Medication Tracker

Technical Answers For Real World Problems
Course Code - CSE-3999
Review - 2

Submitted by

Venuu Maadhav V R	16BCE2255
M Rana Simha Reddy	16BCE0475
Sai Ganesh	16BCE0161
N.Sriram	16BCB0053
D.V.S.Rahul	16BCB0057

Submitted to
VAIRAMUTHU S
Asst. Professor

Abstract

- People suffering from diseases like Alzheimer’s, Huntington’s disease, Cortico basal Degeneration etc often experience with memory loss and difficulty in remembering things.
- Such people and old age people often forget to take their regular medication which might affect their physical as well as mental health.
- A working model with the help of IFTTT and Adafruit Cloud helps patients keep track of their medicine consumption on daily basis and reminds them to take medicines on time and also refill the medicines on weekly basis via an Audio-Visual Stimuli.

Introduction

- This project is aimed at created a working model which can help people who often forget to take medicines on time.
- Prescriptions are accessible to treat these side effects, however the patients frequently neglect to take portions on the off chance that they are not reminded by a relative or guardian
- We will try to integrate Real-time reminders on a daily basis to maintain regularity and a weekly medicine refill reminder so that the patient never runs out of medicines.
- All the reminders are Audio Visual stimuli-based reminders (LEDs + Buzzer alarm). Email notifications are sent to the nearest guardian if the patient forgets to take his/her medicine

Proposed Methodology

- First Step, we created a cloud account in Adafruit.io
- We will be using Adafruit for storing the User Input feed.
- We have created an applet with IFTTT with the same credentials and also integrate the Arduino code with it.
- IFTTT will help in determining if the medication is taken or not.
- In response to that, Adafruit will store that information in the created cloud account.
- The reminders will send either as email or messages via Mobile application.
- If necessary, We can attach a GSM module to the device, so that we can generate a SMS message.

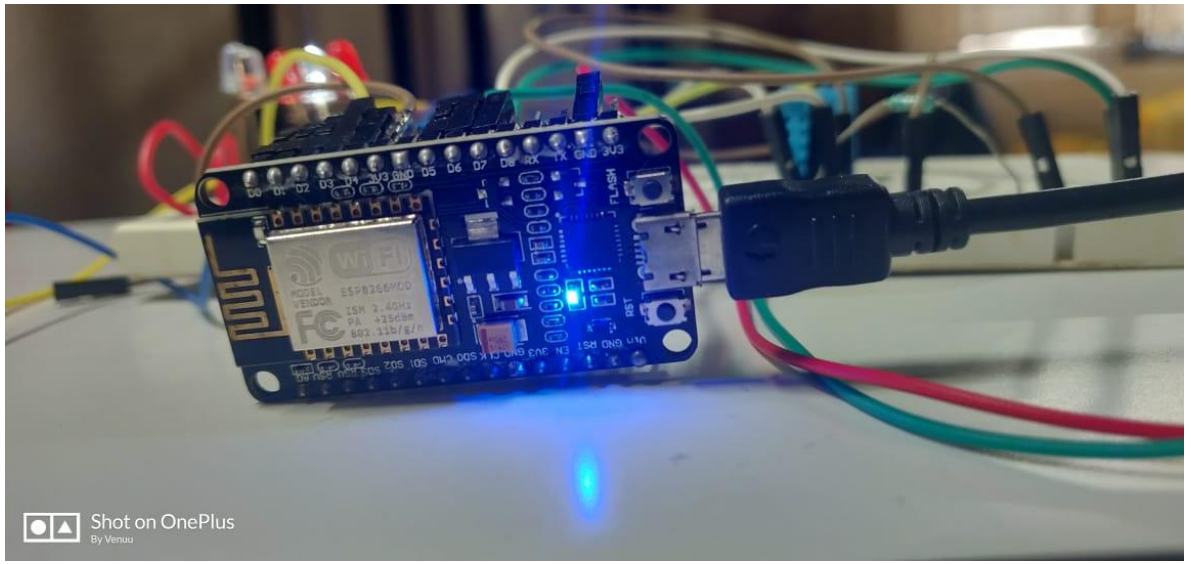
Hardware Components

- Bread Board
- Humidity Sensor
- Temperature Sensor
- Node MCU
- LED
- Buzzer
- Resistors
- Jumper wires

Software Requirements

- Arduino IDE
- Adafruit cloud account
- Mobile Installed with IFTT App

Node MCU

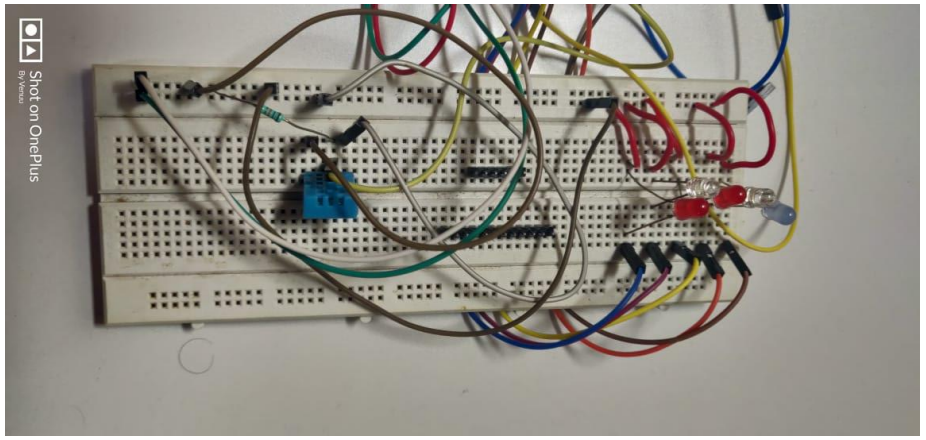


- Node MCU is an open source based firmware developed for ESP8266 WIFI chip.
- By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board.
- Node MCU has inbuilt WIFI Module when compared to Ardiuno and Raspberrypi. So we prefered to use NodeMCU.

