Home Automation using Raspberry Pi and MQTT Cloud

Debanjali Saha
(1810110059)

Hrishika Enugula
(1810110079)

Isha Arora
(1810110084)

06/12/2021

—

Internet of Things (EED-379)

—

Dr. Rohit Singh & Dr. P.C. Jain

Abstract

The aim of this project is to control home appliances from anywhere using an MQTT server (hosted on a cloud) and the Raspberry Pi.

The drawback of a local MQTT server is that we cannot control the GPIOs from anywhere in the world, it only provides services locally. But if this MQTT server is hosted on some cloud then any appliances connected to Raspberry Pi can be controlled globally.

So, we have used the Adafruit IO as an MQTT broker to control an appliance connected to Raspberry Pi GPIO in the lab. LED connected to the Raspberry Pi represents an appliance signifying that any appliance connected to the board can be controlled from anywhere in the world in this fashion.

Table of Contents

List of Figures

Introduction

MQTT (message queue telemetry transport) is a lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP, however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. An MQTT broker is an intermediary entity that enables MQTT clients to communicate. Specifically, an MQTT broker receives messages published by clients, filters the messages by topic, and distributes them to subscribers. In this case, it is connected to a cloud- Adafruit IO.

Adafruit.io is a cloud service. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that such as display your data in real-time and online, make your project internet-connected: Control motors, read sensor data, etc. connect projects to web services like Twitter, RSS feeds, weather services, etc. as well as connect your project to other internet-enabled devices. Adafruit IO supports different hardware like Raspberry PI, ESP2866, and Arduino.

IoT developers prefer Adafruit IO over other IoT cloud providers for the following reasons:

- Powerful API - Provides us libraries for various programming languages, which also provides built-in user interface support.
- Dashboard - Understanding data via charts and graphs enable us to make better decisions.
- Privacy - Data is secured in the cloud platform with better encryption algorithms.

MQTT, or message queue telemetry transport, is a protocol for device communication that Adafruit IO supports. Using an MQTT library or client you can publish and subscribe to a feed to send and receive feed data. Adafruit IO's MQTT API exposes feed data using special topics. You can publish a new value for a feed to its topic, or you can subscribe to a feed's topic to be notified when the feed has a new value. Any one of the following topic forms is valid for a feed:

- (username)/feeds/(feed name or key)
- (username)/f/(feed name or key)

Where (username) is your Adafruit IO username (the same as specified when connecting to the MQTT server) and (feed name or key) is the feed's name or key.

Logistics Required for the Project

The components required for the project are as follows:

- Raspberry Pi with Raspbian Stretch installed in it.
- Relay Module
- Bulb
- Jumper Wires
- A laptop/mobile to access the Adafruit IO platform.

The relay module is an electrically operated switch that can be turned on or off deciding to let current flow through or not. They are designed to be controlled with low voltages like 3.3V or 5V like your Arduino or Raspberry Pi. Relays are used to control high voltage circuits with the help of low voltage signals.

The relay module is basically used for the conversion of voltage between the home appliance and the Raspberry Pi.

Due to the unavailability of a relay module in the lab, it was decided with permission from our project supervisor, Dr. Rohit Singh sir, to directly connect the LED to the Raspberry Pi using a breadboard for the purpose of demonstration.

Following was the circuit diagram for these IoT Controlled Home appliances with MQTT cloud and Raspberry Pi basically connecting a bulb with relay module on GPIO pin 35 of Raspberry Pi.

