Assignment: 3

Company: Smart-Internz

Domain: Internet of Things

Date: 2-6-2023

Name: H. Shyam

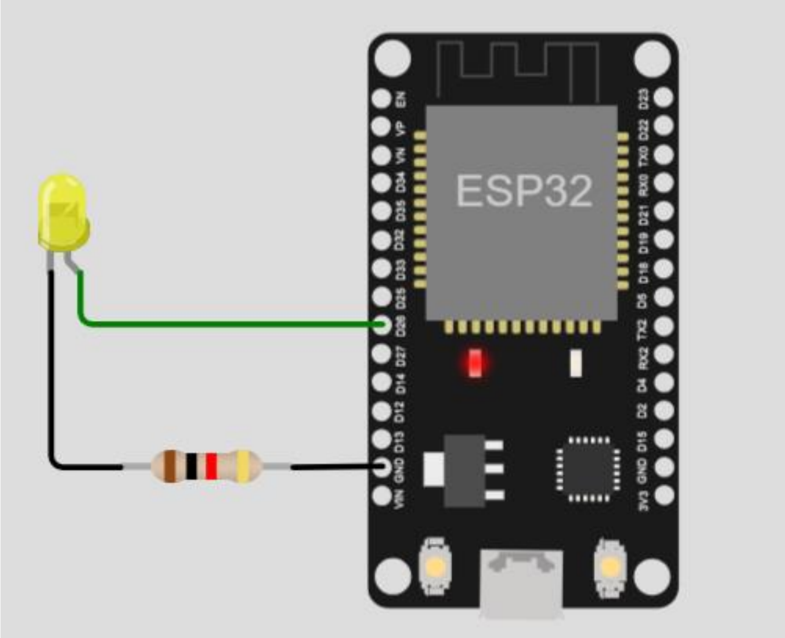
Reg. no.: 20BEC1055

Task: In Wokwi simulator, add an LED & switch it ON & OFF from Node-Red.

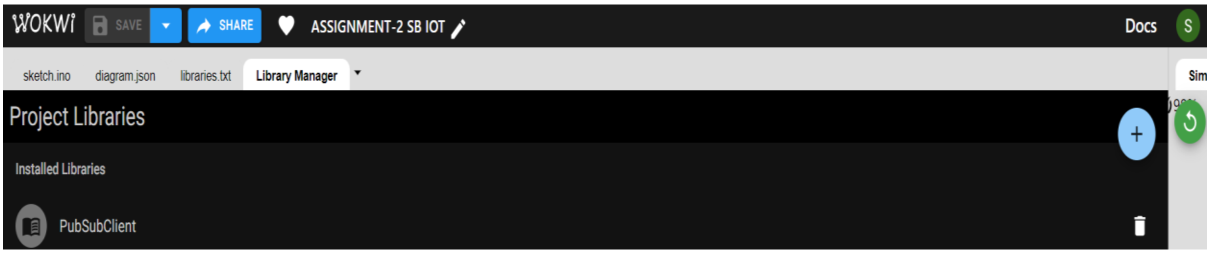
Wokwi Link:

