Lab Report: Project 5 – Mood Cue

Lucas Porter

Spring 2025 TESY 3300

Trevor P. Robinson, PhD

Lab 5 – Mood Cue

2/16/2025

Abstract

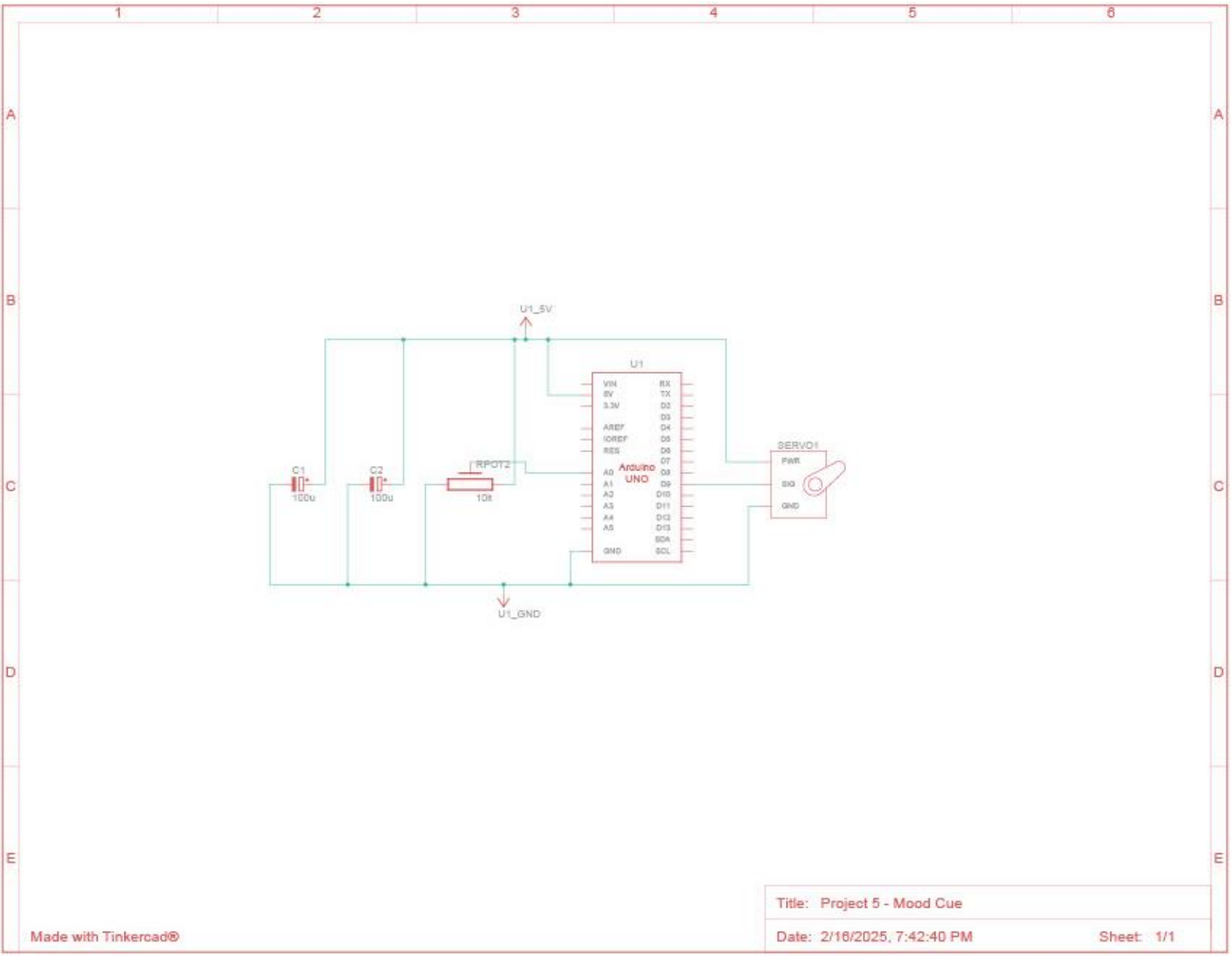
The purpose of this lab was to use an Arduino and a servo motor to create a Mood Cue, a device that moves an indicator based on user input from a potentiometer. The project demonstrated how servo motors operate and how analog input values can be mapped to control movement. By using the map() function, the potentiometer's range of values was converted into an appropriate angle for the servo motor. This experiment provided experience in working with servo motors, analog sensors, and circuit components like capacitors to ensure stable operation.