Circuit Diagrams

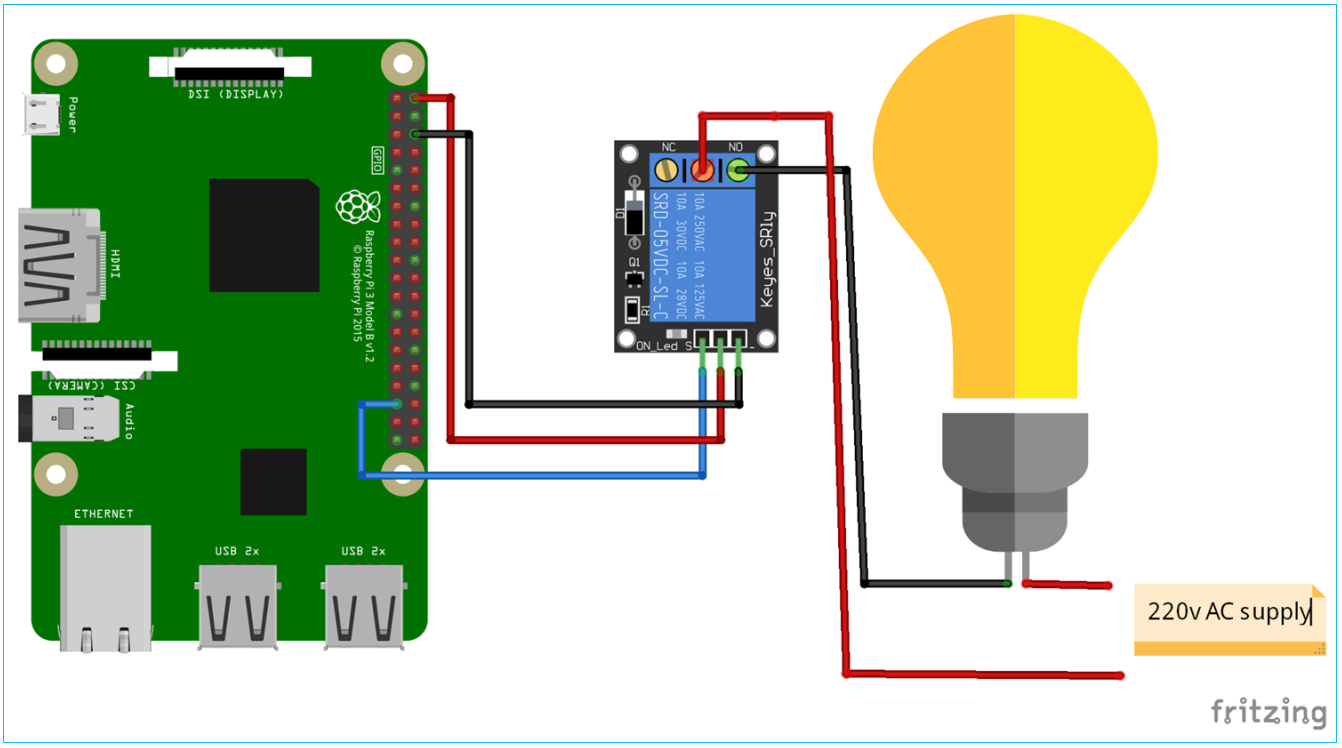

Fig. 1: Initial circuit diagram for IoT Controlled Home appliances with MQTT cloud and Raspberry Pi

Following was the circuit diagram which was eventually made use of. In this, we eliminated the Relay Module.

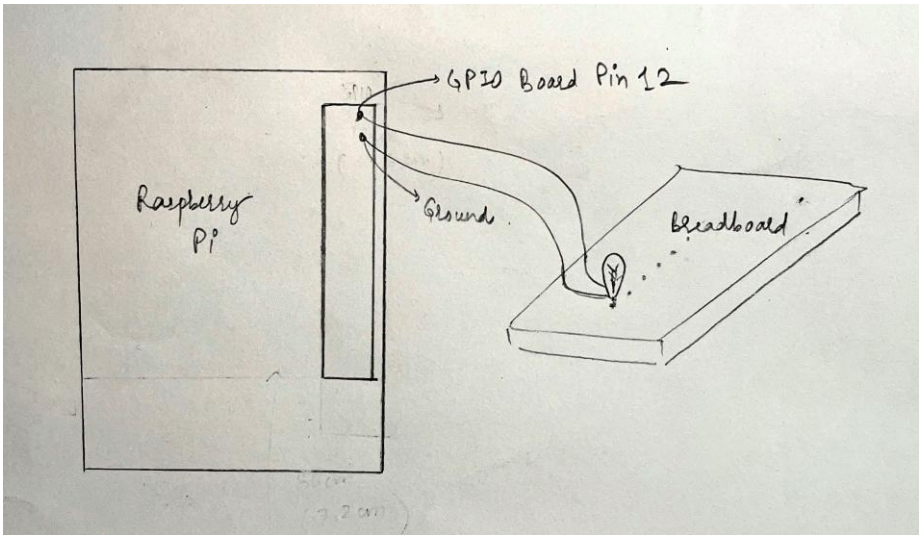

Fig. 2: Final circuit diagram for IoT Controlled Home appliances with MQTT cloud and Raspberry Pi

