timer / counter

* Notes → ① time unit in µc is clk
  ↳ clk is Responsibility for Instruction
→ ② system clock → Processor
  ↳ crystal → 8 MHz
  10 MHz
  ⋮
  16 MHz
→ ③ 1 clk ⟹ ?? sec

clk time = PresCaller / system Freq

ex1. system = 8 MHz

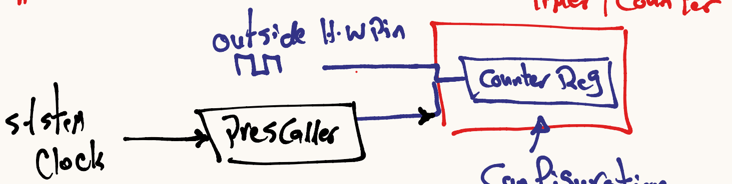
= 1 / 8000 000 = 0.125 µsec

timer / counter in Atmega32

↳ 3 separated timer → 8 bits timer0
→ 16 - bits timer1
→ 8 bit timer2

↳ timer | Count
Count No of clk
* Perodic
* system Clock

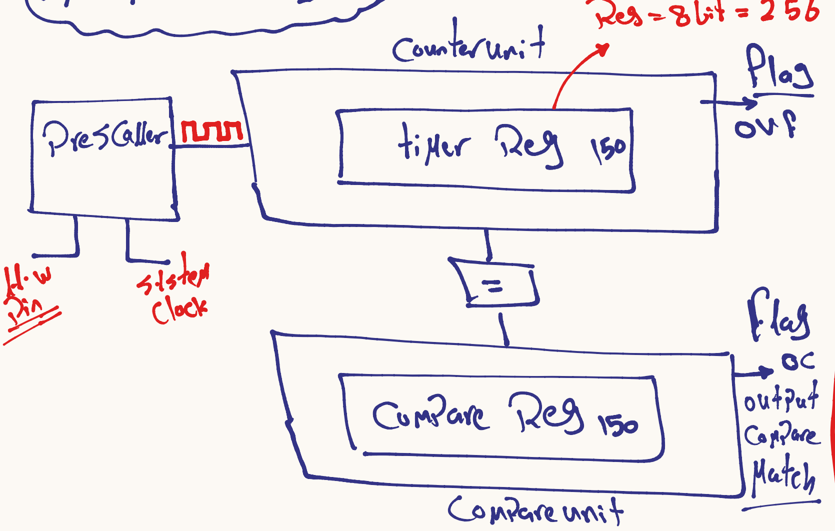
↳ Count Pulse
↳ Not Perodic
↳ out Pulse



timer mode
↳ Calc → clk time = PresCaller / systemclk

Counter mode
↳ Count the rising / Falling

timer / Counter 8 bit [timer0]

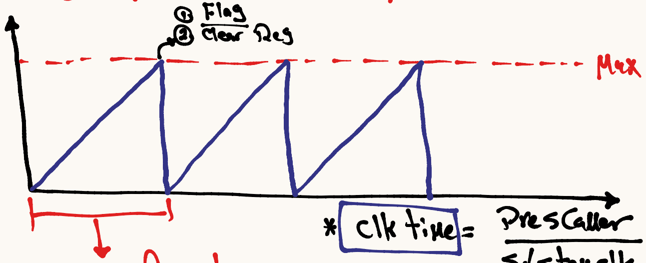


Definition

↳ timer / Counter have H.w limit size
↳ timer Mode
  ↳ ① Normal Mode [over Flow Mode]
  ↳ ② Compare Match Mode
    ↳ clear Compare Match
    ↳ Pulse width Modulation [PWM]
  ↳ ③ watch dog → timer specific

* Normal Mode

↳ Count For Clock → Max size



* clk time = PresCaller / system clk

* over Flow time = → clk time * over Flow value
→ clk time * 2^n
↳ n → # No of bit

[* Real time and over Flow time Relation]

① Real time == over Flow time → Not Common
② Real time > over Flow time
③ Real time < over Flow time

Real time == over Flow

↳ timer Init()
  ↳ over Flow Mode → ①

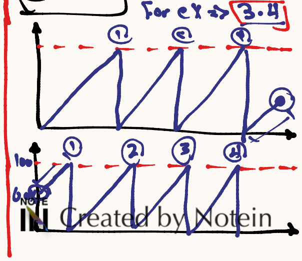
↳ ISR_OVF() ←
{
  Action ←
}

[Real > over Flow]

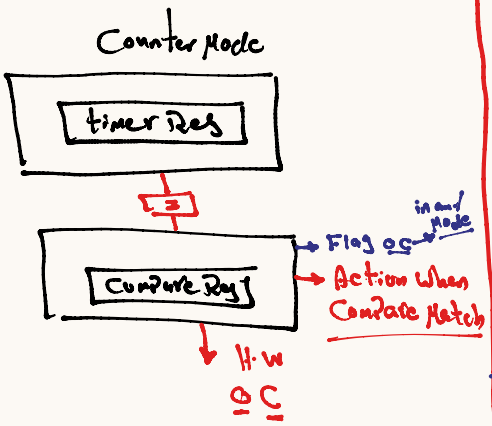
→ # No of over Flow Count = Real time / over Flow time

