PWM

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Normal mode



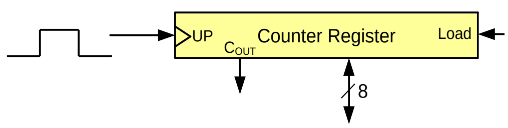


TOV0: **1**

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

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

```
14 = $0E
  $100
 -$0E
 -----
  $F2
```



$F2

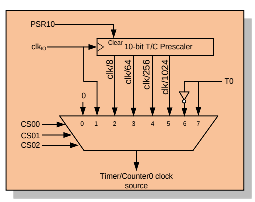
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	TCCR0A
						0	0	

FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B
				0	0	0	1	

TCCR0A = 0x00;
TCCR0B = 0x01;

Timer Mode (WGM)

WGM02	WGM01	WGM00	Comment
0	0	0	Normal
0	0	1	Phase correct PWM
0	1	0	CTC
0	1	1	Fast PWM
1	0	0	Reserved
1	0	1	Phase correct PWM
1	1	0	Reserved
1	1	1	Fast PWM



Department of Mechatronics Engineering
Faculty of Engineering and Material Science

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

```
  $100
 -$0E
  $F2
```

-	-	-	-	-	OCF0B	OCF0A	TOV0	TIFR0

```
DDRB = 1<<5;
PORTB &= ~(1<<5);   //PB5=0
while (1)
{
  TCNT0 = 0xF2;
  TCCR0A = 0x00;
  TCCR0B = 0x01;
  while((TIFR0&(1<<TOV0))==0)
  {}
  TCCR0B = 0;
  TIFR0 = (1<<TOV0);
  PORTB ^= (1<<5);
}
```

Question: How to calculate the delay generated by the timer?

Answer:
1) Calculate how much a machine clock lasts.
   T= 1/f
2) Calculate how many machine clocks it waits.
3) Delay = T * number of machine cycles

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

In example 1 calculate the delay.
XTAL = 10 MHz.

Solution 1 (inaccurate):

1) Calculating T:

T = 1/f = 1/10M = 0.1µs

2) Calculating num of machine cycles:

```
  $100
 -$F2
  $0E = 14
```

3) Calculating delay

14 * 0.1µs = 1.4 0µs

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Finding values to be loaded into the timer

1. Calculate the period of clock source.
   - Period = 1 / Frequency
     - E.g. For XTAL = 16 MHz ➔ T = 1/16MHz
2. Divide the desired time delay by period of clock.
3. Perform 256 - n, where n is the decimal value we got in Step 2.
4. Set TCNT0 = 256 - n

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Example 2: Assuming that XTAL = 10 MHz, write a program to generate a square wave with a period of 10 ms on pin PORTB.3.

- For a square wave with T = 10 µs we must have a time delay of 5 µs. Because XTAL = 10 MHz, the counter counts up every 0.1 µs. This means that we need 5 µs / 0.1 µs = 50 clocks. 256 - 50 = 206.

```
DDRB = 1<<3;
PORTB &= ~ (1<<3);
while (1)
{
  TCNT0 = 206;
  TCCR0A = 0x00;
  TCCR0B = 0x01;
  while((TIFR0&0x01) == 0)
  {}
  TCCR0B = 0;
  TIFR0 = 1<<TOV0;
  PORTB = PORTB ^ (1<<3);
}
```

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Example 3: Modify TCNT0 in Example 2 to get the largest time delay possible with no prescaler. Find the delay in µs. In your calculation, do not include the overhead due to instructions.

- To get the largest delay we make TCNT0 zero. This will count up from 00 to 0xFF and then roll over to zero.

```
DDRB = 1 << 3;
PORTB &= ~(1<<3);
while (1)
{
  TCNT0 = 0x00;
  TCCR0A = 0x00;
  TCCR0B = 0x01;

while((TIFR0&(1<<TOV0))==0)
{}
  TCCR0B = 0;
  TIFR0 = 0x01;
  PORTB = PORTB^(1<<3);
}
```

Solution

1) Calculating T:

T = 1/f = 1/10MHz = 0.1µs

2) Calculating delay

256 * 0.1µs = 25.6µs

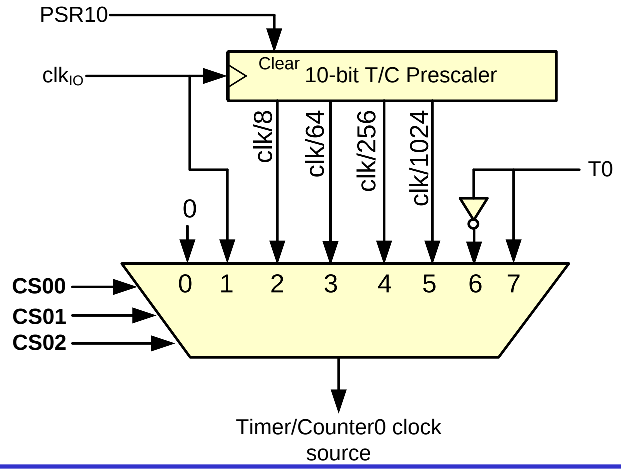
GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Generating Large Delays

