

Sprint-3
Team ID : PNT2022TMID08020

Wokwi.com Reference

<https://wokwi.com/projects/348680991769887315>

NodeMCU Code :

```
#include <ESP32Servo.h>
#include <WiFi.h>
#include <Stepper.h>
#include <PubSubClient.h>
#include <DHTesp.h>
#define DHTPIN 15
#define GAS_LEVER 34
#define buzzer 13
#define LED 5
const int servoPin = 12;
Servo valve;
DHTesp dhtsensor;
Stepper stepper(1000, 19,21,22,23);

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

#define ORG "8o05yw"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID"sam123"
#define TOKEN "123456789"

String data3;
float h, t, g;
int pos=0;
boolean valve_open=true;

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/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);
  dhtsensor.setup(DHTPIN,DHTesp::DHT22);
  stepper.setSpeed(100);
  valve.attach(servoPin);
  pinMode(GAS_LEVER, INPUT);
  pinMode(buzzer,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
  valve.write(90);

}

void loop()
{
  TempAndHumidity data=dhtsensor.getTempAndHumidity();
  t=data.temperature;
  h=data.humidity;
  g=map(int(analogRead(GAS_LEVER)), 0, 4095, 200,
2000);
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Humidity:");
  Serial.println(h);
  Serial.print("Gas Level:");
  Serial.println(g);

  if(g>500){
    tone(buzzer, 1000);
    stepper.step(1000);
    valve.write(180);
  }
  else{
    valve.write(90);
    noTone(buzzer);
  }
}

```

```

    PublishData(t, h, g);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

```

```

void PublishData(float temp, float humid, float
gas_level) {
    mqttconnect();
    String payload = "{"\"temperature\":";
    payload += temp;
    payload += "," \" \"\"humidity\":";
    payload += humid;
    payload += "," \" \"\"gas_level\":";
    payload += gas_level;
    payload += "}";
}

```

```

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

```

```

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

```

```

}
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 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);

    data3="";
}

```