Buzzer

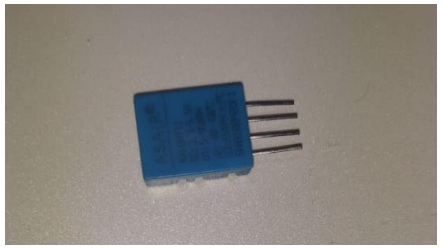
- A buzzer or beeper is a signaling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows.
- If the medicine hasn’t been taken at the respective time the Buzzer will beep.

Bread Board



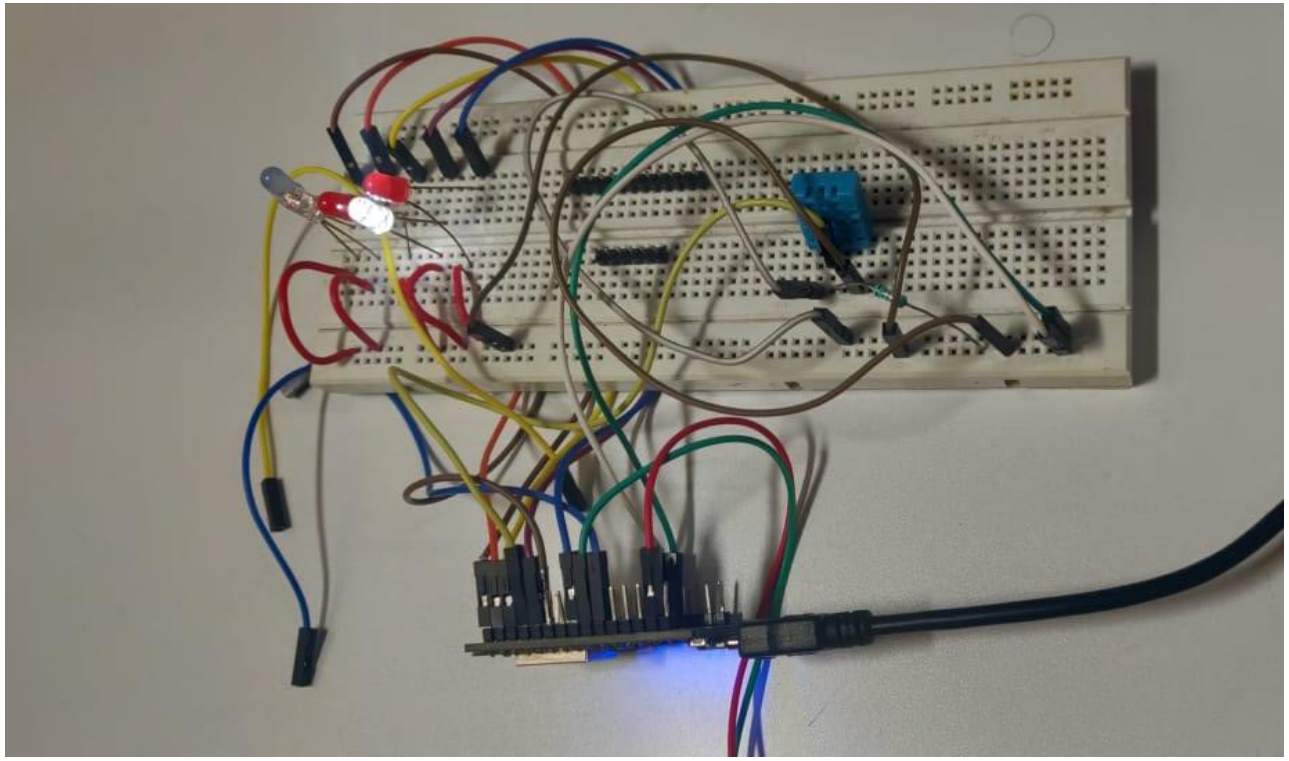
- A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.
- Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.
- LED’S are used to represent which medicine has to be taken by he patient on the particular day