- Using loop
- Prescaler
- Bigger counters

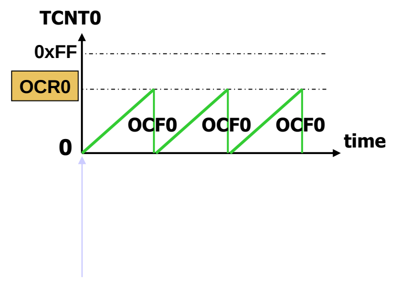
GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

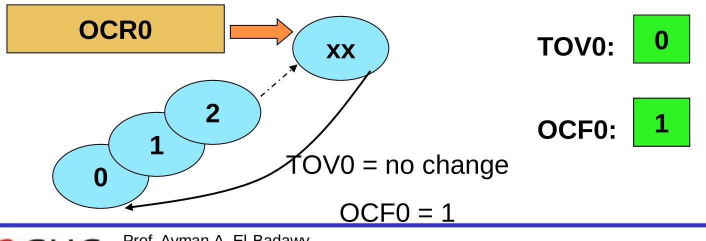
Prescaler and generating a large time delay



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

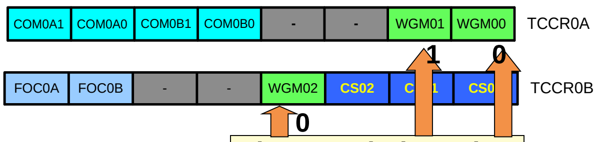
CTC (Clear Timer on Compare match) mode





Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

CTC mode (Cont.)



Timer Mode (WGM)

WGM02	WGM01	WGM00	Comment
0	0	0	Normal
0	0	1	Phase correct PWM
0	1	0	CTC
0	1	1	Fast PWM
1	0	0	Reserved
1	0	1	Phase correct PWM
1	1	0	Reserved
1	1	1	Fast PWM

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Rewrite example 2 using CTC

(Assuming XTAL = 10MHz, write a program to generate a square wave with a period of 10 us.)

- For a square wave with T = 10 µs we must have a time delay of 5 µs. Because XTAL = 10 MHz, the counter counts up every 0.1 µs. This means that we need 5 µs / 0.1 µs = 50 clocks. Therefore, we have OCR0A= 49.

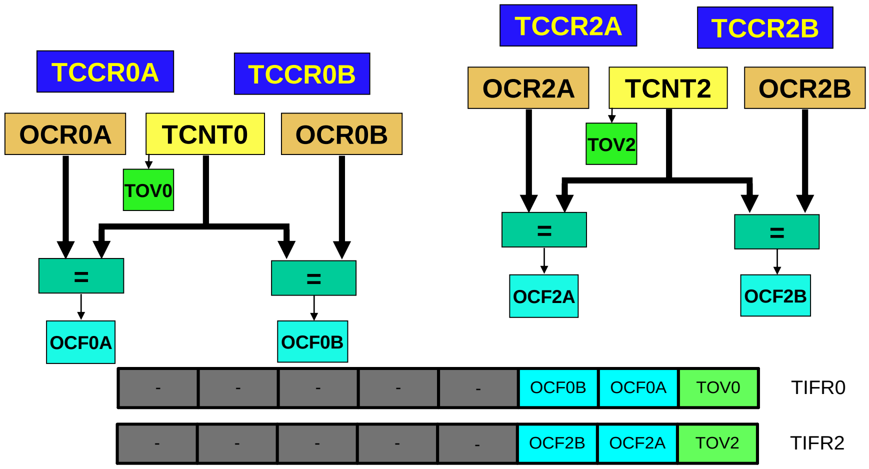
```
DDRB |= 1<<3;
while (1)
{
  OCR0A = 49;
  TCCR0A = (1<<WGM01); //CTC
  TCCR0B = 1;  //N = 1
 while((TIFR0&(1<<OCF0A))==0)
  { }
  TCCR0B = 0; //stop timer0
  TIFR0 = 1<<OCF0A;
  PORTB ^= 1<<3; //toggle PB3
}
```

