

SPRINT-4

PROJECT	INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM
TEAM ID	PNT2022TMDI36769

PROGRAM:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include "DHT.h" // Library for dht11
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
#define LED 2
DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht connected

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

//-----credentials of IBM Accounts-----

#define ORG "1vqi0j" //IBM ORGANITION ID
#define DEVICE_TYPE "node-red" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "123456789" //Token
String data3;
float t;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and format
in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND COMMAND
IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by
passing parameter like server id,portand wificredential
void setup() // configureing the ESP32
{
    Serial.begin(115200);
    dht.begin();
    pinMode(LED,OUTPUT);
```

```

    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop()// Recursive Function
{

    t = dht.readTemperature();
    Serial.print("temperature:");
    Serial.println(t);

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

/*.....retrieving to Cloud.....*/

void PublishData(float temp) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"temperature\":\"";
    payload += temp;
    payload += "\"}";

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

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
        publish ok in Serial monitor or else it will print publish failed
    } 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() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
    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++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }

    Serial.println("data: "+ data3);
    if(data3=="lighton")
    {
        Serial.println(data3);
        digitalWrite(LED,HIGH);

    }
}

```

```
    else
    {
Serial.println(data3);
digitalWrite(LED,LOW);

    }
data3="";

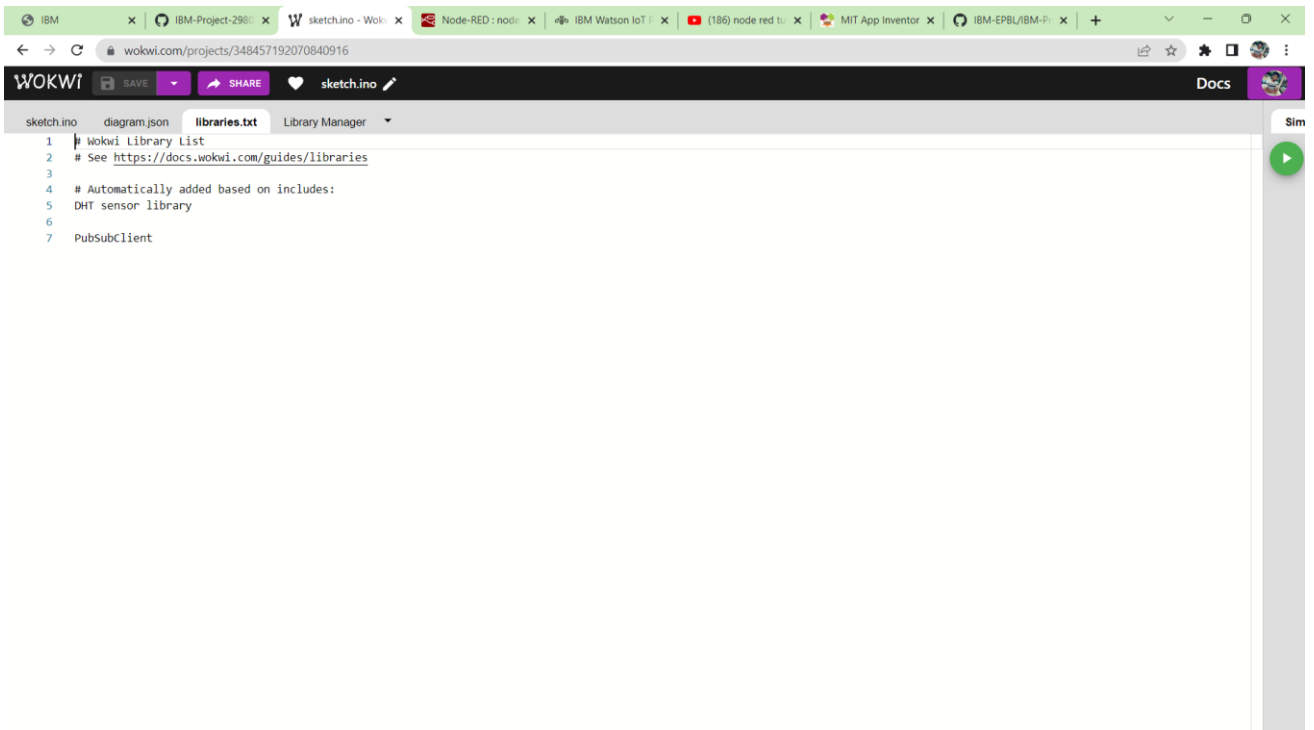
}
```

DIAGRAM.JSON:



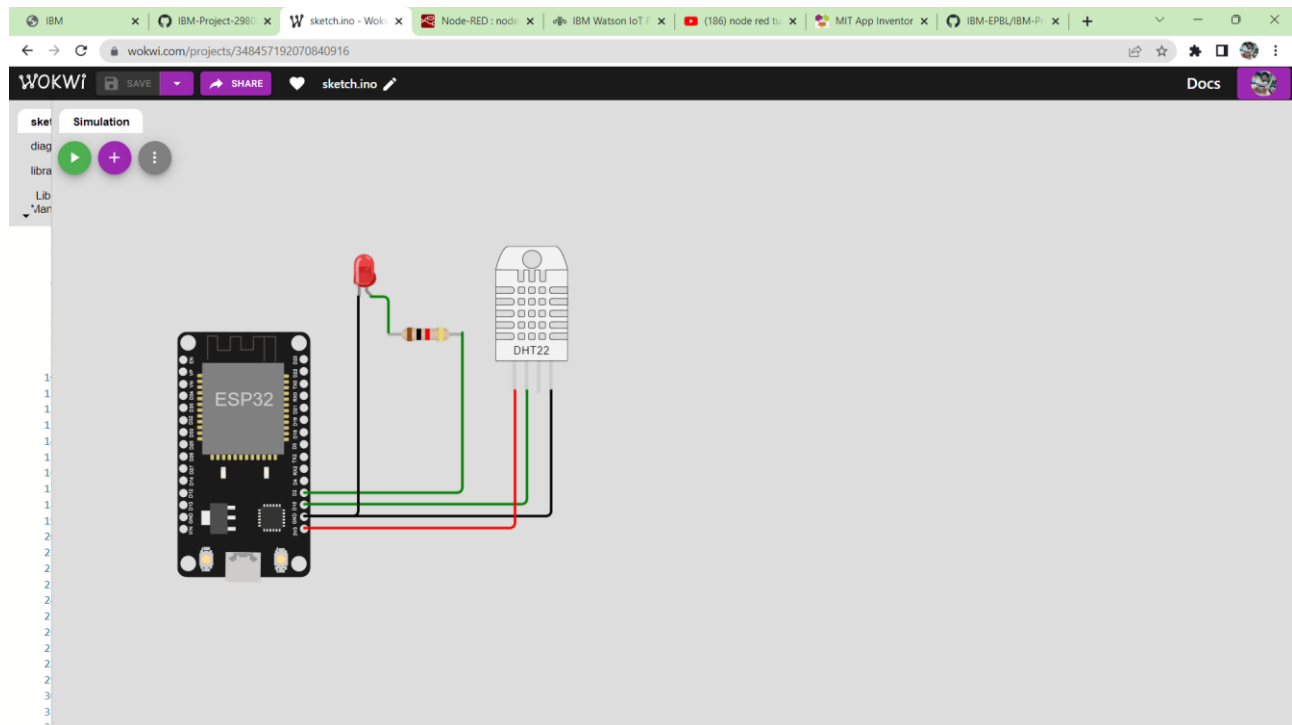
```
1 {
2   "version": 1,
3   "author": "Fershi",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 20.66, "left": -128, "attrs": {} },
7     { "type": "wokwi-dht22", "id": "dht1", "top": -45.26, "left": 125.9, "attrs": {} },