Procedure to access Adafruit IO Platform

The process we used to access the cloud platform and use it for our project is as follows:

1. Made a free account on the Adafruit IO platform.
2. Logged in to Adafruit IO with personal credentials. An Adafruit IO Key is generated by the platform which will be used in our code. It can be regenerated with a click of a button.
3. Created a new dashboard called IoT Project
4. Created a new feed and named it appropriately for the appliance to be controlled (LED light in our case and so is named 'Light')
5. Next, in this feed called 'Light', a new block of type 'Toggle' is created to be used as a switch. It is named 'Bulb Light'. Button on is set up as '1' and off is set up as '0'.

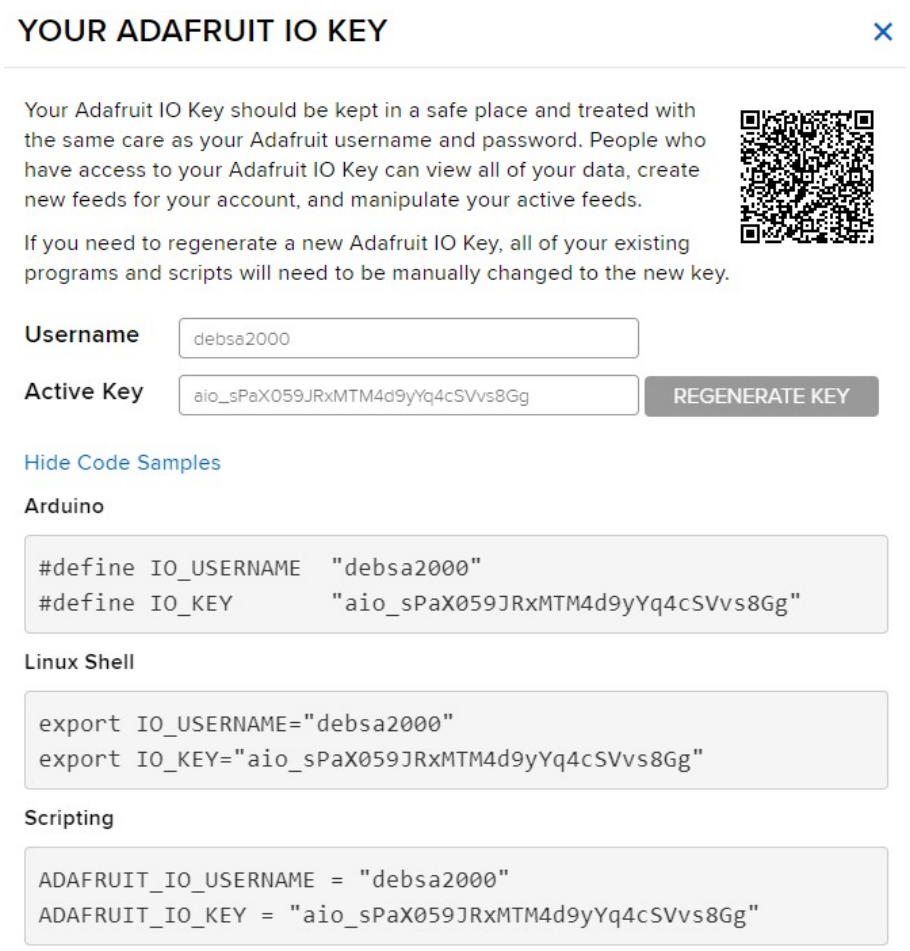

Fig. 3: Generation of Adafruit IO Key

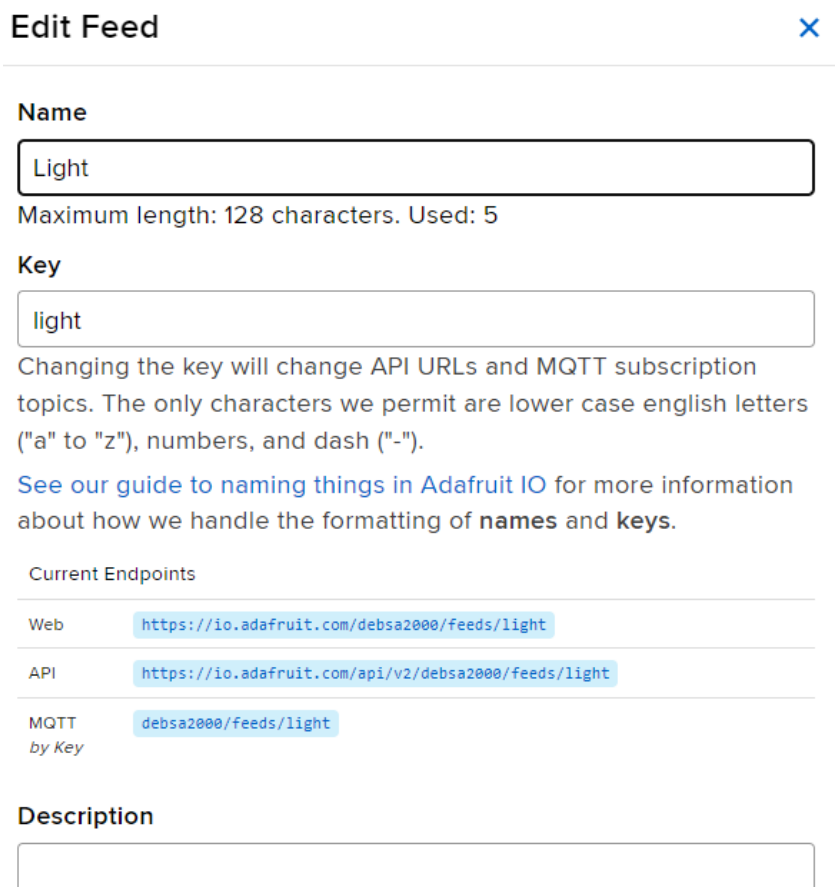

Fig. 4: Feed

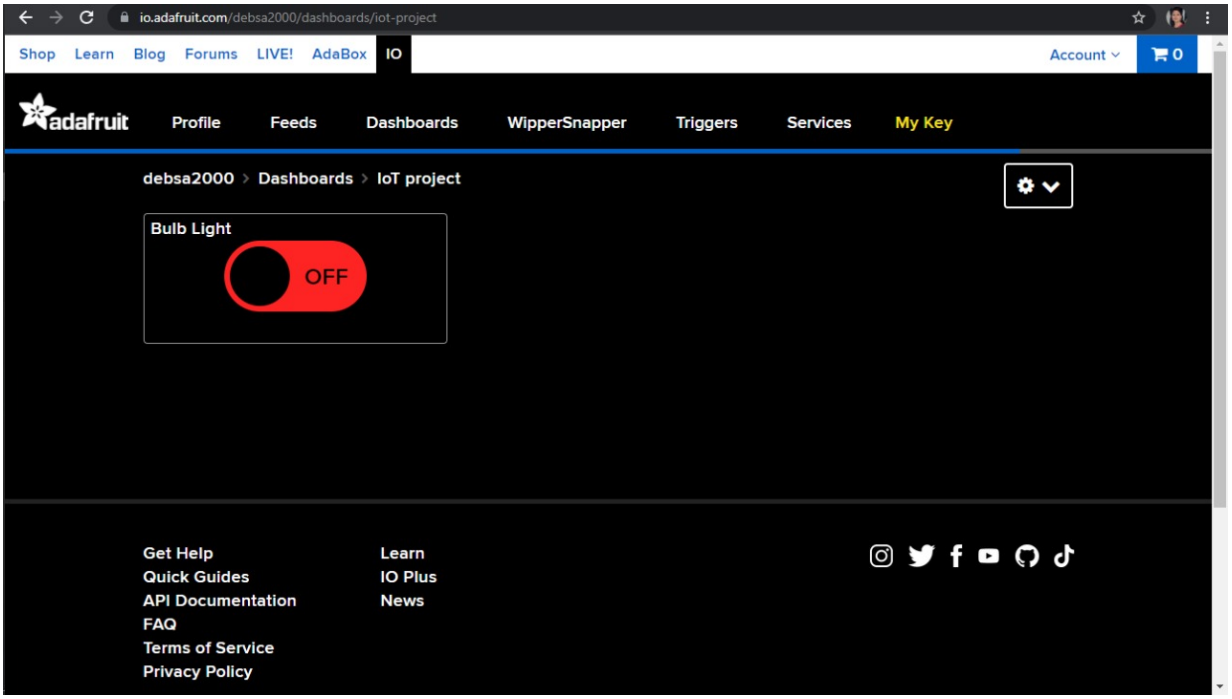

Fig. 5: Adafruit Dashboard with the toggle used to control the appliance

Code Description

PART-1

- All the required libraries to use GPIO pins and Adafruit MQTT client are imported.