[Case1] → No of Count => int
* time_Init()
  ↳ Normal (ovf)
* ISR()
{
static u32 Count = 0;
Count ++;
if (Count == No of ovf)
{
Action
}

[Case 2] Count => Float
For ex => 3.4



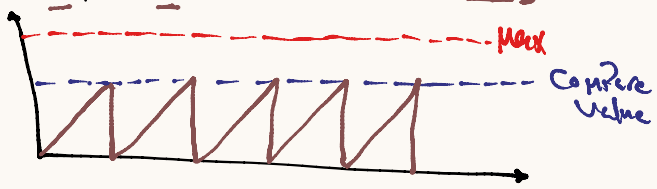
Compare Match



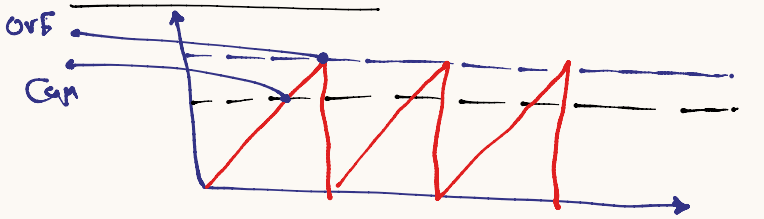
* ① Action when Compare Match
* ② Action on output Compare Match

* Action on Compare Match

① clear time Compare Match [CTC]



② Pulse width Modulation



* Action on H.w Pin

CTC Mode
- Disable
- clear
- set
- toggel

PWM
- Inverter Mode
- Non-Inverter Mode

---

① CTC_Mode

① $Clk\ time = \frac{Prescaller}{System\ Freq}$

② Compare time ⇒ Clk time * Compare value

③ #No of CTC Count $= \frac{Req\ time}{Compare\ time}$

Req time ✓ #No of CTC → Assume

$Compare\ time = \frac{Req\ time}{No\ of\ CTC\ Count}$

Clk time * Compare value = Req time / No of CTC Count

In side Range

---

* Pre load value → $2^n * (1 - 0.y)$
→ $2^8 * (1 - 0.4)$ ⇒ [ ]
↳ will be stored in timer Reg

① No of Count

② Pre load value → if Count = int → 0
if Count ⇒ Floating → law

Req < over flow

① No of Count = $\frac{Req}{over\ flow\ time}$ ⇒ Floating

② Pre load value → $2^n (1 - 0.y)$

---

timer = 16 bit - system = 8 MHz,
Prescaller 8 → Req → 5 sec

① Clk time = $\frac{Prescaller}{system\ F} = \frac{8}{8000000} = 1\ \mu sec$

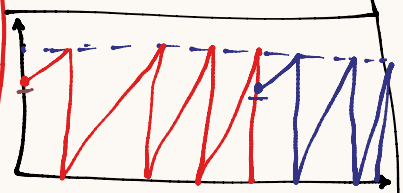
② over flow time = Clk time * 2^n = 1 μsec * 2^{10} =
Req > over flow ⇒ 1024 μsec

③ No of Count ⇒ $\frac{Req}{over\ flow} = \frac{5}{1024 \times 10^{-6}} =$
4882.8125 →

④ Pre load value
= $2^n * (1 - 0.8125)$ ⇒
$1024 * (1 - 0.8125)$ ⇒ 192

timer_Init
↳ timer Reg = 192 ←
↳ timer Normal mode

```
ISR ()
{
  static u32 Count=0;
  Count ++;
  if(Count == 4883)
  {
    // Action; ←
    // Update Preload
    // Counter=0
  }
}
```



Normal Mode (over Flow Mode).

① Clk time = $\frac{Prescaller}{System\ Freq}$

② over Flow time = Clk time * 2^n

③ #No of ovf Count = $\frac{Req\ time}{over\ Flow\ time}$

④ Pre load value = $2^n * (1 - 0.y)$

ex toggel led each 1000 μsec
timer → 8 bit | system clk → 8 MHz
Prescaller = 8

→ Compare value = $\frac{\text{Real time}}{\text{clk time} \times \text{No of ctc}}$

