1. Year and Semester : 2013 SPRING

2. Course Number : CS-492

3. Course Title : Kinect Programming

4. Work Number : LA-02

5. Work Name : Arduino Laboratory Report

6. Work Version : Version 1

7. Long Date : Sunday, 3, February, 2013

8. Author(s) Name(s) : Jake Waffle

Circuit #1

The first circuit was pretty simple. The It produces a blinking LED light. The LED component simply had its negative lead (the shorter one) connected to the ground through a resistor (so that the voltage is limited and doesn't bust the LED, I think.) The resistor is linked to the ground through the designated negative bar on the breadboard. And the LED's positive lead (the longer one) is connected to PIN 13 by a wire (which seems to be connected to a timer by default.)

After getting that to work, I copied the code from the book for Project #1 into our processing IDE. Then I altered the argument passed to the delay() function to see if the upload to the Arduino was successful. And sure enough, the speed of the blinking for the LED changed.

The circuit and code for this project can also be found at (http://arduino.cc/en/Tutorial/Blink.)

Circuit #2

The second circuit was much like the first one in that it involved turning on an LED light. But the difference was in the amount of LED’s that were being turned on.

First off the ground PIN needs to be connected to one of the bars on the side of the breadboard (I think there is a negative sign next to the bar) with a wire. Then a resistor needs to be plugged into that bar and connected to another bar with an LED's shorter lead (negative) on it. And that needs to be done for every LED that exists on the board. Then the bars that the LED's longer lead (positive) need to have a wire plugged into them and connected to a PIN # on the Arduino (these then are able to be controlled through the code.)

Then I copied the code from out book into our Processing IDE. The code originally would just loop through the LEDs and turn them on again and again (which only really needed to be done once.) So I changed the code so that after turning the LED lights on with one function and it would then turn them all off with another function. The function simply looped through all of the PINs attached the LEDs and changed their output to LOW (which turns the LEDs off.)

Circuit #3

This circuit is a little more complex. When completed it will run a motor for a time period, turn the motor off for another period and repeat.

Looking at the circuit, when the motor is connected to a ground and is getting a HIGH input from a PIN (or if the analogWrite() is being used, the speed of the rotation is given instead,) the motor will run. PIN 9 is connected to a resistor that is connected to the middle lead of the transistor. This makes it so that when PIN 9 is set to HIGH, the ground will be able to connect to the negative wire on the motor. So two wires need to connect the negative wire on the motor and the ground to the outside leads on the transistor.

The Transistor has three wires connected to it and when the middle wire gives it input, the other two wires will be connected (this could be compared to a switch.) And the 5v PIN is connected to the positive wire on the motor at all times with a diode in-between (to convert the AC power to DC power.) All of this together allows us to run a motor while PIN #9 gives input to the transistor. Otherwise the transistor will cease the connection between the ground and the motor.

Then after figuring out what the circuit did, I uploaded the example code (copied from the book from project #3) for accelerating the motor's speed and running the motor at a specific speed. The code seemed pretty straight-forward (turn PIN #9 on/off to turn the motor on/off.) But I noticed that the speed of the motor is handled by the analogWrite() function. That

seemed interesting because it gave more control over the motor than the digitalWrite() did (which just turns it on/off.)

The included example seemed to go over a lot on its own, so I just tried to understand it and uncommented out the different functions that were there. It definitely has given me a better idea about the analogWrite() and digitalWrite() functions.