

SPRINT-2

WhatsApp x sprint 2 - Wokwi Arduino and ESP32 x

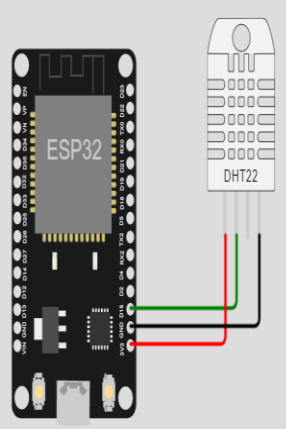
wokwi.com/projects/347959858838897236

WOKWI SAVE SHARE sprint 2 Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #include <stdlib.h>
5 #include <time.h>
6 #define DHTPIN 15 // what pin we're connected to
7 #define DHTTYPE DHT22 // define type of sensor DHT 11
8
9
10 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht connect
11
12 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
13
14 //-----credentials of IBM Accounts-----
15
16 #define ORG "sms611"
17 #define DEVICE_TYPE "3114"
18 #define DEVICE_ID "14"
19 #define TOKEN "98765432"
20
21 String data3 = "";
22 String accidentstatus = "";
23 String sprinkstatus = "";
24 float temp = 0;
25 bool isfanon = false;
26 bool issprinkon = false;
27 int gas = 0;
28 int flame = 0;
29 int flow = 0;
30
31
32 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
33 char publishTopic[] = "iot-2/evt/data/fmt/json";
34 char subscribetopic[] = "iot-2/cmd/command/fmt/String";
35 char authMethod[] = "use-token-auth";
```

Simulation



ESP32

DHT22

19:06 18-11-2022

OUTPUT:

The screenshot displays the Wokwi IoT simulator interface. On the left, the code editor shows the file `esp32-dht22.ino` with the following code:

```
15 char token[] = "TOKEN";
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17
18 WiFiClient wifiClient;
19 PubSubClient client(server, 1883, wifiClient);
20
21 const int DHT_PIN = 15;
22
23 bool is_exhaust_fan_on = false;
24 bool is_sprinkler_on = false;
25
26 float temperature = 0;
27
28 int gas_ppm = 0;
29 int flame = 0;
30 int flow = 0;
31
32 String flame_status = "";
33 String accident_status = "";
34 String sprinkler_status = "";
35
36 DHTesp dhtSensor;
37
38
39 void setup() {
40   Serial.begin(999000);
41
42   /* sensor pin setups */
43   dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
44   //if real gas sensor is used make sure the sensor is heated up for accurate readings
45   /*
46    - Here random values for readings and stdout were used to show the
47    working of the devices as physical or simulated devices are not
48    available.
49    */
50 }
```

On the right, the simulation window shows a visual representation of the ESP32 microcontroller connected to a DHT22 temperature and humidity sensor. The sensor is connected to the ESP32 via three wires: a red wire to the VCC pin, a green wire to the GND pin, and a black wire to the data pin.

Below the simulation window, the console output shows the following messages:

```
Publish failed
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
```

The bottom of the interface shows a Windows taskbar with various application icons and a system clock indicating 19:52 on 18-11-2022.