# Process:

First of all I needed to remember everything related to timers,

So I first watched timer videos of this play list:

<https://www.youtube.com/watch?v=Sr7aYoWgMbg&list=PLoiqjtgvXf9e2VJk8GWEXwECPM_7JRwkE&index=32>

# Content:

Here I will be writing all the content I need to cover in the timer session:

First of all we need to revise what’s a clk in a microprocessor:

Clk is the ticks that are digital square waves which make microprocess progress through its functionality like executing line-by-line code that helps the microprocessor to keep in track

In short, it’s like a heart beat for us humans.

The clk has a period which we call a microprocessor clk or a tick which is usually very quick like some micro seconds or even faster.

Ok now let’s talk about timers:

First of all it’s a prephiral so we will find itsdriver in the MCALlayer

Like any other prephiral it has:

* HW circuit : that contains of system clk and prescaler

A prescaler it’s just a frequency divider that means it slows down the system clock

So

* Register: where we can control this prephiral and we will be using systick register

The timer hardware circuit has two modes:

1. Timer mode: which is just a timer like a stop watch if the time run out it will do an interrupt so this is periodic and internal
2. Counter mode: Which just used as a counter here you could use like an external switch to count from the user so this is not periodic and external

Now as for the register:

It contains R= 8 or 10 bit for counting as it stores the # of ticks, and has an enable pin attached to an or gate which is used to determine which mode you will use

So we can say that the total time:

Moving on to the timer modes:

First we got the overflow (OV-F) or normal mode:

Where we begin at zero (if it counts up ++) till we hit the over flow number which is notice that we move from a number to another each tick

So for example if we have a required time for application = X ms we can have three cases:

X = Tov or less or bigger

1) If X=Tov it’s the ideal case that never happens

2) If X> Tov:

That’s actually a good thin we could make it using a SW counter:

#OV = X/Tov

So the result will be = X.Y

The integer part X is easy it’s the number of SW counter as for the float part Y it’s like saying half an hour it’s not a complete cycle it’s a preload value

So our ISR will be in this shape:

ISR()

Counter++;

If(counter == 3)

{

Counter = 0

Time register = Pre load value // notice that we started with the timer fisrt for more accuracy

APP code

}

3) X < Tov

It’s the same as X.Y but always X = 0 that means we don’t need the SW counter

But that has some accuracy problems:

1. App takes time

2. Ov flow will take time in contex switching

That’s why we will move into CTC comapar match

But a little note:

Timer has two kinds:

1. GPT (general Purpose Timer) which is OV-F and CTC

2. SPT which is PWM and WDT (watch dog)

Ok for the CTC:

Compare Timer on Compare match

It’ll use ISR + OC pin so it’s accurate as it uses the hardware pin directly