DHT11 Sensor

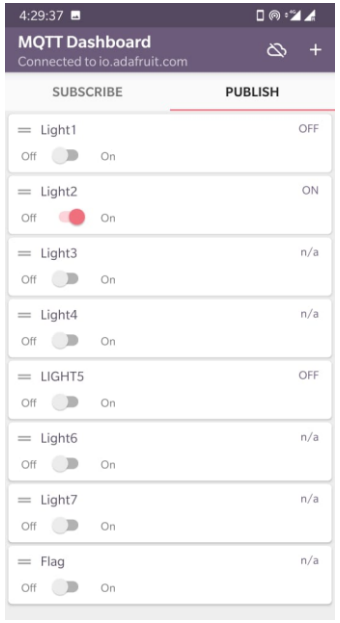


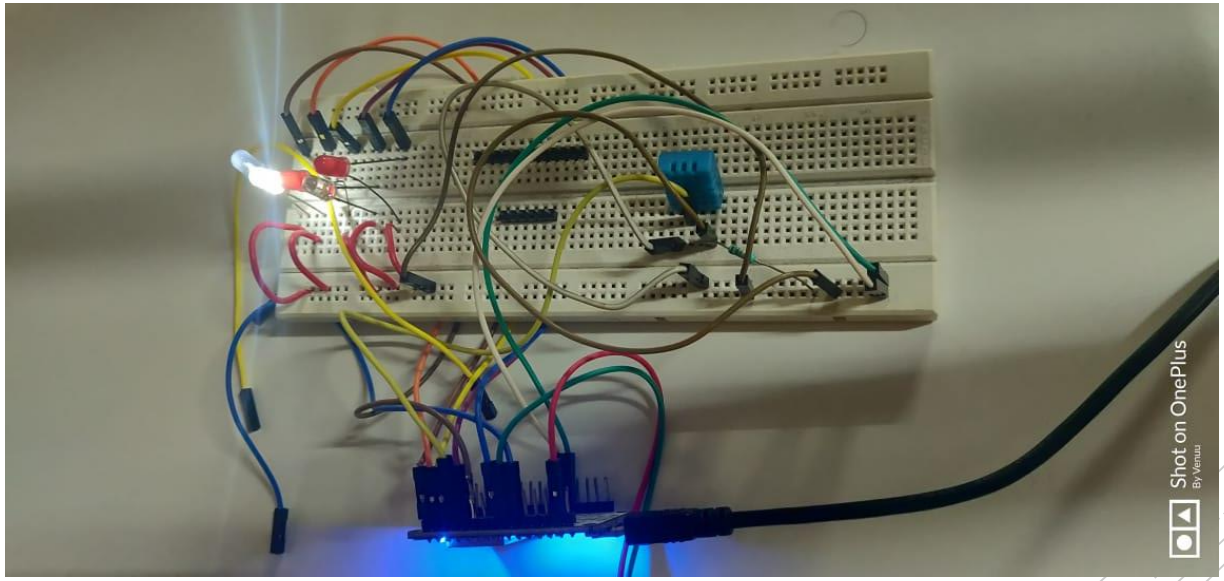
- This DHT11 Temperature and Humidity Sensor features a calibrated digital signal output with the temperature and humidity sensor capability.
- It is integrated with a high-performance 8-bit microcontroller. Its technology ensures the high reliability and excellent long-term stability.
- This sensor includes a resistive element and a sensor for wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high performance.
- Tablets must be stored in a particular temperature so we will monitor the temperature and humidity around them

Circuit Diagram:

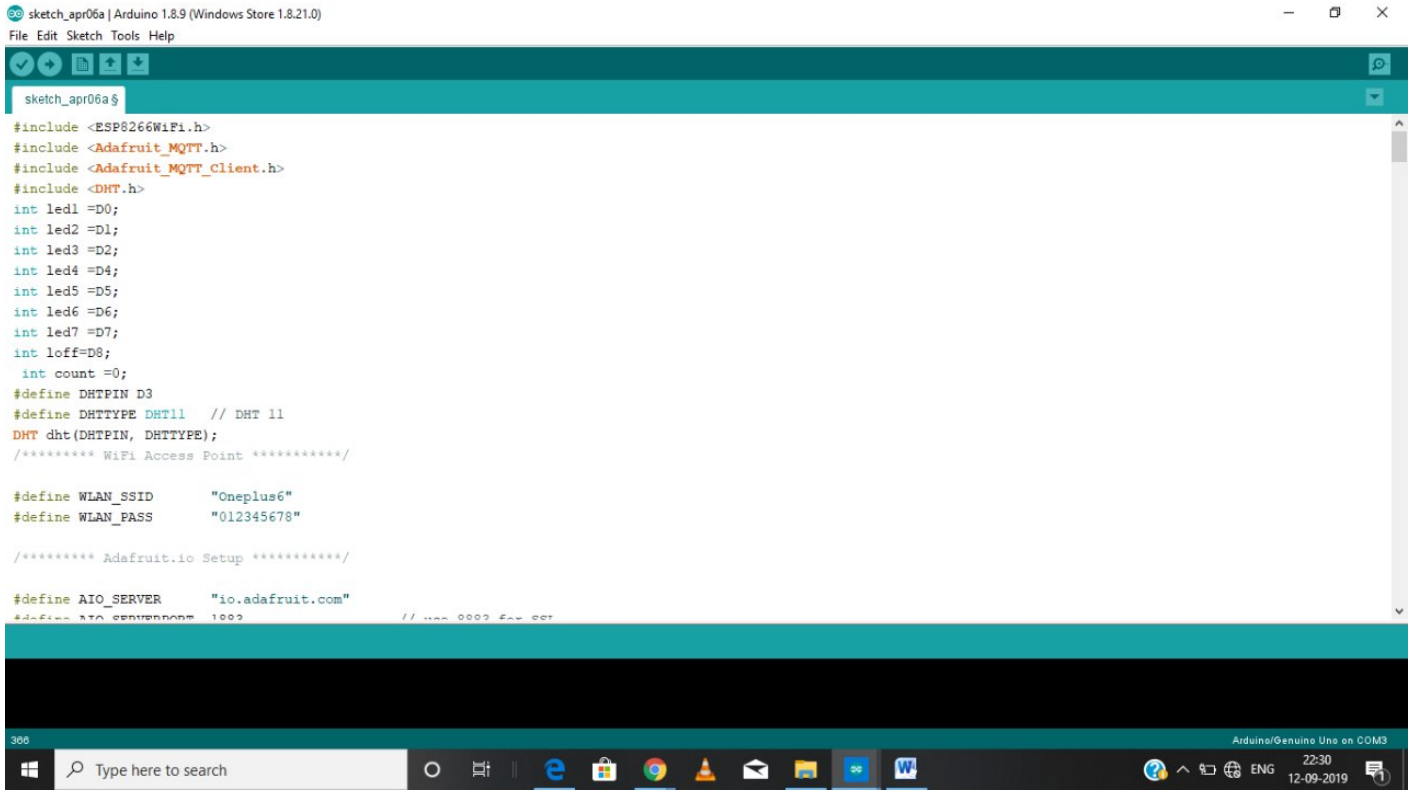


Output (Screen Shots)