- GPIO mode is set to Board numbers and LED pin number 12 is set as output.
- Next, we have set the AdafruitIO key of our dashboard and the Username of the account as ***aio_sPaX059JRxMTM4d9yYq4cSVvs8Gg*** and ***debsa2000***, respectively.
- Enter the feed name to turn on and off the ***light*** is entered.

subscribe.py

```
1  import RPi.GPIO as GPIO
2  import sys
3  from Adafruit_IO import MQTTClient
4
5  GPIO.setmode(GPIO.BOARD)
6  GPIO.setwarnings(False)
7  ledPin = 12
8  GPIO.setup(ledPin, GPIO.OUT)
9
10 ADAFRUIT_IO_KEY = 'aio_sPaX059JRxMTM4d9yYq4cSVvs8Gg'
11
12 ADAFRUIT_IO_USERNAME = 'debsa2000'
13
14 FEED_ID = 'Light'
15
```

Fig. : Code- Part 1

PART-2

- A function called ***connected*** that will be called when the event will happen is defined. So, we will subscribe to the Feed using ***client.subscribe(FEED_ID)***

- After subscribing to the feed, we have to check for the new value and store it into a ***payload*** variable. For this, the ***message*** function is called. So, whenever there is "1" in the payload variable, the led pin is set to '1' for HIGH and for '0' it is set to LOW.

- These values of '1' and '0' were set during the creation of the feed for the light. These are checked using *if statements* and are string values.

subscribe.py

```
16 def connected(client):
17     client.subscribe(FEED_ID)
18     print('Connected..')
19     print('Waiting for feed data...')
20 
21 def message(client, feed_id, payload):
22     print('Feed {0} received new value: {1}'.format(feed_id, payload))
23     if (payload == '1'):
24         GPIO.output(ledPin, GPIO.HIGH)
25         print ("High")
26     if (payload == '0'):
27         GPIO.output(ledPin, GPIO.LOW)
28         print ("Low")
29 
30 def disconnected(client):
31     print('Disconnected...')
32     sys.exit(1)
```

Fig. 7: Code- Part 2

- In the Adafruit IO dashboard when the switch is toggled, a payload is sent and the corresponding if statement is entered and executed.

- The bulb turns on when '1' is received and a "High" message is displayed and turned off when '0' is received and a "Low" message is displayed as can be seen in Fig.8.

