Alex Harvey - 25%

Amy Lee - 25%

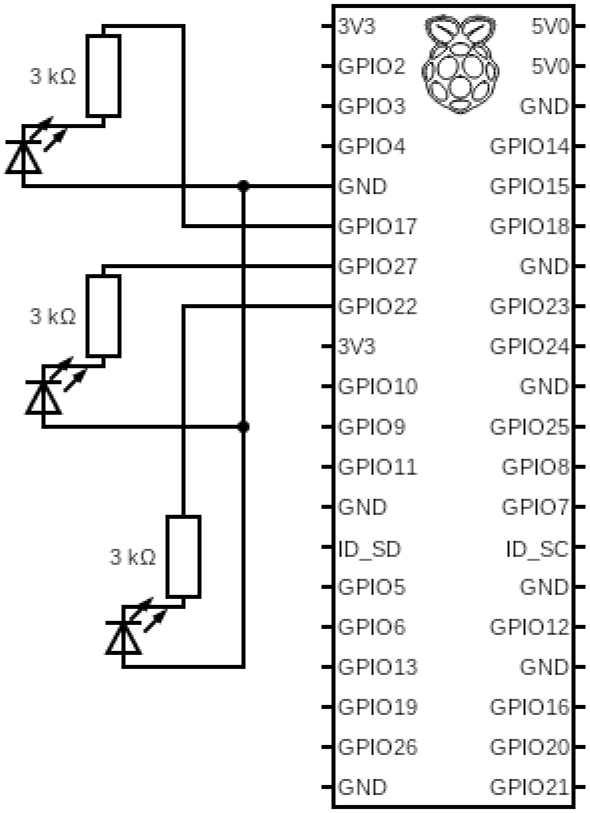
Sayali Patukale -25%

Seth Schallau - 25%

	Seth	Sayali	Alex	Amy
Broker setup code	Figured out broker install/testing commands	Tested and install code	Tested and install code	Tested and install code
Raspberry Pi A	Proofread code to ensure it met requirements	Wired up the adc/light sensor	Proofread code to ensure it met requirements	Wrote the code to communicate with the broker
Raspberry Pi B	Wrote the code, wired the LED’s, uploaded schematic	Proofread code to ensure it met requirements	Proofread code to ensure it met requirements	Proofread code to ensure it met requirements
Raspberry Pi C	Proofread code to ensure it met requirements	Proofread code to ensure it met requirements	Wrote the code	Wrote example code and proofread code to ensure it met requirements
Laptop #2	Wrote the code	Proofread code to ensure it met requirements	Proofread code to ensure it met requirements	Wrote example code and proofread code to ensure it met requirements

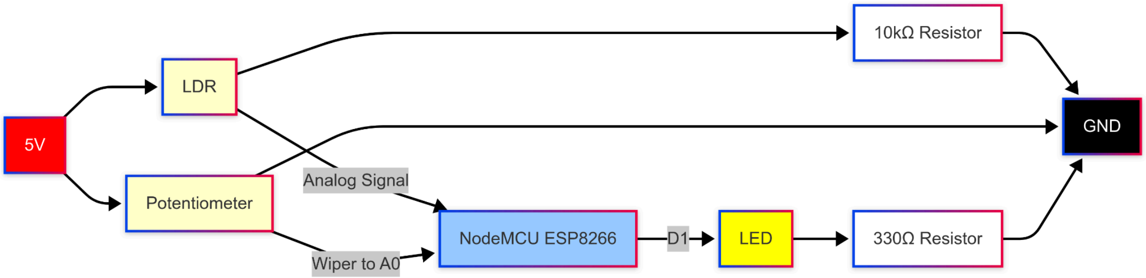
Report	Made sure to fill in gaps of teammates contributions/ helped wrote the paper	Proofread	Proofread	Managed the team and provided guidance when issues arose/ helped wrote the paper

1. For the schematics diagram of connection of LEDs to Raspberry Pi B:

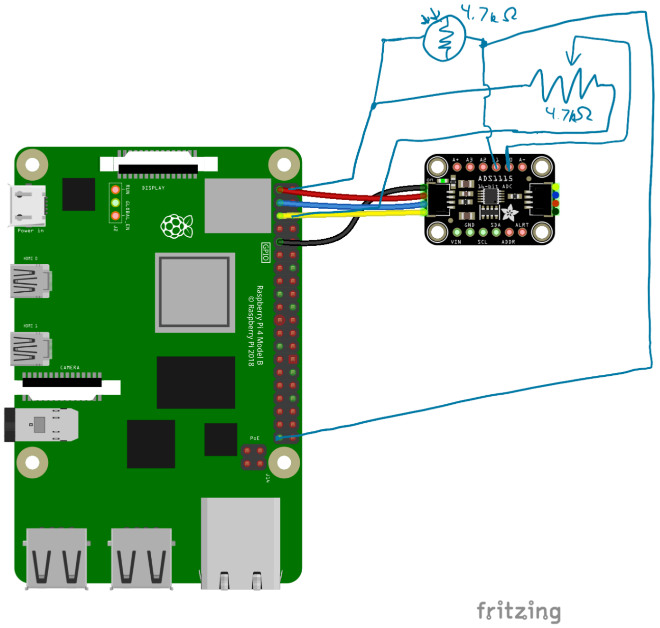


   a.

2. For the schematics diagram of connection of LDR and potentiometer to Raspberry Pi A:



a.



Our setup involves using Mosquitto as the MQTT broker, installed onto Raspberry Pi and a computer. Mosquitto was chosen due to its lightweight nature, ease of installation, and broad community support. We configured it to listen on port 1883 with anonymous access enabled for simplicity in local network testing. The message queue was limited to 1000 messages, with queue_qos0_messages disabled to optimize real-time message delivery. The ADC is sampled at around 10 Hz. The sample frequency is determined by the time.sleep(0.1) at the end of the loop. Since the code pauses for 0.1 seconds between iteration, it samples the ADC roughly every 0.1 seconds which translates to about 100 samples per second(10 Hz). The code scales the LDR's raw values (10–100) and the potentiometer's raw values (90–250) into a uniform range of 0–100 for consistent and comparable sensor readings.The function scale_value performs a linear mapping, it subtracts a sensor-specific minimum, divides by the range (max minus min), and then multiplies by 100 (the new range), finally clamping the result between 0 and 100. For the scaled values range, after scaling with the function, regardless of the raw input, the values are mapped to a range of 0 to 100. This range makes it easier for the Raspberry Pi C to compare the light sensor and potentiometer readings. For step by step installation and setup instructions you do the following:

Open your cmd or terminal window:

1. Install Mosquitto and Mosquitto clients:
    a. `sudo apt install mosquitto mosquitto-clients -y`
2. Edit the Mosquitto configuration file:
    a. `sudo nano /etc/mosquitto/mosquitto.conf`
3. Add the following lines:

```
listener 1883
allow_anonymous true
max_queued_messages 1000
queue_qos0_messages false
```

Save the file and exit

4. Restart the Mosquitto service:
    a. `sudo systemctl restart mosquitto`
5. Start Mosquitto in verbose mode:
    a. `mosquitto -c /etc/mosquitto/mosquitto.conf -v`
6. Test the broker in two separate terminals:
    a. One terminal : `mosquitto_sub -h localhost -t "test/topic" -q 2`
    b. Second terminal: `mosquitto_pub -h localhost -t "test/topic" -m "Going Test Mode" -q 2 -r`