Lab 10: **40 points.**

Objective: Develop an understanding of the Digital to Analog Converters (DAC's), and the effect of the RC circuit and the period of the PWM on the output waveform.

Description of Lab 10: In this lab we will be trying to create a voltage, using the Pulse Width Modulation (PWM) output pins on the Arduino. As I stated in class, the analogWrite() function supplied by Arduino has a couple short comings. Thus we will be using the Timer1 system, that has been installed in the lab, to generate the PWM for our experiment.

For this lab the first thing to do is hook up the circuit shown in Figure 10-1. The connection to the Analog Discovery is similar to that done in Lab 9, except now we are watching the input to the Arduino, the output from the PWM and we will use the Analog Discovery to generate a waveform as an input to our system. Also, note that no values were given for the R and C in the drawing. That is because we will be changing them through the course of the experiment. Initially they should be set to a 10 K Ω (Brown, Black, Orange color coding) and a 0.1 μF (104 marking) capacitor.

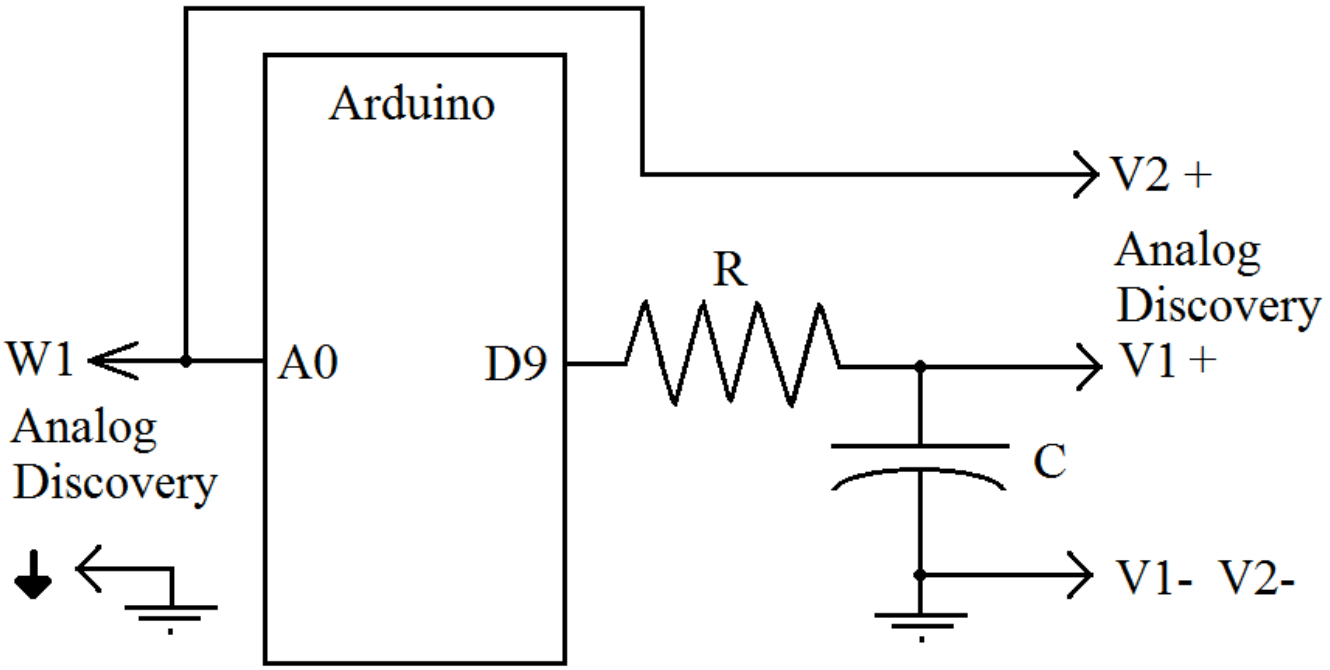

Figure 10-1. Analog Input and Output.

The input to the Arduino is the waveform generator from the Analog Discovery, which can generate a variety of signals. The signal we will be using in this lab is a sine wave, but one of the challenges here will be the fact that sine characteristically swing positive and negative. However the waveform generator has the ability to apply an offset to the waveform, thus we will be generating an input that swings from about 0 to 4 volts. The settings for the Waveform Generator are shown in Figure 10-2. Note that we have a setting of 10 Hz, an amplitude of 2 volts, an offset of 2 volts and a symmetry of 50%.

One last point, in the program described below it calls for the LED or pin 13 to be turned on before the sampling operation and off after it is complete. This signal from the Arduino allows us to monitor the sampling rate we are actually getting from the Arduino.

Documenting the results of your experiments with screen shots will be a critical part of this lab!

