

/*

Experiment No. : 16

Statement : MQTT protocol with ESP8266 Witty Cloud
Development Board and Adafruit IO.

Date of Exp. : xx/xx/xxxx

Author : Mansi Mandhane (A-24)

*/

#include <Adafruit_BusIO_Register.h>

#include <Adafruit_I2CDevice.h>

#include <Adafruit_I2CRegister.h>

#include <Adafruit_SPIDevice.h>

#include <Adafruit_Sensor.h>

#include <ESP8266WiFi.h> // library file for ESP8266

#include "Adafruit_MQTT.h" // library included through
Adafruit IO Arduino

#include "Adafruit_MQTT_Client.h" // library included through
Adafruit IO Arduino

// pinout for wittyBoard

#define led 2 // debug LED, tiny blue

#define red 15 // RGB LED red

#define green 12 // RGB LED green

#define blue 13 // RGB LED blue

#define ldr A0 // LDR

#define WLAN_SSID "Mansi's S22"

#define WLAN_PASS "mansil702"

```

#define AIO_SERVER      "io.adafruit.com"

#define AIO_SERVERPORT  1883    // mqtt: 1883, secure-mqtt: 8883

#define AIO_USERNAME    "mansil702"

#define AIO_KEY          "aio_FpZr08AVJpmbY0GZ6yD6gVlMc8E9"

WiFiClient client;        // declare client

Adafruit_MQTT_Client  mqtt(&client,  AIO_SERVER,  AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);    // declare MQTT client

Adafruit_MQTT_Publish  lightintensity  =  Adafruit_MQTT_Publish(
&mqtt, AIO_USERNAME "/feeds/lux-meter");    // declare publisher

Adafruit_MQTT_Subscribe redbutton=Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/red");    // declare subscriber

Adafruit_MQTT_Subscribe greenbutton =
Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/green");    //
declare subscriber

Adafruit_MQTT_Subscribe bluebutton =
Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/blue");
// declare subscriber

void MQTT_connect(); // bug fixes

void setup() {

// put your setup code here, to run once:

pinMode(led, OUTPUT);

pinMode(red, OUTPUT);

pinMode(green, OUTPUT);

pinMode(blue, OUTPUT);

Serial.begin(115200);

delay(10);

```

```

Serial.println(F("Adafruit MQTT demo"));

// Connect to WiFi access point.

Serial.println();

Serial.print("Connecting to ");

Serial.println(WLAN_SSID);

WiFi.begin(WLAN_SSID, WLAN_PASS);

while (WiFi.status() != WL_CONNECTED) {

  delay(500);

  Serial.print(".");

}

Serial.println();

Serial.println("WiFi connected");

Serial.println("IP address: "); Serial.println(WiFi.localIP());

// Setup MQTT subscription for onoff feed.

mqtt.subscribe(&redbutton);

mqtt.subscribe(&greenbutton);

mqtt.subscribe(&bluebutton);

}

void loop() {

  // put your main code here, to run repeatedly:

  MQTT_connect();

  Adafruit_MQTT_Subscribe *subscription;

  while ((subscription = mqtt.readSubscription(5000))) {

```

```
if (subscription == &redbutton) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)redbutton.lastread);  
    if(strcmp((char*)redbutton.lastread, "ON"))  
        digitalWrite(red, LOW);  
    else  
        digitalWrite(red, HIGH);  
}  
  
if (subscription == &greenbutton) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)greenbutton.lastread);  
    if(strcmp((char*)greenbutton.lastread, "ON"))  
        digitalWrite(green, LOW);  
    else  
        digitalWrite(green, HIGH);  
}  
  
if (subscription == &bluebutton) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)bluebutton.lastread);  
    if(strcmp((char*)bluebutton.lastread, "ON"))  
        digitalWrite(blue, LOW);  
    else  
        digitalWrite(blue, HIGH);  
}
```

```

}

}Serial.print(F("\nSending light val "));

Serial.print(analogRead(ldr));

Serial.print("...");

if (! lightintensity.publish(analogRead(ldr)))

Serial.println(F("Failed"));

else

Serial.println(F("OK!"));

}

// Function to connect and reconnect as necessary to the MQTT
server.

void MQTT_connect() {

int8_t ret;

// Stop if already connected.

if (mqtt.connected()) {

return;

}

Serial.print("Connecting to MQTT... ");

uint8_t retries = 3;

while ((ret = mqtt.connect()) != 0) { // connect will return 0
for connected

Serial.println(mqtt.connectErrorString(ret));

Serial.println("Retrying MQTT connection in 5 seconds...");

mqtt.disconnect();

