ASSIGNMENT# 3
Smart LED Lighting

Design:

MQTT Broker

In our implementation, we used “mosquitto” as MQTT Broker. The following are some of its features:

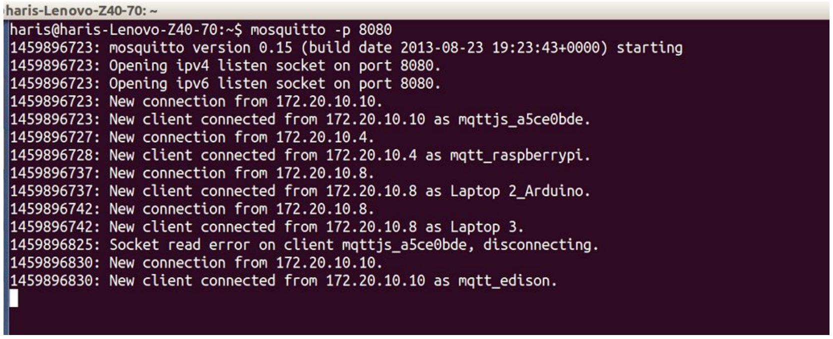
1. It is an open source message broker which supports MQTT protocol versions 3.1 and 3.1.1. This helps any user to pull the repository from Git, and can experiment as well as make changes to the Broker as per requirement.
2. One of the most important feature of mosquitto is that it is written in C, therefore MQTT server can run on machines which do not have capacity for running a JVM (Java Virtual Machine).
3. It is extremely lightweight. The current release of mosquitto executable is of 120 KB that consumes around 3MB RAM with 1000 clients connected.
4. Mosquitto has a bridge which allows to connect to other mosquitto MQTT servers and form a network which helps in passing MQTT messages from any location in a network to another depending upon the configuration of bridge.
5. Simple and easy-to-use command line Broker.

Installation:

1. We added mosquitto-dev PPA to our Ubuntu machine's repository list and updated the machine.
   - sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
   - sudo apt-get update
2. Now through command-line, we can run the mosquitto MQTT server:

   mosquitto -p 8080

Note: By default, the port number used is 1883.



```
haris-Lenovo-Z40-70: ~
haris@haris-Lenovo-Z40-70:~$ mosquitto -p 8080
1459896723: mosquitto version 0.15 (build date 2013-08-23 19:23:43+0000) starting
1459896723: Opening ipv4 listen socket on port 8080.
1459896723: Opening ipv6 listen socket on port 8080.
1459896723: New connection from 172.20.10.10.
1459896723: New client connected from 172.20.10.10 as mqttjs_a5ce0bde.
1459896727: New connection from 172.20.10.4.
1459896728: New client connected from 172.20.10.4 as mqtt_raspberrypi.
1459896737: New connection from 172.20.10.8.
1459896737: New client connected from 172.20.10.8 as Laptop 2_Arduino.
1459896742: New connection from 172.20.10.8.
1459896742: New client connected from 172.20.10.8 as Laptop 3.
1459896825: Socket read error on client mqttjs_a5ce0bde, disconnecting.
1459896830: New connection from 172.20.10.10.
1459896830: New client connected from 172.20.10.10 as mqtt_edison.
```

Figure 1: Snapshot of MQTT Broker in listening state

Setup:

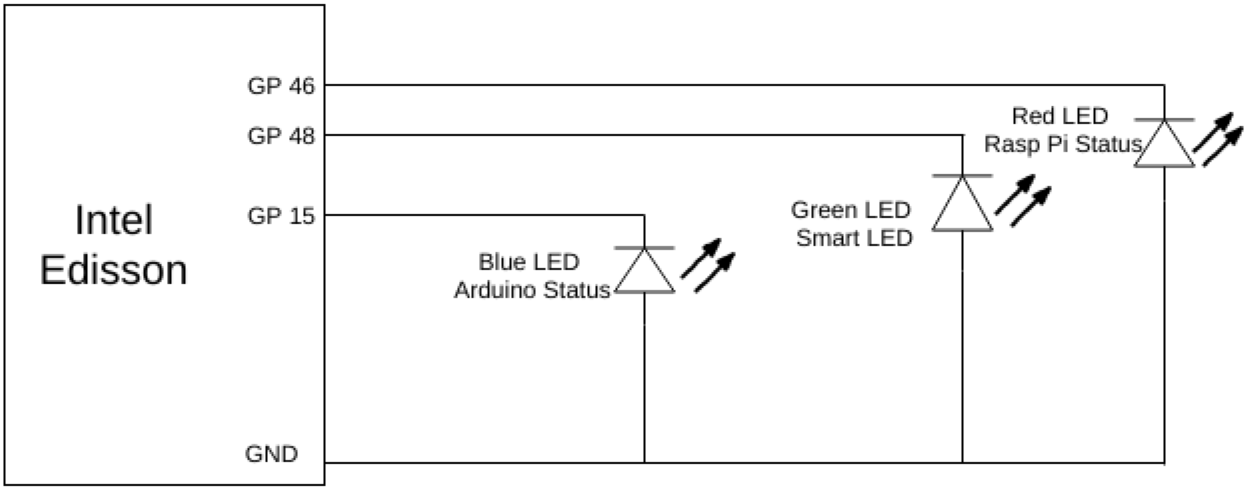


Figure 2: Schematics diagram of connection of LEDs to Intel Edison

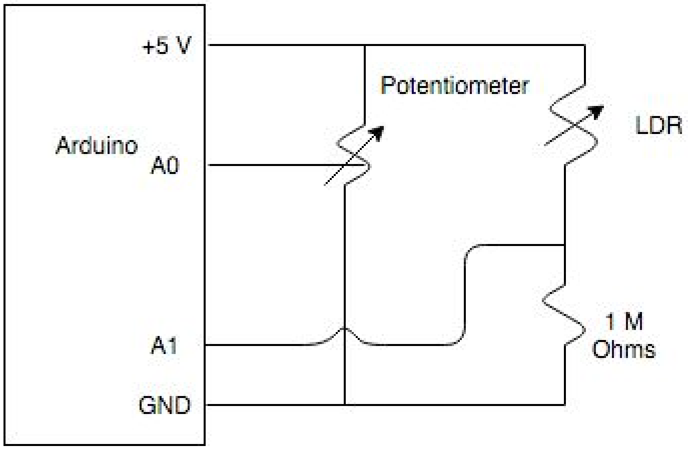


Figure 3: Schematics diagram of connection of LDR and potentiometer to Arduino Uno

Arduino Configuration:

What frequency did you sample the ADC at?
The Arduino is sending 1 sample every 100ms.
Therefore the sampling frequency is 10Hz.

What baud rate did you use to send data from Arduino to laptop # 2?
9600

How did you scale values from potentiometer/LDR before posting to their corresponding topics so that they could be compared by Rasp Pi?

The values were scaled using a scaling factor. Answers for the following questions were used to determine the scaling factor.
As in the presence and absence of light, the scaled values of LDR drastically change. Hence, on laptop 2 the threshold for difference between previous value and new value is kept high. i.e. 0.25.
As the Potentiometer value acts as the threshold and controls the lightening of Smart LED, on laptop2 where it is compared with previous value of threshold, the difference checked for (0.15)is comparatively small to give more control. If the difference is greater than this threshold, then the value is published to topic threshold.

What was the range of raw values (min and max) that your ADC got from the LDR?
Environment: The room used to capture data for minimum and maximum value has ambient light. Analog value from LDR is given to the analog input pin of arduino which converts it to digital value and displays on the serial communication monitor for Arduino.

Minimum Value was observed on covering the LDR with hand.
Min: 1
Maximum Value was observed on focusing the light from the Nexus 6 flashlight onto the LDR at a distance of approximately 1 cm above it for 4 seconds.
Max: 152

What are the range of raw values that your ADC got from potentiometer(min and max)?
Analog value from Potentiometer is given to the analog input pin of arduino which converts it to digital value and displays on the serial communication monitor for Arduino.
Minimum value is 0
Maximum value is 1022.

What are the range of scaled values (min and max) that resulted after you scaled the values from the potentiometer and/or LDR
For scaling, the raw values are divided by the scaling factors.
Scaling factor for Potentiometer: 1024.
Scaling factor for LDR: 160.

For Potentiometer:
Min. Scaled value: 0.00
Max Scaled Value: 0.99

For LDR:
Min. Scaled value: 0.00
Max Scaled Value: 0.95

Submitted by:

Haris Ansari
hmansari@ncsu.edu

Jina Jain
jhjain@ncsu.edu

Arpita Awasthi
aawasth2@ncsu.edu

Pranjali Chumbhale
pchumbh@ncsu.edu