

# **Assignment -3**

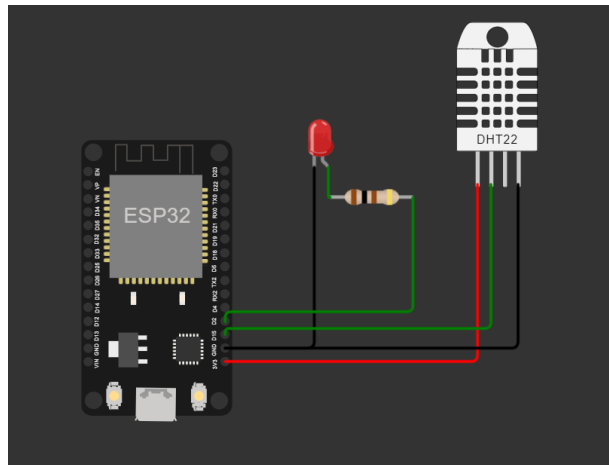
**Apoorv Gupta**  
**20BCT0117**

Date: 02/06/23

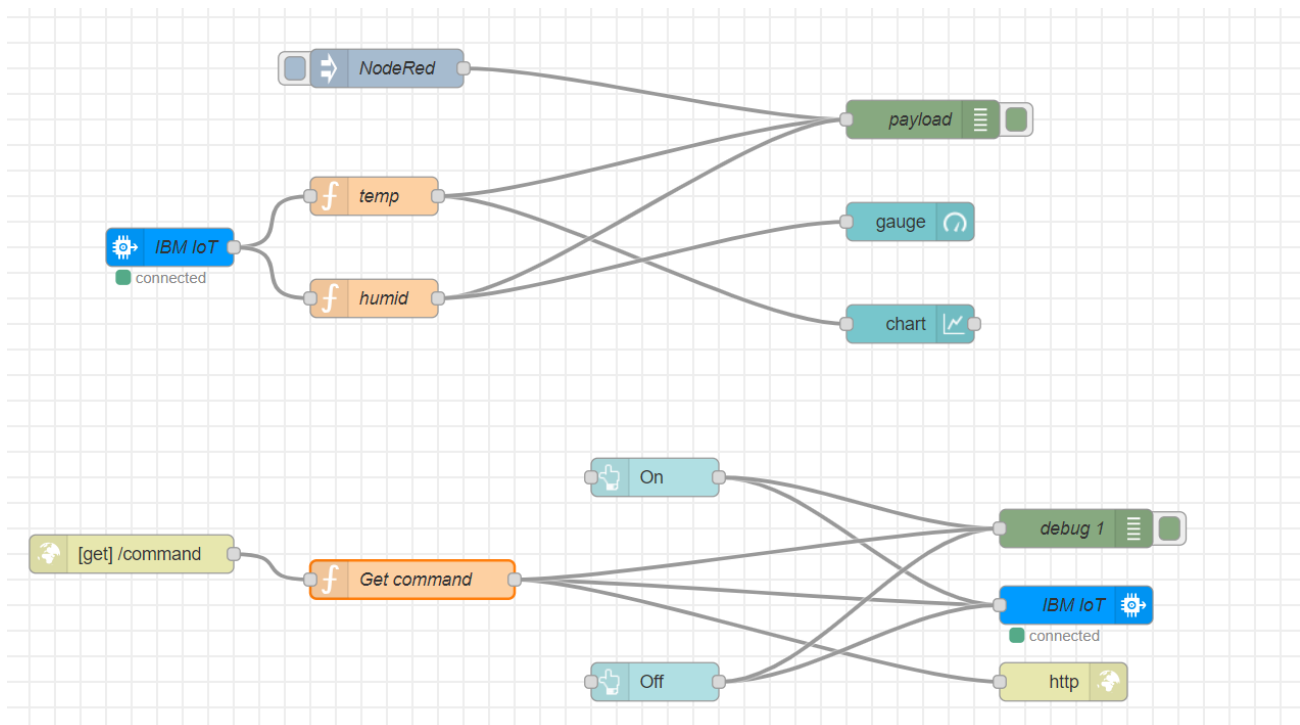
## Control LED using Node-Red

*Q- Using ESP32, make a circuit in Wokwi to control LED through NodeRed(turn the led 'on' and 'off').*