https://wokwi.com/projects/366776415060669441

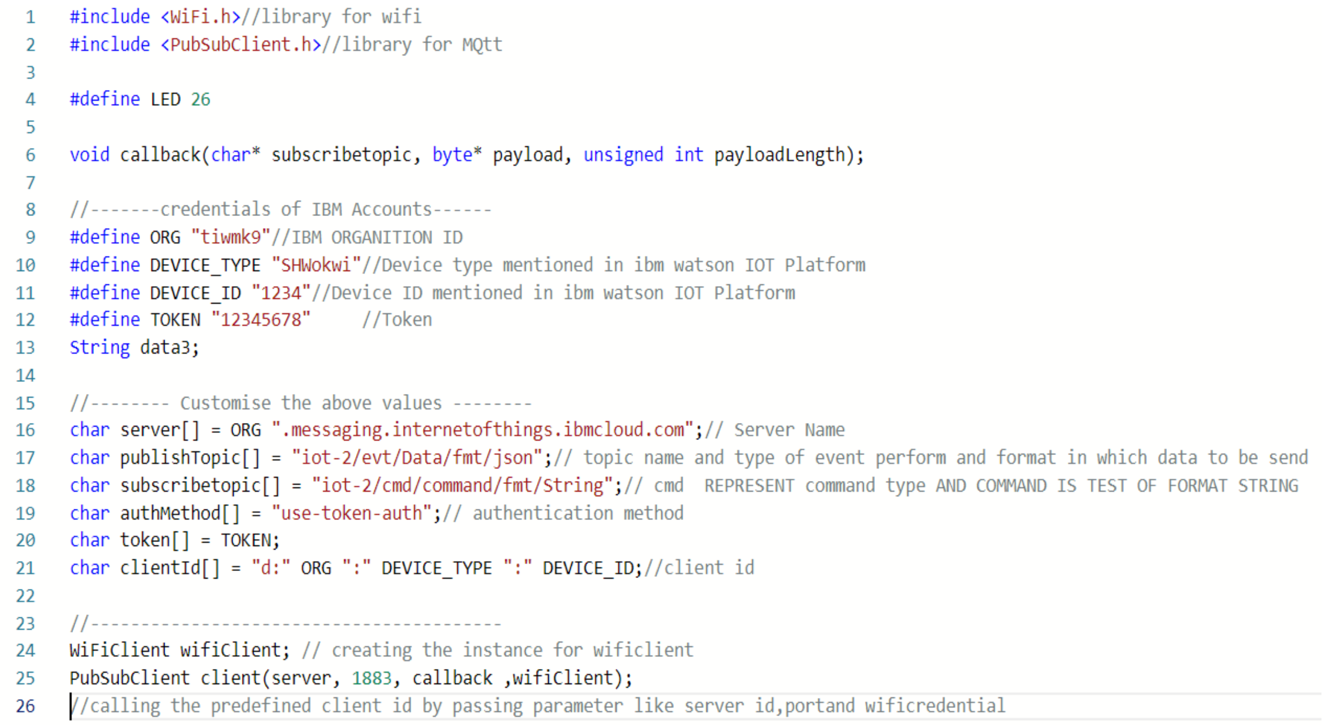
Circuit:



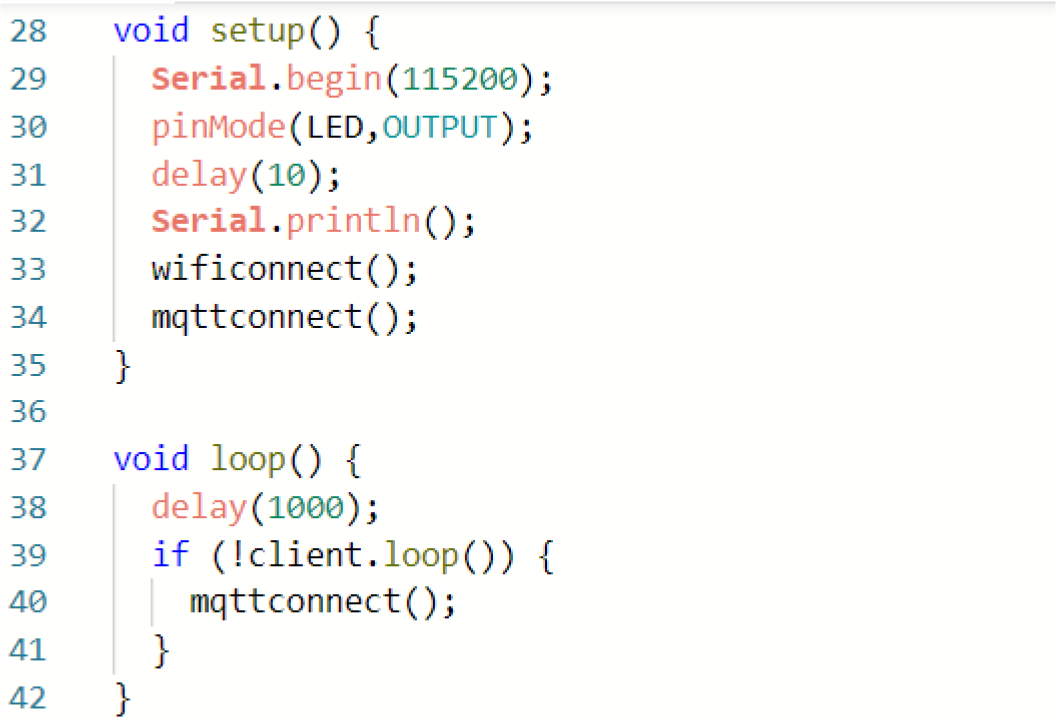
Add the PubSubClient library to the Library manager pane & run the code below



