

## Hi , There

Today , I'll discuss my new Design of Software PWM It's not actually a Separated peripheral it's still depend on TIME , but we may be used for hardware constrains only due to the overhead that add to AVR interrupt

So here we Can chose between two modes first one Based on PWM mode

You can choose this mode By setting **SOFTWARE\_PWM\_CTC\_MODE** macro to Zero , and you need to check this option during compile time

Here in this mode we use two Separated ISR one for compare and the another

one for Overflow (hint this is applied to (TIM0 and TIM2) only so timer one will work only in one mode

Here this is the one of the configuration structure that you need to pass it to all Software PWN subroutines

```
TIMInit_t  Tim_PWM_Handler ={  
    .Instance = TIM0  
    ,.COMPConfig.TIM8Bit.CompAction  
    n = TIM_COMP_PIN_OUT_Normal  
    , .TIM_Interrupt =  
    TIM_0_IT_COMP ,.TimPreScaler =  
    TIM_0_Prescaler_256 ,.TIMMode  
    = TIM_MODE_CTC}; (mode here is  
    neglected )
```

## Another example for timer (0/2)

```
TIMInit_t  Tim_PWM_Handler ={  
  .Instance = TIM0  
  ,.COMPConfig.TIM8Bit.CompAction =  
  TIM_COMP_PIN_OUT_Normal  ,  
  .TIM_Interrupt = TIM_0_IT_COMP  
  ,.TimPreScaler = TIM_0_Prescaler_256  
};
```

On the other hand Timer will work on CTC mode and also I prefer to use this mode because it's need only one ISR

Also you can modify this module by using it to toggle any number of pins as you need