Lab Report: Project 3 - Love-O-Meter

Lucas Porter

Spring 2025 TESY 3300

Trevor P. Robinson, PhD

Lab 3 – Love – O – Meter

2/2/2025

Abstract

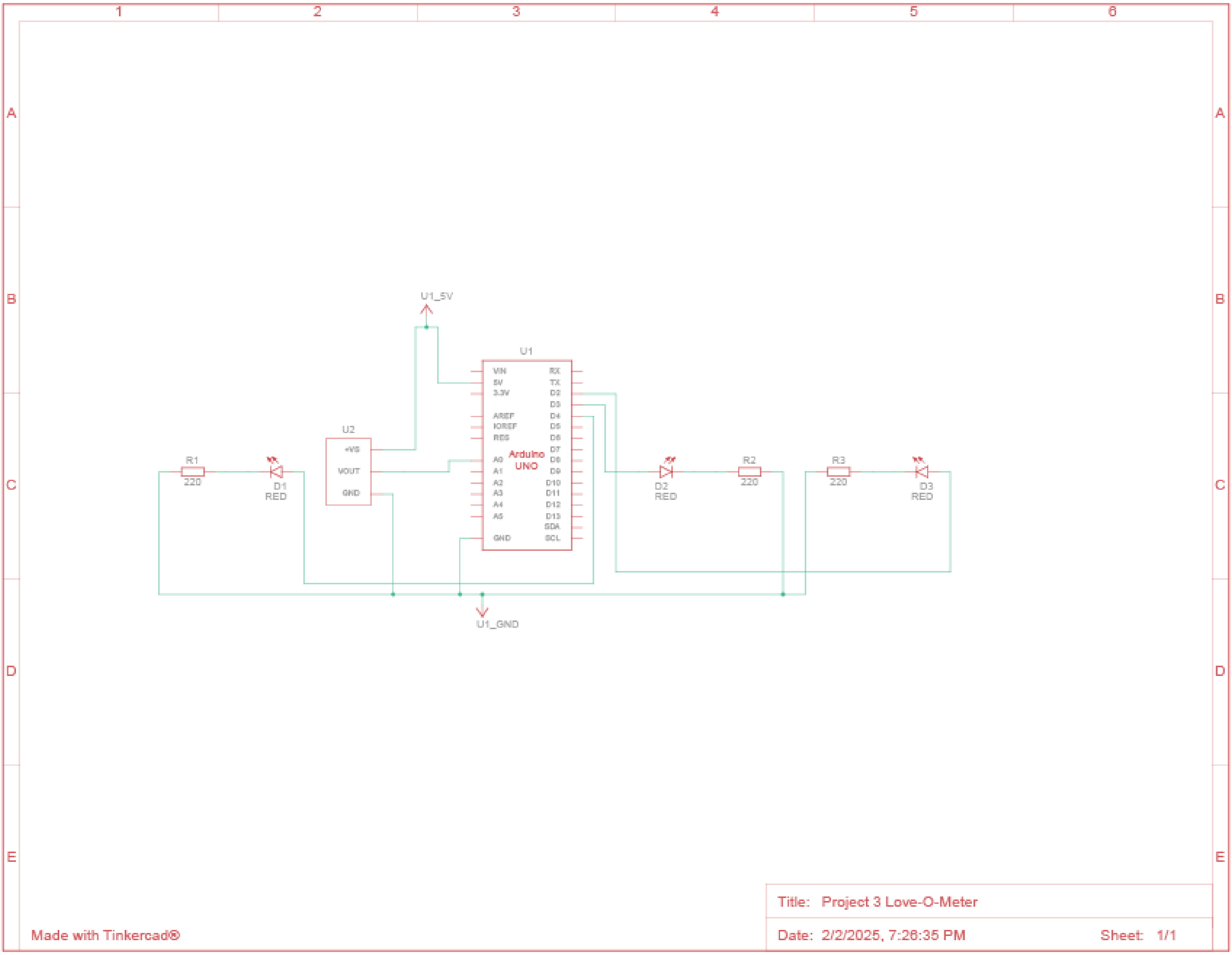
The objective of this lab was to use an Arduino microcontroller and a temperature sensor to create a Love-O-Meter, an interactive circuit that responds to temperature changes by lighting up LEDs. The circuit utilized a TMP36 temperature sensor, which provided an analog input to the Arduino. The program read the sensor's output and determined how many LEDs should light up based on the detected temperature. This lab reinforced the importance of calibrating sensors correctly to produce meaningful results.