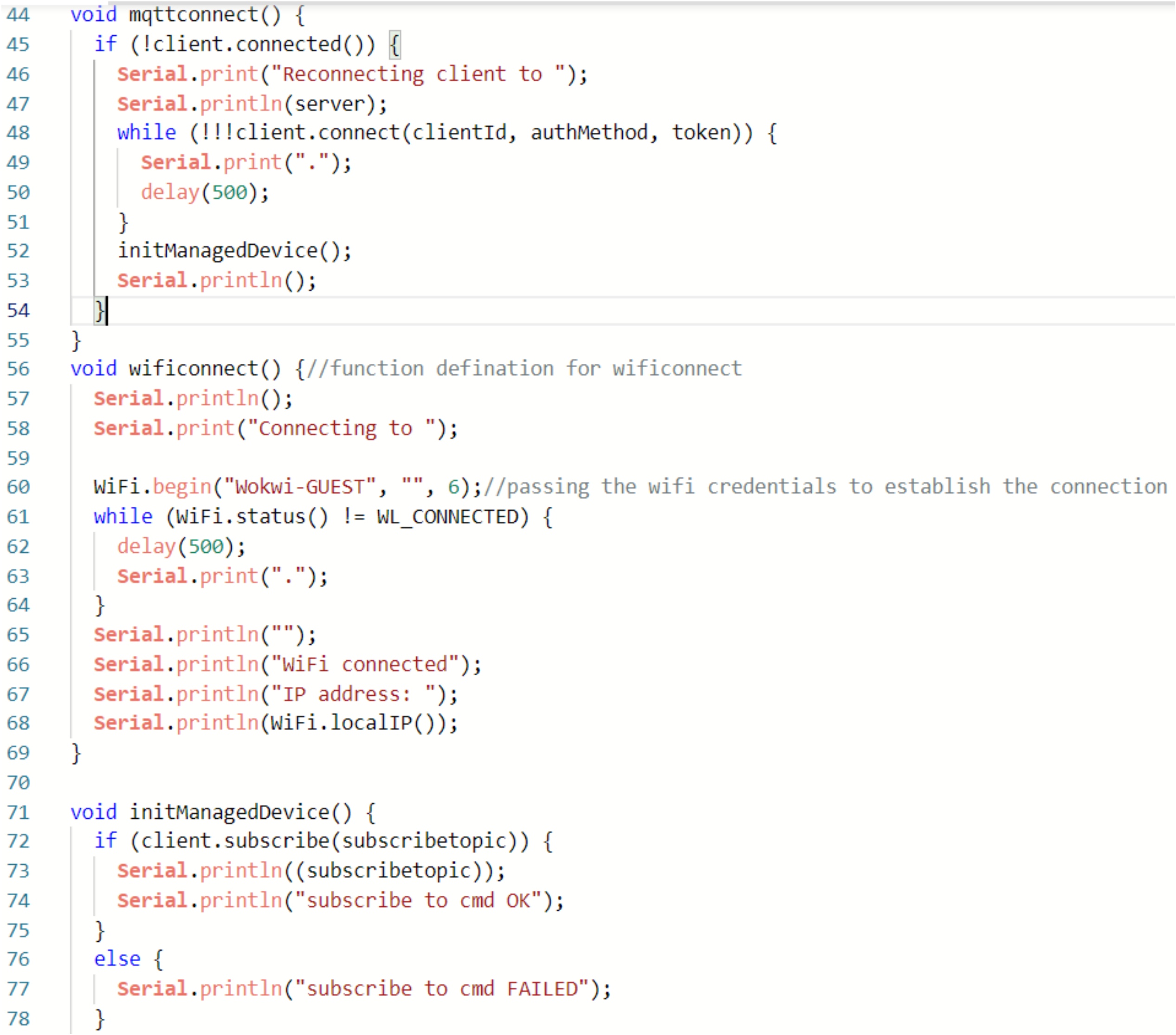
CODE:



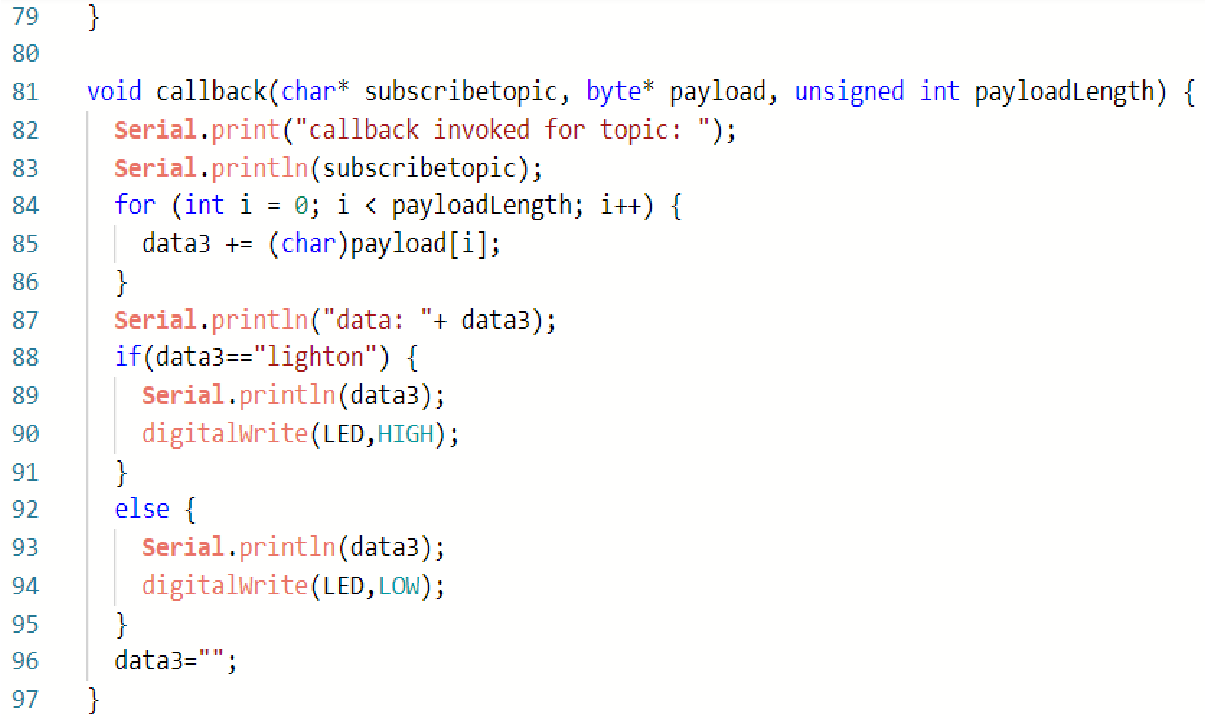
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

#define LED 26

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------
#define ORG "tiwmk9"//IBM ORGANITION ID
#define DEVICE_TYPE "SHWokwi"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient);
//calling the predefined client id by passing parameter like server id,portand wificredential
```



```
void setup() {
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop() {
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
```

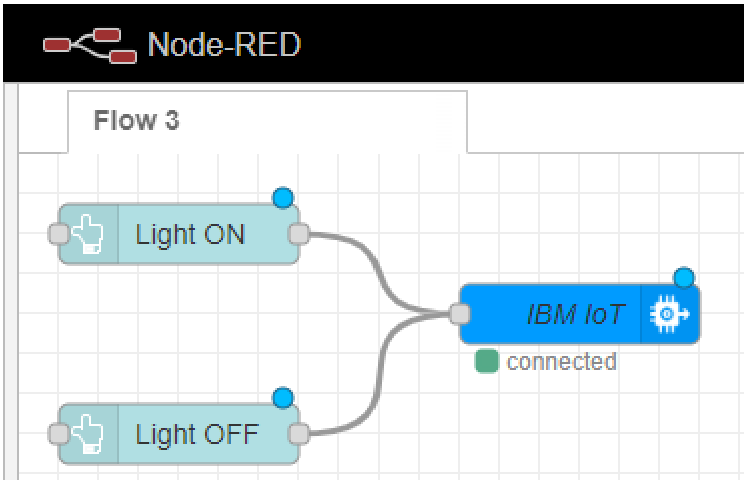