### Circuit:



### Node-Red Flow:



## Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#define OUT 4

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

#define ORG "#####" // it should remain confidential
#define DEVICE_TYPE "VirtualIoT"
#define DEVICE_ID "5901"
#define TOKEN "#####" // it should remain confidential
String data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, callback, wifiClient);

void setup(){
  Serial.begin(115200);
  pinMode(OUT, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop(){
  int state = digitalRead(OUT);
  digitalWrite(LED_BUILTIN, state);
  PublishData(state);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

void PublishData(int value) {
  mqttconnect();
  String payload = String(value);

  Serial.print("Sending payload: ");
```

```

Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
} else {
    Serial.println("Publish faiOUT");
}
}

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(){
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6);
    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 FAIOUT");
    }
}

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);
}

```

## Node-Red Configuration:

After Deploying, we can see that we are receiving data from the IBM Cloud to Node-Red in the debug tab, also the output after pressing the buttons.

The left screenshot shows the debug console with a list of incoming messages. Each message has a timestamp, a node name 'payload', and a detailed log showing the IoT cloud event and the message payload (a number). The payloads shown are 44.3, 44.3, and 34.1.

The right screenshot shows the debug console with a list of outgoing messages. Each message has a timestamp, a node name 'debug 1', and a detailed log showing the command being sent to the IoT cloud. The commands shown are 'lightoff' and 'lighton'.

Two different types are visible as I have done it using http and buttons both.

This function receives it using http request and sets the string value in a global variable “com”. (Its basically a Buffer data type, which contains character-wise ascii value), which we receive in the form of byte in wokwi.

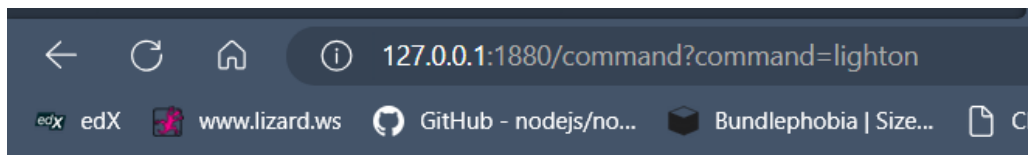
The screenshot shows a function node in Node-Red. The node is named 'Get command'. The code inside the function node is as follows:

```

1 msg.payload = msg.payload.command;
2 global.set("com", msg.payload);
3 return msg;
4

