EE333 Introduction to Microcontrollers

Lab#5 Pulse Width Modulation

Allan A. Douglas, Associate Professor

**Lab Overview**

In this lab, students will learn how to use microcontroller timer/counters in Pulse Width Modulation (PWM) mode to create a variable DC voltage, to vary the intensity of an LED, and to control to speed of a DC motor.

**Required Materials**

|  |  |
| --- | --- |
| Breadboard | Various Resistors |
| Arduino UNO (or Spark Fun RedBoard equivalent) | Various Capacitors |
| ATmega328P Microcontroller Datasheet | Oscilloscope |
| LED | Arduino ISE Software |
| DC Motor (M260) Sparkfun ROB-11696 | Flyback Diode (1N5820) |
|  | Transistor- NPN (BC337) |

**Required Work**

The areas of **bold** text below are items required in your Lab Report.

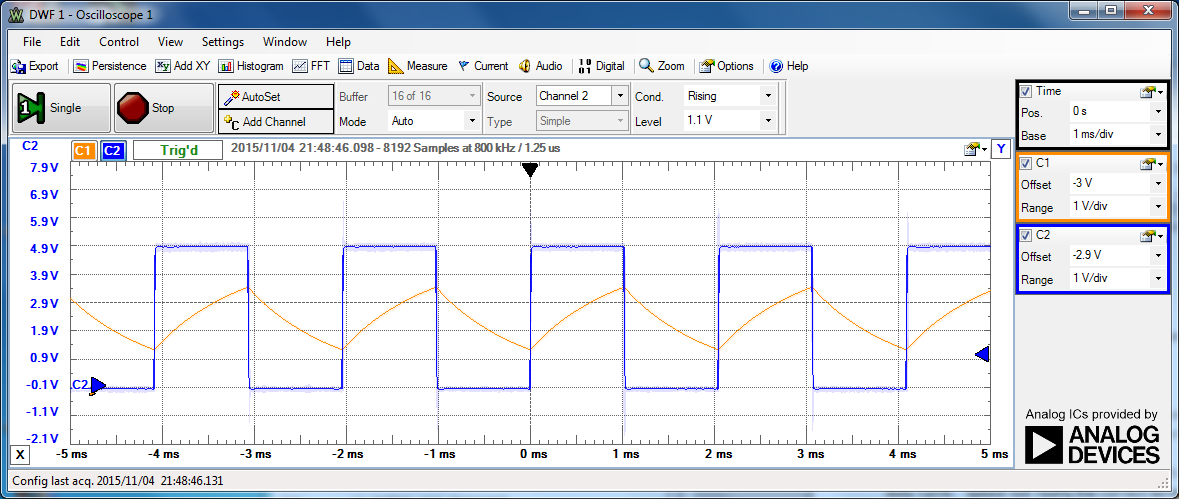
Part 1: Variable DC Voltage with PWM

The goal of this part of the lab is two write a sketch to use PWM to charge a 0.1uF capacitor through a 10K Ohm resistor. The external RC circuit forms a low-pass filter for the PWM signal.

Write an Arduino sketch to generate a PWM signal using pin 9. Use the analogWrite(motorPin, speed); function to set the PWM duty cycle. Speed not really the correct term. We are actually setting the PWM duty cycle (range is 0-255).

Set the duty cycle to 50% (127/256) and measure the frequency of the PWM signal and the voltage on the capacitor with an oscilloscope. Your frequency should be approximately 490Hz.

Your waveforms should look similar to the following figure.



**Include a copy of your sketch in your Lab Report.**

**Include an oscilloscope screenshot in your Lab Report.**

You should see the capacitor charging and discharging. Examine the average voltage on the capacitor.

Now, step the duty cycle value from 0 to 255 in steps of 15. Measure the average value of the capacitor voltage for each step with the oscilloscope.

**Create a table with the following columns: duty cycle, expected capacitor voltage, measured average capacitor voltage.**