```
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}
void wificonnect() {//function defination for wificonnect
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  }
  else {
    Serial.println("subscribe to cmd FAILED");
  }
```

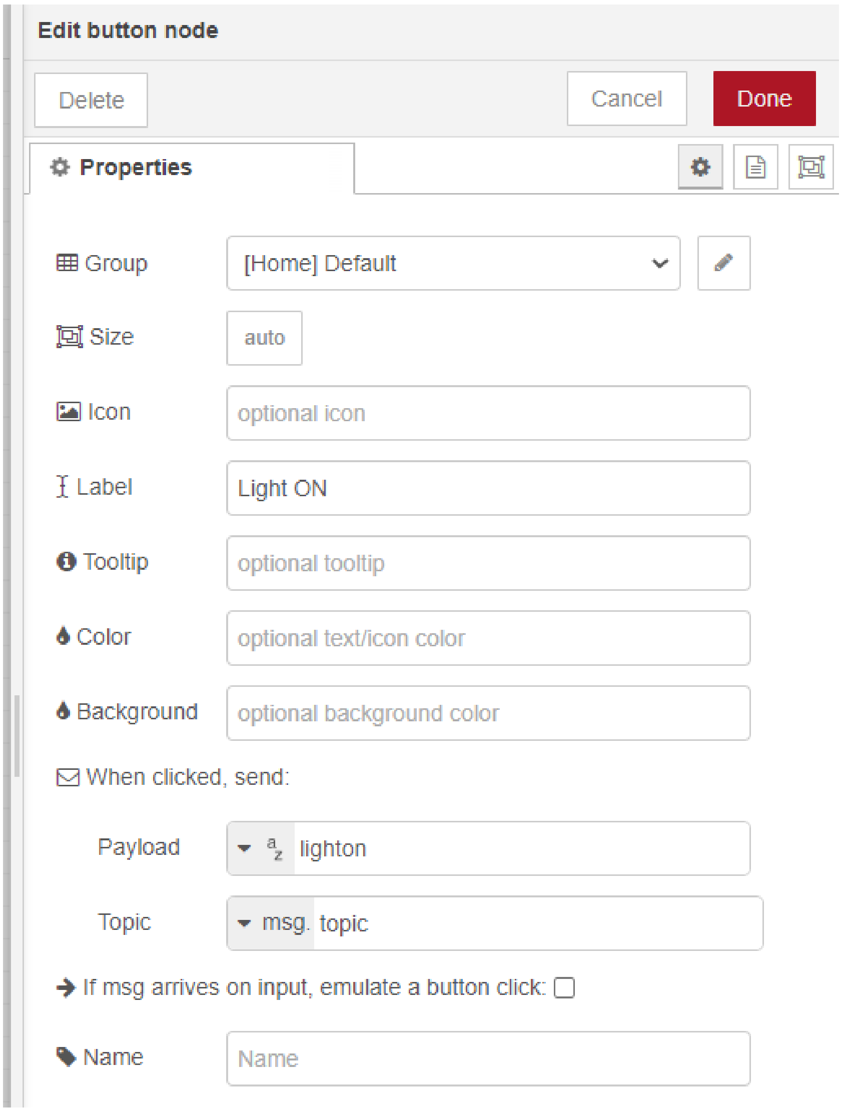