```

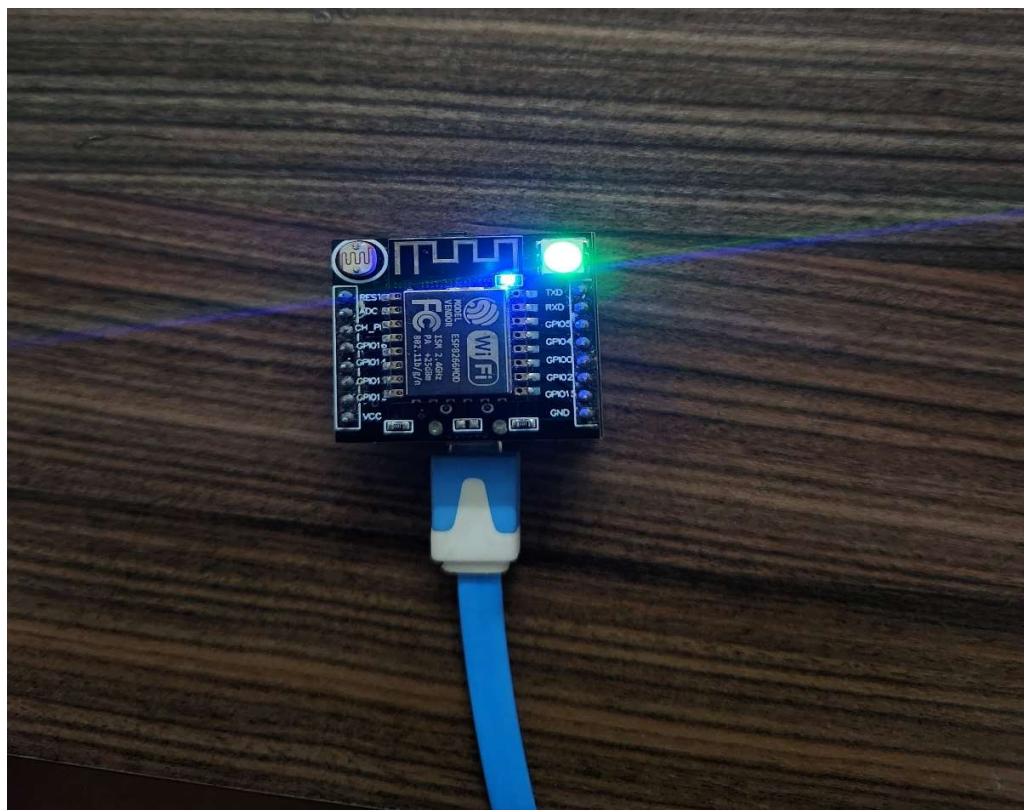
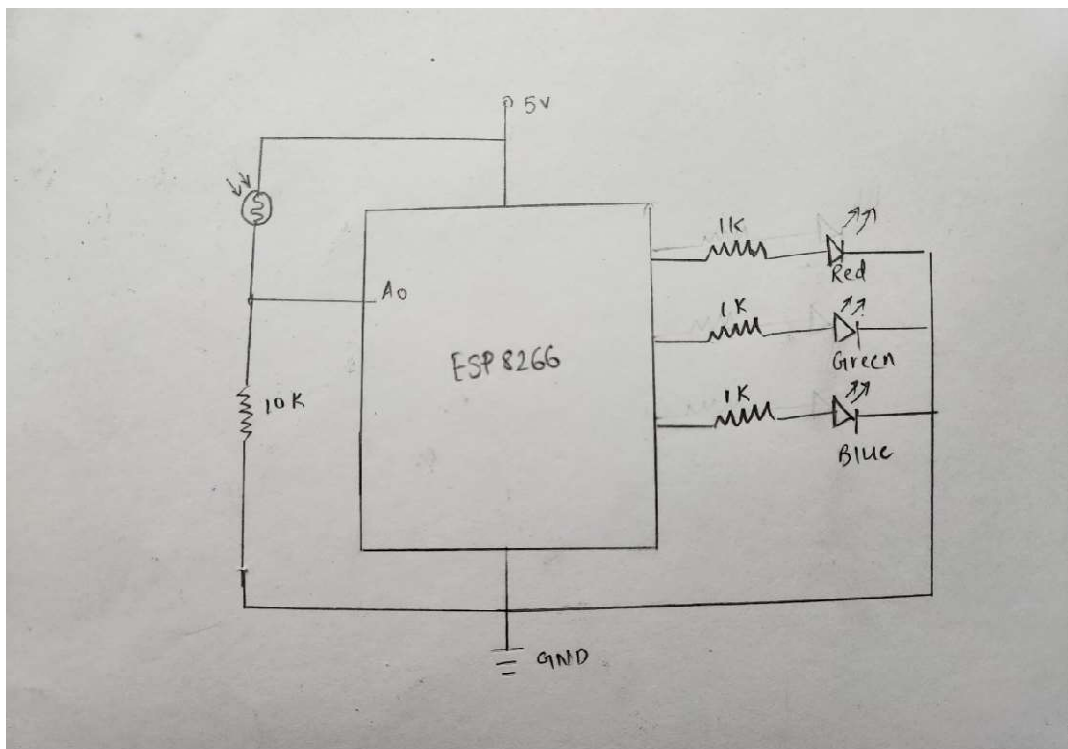
```
delay(5000); // wait 5 seconds

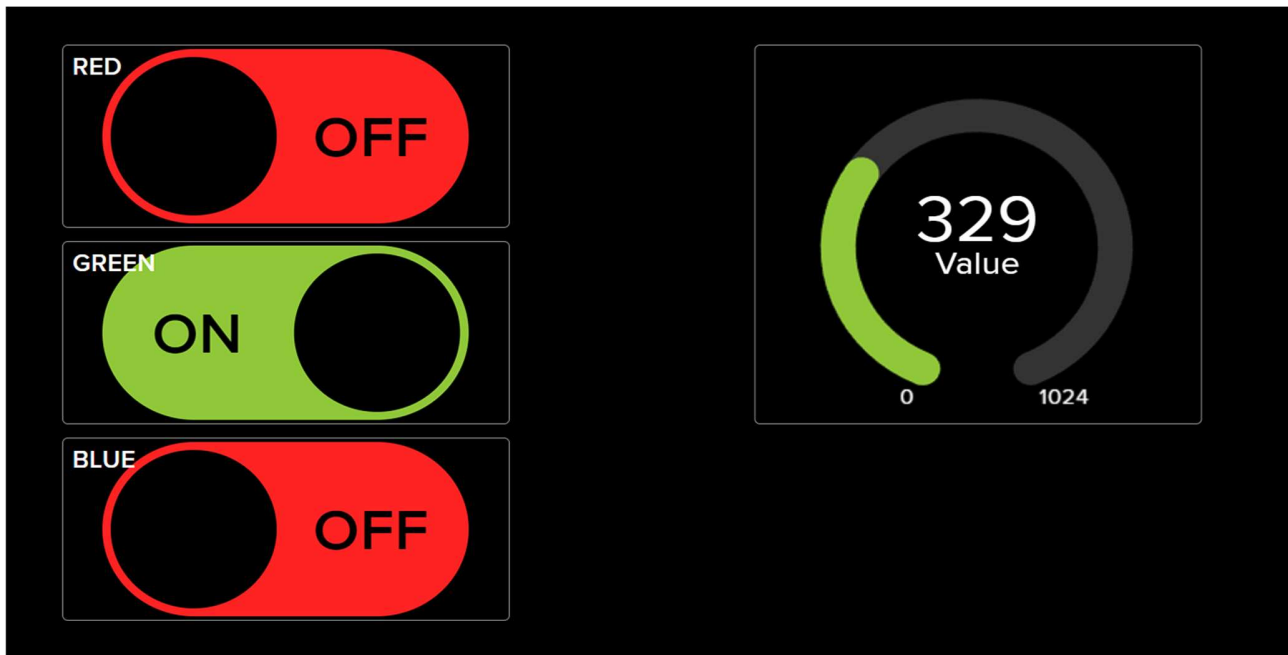
retries--;

if (retries == 0) {
  // basically die and wait for WDT to reset me
  while (1);
}

}

Serial.println("MQTT Connected!");
}
```





```
112 1
Output Serial Monitor x
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM5')
17:59:06.836 -> ....
17:59:08.311 -> WiFi connected
17:59:08.311 -> IP address:
17:59:08.311 -> 192.168.150.51
17:59:08.311 -> Connecting to MQTT... MQTT Connected!
17:59:15.151 ->
17:59:15.151 -> Sending light val 84...OK!
17:59:20.196 ->
17:59:20.196 -> Sending light val 89...OK!
```