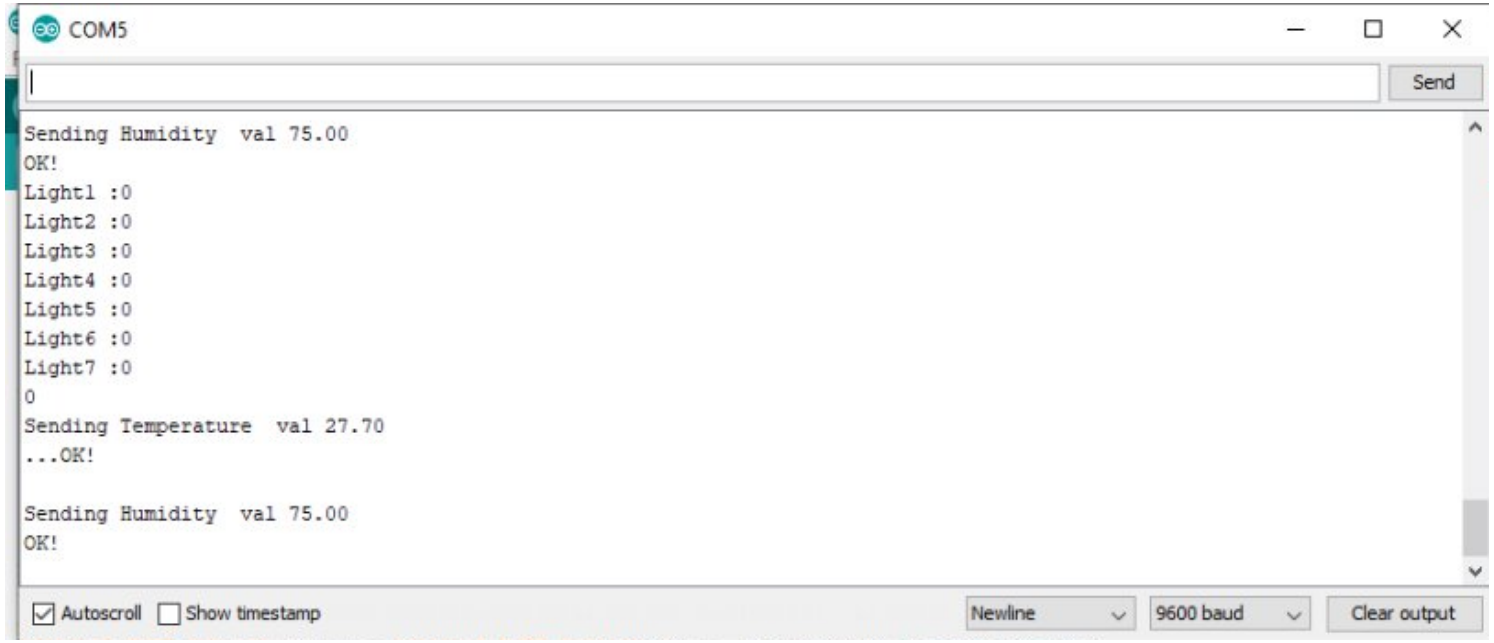


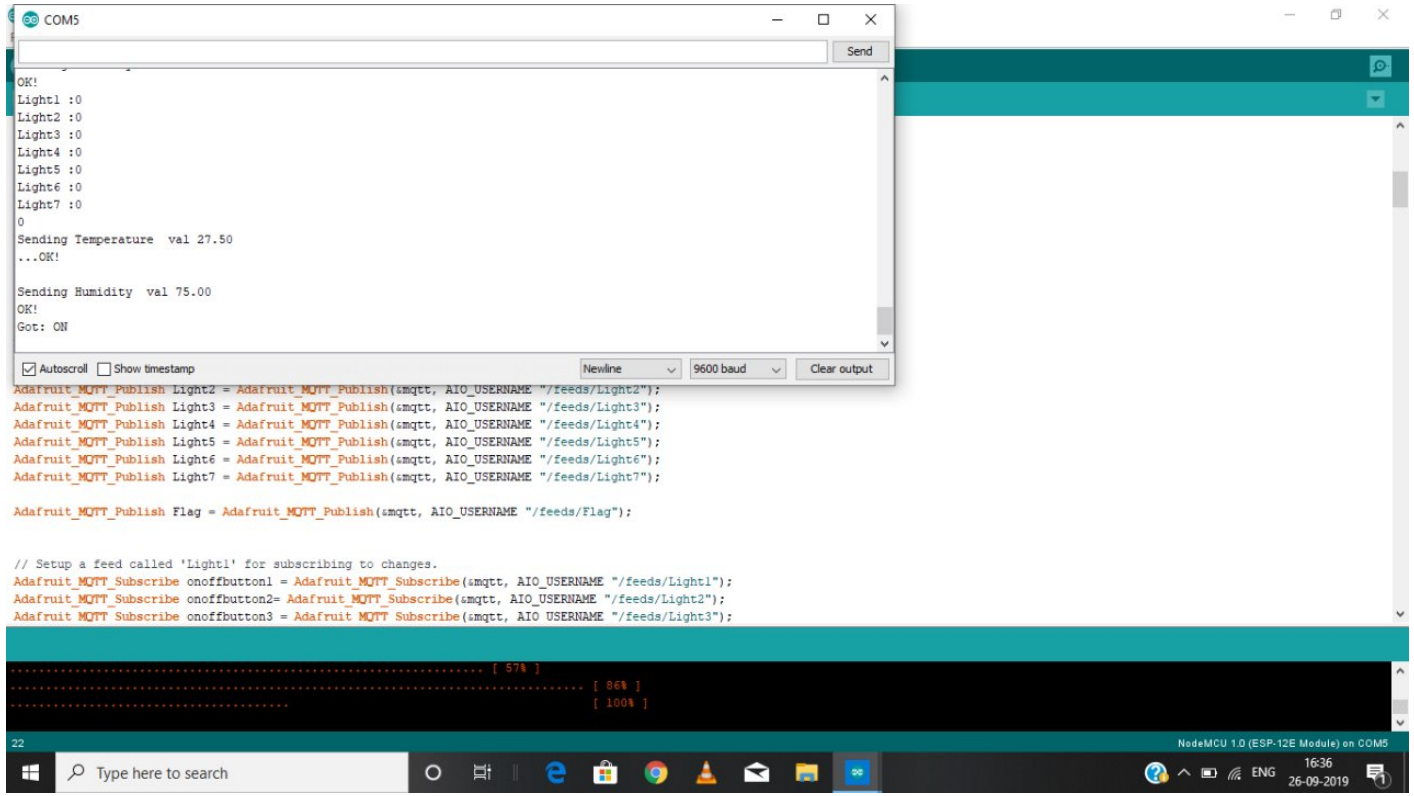


