Week 3 Assignment

L.ROSHAN KUMAR

Link: https://wokwi.com/projects/365072974192459777

TASK:

In Wokwi, add a LED and switch it ON and OFF from Node-Red.

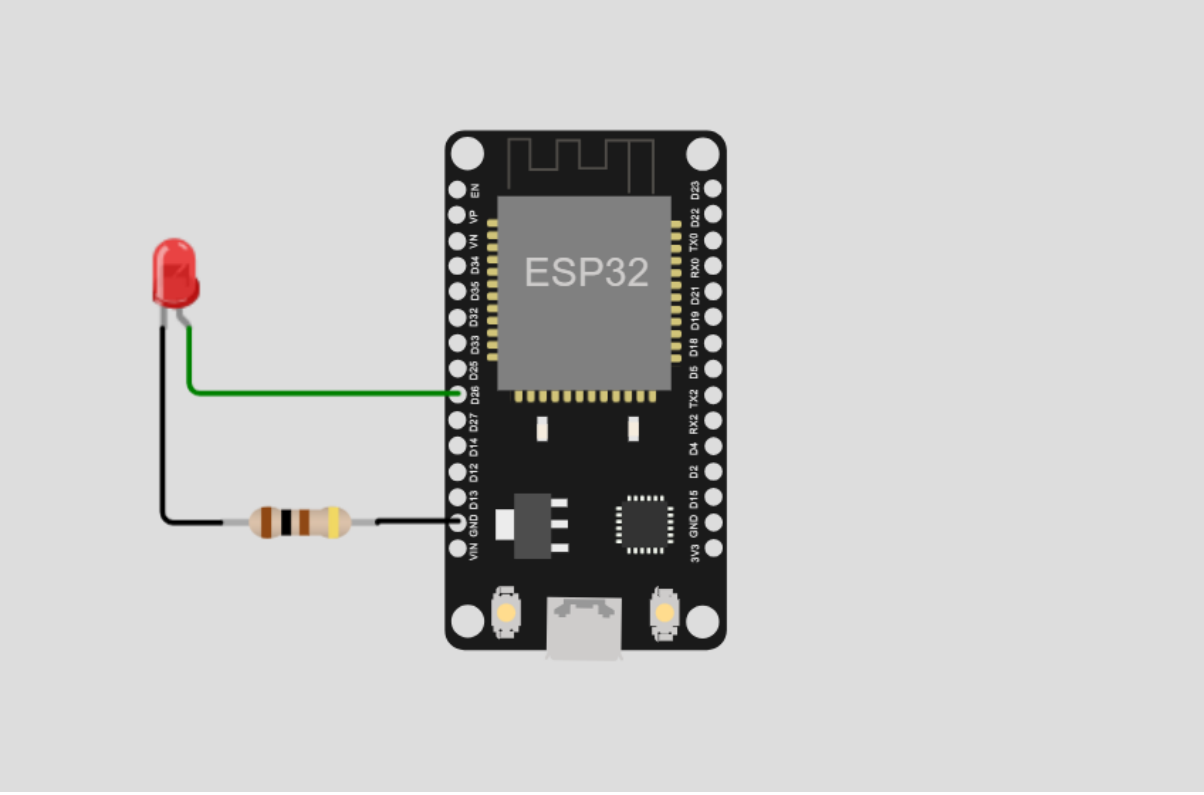
CODE:

```
#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 "1uw3rp"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//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);
```

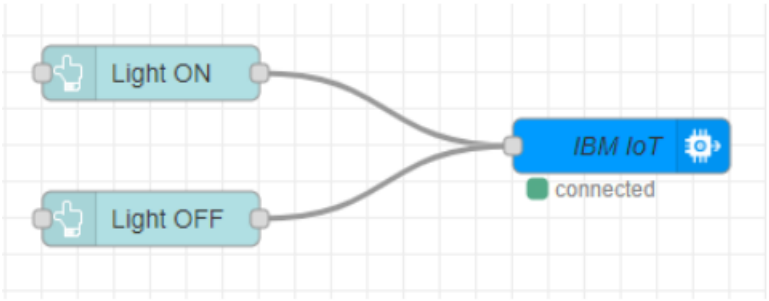
```
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

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 {
```

