CS281: Computer Systems

# Lab 06: Ohm's Law and Servo Motor Control

The main purpose of this lab is to build a servo motor control circuit. As with prior labs, there will be some exploratory sections designed to familiarize you with the basic operation of components. Then there will be a design section in which you apply your knowledge to solve a design challenge. **The lab report should focus solely on your design for the design portion of the lab, not on the exploratory sections.**

## Exploration: Potentiometer

In this first section, we will familiarize ourselves with the operation of the potentiometer. Your breadboard has two of them in the lower center; we will use the 1K POT.

1. Your 1K POT has three different connection points: a left column, a right column and a bunch of center columns.
2. Wire one of the pins on the left column to +5 volts.
3. Wire one of the pins on the right column to Ground.
4. Plug a wire into any one of the pins on any of the middle columns. Leave the other end of this wire unconnected for the moment.
5. Obtain a Multimeter. There are only a few of these, so please be patient and share when you are not using it.
6. Hook the ground connection of the multimeter to Ground. This is the black connector. It should have an alligator clip on it that can connect to the black Ground post at the top right of your breadboard.
7. Hook the red connection of the multimeter to your wire that you have coming out of the middle of your 1K POT.
8. Turn on the multimeter to the $\bar{V}$ setting. This is DC voltage – it is one "click" to the right of the off position.
9. Turn on your breadboard and measure the voltage on the 1K POT. Turn the know to different positions and record a few position/voltage readings in a table. Be sure to measure the max and min readings.

Now let's take readings with our Arduino board. Hook the wire coming out of the 1K POT to your Analog0 pin on the Arduino. Wire up the Arduino with connections to +5 and Ground on your breadboard. Write an Arduino program that reads and prints values from this pin. You should use the `analogRead()` function. Look at the reference guide if you have questions. Again, make a table in which you record knob position and Arduino reading; be sure to measure both extremes.

## Exploration: Photoresistor

We now turn our attention to a sensor. The photoresistor measures light brightness; it is a resistor that changes its resistance value according to the brightness of the light impinging upon its surface. Your kit should have two photoresistors; choose one and keep it separate from the other as the two are likely to have subtly different behaviors.

**Test 1**

1. Obtain the multimeter.

2. The multimeter has two alligator clips, black and red. Clamp one alligator clip to one lead of the photoresistor and clamp the other alligator clip on the other lead of the photoresistor. In this case, it doesn't matter which alligator clip is connected to which lead.

3. Turn the dial on the multimeter to resistance. It is the Ω symbol, two clicks to the left from the center/off position. You should now see a value on the display. If the light in the room is on, it will probably read something close to 7kΩ, read as 7 kilo-Ohms of resistance. [1]

4. Put your hand over the photoresistor (darken it). The reading on the multimeter should drop considerably.

5. Measure the photo resistance for different kinds of brightness. Make a little table such as the following:

[1] We've used 10k resistors for our LEDs in the past – these are 10 kilo Ohms, just slightly more resistance than this photoresistor.

| Lighting Conditions | Measurement (kΩ) | Measurement (Arduino Pin 0) – see Test 2 below |
|---|---|---|
| Dark Room | | |
| Completely Covered | | |
| Partially Covered (cupped hand over top) | | |
| Ambient lit room | | |
| With flash light 1 foot from face (I used my iPhone light) | | |
| With flash light directly on face (I used my iPhone light) | | |

**Test 2**

1. Build the circuit with the photoresistor in series with a 1kΩ resistor (it should have brown, black, red for the stripe colors). The 1kΩ resistor should go to +5v. The other end of the photoresistor to ground.

2. One of Kirchoff's Loop Laws tells us how to compute the voltage drop across the photoresistor. Let $x$ be the resistance (in Ohms) of the photoresistor. The voltage drop is proportional to the photoresistors resistance as compared to the whole resistance. If we have $V = IR$, then we can compute the voltage drop across the photoresistor as

$$v_{pr} = 5v \cdot \frac{x}{x + 1000}$$

3. Connect the positive end of the photoresistor (where it connects to the 1k resistor) to Arduino analog input 0.

4. Wire the Arduino to power and ground as usual.

5. Run the following program and observe the results in the serial window. Make a table of different lighting conditions and record the readings from the `analogRead()` function.

```
#define V_PIN 0

void setup ()
{
  Serial.begin(9600);
  pinMode(V_PIN,INPUT);
}

void loop ()
{
  int val = analogRead(V_PIN);
  Serial.println(val);
  delay(1000);
}
```

### Exploration: Servo Motor

Obtain your servo motor. Wire the servo motor to your Arduino board with the brown wire going to Ground, red wire to +5V and yellow wire to Digital Pin 9 on your Arduino. Secure one of the three attachments to the shaft; I like the one with an unbalanced, single arm as this will be useful in your lab later. Write the following program and observe the results.

```
#include <Servo.h>
#define SERVO_PIN 9

Servo myservo;

void setup()
{
   Serial.begin(9600);
   myservo.attach(SERVO_PIN);
}

void loop ()
{
   int i;
   for (i = 0; i < 180; i += 10)
   {
      myservo.write(i);
```

```
        delay(1000);
    }
}
```

## Design: Brightness Indicator

The main purpose of this design portion of the lab is to construct a robotic device that measures the ambient lighting conditions and indicates those by positioning a dial. Obtain one of the ”servo cover cards” from the class professor. This fits over your servo motor; you might want to secure it in place with a little piece of tape. If you haven’t done so already, put on the ”single arm” attachment device to the shaft of the motor. It now acts like a dial that points to some place on the arc drawn on the card.

Your job is to construct a device that positions the servo arm correctly on the card to indicate the current ambient lighting conditions. The servo should be fully to the left in a dark room (lights off). The servo should be straight up in a regular classroom condition with the lights on. The servo should be all the way to the right when a flash light is held directly on the servo. Have it respond in a reasonable way to lighting conditions in between these three conditions.

Make sure your servo controller does not try to push the servo too far left (beyond 0) or too far right (beyond 179) as this will damage the servo motor internals. Note: since the scale on the card does not vary linearly with the lighting conditions being read from the photoresistor, you will have to be creative (mathematically) to map the input to the right output condition. It is unlikely you will get good results from either using the `map()` built-in function or by a simple switch statement.

Demonstrate the correct operation of your device to the class professor. Your lab report should focus exclusively on this design portion of the lab. Describe fully the problem given to you. Describe fully your hardware design and construction. Describe fully your software solution (include your code in the lab report). Pay special attention to the design challenges that you overcame and that makes your solution unique/better than other solutions. Include graphs, pictures, charts, tables or other visuals that will help convey your design.