

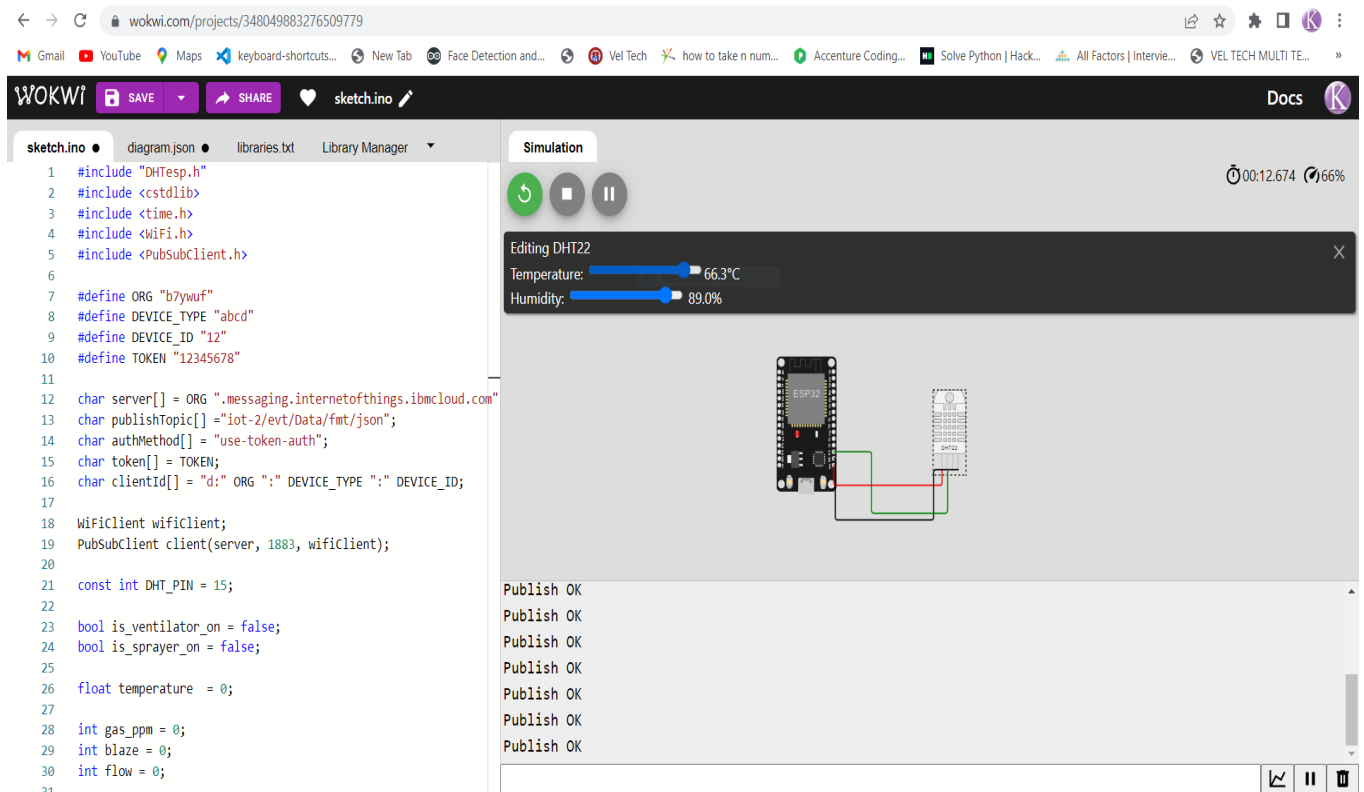
Industry-Specific Intelligent Fire Management System

TMID:- PNT2022TMID22485

SPRINT-2

Sending the data of the sensor from the microcontroller to the IBM Watson Dot platform

1. Write code in WOKWI platform



The screenshot displays the WOKWI web interface. On the left, the 'sketch.ino' file is open in a code editor, showing the following code:

```
1 #include "DHTesp.h"
2 #include <stdlib>
3 #include <time.h>
4 #include <WiFi.h>
5 #include <PubSubClient.h>
6
7 #define ORG "b7ywuf"
8 #define DEVICE_TYPE "abcd"
9 #define DEVICE_ID "12"
10 #define TOKEN "12345678"
11
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"
13 char publishTopic[] = "iot-2/evt/Data/fmt/json";
14 char authMethod[] = "use-token-auth";
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_ventilator_on = false;
24 bool is_sprayer_on = false;
25
26 float temperature = 0;
27
28 int gas_ppm = 0;
29 int blaze = 0;
30 int flow = 0;
31
```

On the right, the 'Simulation' window is active, showing a visual representation of the ESP32 microcontroller connected to a DHT22 sensor. The 'Editing DHT22' panel displays the following values:

- Temperature: 66.3°C
- Humidity: 89.0%

Below the simulation, the 'Publish OK' message is repeated multiple times, indicating successful data transmission.

2. Sending data to IBM Watson platform

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device, ID 12, is 'Connected' and has a 'Recent Events' tab selected. The events table shows two entries, both labeled 'Data', with values in JSON format: `{"sensor_values":{"gas_ppm":249,"temperature":...`. The second device, ID 22, is 'Disconnected'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12	Connected	abcd	Device	Oct 10, 2022 1:26 PM	
22	Disconnected	py_code	Device	Nov 13, 2022 8:35 PM	

Event	Value	Format	Last Received
Data	<code>{"sensor_values":{"gas_ppm":249,"temperature":...</code>	json	a few seconds ago
Data	<code>{"sensor_values":{"gas_ppm":316,"temperature":...</code>	json	a few seconds ago

3.Data in JSON format

The 'Event Payload' window displays the following information:

- Event Name: Data
- Time Received: Nov 16, 2022 2:05 PM

```
1 {  
2   "senor_values": {  
3     "gas_ppm": 31,  
4     "temperature": 66,  
5     "blaze": 861,  
6     "flow": 0  
7   },  
8   "output": {  
9     "is_ventilator_on": false,  
10    "is_sprayer_on": true  
11  },  
12  "messages": {  
13    "fire_status": "Close Fire",  
14    "flow_status": "not working",  
15    "hazard_status": "severe"  
16  }  
17 }
```

4.Output is displayed in boards



< skboard



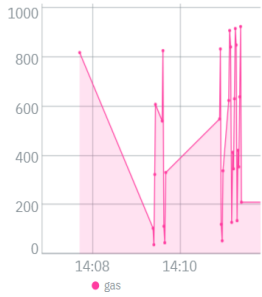
+ Add New Card



Settings



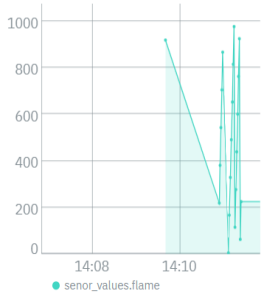
Line chart



gas



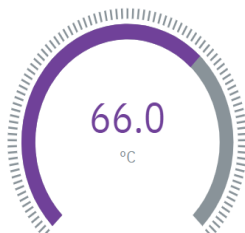
Line chart



senor_values.flame



Gauge



66.0

°C