

LAB 5

Non-Blocking PWM LED Dimmer

NAME:

POSSIBLE POINTS: 10

STUDENT ID:

COURSE DATE & TIME:

OBJECTIVE:

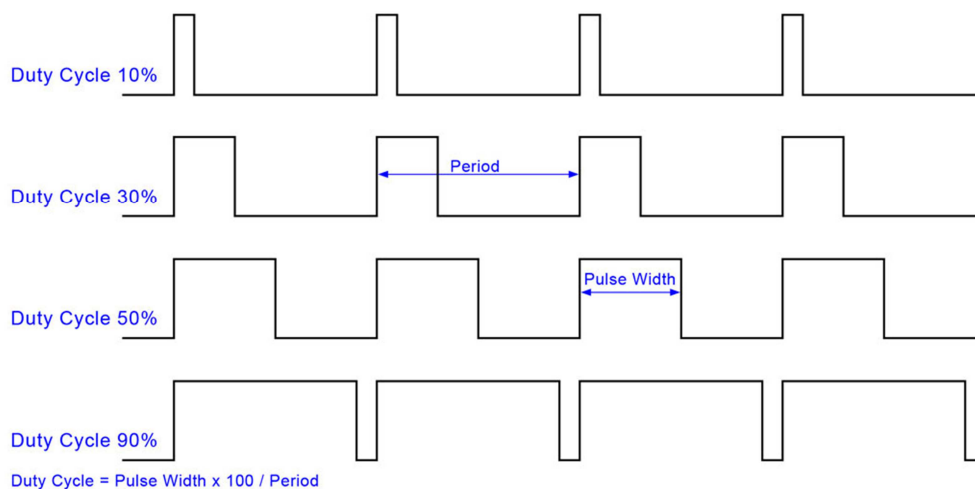
- To change the brightness of an LED with a variable non-blocking PWM (Pulse Width Modulated) signal.
- As opposed to lab4 in which a blocking piece of code (delay routine) was used to generate the duration of the positive pulse, this lab will take a completely interrupt driven approach in order to free up a considerable amount of CPU time.

OPERATION:

The program will take a binary input from the switches on the 8051 Lab Pro-51 Board. The valid input will range from 0 to 255 decimal. We conventionally refer to the duty cycle of a PWM signal as a percentage from 0% to 100%. Since our Timer peripheral in mode 0 acts as an 8-bit timer and our 8051 MCU is an 8-bit processor we will use decimal 0 as a duty cycle of 0% and decimal 255 as a duty cycle of 100%. When PWM is used to drive an LED, a duty cycle of 0% will make the LED off while a duty cycle of 100% will cause the LED to be completely on. PWM values in between 0% and 100% will cause a varying brightness.

THEORY:

When using the Timer peripheral in Mode 1 to generate an Interrupt we can manually load the timer register with a value. If we also use a flag inside of the ISR and check it when this interrupt occurs we can use it to toggle an I/O port pin.



LAB 5

Non-Blocking PWM LED Dimmer

The following psuedo-code should explain how the ISR should operate.

```
Timer0_ISR()
{
    if(Flag == 0)
    {
        //Start of High level
        //Flag = 1
        //SET Pin High
        //TIMER_REG = 255 - PWM_DUTY_CYCLE;
    }
    else
    {
        //Start of Low level
        //Flag = 0
        //SET Pin Low
        //TIMER_REG = PWM_DUTY_CYCLE;
    }
}
```

LAB WRITE-UP:

The lab write-up will include this page as the cover sheet with any questions answered and the source code. Also a schematic of your specific implementation will be included. You can draw your schematic by hand or with any EDA (Electronic Design Aid) or Cad software you like. You must label all devices, values, and power.

Example Programs:

CircuitMaker by Altium – This is the free version of my favorite circuit board design tools

Orcad by Cadence – Another very popular and industry used tools

KiCad – One of my favorite free PCB (Printed Circuit Board) design tools.

Eagle – Another popular free PCB design tools.

DEMO AND GRADING:

When your project is ready, you will demonstrate the functionality to the instructor and hand in the write-up. Your demo will be graded on its ability to correctly resolve a desired PWM duty cycle from the value set on the switches. Also I will be looking for good coding techniques which will be discussed during the lab period. Code structure and flow of data in your programs become crucial with the use of interrupts.