

# Homework #4 – Internal Interrupts



Timers are one of the most essential parts in every microcontroller unit. By utilizing them the designers can perform time sensitive operations such as creating PWM signals or executing a critical code on a regular basis. ATMEGA328p (Arduino Uno's microcontroller) contains three timers labeled as TIMER0, TIMER1 and TIMER2. TIMER0 and TIMER 2 are 8-bit timers and TIMER1 is a 16-bit timer.

In this experiment we want to use **TIMER1 to generate a PWM signal on pin 4**. This pins is not connected to any hardware PWM generation unit thus we can't use `analogWrite()` function in order to generate PWM signal on this pin.

TIMER1 has three different operating modes: **CTC**, **Fast PWM** and **Phase corrected PWM**. As mentioned, pin 4 is not connected to hardware PWM unit so we cannot use the timer in the PWM mode, and we should set the timer's operating mode to **CTC**. In this mode the timer starts to count **from zero and is incremented on each timer's cycle**. The value of timer counter register (**TCNT**) is then compared to a compare register (**OCR**) and if it matches the value stored in this register, it will trigger a **compare match interrupt**. By setting the appropriate registers you can register an ISR for servicing the interrupt (Refer to the device datasheet for further reading).

The last thing you need to configure is the timer's prescaler. The prescaler simply determines the counting frequency of the timer. For example, if the prescaler is set to one, the timer counts on each microcontrollers' clock cycle and if it is set to 1/8, it counts with 1/8 speed of the microcontroller clock speed.

For this experiment you should set the timer to **CTC mode with the prescaler value of 1/8** and write an ISR to **switch pin 4 in a manner that creates a PWM signal with adjustable duty cycle between 10% and 90%**. The PWM resolution should be **10-bits** and the duty cycle should be adjusted using a potentiometer. You can monitor the generated signal using an oscilloscope in Proteus.

NOTE: Arduino configures the timer configuration registers on boot and you need to reset them before applying your own settings.

Components used in proteus:

SIMULINO UNO, POT-HG, oscilloscope

Good Luck

Graders Team  
Shiraz University  
Computer Science and Engineering Dept.  
Microprocessors Lab  
Spring 2021