```
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength) {
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton") {
    Serial.println(data3);
    digitalWrite(LED,HIGH);
  }
  else {
    Serial.println(data3);
    digitalWrite(LED,LOW);
  }
  data3="";
}
```

NODE RED FLOW DIAGRAM:



NODE RED FLOW PROPERTIES:



Group [Home] Default

Size auto

Icon optional icon

Label Light ON

Tooltip optional tooltip

Color optional text/icon color

Background optional background color

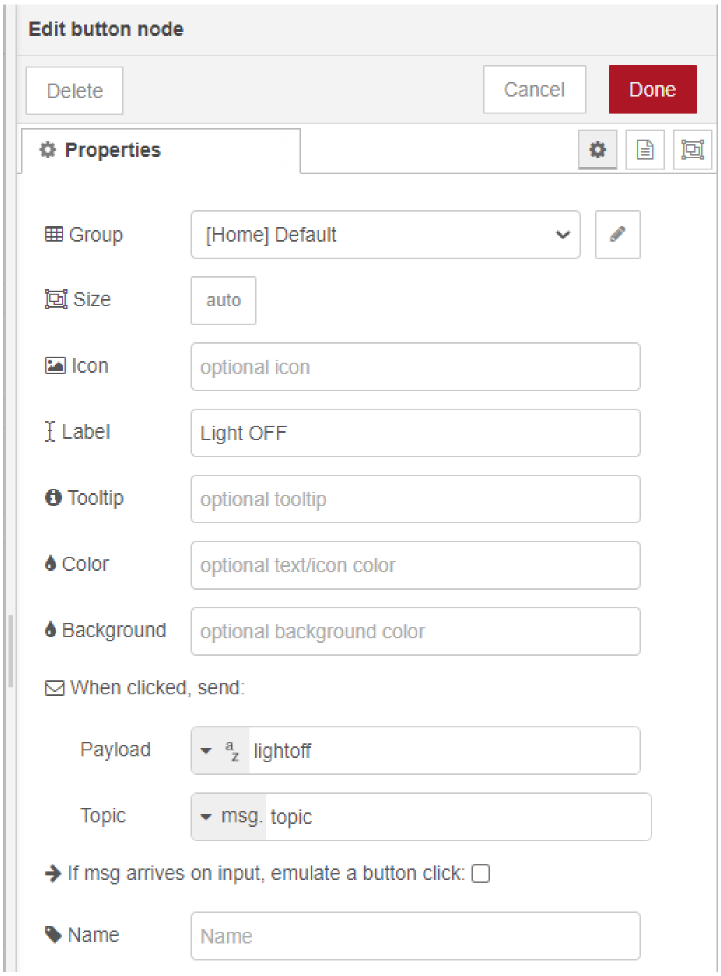
When clicked, send:

Payload lighton

Topic msg. topic

If msg arrives on input, emulate a button click: □

