

NAME: R. GAUTHAM

REG NO: 20BEC1336

COURSE: Internet of Things (IOT)

WEEK 3 ASSIGNMENT

AIM:

Switch ON and OFF an LED from NODE RED

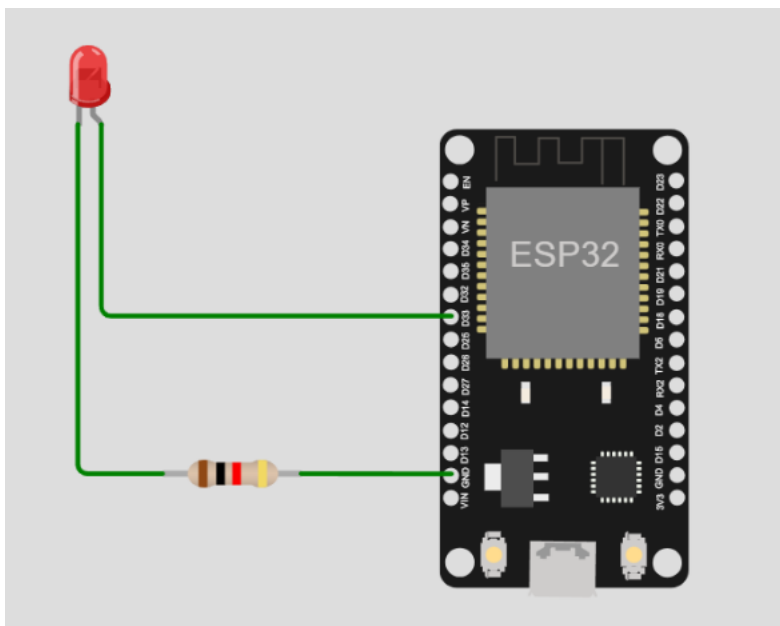
SOFTWARE REQUIRED:

WOKWI and NODE RED

WOKWI LINK:

<https://wokwi.com/projects/366879562597013505>

CIRCUIT DIAGRAM:



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 "t39hi2"//IBM ORGANITION ID
#define DEVICE_TYPE "Learning"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "31929095"//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 ");

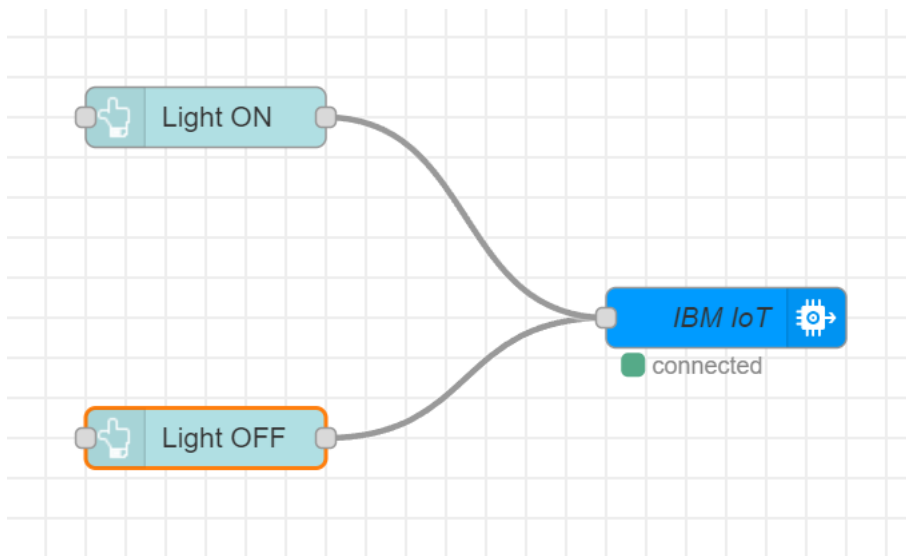
    WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish
the connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());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

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:



NODE PROPERTIES:

Edit button node

Delete Cancel Done

⚙ 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

☐ Enabled

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

Enabled

Edit ibmiot out node

Delete

Cancel

Done

Properties

Authentication

API Key

API Key

IBMiotapi

Output Type

Device Command

Device Type

Learning

Device Id

31929095

Command Type

command

Format

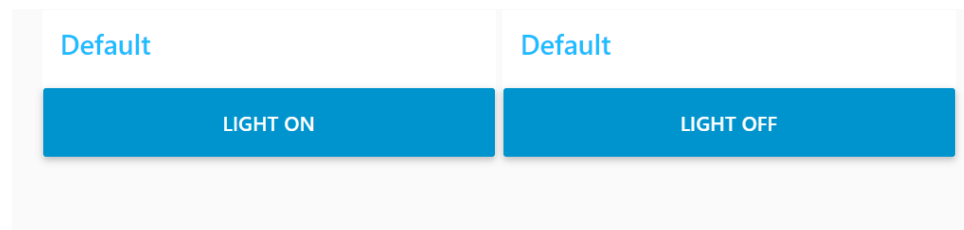
String

Data

Data

Enabled

NODE RED DASHBOARD:



OUTPUT:

When “LIGHT ON” button is clicked:

The diagram shows an ESP32 microcontroller board connected to a red LED. The LED's anode is connected to a digital pin on the ESP32, and its cathode is connected to ground through a resistor.

```
callback invoked for topic: iot-2/cmd/command/fmt/String
data: lighton
lighton
Reconnecting client to t39hi2.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK
```

When “LIGH OFF” button is clicked:

The diagram shows an ESP32 microcontroller board connected to a red LED. The LED's anode is connected to a digital pin on the ESP32, and its cathode is connected to ground through a resistor.

```
callback invoked for topic: iot-2/cmd/command/fmt/String
data: lightoff
lightoff
Reconnecting client to t39hi2.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK
```

RESULT:

Given task was carried out successfully.