Results

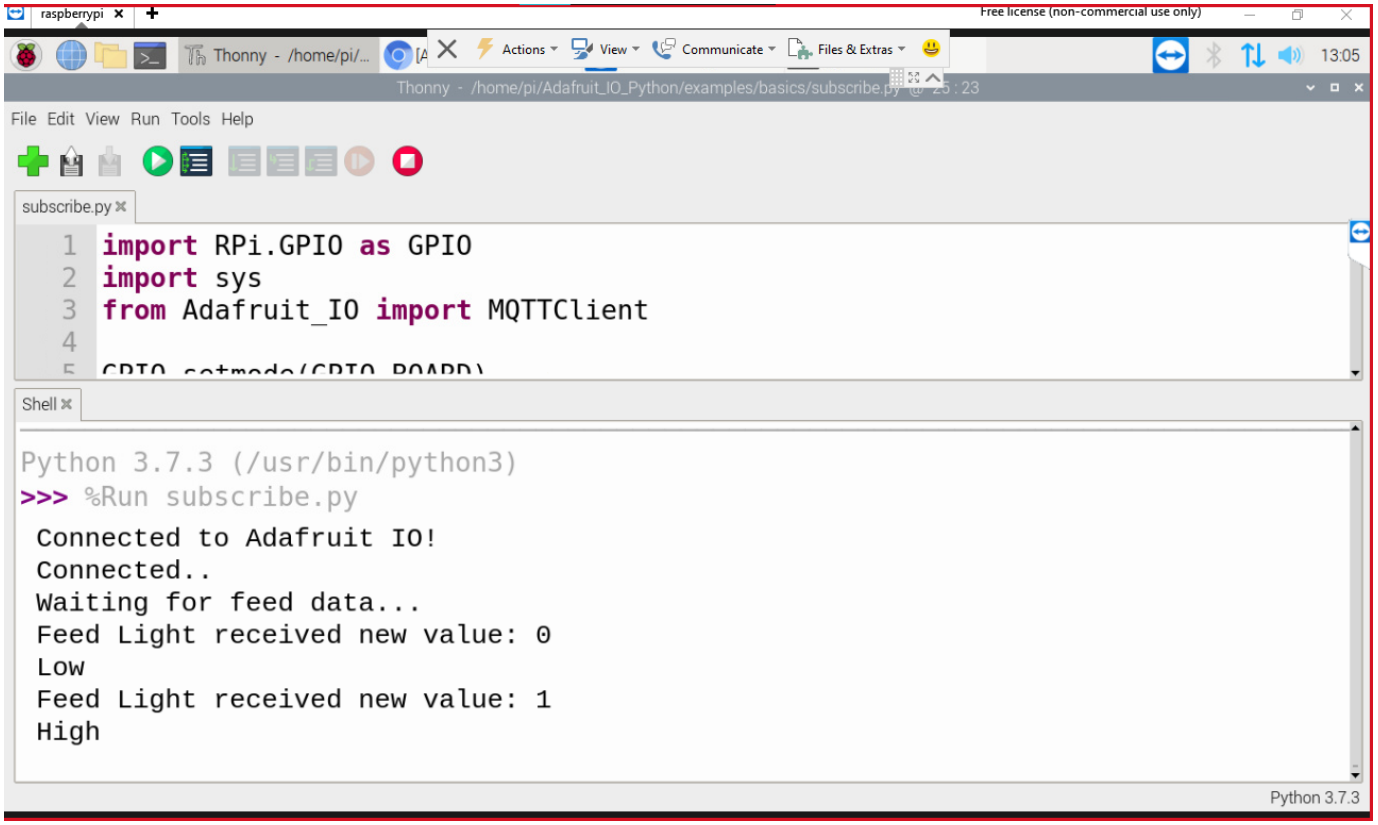

Fig. 8: Output Display

- It can be observed from the snippet above that upon running the code, connection to Adafruit IO was established and it waits for feed data.

- Upon toggling the button in the feed on the Adafruit IO Dashboard from anywhere in the world, the Feed Light receives a new value of either 1 or 0 depending upon whether it has been turned on or off.

- It was observed that upon receiving 1, the LED connected to the Raspberry Pi was turning ON and upon receiving 0 it was turning OFF appropriately. A picture of the lab setup for the LED connected to the Raspberry Pi module turning on and off upon receiving data from Adafruit IO h can be seen in Fig.9.

- In the same way, any home appliance can be connected to a Raspberry Pi through a Relay in between and can be controlled from anywhere in the world.