Name Name



Edit button node

Group: [Home] Default

Size: auto

Icon: optional icon

Label: Light OFF

Tooltip: optional tooltip

Color: optional text/icon color

Background: optional background color

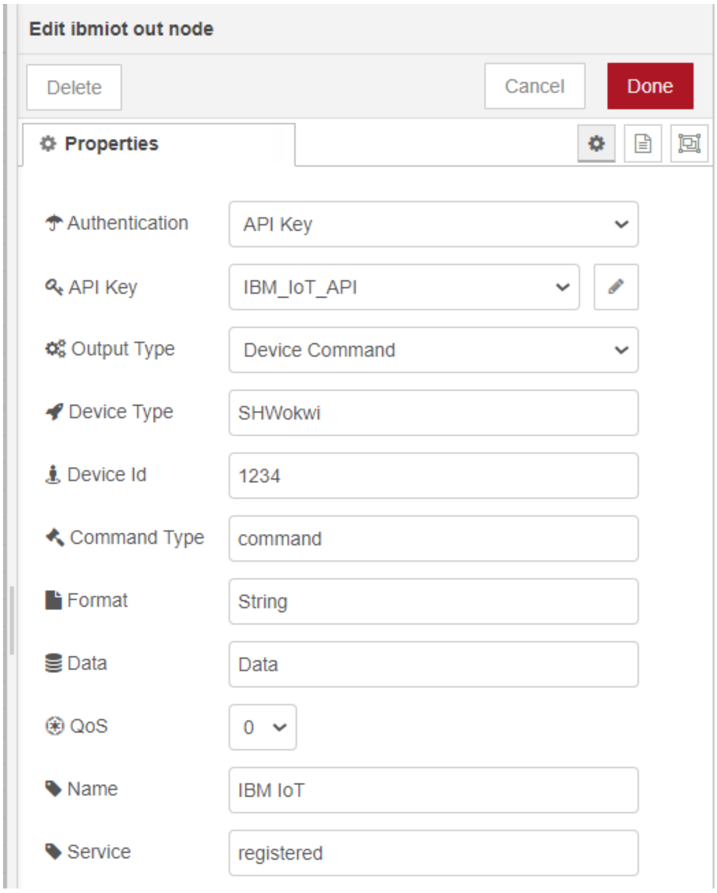
When clicked, send:

Payload: lightoff

Topic: msg. topic

If msg arrives on input, emulate a button click: □

Name: Name



Edit ibmiot out node

Authentication: API Key

API Key: IBM_IoT_API

Output Type: Device Command

Device Type: SHWokwi

Device Id: 1234

Command Type: command

Format: String

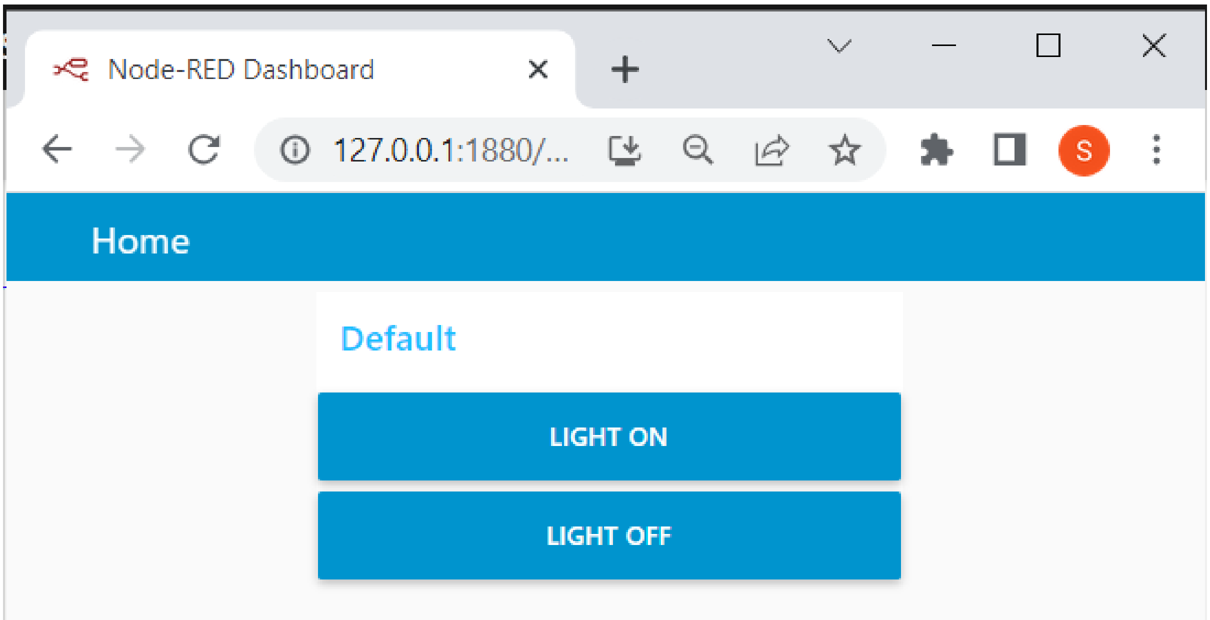
Data: Data