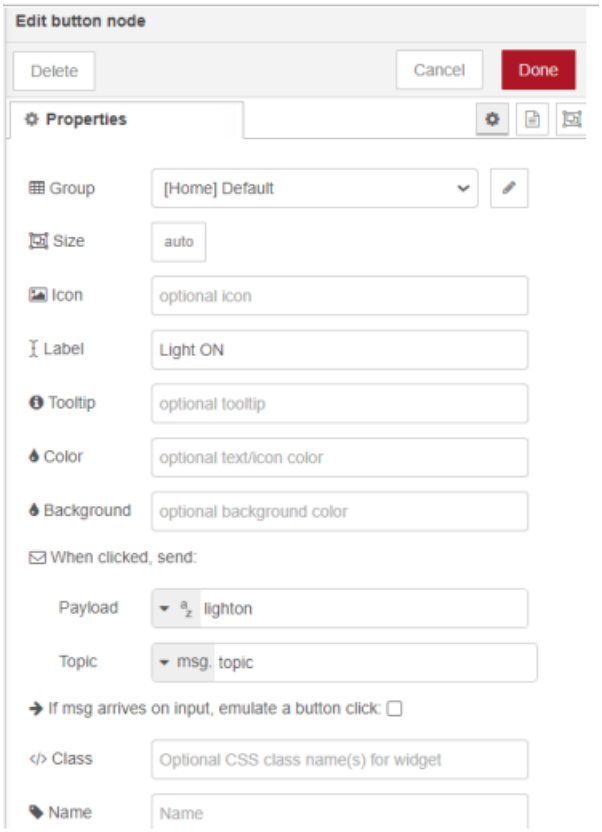
```
    Serial.println("subscribe to cmd FAILED");
  }
}

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="";
}
```