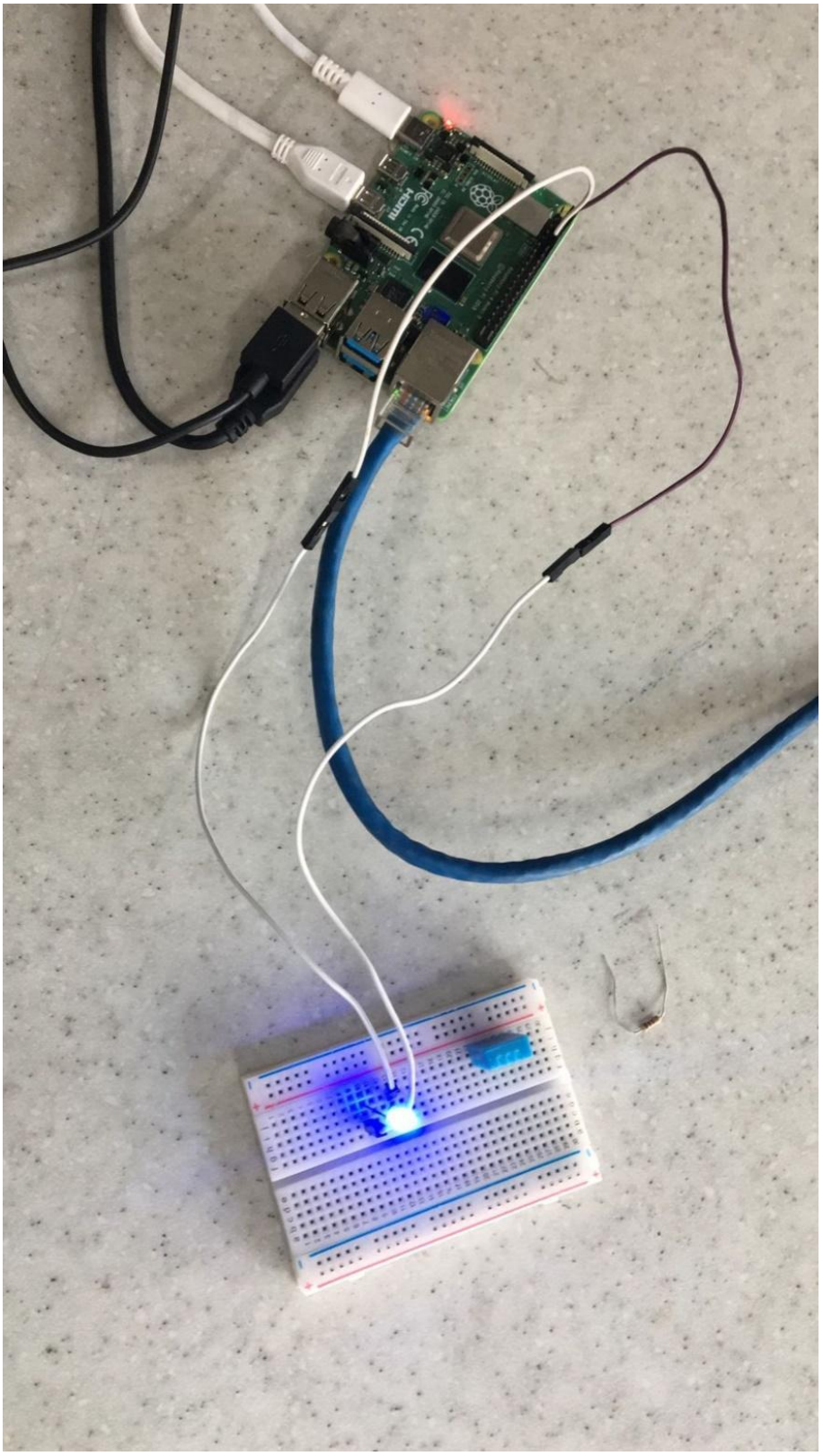

Fig. 9: LED connected to Raspberry Pi, glowing when the toggle button in Adafruit IO is positioned at ON

Acknowledgments

We would like to acknowledge the support that our Professor Rohit Singh, Lab Teaching Assistant Kiran G, and Lab personnel Vakil Khan have provided to us in completing this project. This project would not have been possible without their constant advice and guidance, especially due to the partially remote nature of the project.

They have helped us at any step we got stuck at and helped us resolve potential problems that fell on the way during the progress of the project.

We truly thank them for all that they have done for us.