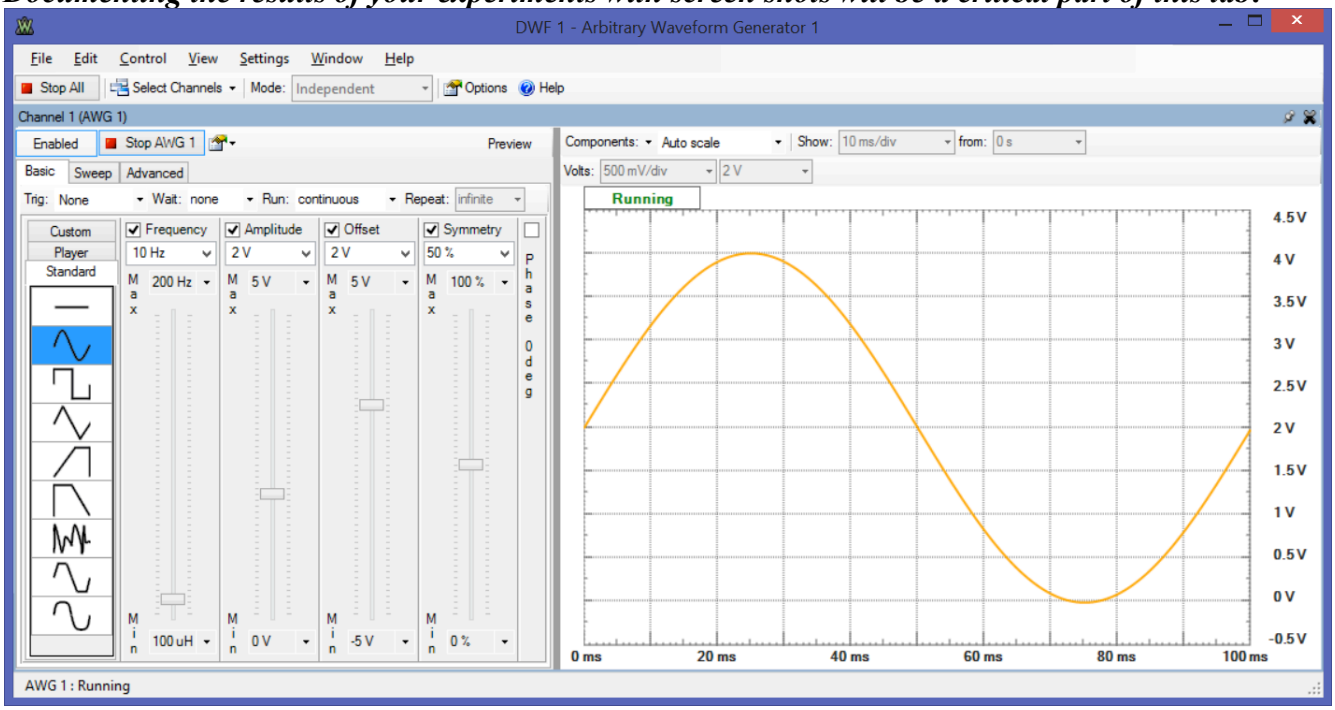

Figure 10-2. Waveform Control Display

Lab Assignment: The code for this lab is very simple. It is intended to read the voltage every 10 milliseconds and then send it out to the PWM. Note we will be using the PWM from the Timer1 software. Demonstrations of its use can be found on the Files->HardwareInterfacing link, in TimerOneDemo.pdf .

Setup:
    Set up your Timer used for 10 milliseconds
    Set Timer1 to have a 500 microseconds period
        and a PWM on pin 9.
Loop:
    if 10 milliseconds has passed
        Turn on pin 13 (the LED)
        Read analog voltage on A0
        Set Duty Cycle on the PWM pin
        Turn off pin 13 (the LED)

Prelab: Write the program described above and upload it to the Prelab 10 location.

Lab 10:

1) Setup the circuit in Figure 10-1, using a 10 K Ω resistor and a 0.1 μF capacitor. Set the

function generator as shown in the Figure 10-2, with a 10 Hz sine wave, 2 volt amplitude, and a 2 volt offset. Demonstrate this to your lab instructor, and also show the effect for an input waveform at 90 Hz.

2) Change the capacitor to 0.2 μF's (placing two capacitors in parallel) and the resistor to a 1 K Ω resistor. Capture the waveform for an input of 10 Hz, recording the results as a screen shot of the oscilloscope

3) Change the program so that the PWM has a period of 250 microseconds, and again capture the output waveform for the 10 Hz signal.

4) Change the circuit back to a 10 K Ω resistor and a 0.1 μF capacitor, leaving the period of the PWM as 100 microseconds and capture the 10 Hz signal again.

5) Connect channel 1 from the Analog Discovery to pin 13 on the Arduino, and using the oscilloscope's measurement settings, measure the frequency (cycles per second) of this signal.

Questions:

1) Which of the changes had the most effect on the "cleanness" of the output waveform, the R and C values or the period?

2) How close to the desired 100 Hz (1 / (10 milliseconds)) sampling rate did you achieve?