

Sprint 3

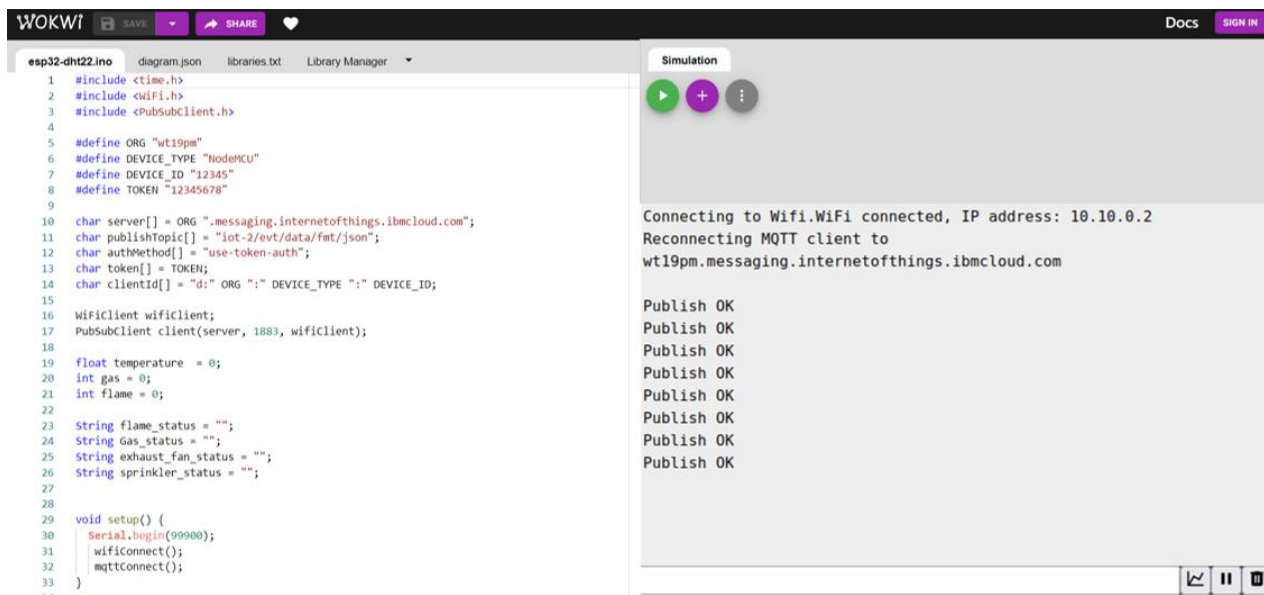
Check the Project:

WOKWI SIMULATOR LINK:

<https://wokwi.com/projects/347212137885598290>

WEB PAGE LINK: <https://node-red-dashboard059.eu-gb.mybluemix.net/fire>

Results:



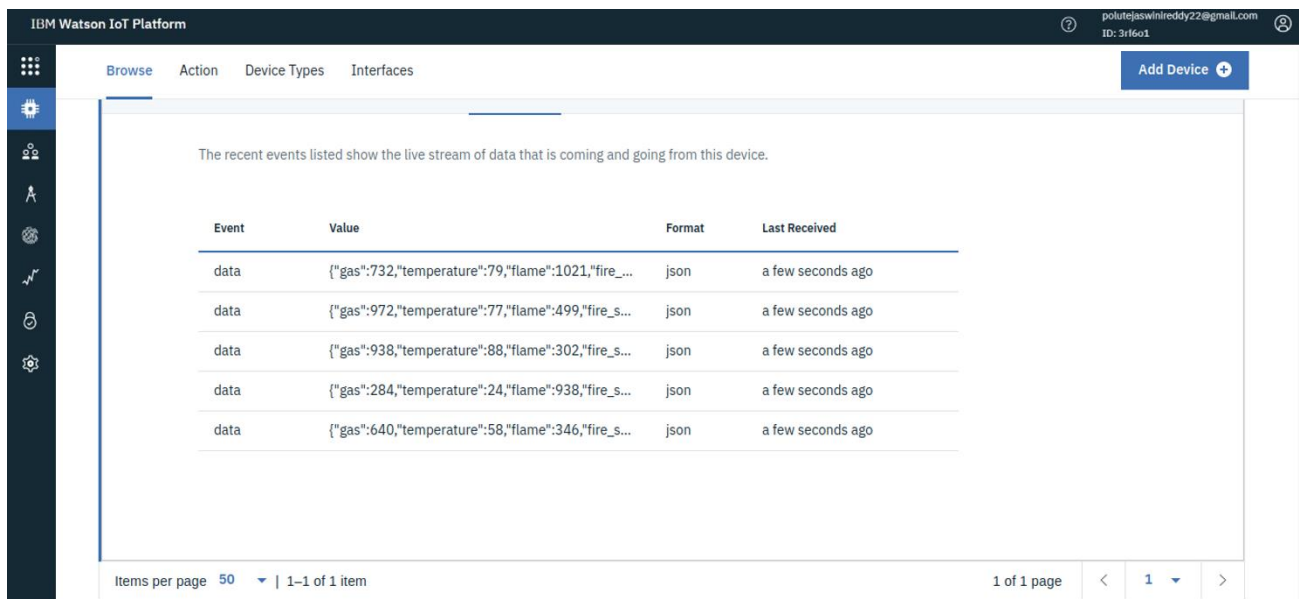
The screenshot shows the Wokwi web-based IDE. On the left, the Arduino IDE editor displays a sketch named 'esp32-dht22.ino'. The code includes libraries for time, WiFi, and PubSubClient, and defines MQTT server details and device information. The setup function initializes the serial port, WiFi, and MQTT client. On the right, the 'Simulation' window shows the execution logs. It indicates a successful WiFi connection to 10.10.0.2 and the reconnection of the MQTT client to the specified broker. Below the logs, a series of 'Publish OK' messages are shown, indicating successful data transmission.

```
1 #include <time.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 #define ORG "wt19pm"
6 #define DEVICE_TYPE "NodeMCU"
7 #define DEVICE_ID "12345"
8 #define TOKEN "12345678"
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/data/fmt/json";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15
16 WiFiClient wificlient;
17 PubSubClient client(server, 1883, wificlient);
18
19 float temperature = 0;
20 int gas = 0;
21 int flame = 0;
22
23 String flame_status = "";
24 String gas_status = "";
25 String exhaust_fan_status = "";
26 String sprinkler_status = "";
27
28
29 void setup() {
30   Serial.begin(99900);
31   wificlient.connect();
32   mqttConnect();
33 }
```

Simulation

Connecting to Wifi.WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
wt19pm.messaging.internetofthings.ibmcloud.com

Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK



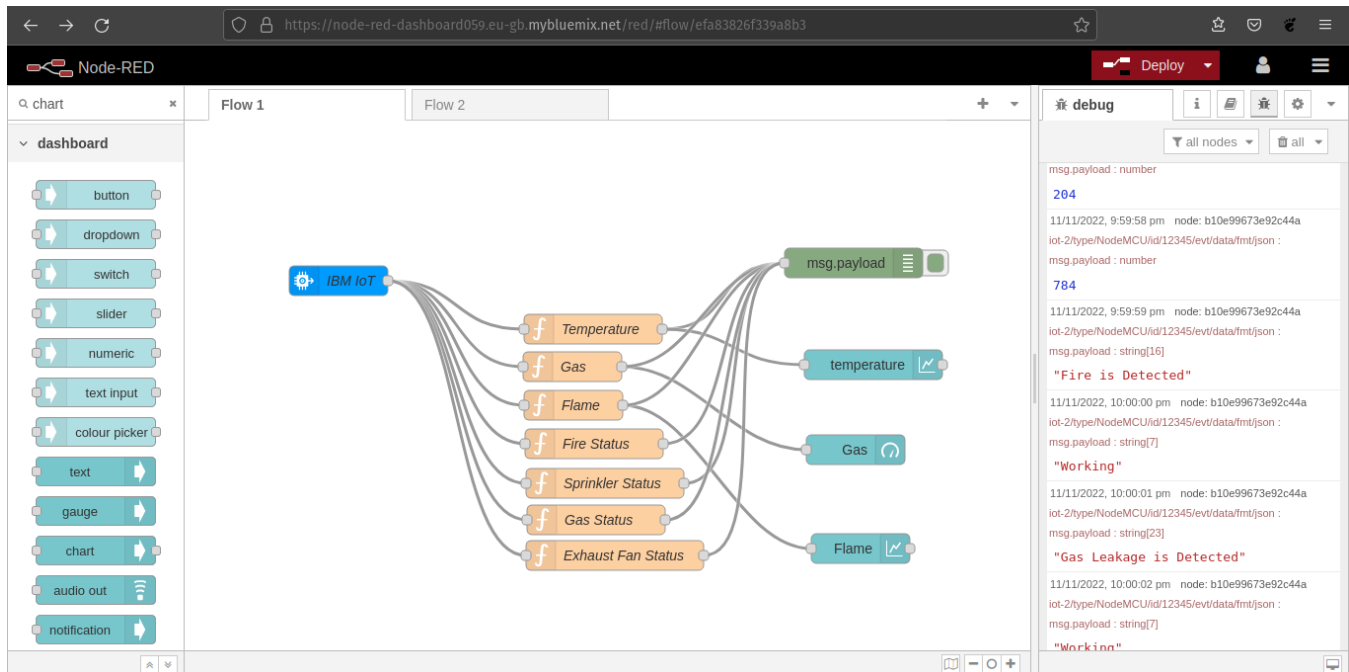
The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area displays a table of recent events for a device. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events are listed in descending order of time, with the most recent at the top. The 'Value' column contains JSON strings representing the data points. The 'Format' column shows 'json' for all events. The 'Last Received' column indicates the time since the event was received, all showing 'a few seconds ago'.