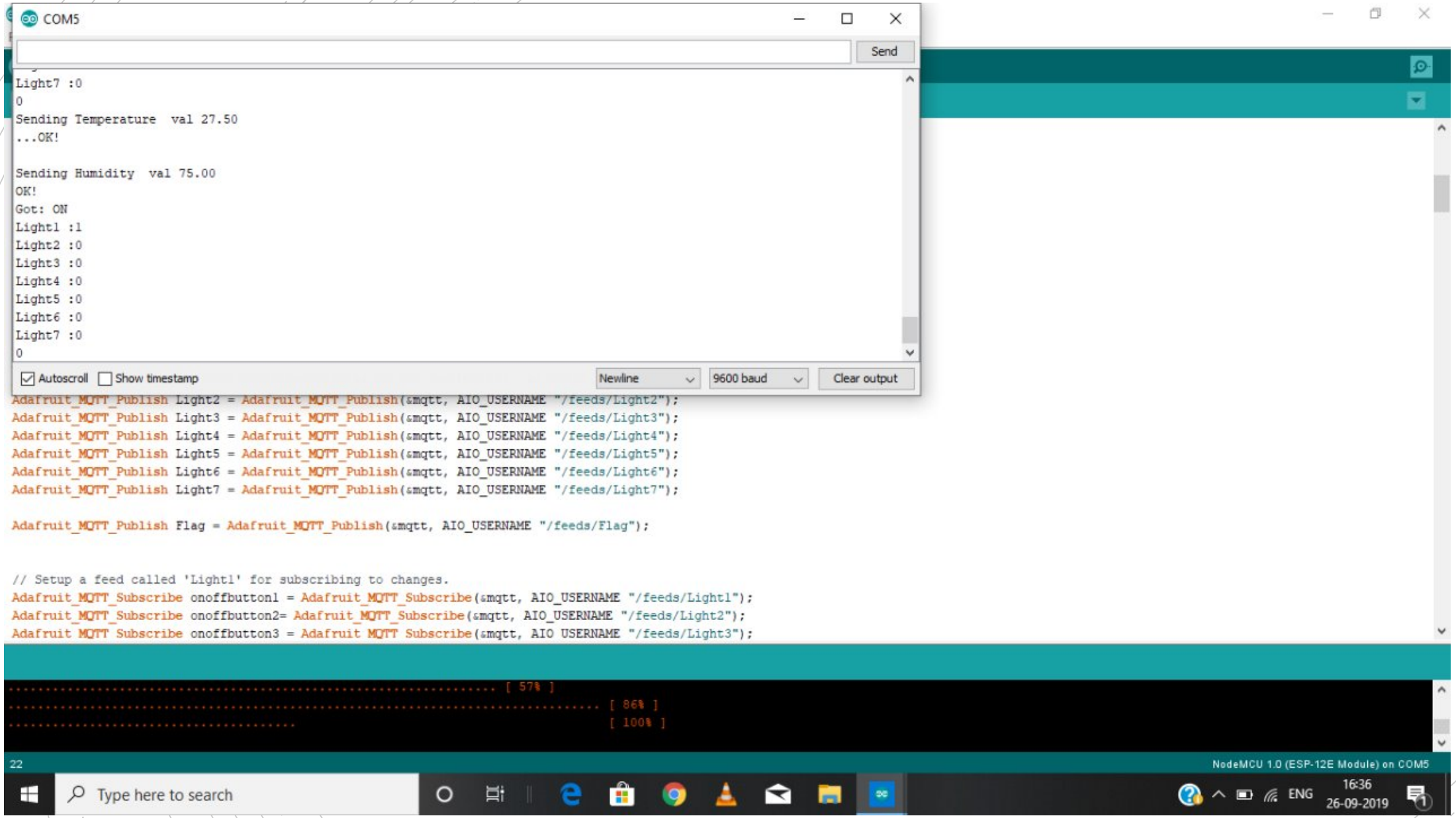
Code:

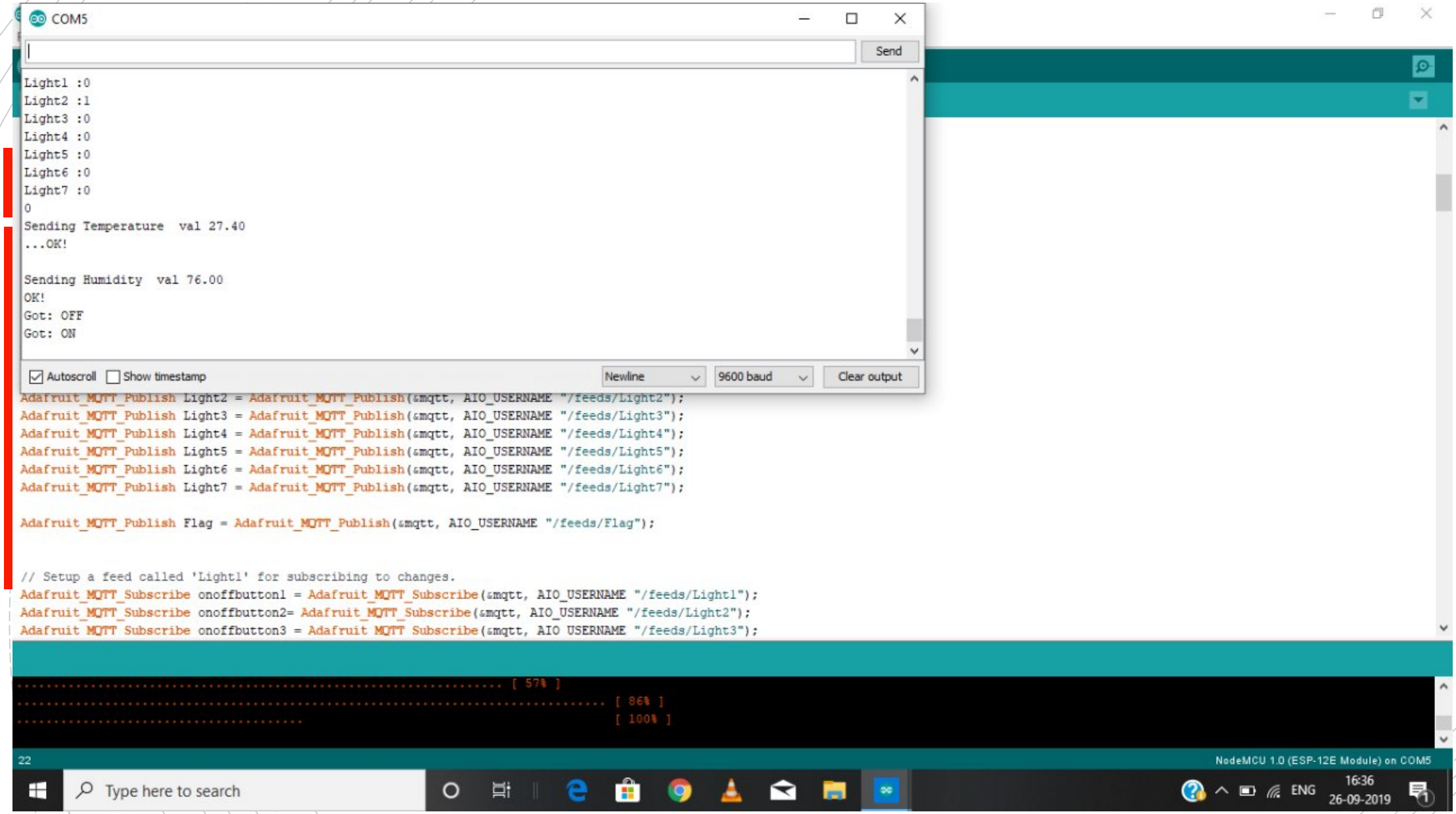

sketch_apr06a | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
sketch_apr06a §
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <DHT.h>
int led1 =D0;
int led2 =D1;
int led3 =D2;
int led4 =D4;
int led5 =D5;
int led6 =D6;
int led7 =D7;
int loff=D8;
int count =0;
#define DHTPIN D3
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
/********* WiFi Access Point **********/
#define WLAN_SSID "Oneplus6"
#define WLAN_PASS "012345678"
/********* Adafruit.io Setup **********/
#define AIO_SERVER "io.adafruit.com"
366
Arduino/Genuino Uno on COM3
Type here to search
ENG
22:30
12-09-2019