**Explain how the duty cycle relates to the average voltage on the capacitor. Discuss your results.**

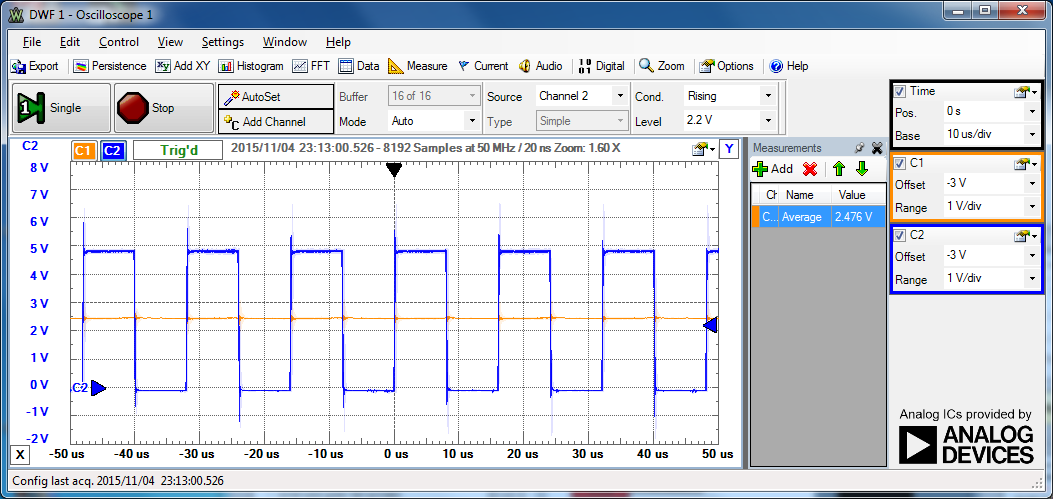
If we wanted to generate a smoother voltage on the capacitor, we need to increase the frequency of the PWM signal.

Now, write another Arduino sketch to control the PWM directly using the timer/counter registers. Do not use the analogWrite() function. Use pin 9 as your PWM output.

Setup Timer1 for Fast 8-bit PWM. Set the prescaler to 1 (no prescaling). Use OCR1A to set the duty cycle.

You should only need to manipulate the following Timer1 registers TCCR1A, TCCR1B, TCCR1C, TCNT1, OCR1A, and OCR1B.

Your goal is to increase the frequency until the voltage on the capacitor is smooth (minimize the ripple) as shown in the diagram below.



Capacitor

Voltage

Microcontroller

Output

**Include a copy of your sketch in your Lab Report.**

Now, step the duty cycle value from 0 to 255 in steps of 32. Measure the average value of the capacitor voltage for each step with the oscilloscope.

**Include an oscilloscope screenshot in your Lab Report with the duty cycle set to 50%. Measure the PWM frequency.**

**Explain why the capacitor voltage is smooth now.**

**Include a picture of your Arduino and RC circuit in your Lab Report.**

Part 2: Variable LED Intensity with PWM

In this part of the lab, you will use PWM to vary the intensity of a LED. When you are done with this part the LED intensity should follow a sinusoid function.

Disconnect the RC circuit from the PWM output.

Connect a 330 Ohm to the PWM output (pin 9). Connect a LED from the resistor to ground. Drive the LED with your previous Arduino sketch to make sure you can vary the LED intensity by setting a constant PWM duty cycle.

Create a new Arduino sketch. Write your loop() function so that you dynamically adjust the PWM duty cycle using a sin() function that varies slowly between 0 and 255. To use the sin() function you will need to include the <math.h> library. You may use the delay() function from Timer0 to control the speed.

Adjust your sin() and delay() functions until the LED is blinking at about 1 Hz.

**Include a copy of your sketch in your Lab Report.**

**Include a picture of your Arduino and LED circuit in your Lab Report.**

**Record a few seconds of video showing the LED blinking. Include this video (or a link) with your Lab Report submission.**

You can do this part of the lab with direct Timer/Counter register manipulation or use the Arduino analogWrite() function.

**For extra credit, implement the variable LED intensity with both methods.**

Part 3: Variable DC Motor Speed

Disconnect your previous LED circuit from the PWM output.

Connect the following DC Motor and drive circuitry to the PWM output (pin 9).

Write a new Arduino sketch to drive the DC motor with PWM. In this case, you may use the direct register writes to the Timer/Counter control registers or analogWrite().

Examine your +5V power supply voltage and the PWM signal while running the motor. Make sure that the +5V supply is not dropping when the transistor is on. If the +5V is starting to sag, use an external +5V supply for the motor and connect the ground connections to the Arduino ground.

**Include an oscilloscope screenshot in your Lab Report showing both the PWM signal and the +5V supply.**

**Include a copy of your sketch in your Lab Report.**

**Include a picture of your Arduino and motor circuit in your Lab Report.**

**Record a few seconds of video showing the motor running. Include this video (or a link) with your Lab Report submission.**

**Describe the operation of the motor drive circuit. Explain why the diode required in the drive circuit.**