| Event | Value | Format | Last Received |
|-------|---|--------|-------------------|
| data | {"gas":732,"temperature":79,"flame":1021,"fire_s... | json | a few seconds ago |
| data | {"gas":972,"temperature":77,"flame":499,"fire_s... | json | a few seconds ago |
| data | {"gas":938,"temperature":88,"flame":302,"fire_s... | json | a few seconds ago |
| data | {"gas":284,"temperature":24,"flame":938,"fire_s... | json | a few seconds ago |
| data | {"gas":640,"temperature":58,"flame":346,"fire_s... | json | a few seconds ago |

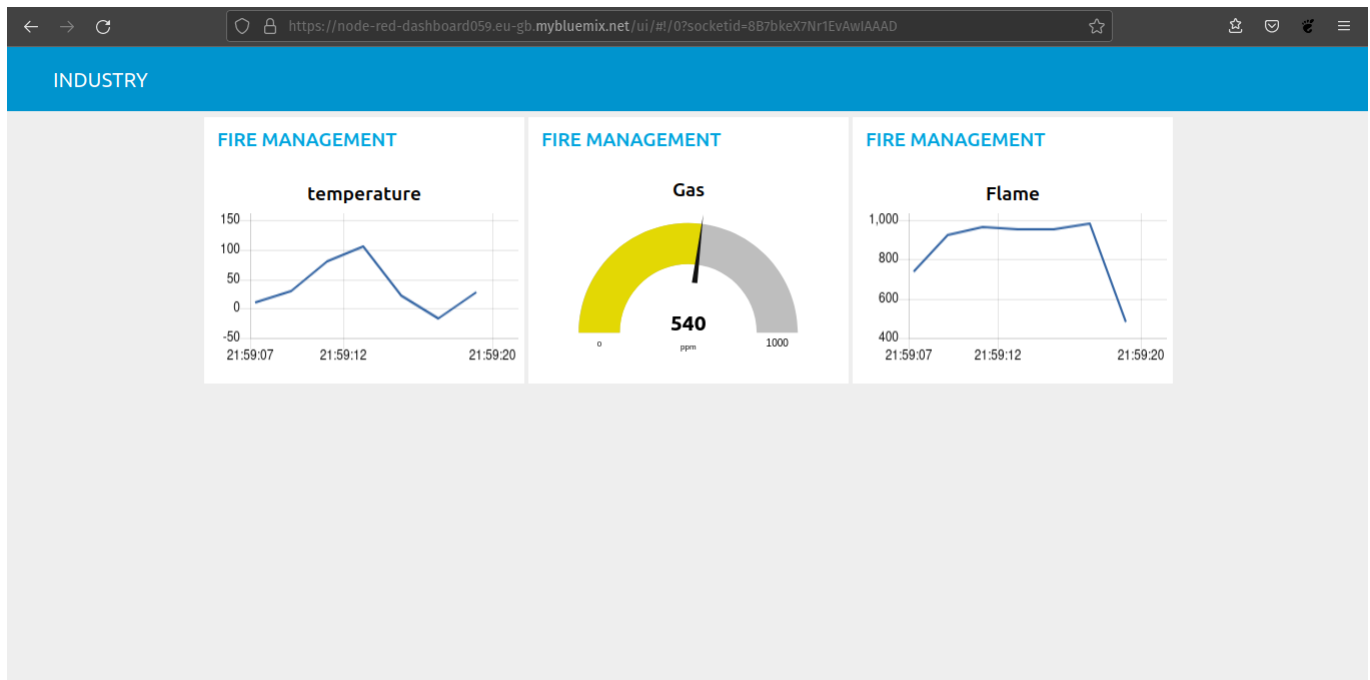
Items per page 50 | 1-1 of 1 item

1 of 1 page < 1 >

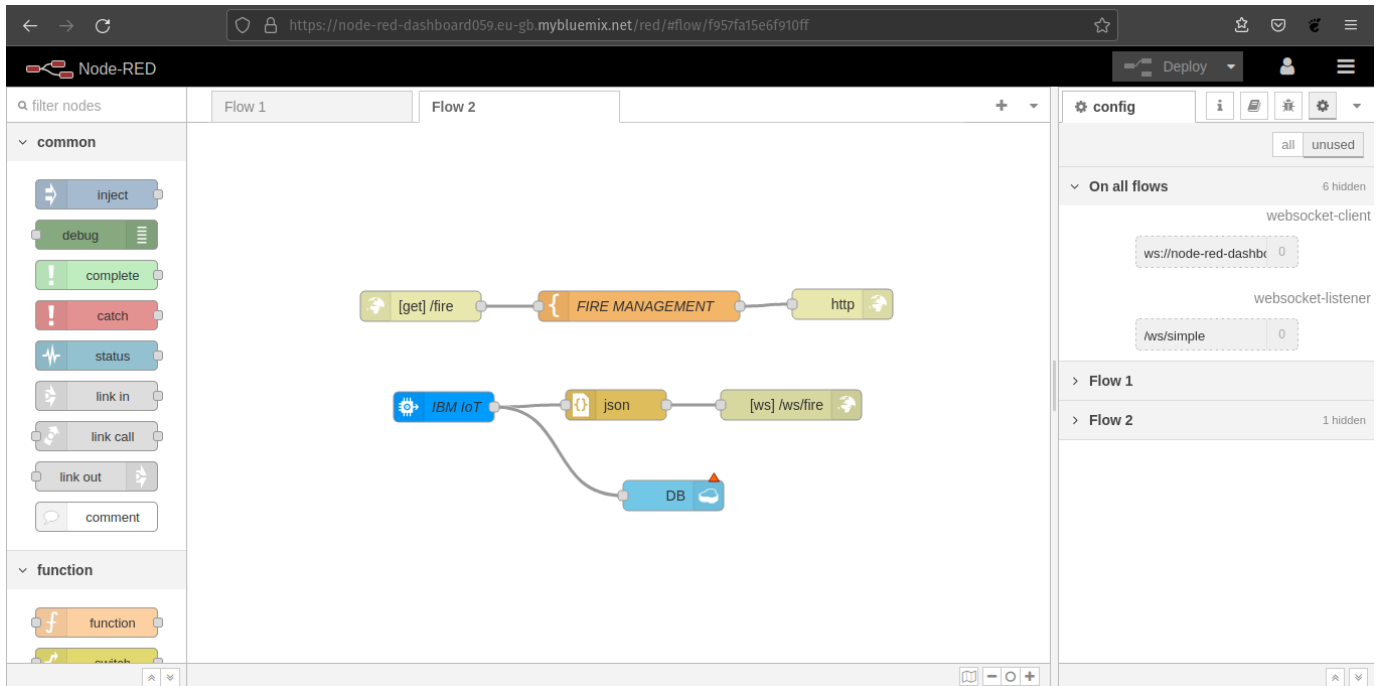
TRANSFERRING DATA FROM IBM WATSON INTO NODE-RED:



NODE-RED DASHBOARD:



TRANSFERRING DATA FROM NODE-RED INTO WEB PAGE USING WEB SOCKET:



WEB UL:

DESKTOP VIEW:



MOBILE VIEW:


🔒

eu-gb.mybluemix.net


+

29


FIRE MANAGEMENT SYSTEM




TEMPERATURE : 121



GAS : 3



FLAME : 650



FIRE STATUS : Fire is Detected
SPRINKLER STATUS: Working
GAS STATUS : No Gas Leakage is Detected
EXHAUST FAN STATUS : Not Working