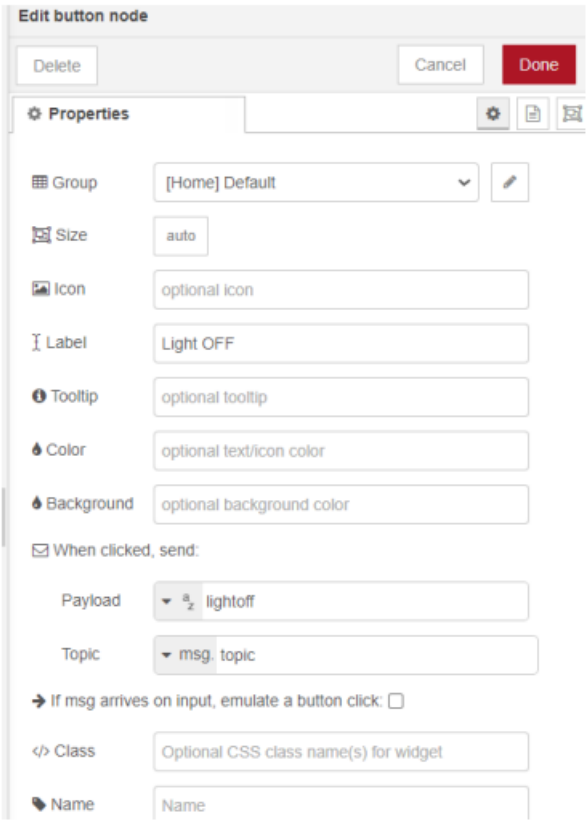
SCHEMATIC

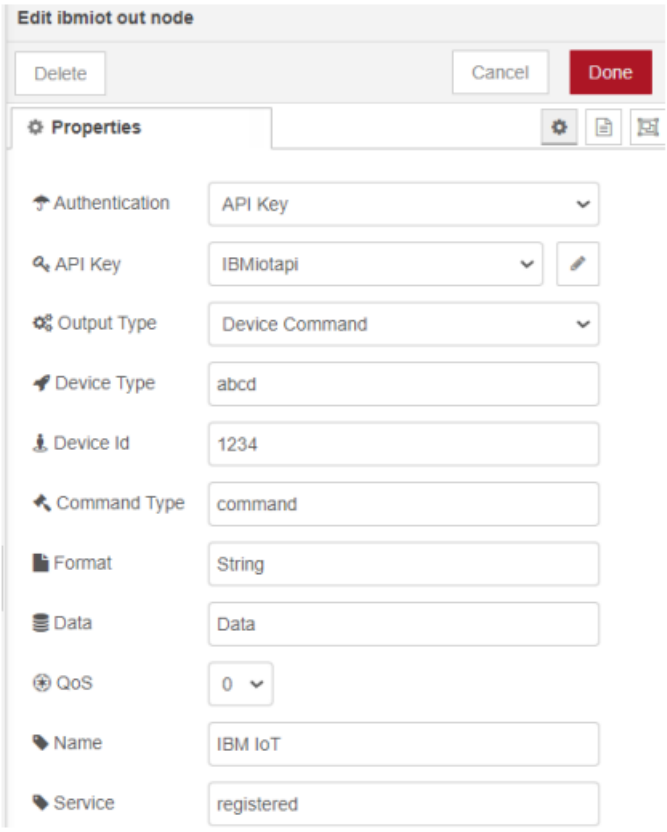


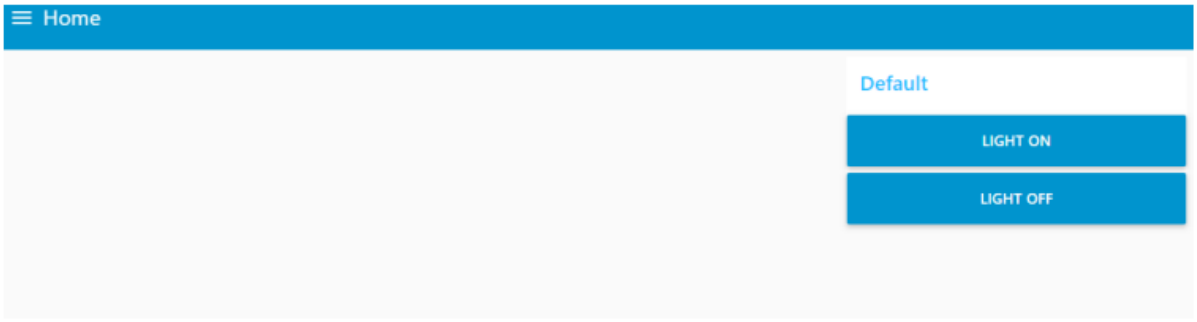
NODE RED FLOW DIAGRAM :



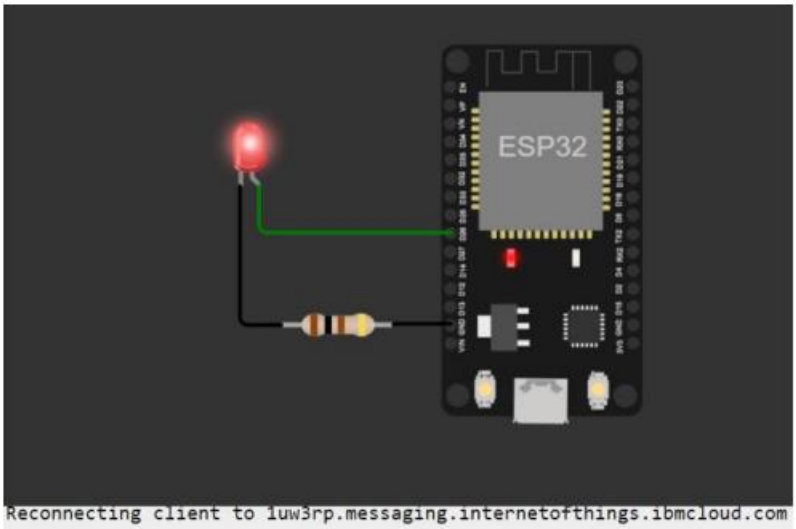



Edit button node
Delete
Cancel
Done
Properties
Group
[Home] Default
Size
auto
Icon
optional icon
Label
Light OFF
Tooltip
optional tooltip
Color
optional text/icon color
Background
optional background color
When clicked, send:
Payload
lightoff
Topic
msg.
topic
If msg arrives on input, emulate a button click:
Class
Optional CSS class name(s) for widget
Name
Name


Edit ibmiot out node
Delete
Cancel
Done
Properties
Authentication
API Key
API Key
IBMiotapi
Output Type
Device Command
Device Type
abcd
Device Id
1234
Command Type
command
Format
String
Data
Data
QoS
0
Name
IBM IoT
Service
registered

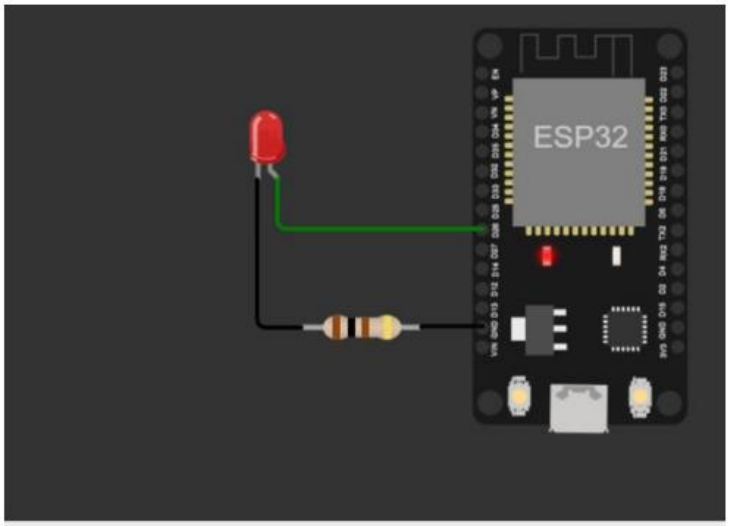

Home
Default
LIGHT ON
LIGHT OFF

OUTPUT




```
Reconnecting client to 1uw3rp.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 IS ON




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

WHEN LIGHT IS OFF