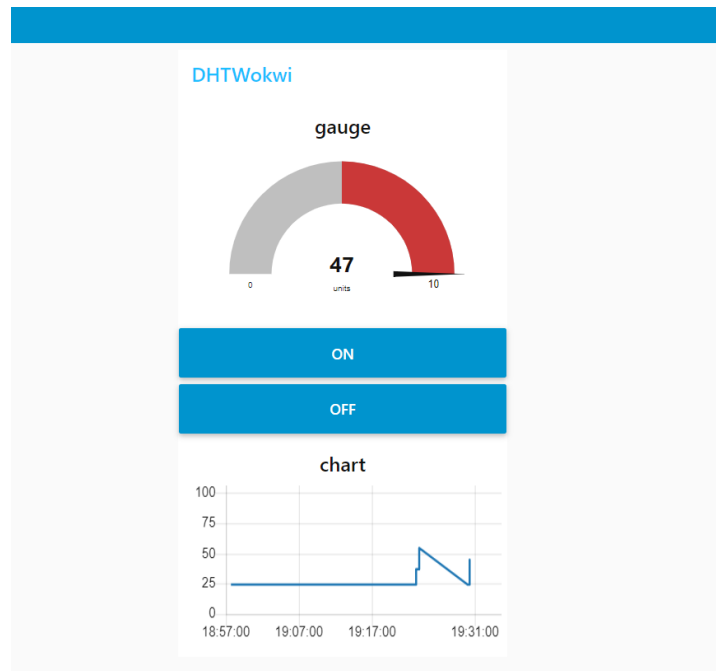
```

I can either give command here or use the buttons.

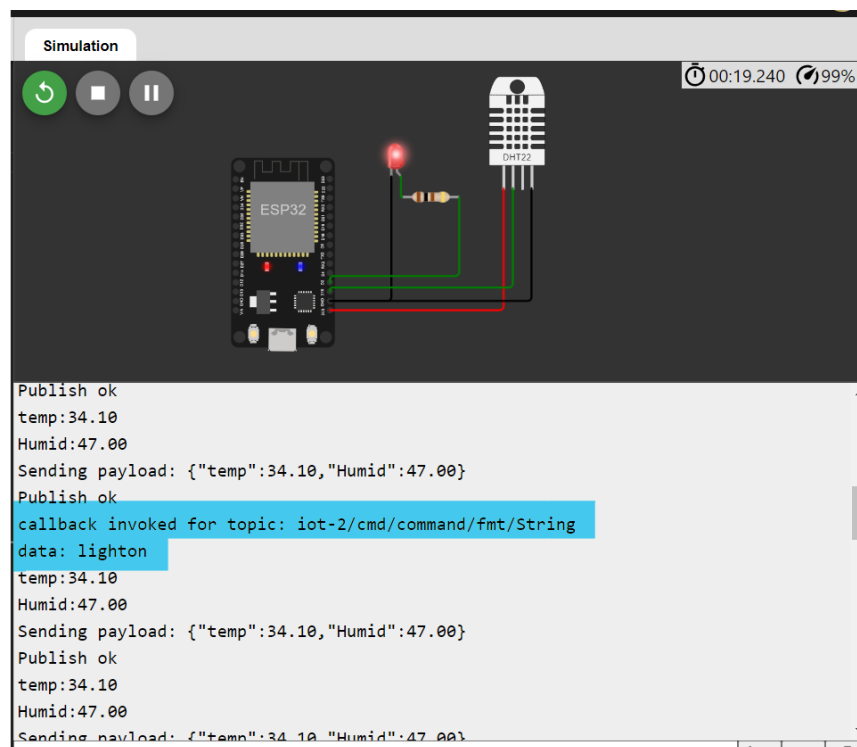


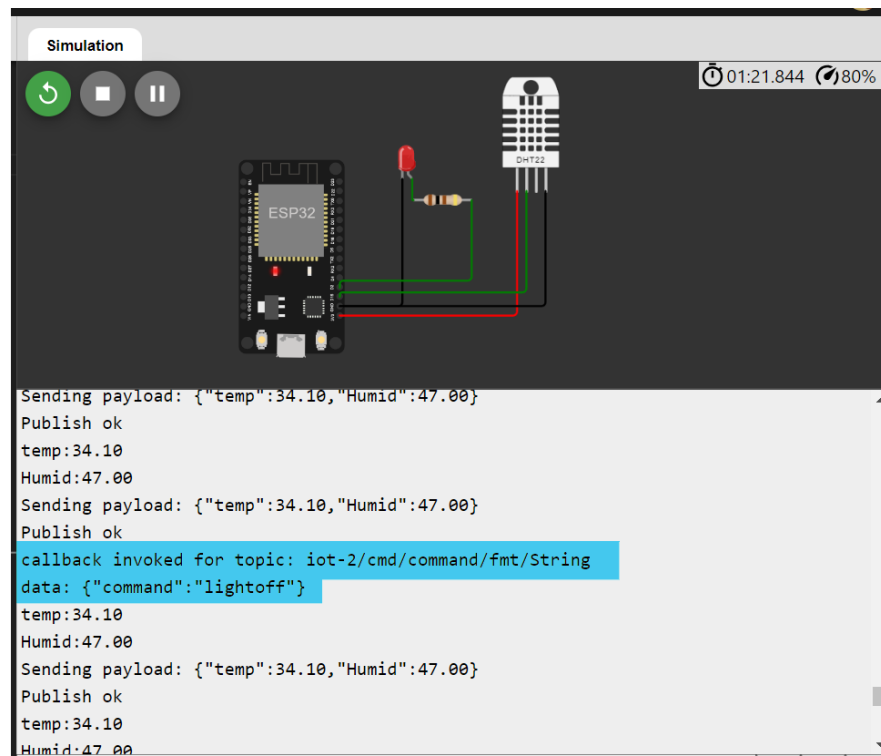
lighton

## Dashboard:



## OUTPUTS:





**Callback function gets invoked whenever I give the command that subscribes to the required topic.**

**As the condition is given in if-else... So any other data except “lighton” will make the led off.**

## **Wokwi Project Link:**

[Assignment3-DHT GlowLED - Wokwi ESP32, STM32, Arduino Simulator](#)