

# **Lab Report: Project 14 – Tweak the Arduino Logo**

Lucas Porter

Spring 2025 TESI 3300

Trevor P. Robinson, PhD

Lab 14 – Tweak the Arduino Logo

04/13/2025

## **Abstract**

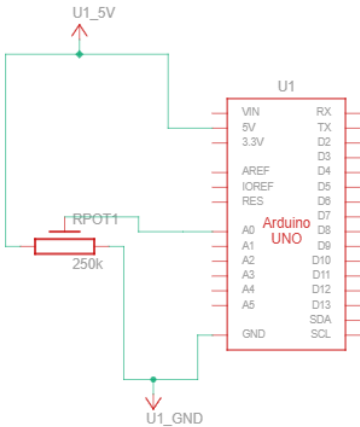
The purpose of this lab was to use an Arduino Uno and a potentiometer to control the color properties of the Arduino logo displayed in a Processing sketch. This project demonstrated basic serial communication between an Arduino and a computer program, how to manipulate visual outputs using input sensors, and the basics of modifying Processing sketches to respond to serial data.

## **Materials**

- Arduino Uno Board
- Breadboard
- Potentiometer
- Jumper Wires
- USB Cable
- Computer with Arduino IDE and Processing IDE

Procedure

Circuit Diagram



## Steps

1. Connected the Arduino's 5V and GND pins to the power and ground rails of the breadboard.
2. Connected the center pin of the potentiometer to an analog input pin (A0) on the Arduino.
3. Connected the two side pins of the potentiometer to 5V and GND.
4. Uploaded the Arduino sketch to read analog values and send them over serial communication.
5. Loaded the Processing sketch to read serial data and modify the displayed logo accordingly.
6. Adjusted the Arduino and Processing code for better responsiveness.

## Code

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   Serial.write(analogRead(A0) / 4);  
7 }  
8  
9
```

```
1 import processing.serial.*;  
2 Serial myPort;  
3 PImage logo;  
4 int bgcolor = 0;  
5  
6 void setup() {  
7   size(1, 1);  
8   surface.setResizable(true);  
9   colorMode(HSB, 255);  
10  logo = loadImage("https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Arduino_Logo.svg/720px-Arduino_Logo.svg.png?20200922062315");  
11  surface.setSize(logo.width, logo.height);  
12  
13  println("Available serial ports:");  
14  println(Serial.list());  
15  
16  myPort = new Serial(this, Serial.list()[0], 9600);  
17 }  
18  
19 void draw() {  
20   if ( myPort.available() > 0) {  
21     bgcolor = myPort.read();  
22     println(bgcolor);  
23   }  
24  
25   background(bgcolor, 255, 255);  
26   image(logo, 0, 0);  
27 }
```

## Discussion

1. **In serial communication, how many bits of data are sent at a time?**

One bit is sent at a time.

2. **What is meant by the term baud rate?**

The baud rate is the speed of communication, measured in bits per second.

3. **For what does the acronym USB stand?**

Universal Serial Bus.

4. **What is the typical baud rate when communicating between the Arduino board and the computer?**

9600 bits per second.

5. **How do the functions `Serial.write()` and `Serial.print()` differ in terms of actual performance/function in the Arduino software?**

`Serial.write()` sends raw binary data, while `Serial.print()` formats the data as readable ASCII characters.

6. **Once you have twisted and turned to your heart's desire (changing the color), try replacing the pot with an analog sensor. Find something you find interesting to control the color. What did you use? How well did it work?**

I replaced the potentiometer with a photoresistor. It worked fairly well; the color changed depending on the amount of light hitting the sensor. Bright light made the logo lighter, and shadow made it darker.

7. **On what programming language are Processing and Arduino based?**

Arduino is based on C++ and Processing is based on Java

## Troubleshooting

1. **Issue:** There was a noticeable delay between turning the potentiometer and seeing the color change on the screen.

**Solution:** Tweaked the Arduino and Processing code to reduce delay times and increase responsiveness, improving the interaction.

## **Conclusion**

The project initially had a significant delay between adjusting the potentiometer and the logo changing colors, but after some adjustments, the responsiveness improved considerably. Processing was able to communicate successfully with the Arduino, and the overall interaction worked as expected. Although some minor lag remained, the system accurately reflected changes in input. Swapping the potentiometer for a photoresistor demonstrated another way sensor input can dynamically control on-screen elements. This project provided good practice in serial communication and visual output manipulation.