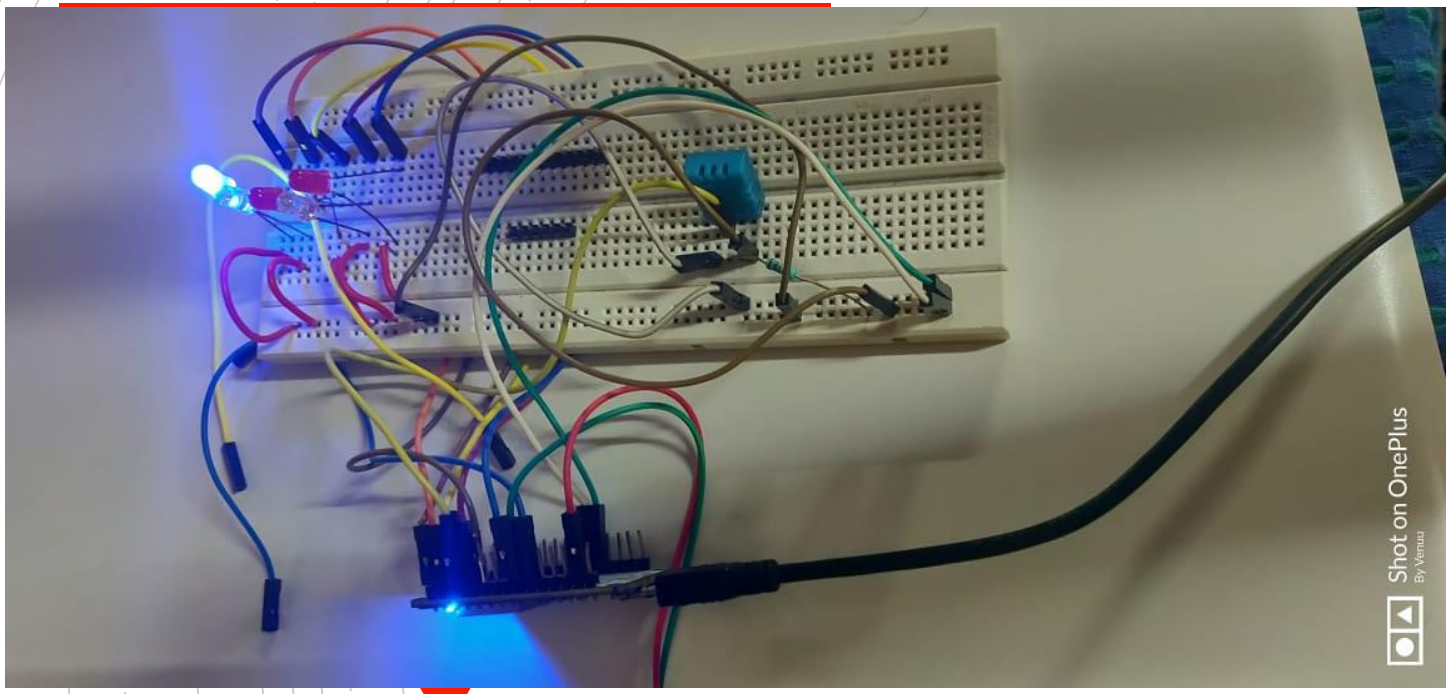
Temperature & Humidity

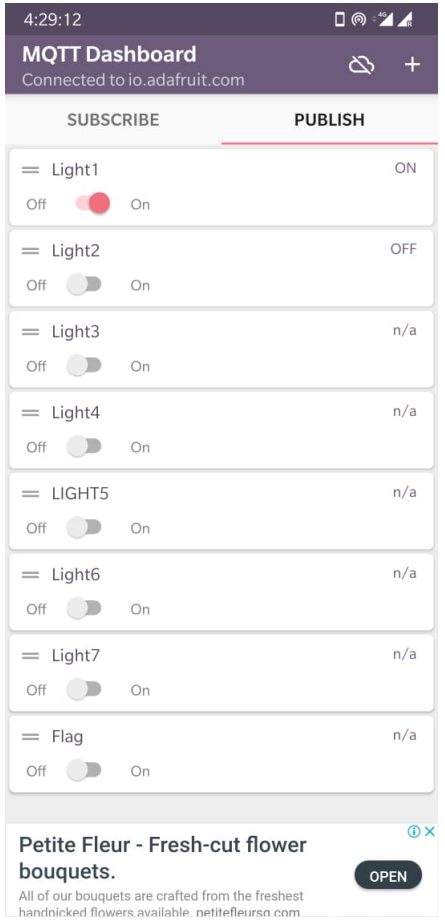



COM5
Send
OK!
Light1 :0
Light2 :0
Light3 :0
Light4 :0
Light5 :0
Light6 :0
Light7 :0
0
Sending Temperature val 27.50
...OK!
Sending Humidity val 75.00
OK!
Got: ON
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Adafruit_MQTT_Publish Light2 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Publish Light3 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light3");
Adafruit_MQTT_Publish Light4 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light4");
Adafruit_MQTT_Publish Light5 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light5");
Adafruit_MQTT_Publish Light6 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light6");
Adafruit_MQTT_Publish Light7 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light7");
Adafruit_MQTT_Publish Flag = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Flag");
// Setup a feed called 'Light1' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light1");
Adafruit_MQTT_Subscribe onoffbutton2= Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Subscribe onoffbutton3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light3");
[57%]
[86%]
[100%]
22
NodeMCU 1.0 (ESP-12E Module) on COM5
Type here to search
ENG
16:36
26-09-2019