Materials

- Arduino Uno Board
- Breadboard
- Three Red LEDs
- One TMP36 Temperature Sensor
- Three 220Ω Resistors
- One 10kΩ Resistor
- Jumper Wires
- USB Cable
- 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 TMP36 temperature sensor on the breadboard with its correct orientation (VCC, OUT, GND).
3. Connected the VCC pin of the sensor to the 5V rail and the GND pin to the ground rail.
4. Connected the OUT pin of the sensor to the Arduino's analog input pin (A0).
5. Placed three LEDs on the breadboard, connecting their anodes to digital pins 2, 3, and 4, respectively, through 220Ω resistors.
6. Connected the cathodes of all LEDs to the ground rail.
7. Uploaded the provided Arduino sketch and tested the circuit.

Code

```
const int sensorPin = A0;
const float baselineTemp = 27.0;

void setup() {
  Serial.begin(9600);

  for (int pinNumber = 2; pinNumber < 5; pinNumber++) {
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}

void loop() {

  int sensorVal = analogRead(sensorPin);
  Serial.print("sensor Value: ");
  Serial.print(sensorVal);

  float voltage = (sensorVal / 1024.0) * 5.0;
  Serial.print(", Volts: ");
  Serial.print(voltage);


  Serial.print(", degrees C: ");
  float temperature = (voltage - .5) * 100;
  Serial.println(temperature);

  if (temperature < baselineTemp + 2) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  else if (temperature >= baselineTemp + 2 && temperature < baselineTemp + 4) {
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  else if (temperature >= baselineTemp + 4 && temperature < baselineTemp + 6) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  }
  else if (temperature >= baselineTemp + 6) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }
  delay(250);
}

```

Discussion

1. How does the temperature sensor work?
The TMP36 temperature sensor converts temperature into an analog voltage. The sensor outputs a voltage that increases linearly with temperature, which the Arduino reads as an analog input. The voltage is then converted to a temperature value using a predefined formula.

2. What does a "for loop" do?
A for loop is used to execute a block of code multiple times. It is typically used when a task needs to be repeated a specific number of times, such as setting up multiple pins or iterating through an array of values.

3. What is the benefit of using the serial monitor in a project like this?
The serial monitor allows real-time debugging and observation of sensor values. It helps verify that the sensor is reading accurate data and aids in troubleshooting by displaying temperature values in real time.

4. What is the difference between digital and analog inputs?
Digital inputs can only detect two states: HIGH (5V) or LOW (0V), whereas analog inputs can detect a range of values between 0V and 5V, allowing for more precise readings from sensors like temperature or light sensors.

5. List three other devices from your everyday life that are analog devices.

- Thermostat (measuring room temperature)
- Volume control on a radio (continuous level adjustment)
- Light dimmer switch (gradual brightness control)

6. What other situations can you see using analog sensors?
Analog sensors can be used in various real-world applications, such as monitoring environmental conditions, adjusting lighting based on ambient brightness, and measuring fluid levels in containers.

Troubleshooting

1. Issue: Baseline temperature was initially set too low, causing LEDs to always be on.

   Solution: Adjusted the baseline temperature to better reflect the actual room temperature.

2. Issue: LEDs were slow to respond to temperature changes.

   Solution: Adjusted the temperature differentials in the code to make the response more noticeable and dynamic.

Conclusion

The circuit-building aspect of this project was straightforward, I didn't encounter any issues with this aspect. However, fine-tuning the software required adjustments. Initially, the baseline temperature was set too low, causing the LEDs to always remain on. Adjustments to the temperature baseline improved the response of the LEDs. Something I noticed was the gradual dimming and flickering of the last LED as the temperature dropped below its threshold, showcasing how analog signals provide continuous rather than binary feedback. This project highlighted the importance of data calibration, real-time monitoring, and understanding analog input behavior.