Materials

1. Arduino Uno Board
2. Breadboard
3. Servo Motor
4. Potentiometer
5. Capacitor
6. Jumper Wires
7. USB Cable
8. Computer with Arduino IDE

Procedure

Circuit Diagram



Steps

1. Connected the Arduino's 5V and GND pins to the power and ground rails of the breadboard.
2. Placed the potentiometer on the breadboard and connected its middle pin to an analog input pin (A0).
3. Connected one side of the potentiometer to 5V and the other to GND.
4. Connected the signal wire of the servo motor to a digital PWM pin (9).
5. Connected the power and ground wires of the servo motor to the Arduino's 5V and GND, respectively.
6. Placed capacitors in the circuit to help smooth power fluctuations.
7. Uploaded the Arduino sketch and tested the circuit.

Code

```
#include <Servo.h>

Servo myServo;

int const potPin = A0;
int potVal;
int angle;

void setup() {
  myServo.attach(9);
  Serial.begin(9600);
}

void loop() {
  potVal = analogRead(potPin);
  Serial.print("potVal: ");
  Serial.print(potVal);

  angle = map(potVal, 0, 1023, 0, 179);

  Serial.print(", angle: ");
  Serial.println(angle);

  myServo.write(angle);

  delay(15);
}

```

Discussion

1. **How does a servo motor function differently than a D.C. electric motor?**
   A servo motor moves to a specific position based on a control signal, whereas a DC motor continuously spins in either direction depending on voltage polarity.
2. **How many degrees of motion do typical servo motors have?**
   Most standard servo motors have a range of 0 to 180 degrees of motion.
3. **How does the map() function allow the analog input (potentiometer) to be transferred into usable data for the servo motor?**
   The map() function converts the potentiometer's 0-1023 range into a 0-180 degree range, making it suitable for servo motor control.
4. **If you set the potentiometer value to ~500, what is the equivalent servo motor angle?**
   Using the map() function, an input of 500 would map to approximately 88/90 degrees.
5. **The potentiometer is not the only sensor you can use for controlling the servo. Using the same physical setup (an arrow pointing to a number of different indicators) and a different sensor, what sort of indicator can you make?**
   A temperature sensor could adjust the servo position to indicate hot, warm, or cold conditions, or a photoresistor could point the arrow toward different light levels.
6. **How would this work with temperature (like in the Love-o-Meter)?**
   A temperature sensor could read the current temperature and use map() to convert it into a servo position that moves an indicator.
7. **Could you tell the time of day with a photoresistor?**
   Yes, a photoresistor could be used to detect ambient light levels, which generally correlate with time of day.
8. **How does mapping values come into play with those types of sensors?**
   Mapping values is essential for converting sensor readings (such as temperature or light intensity) into a meaningful output, such as an LED brightness level or a servo motor angle.

Troubleshooting

1. **Issue:** Ensuring capacitors were placed correctly.
   Solution: Since this was the first time using capacitors, I carefully checked their orientation multiple times before finalizing the connections. This prevented potential damage from incorrect placement**.**

Conclusion

The servo motor behaved as expected and responded smoothly to adjustments made with the potentiometer. Most movements were fluid, allowing for precise control of the indicator's position. However, I did notice a slight input delay when making very quick adjustments to the potentiometer. Despite trying to adjust delay times in the code, the delay persisted, which could be due to the servo motor's response time rather than a coding issue. Other than that, the circuit functioned correctly, and I did not have to make any changes to the code to achieve the desired behavior.

This lab provided valuable experience in working with servo motors and mapping sensor values to control physical movement. Understanding how the map() function translates an analog input into a usable servo motor angle was a key takeaway. Additionally, working with capacitors for the first time reinforced the importance of careful component placement. This type of setup could be expanded to use other sensors, such as temperature or light sensors, for dynamic and responsive projects.