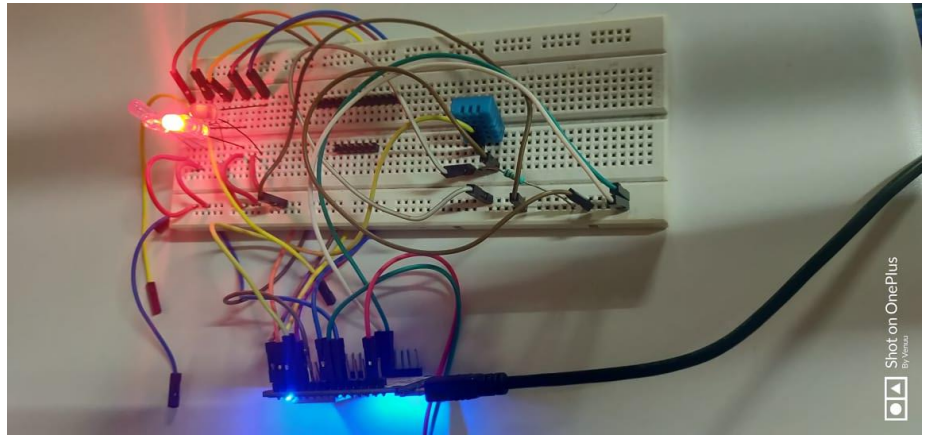

COM5
Send
Light7 :0
0
Sending Temperature val 27.50
...OK!
Sending Humidity val 75.00
OK!
Got: ON
Light1 :1
Light2 :0
Light3 :0
Light4 :0
Light5 :0
Light6 :0
Light7 :0
0
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Adafruit_MQTT_Publish Light2 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Publish Light3 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light3");
Adafruit_MQTT_Publish Light4 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light4");
Adafruit_MQTT_Publish Light5 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light5");
Adafruit_MQTT_Publish Light6 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light6");
Adafruit_MQTT_Publish Light7 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light7");
Adafruit_MQTT_Publish Flag = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Flag");
// Setup a feed called 'Light1' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light1");
Adafruit_MQTT_Subscribe onoffbutton2= Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Subscribe onoffbutton3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light3");
[57%]
[86%]
[100%]
22
NodeMCU 1.0 (ESP-12E Module) on COM5
Type here to search
ENG
16:36
26-09-2019


COM5
Send
Light1 :0
Light2 :1
Light3 :0
Light4 :0
Light5 :0
Light6 :0
Light7 :0
0
Sending Temperature val 27.40
...OK!
Sending Humidity val 76.00
OK!
Got: OFF
Got: ON
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Adafruit_MQTT_Publish Light2 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Publish Light3 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light3");
Adafruit_MQTT_Publish Light4 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light4");
Adafruit_MQTT_Publish Light5 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light5");
Adafruit_MQTT_Publish Light6 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light6");
Adafruit_MQTT_Publish Light7 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Light7");
Adafruit_MQTT_Publish Flag = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Flag");
// Setup a feed called 'Light1' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light1");
Adafruit_MQTT_Subscribe onoffbutton2= Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light2");
Adafruit_MQTT_Subscribe onoffbutton3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Light3");
[57%]
[86%]
[100%]
22
NodeMCU 1.0 (ESP-12E Module) on COM5
Type here to search
ENG
16:36
26-09-2019

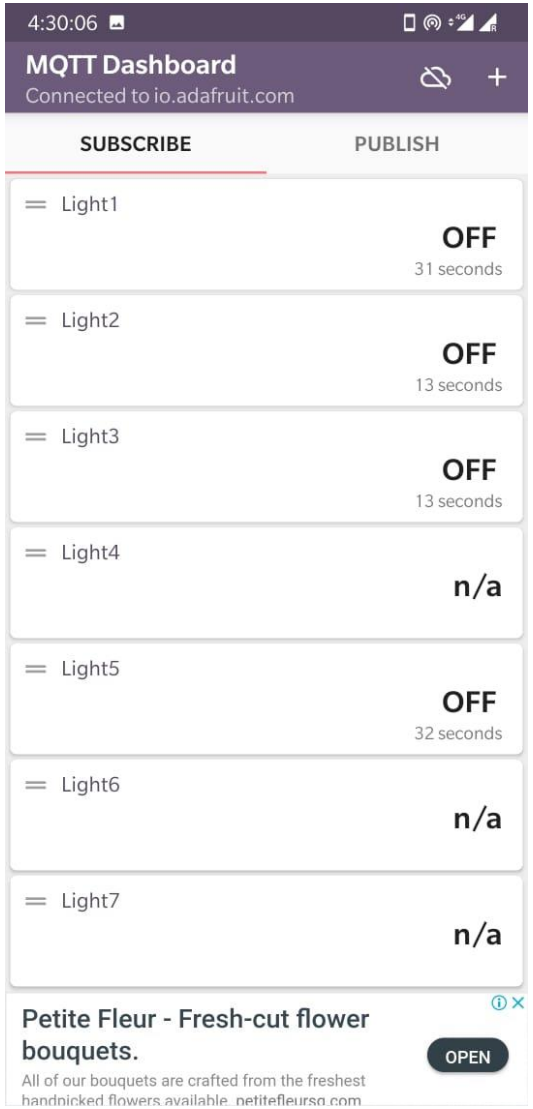

Shot on OnePlus
By Vivo


4:29:12
MQTT Dashboard
Connected to io.adafruit.com
SUBSCRIBE
PUBLISH
Light1
ON
Off
On
Light2
OFF
Off
On
Light3
n/a
Off
On
Light4
n/a
Off
On
LIGHT5
n/a
Off
On
Light6
n/a
Off
On
Light7
n/a
Off
On
Flag
n/a
Off
On
Petite Fleur - Fresh-cut flower bouquets.
All of our bouquets are crafted from the freshest handpicked flowers available. petitefleursg.com
OPEN


4:29:52
MQTT Dashboard
Connected to io.adafruit.com
SUBSCRIBE
PUBLISH
Light1
OFF
Off
On
Light2
OFF
Off
On
Light3
ON
Off
On
Light4
n/a
Off
On
LIGHT5
OFF
Off
On
Light6
n/a
Off
On
Light7
n/a
Off
On
Flag
n/a
Off
On


Shot on OnePlus

Adafruit DashBoard



Future Work

- In the next Review we will make an applet and integrate it to the Adafruit cloud.
- So that IFTT applet sends email notification to the care taker if the patient doesn’t take the Tablet’s on time.

THANK YOU