QoS: 0

Name: IBM IoT

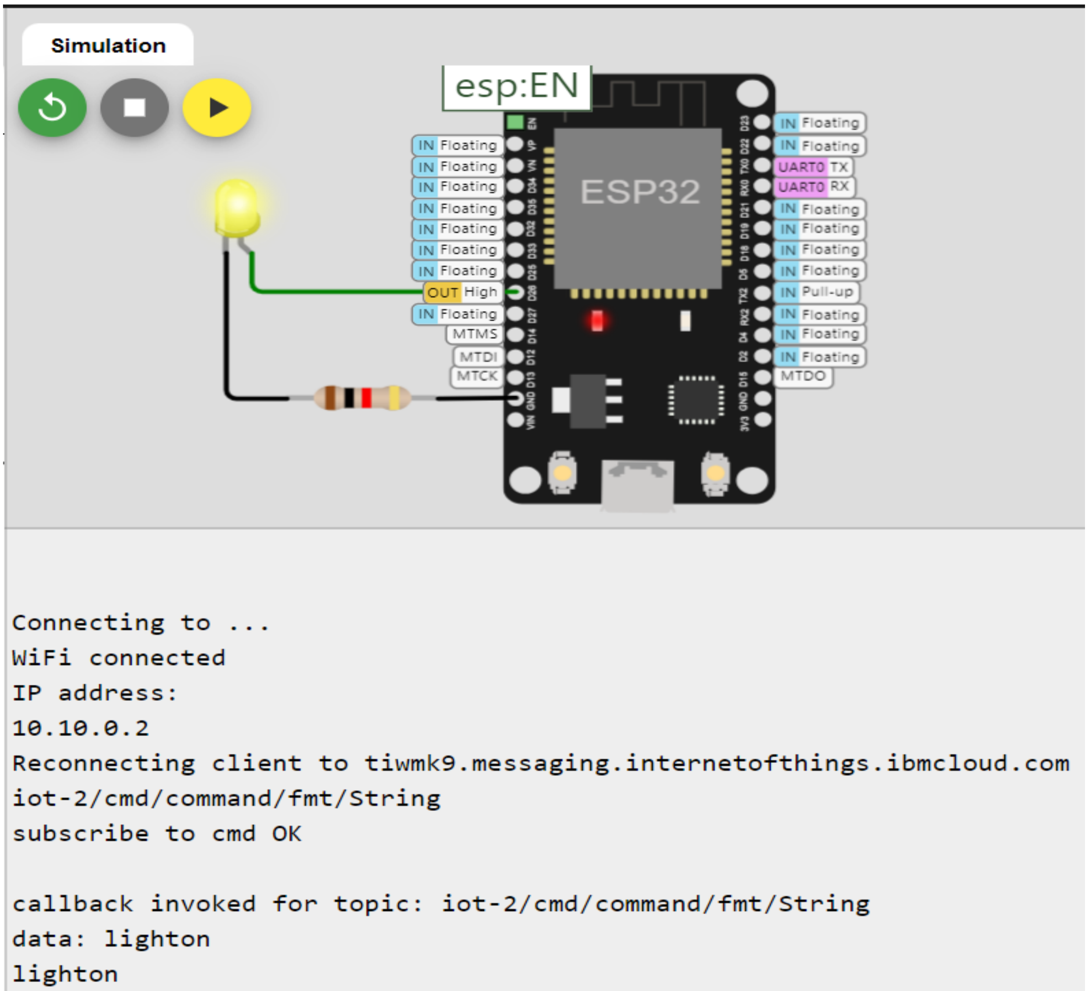
Service: registered

NODE RED DASHBOARD:



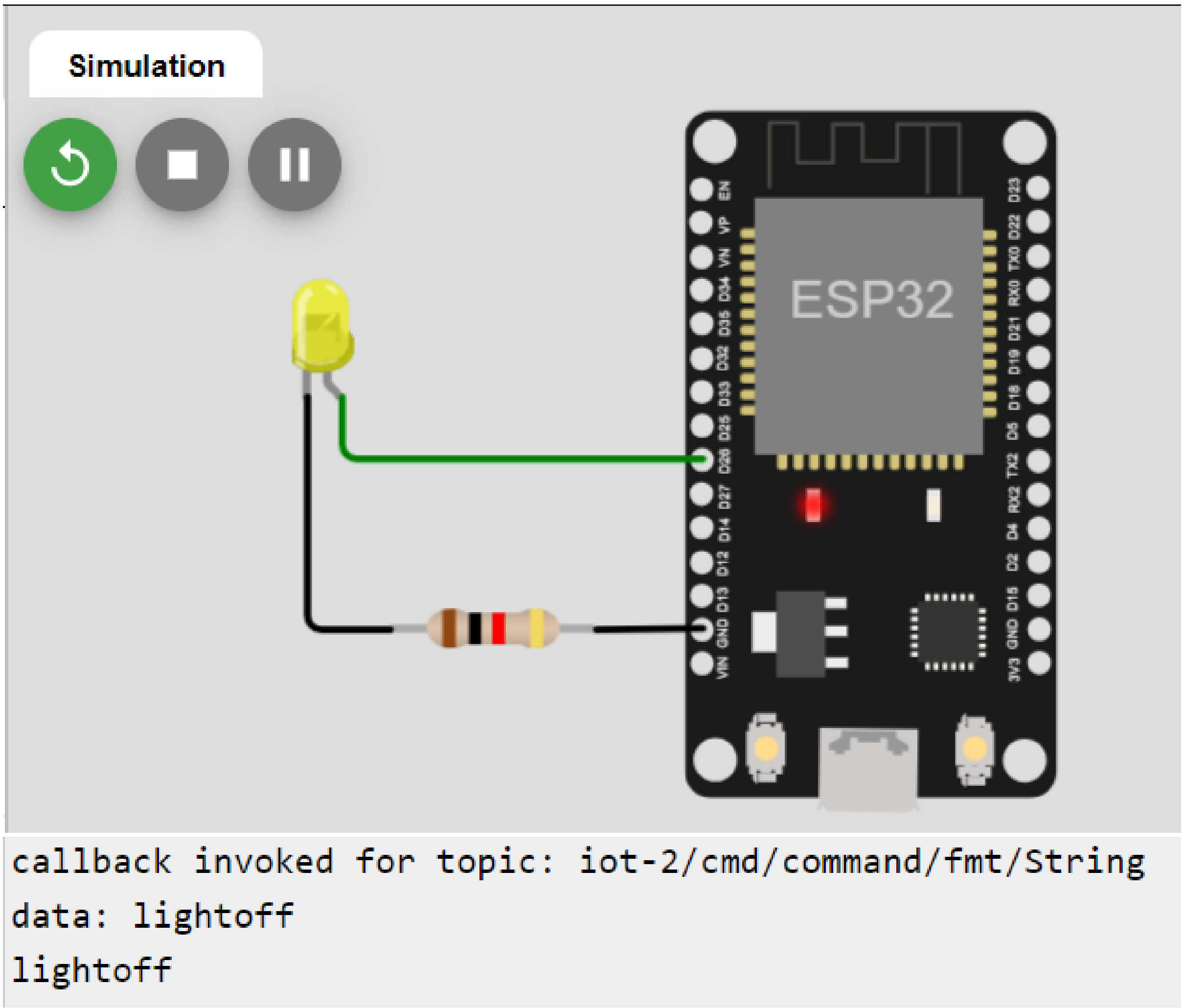
OUTPUT:

When “LIGHT ON” button is clicked on the dashboard-



```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to tiwmk9.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

callback invoked for topic: iot-2/cmd/command/fmt/String
data: lighton
lighton
```

When "LIGHT OFF" button is clicked on the dashboard-



```
callback invoked for topic: iot-2/cmd/command/fmt/String
data: lightoff
lightoff
```

RESULT: Given task was carried out successfully.