→ clk time = $\frac{8}{8\,MHz}$ = 1 μsec

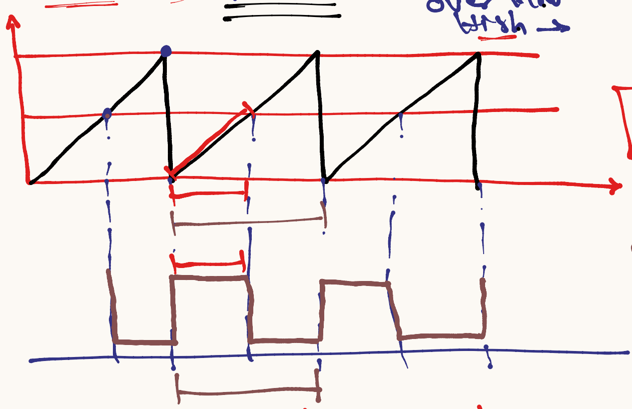
→ Compare value = $\frac{1000}{1 \times 4}$ → 250

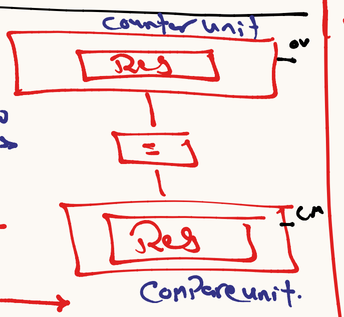
timer_Init
① Compare Reg = 250
② timer Reg = 0
③ CTC_Mode

ISR_CTC()
```
{
static uint16_t Count = 0;
Count ++;
if (Count == 4)
{ Action ;
  Counter = 0;
}
```

Pwm in timer0

* Action on H.w Pin → ① Compar Low, over flow High →

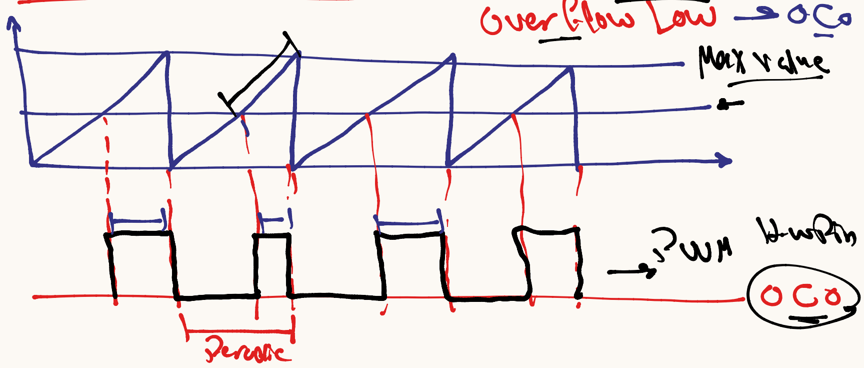




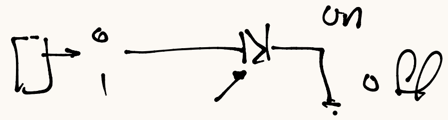
① ontime = Comparetime = clktime * Compare value
② Perodictime = over flow time = 2^n * clk time.
③ Duty cycle = $\frac{\text{clk time} \times \text{Compare}}{\text{clk time} \times 2^n} = \frac{\text{Compare}}{2^n}$

↳ Compare value → 2^n * Duty cycle

* Action on H.w Pin → Compare High → OC0, Overflow Low → OC0

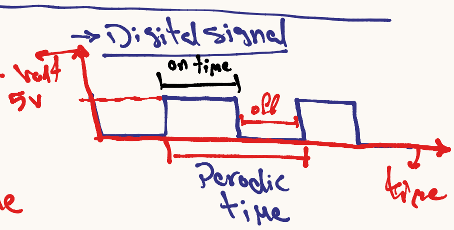


Pwm → Pulse width Modulation



→ out side Analog → DAC
→ Pwm → Simulate Analog Signal

Definition



[1] Perodic time
[2] Frequency → No of cycle in sec
[3] on time → it' time cycle be High
[4] off time → " " " " Low
[5] Duty cycle → Ratio between ontime & Perodic time
= $\frac{\text{on time}}{\text{Perodic time}}$ = $\frac{\text{on time}}{\text{on time} + \text{off}}$
[6] Amplitude ⇒ Max Volt for signal
[7] Root mean squared (RMS)
↳ Analog voltage → H.w
RMS = $A\sqrt{\text{Duty cycle}}$

Pulse width Modulation
↳ Control on width for Pulse by time (on time / Perodic time)

How generate Pwm
① by sw
↳ ① sw. using delay
↳ ② sw. using
  ↳ over flow flag
  ↳ Compare Match flag
② by H.w
↳ H.w using Pwm Circuit.

Fast Pwm   Phase Correct

* Periodic → over flow time = 2^n * clk time

* on time → over flow time − Compare time

→ (2^n * clk time) − (Compare value * clk time)

$$* \text{Duty cycle} = \frac{\text{on}}{\text{Periodic}} = \frac{\text{clk time}\,(2^n - \text{Compare value})}{\text{clk time} * 2^n}$$

$$\text{Duty cycle} = \frac{2^n - \text{Compare value}}{2^n}$$

$$= 1 - \frac{\text{Compare value}}{2^n}$$

$$\boxed{\text{Compare value} = 2^n\,(1 - \text{Duty cycle})}$$