-	-	-	-	-	OCF0B	OCF0A	TOV0	
								TIFR0

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Timer2 vs. Timer0

- Timer0
- Timer2



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

The difference between Timer0 and Timer2

- Timer0

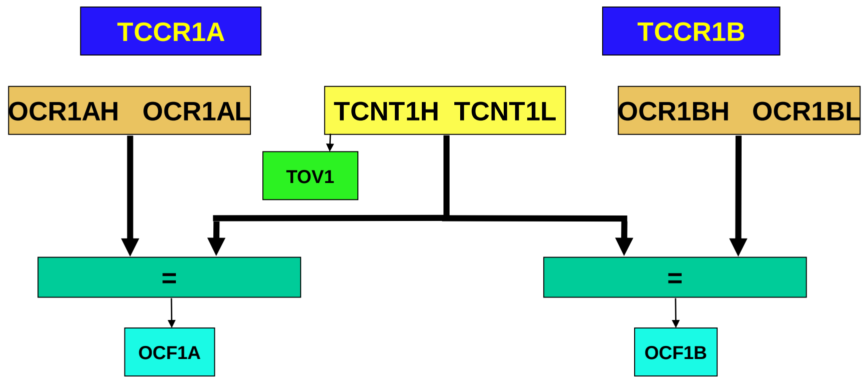
CS02	CS01	CS00	Comment
0	0	0	Timer/Counter stopped
0	0	1	clk (No Prescaling)
0	1	0	clk / 8
0	1	1	clk / 64
1	0	0	clk / 256
1	0	1	clk / 1024
1	1	0	External clock (falling edge)
1	1	1	External clock (rising edge)

- Timer2

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

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Timer 1



-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	TIFR1

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B

Clock Selector (CS)

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

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B

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

Assuming XTAL = 10 MHz write a program that toggles PB5 once per millisecond, using Normal mode.

XTAL = 10 MHz ➔ 1/10 MHz = 0.1 µs
Num. of machine cycles = 1 ms / 0.1 µs = 10,000
TCNT1 = 65,536 – 10,000 = 55,536 = $D8F0

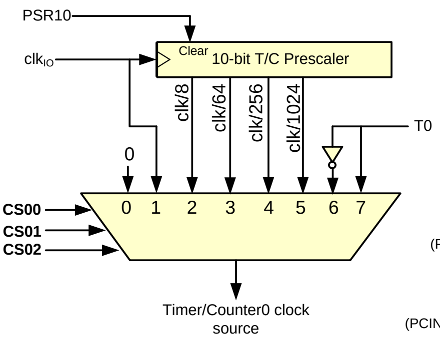
TCNT1H **TCNT1L**

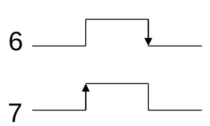
Assuming XTAL = 10 MHz, write a program that toggles PB5 once per millisecond, using CTC mode.

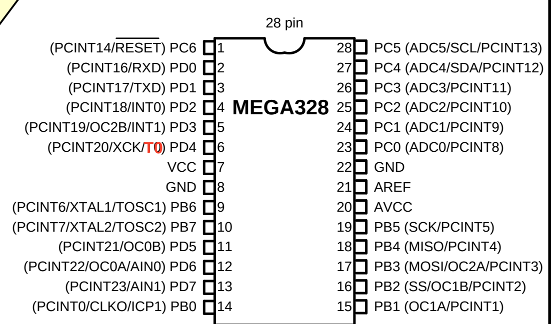
```c
#include <avr/io.h>
void  delay1ms();
int main(){
 DDRB  |= 1<<5;
 while (1) {
 delay1ms();
 PORTB ^= (1<<5); //toggle PB5
 }
}
void  delay1ms()
{
 TCNT1 = 0;
 OCR1A = 10000-1;
 TCCR1A = 0;    //WGM=0100 (CTC)
 TCCR1B = 0x09;  //N = 1
 while((TIFR1&(1<<OCF1A))==0)
 { } //wait until OCF1A is set
 TCCR1B = 0; //stop timer1
 TIFR1 = 1<<OCF1A;//clear flag
}
```

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Counting

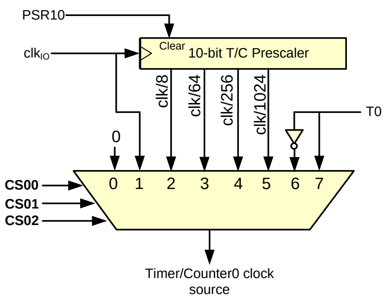






GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Assuming that clock pulses are fed into pin T0, write a program for counter 0 in normal mode to count the pulses on falling edge and display the state of the TCNT0 count on PORTC.




```c
#include <avr/io.h>
int main()
{
 DDRC = 0xFF;
 TCCR0A = 0;    //WGM=0000 (Normal)
 TCCR0B = 0x06;  //CS=6 (Count on fall)
 while (1)
 {
   PORTC = TCNT0; //read TCNT0
 }
}
```

COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	TCCR0A
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Assuming that clock pulses are fed into pin T1. Write a program for Counter1 in CTC mode to make PORTC.0 high every 100 pulses.

```c
#include <avr/io.h>
int main()
{
 DDRD &= ~(1<<5);
 DDRC |= 1<<0;
 TCCR1A = 0;     //WGM=0100 (CTC)
 TCCR1B = 0x0E;  //CS=6 (Count on fall)
 OCR1A = 99;
 while (1)
 {
  while((TIFR1&(1<<OCF1A)) == 0);
  TIFR1 = (1<<OCF1A);
  PORTC |= (1<<0);  //PC0 = 1
  PORTC &= ~(1<<0); //PC0 = 0
 }
}
```


Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science