8     {
9       "type": "wokwi-led",
10      "id": "led1",
11      "top": -47.26,
12      "left": 1.24,
13      "attrs": { "color": "red" }
14    },
15    {
16      "type": "wokwi-resistor",
17      "id": "r1",
18      "top": 14.4,
19      "left": 40.23,
20      "attrs": { "value": "1000" }
21    }
22  ],
23  "connections": [
24    [ "esp:TX0", "$SerialMonitor:RX", "", [] ],
25    [ "esp:RX0", "$SerialMonitor:TX", "", [] ],
26    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ],
27    [ "r1:2", "esp:D2", "green", [ "v0" ] ],
28    [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],
29    [ "r1:1", "led1:A", "green", [ "v0" ] ],
30    [ "led1:C", "esp:GND.1", "black", [ "v0" ] ],
31    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ]
32  ]
33 }
```

LIBRARIES TEXT:

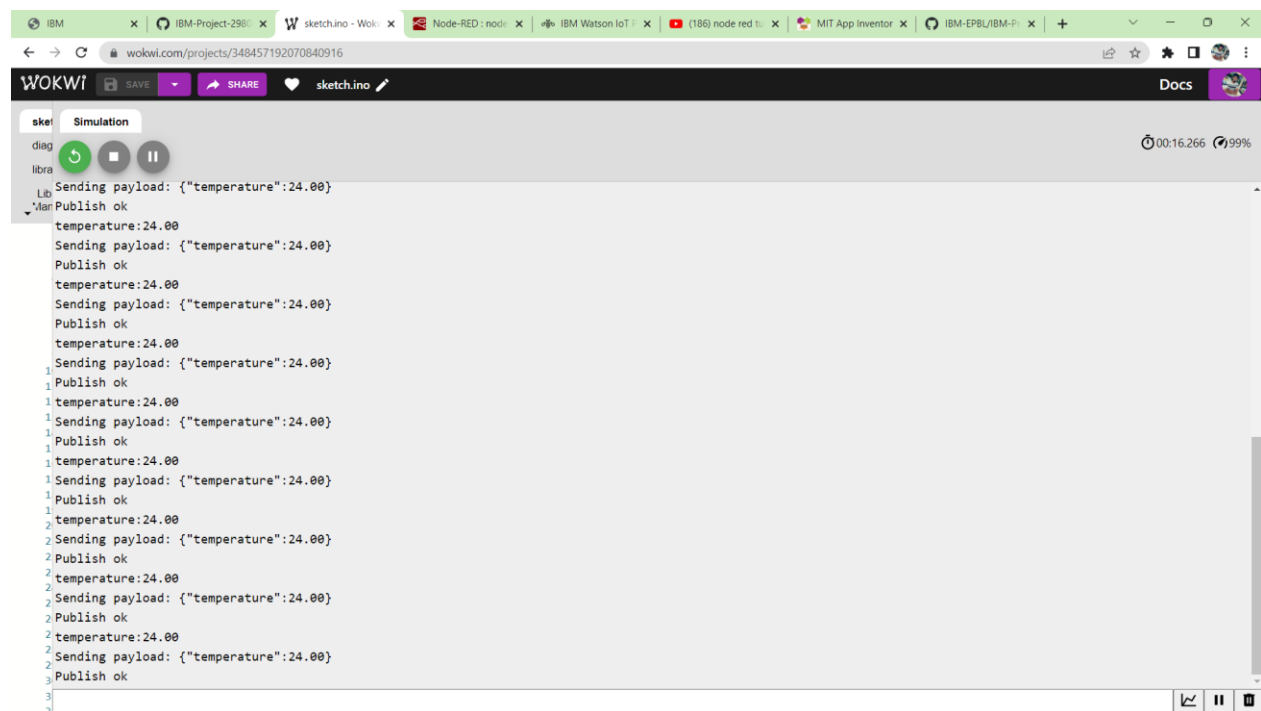


```
1 | Wokwi Library List
2 # See https://docs.wokwi.com/guides/libraries
3
4 # Automatically added based on includes:
5 DHT sensor library
6
7 PubSubClient
```

CIRCUIT:



OUTPUT:



WOKWI LINK:

<https://wokwi.com/projects/348457192070840916>

TESTCASES & OUTPUT

SL.NO	INPUT	OUTPUT	RESULT
-------	-------	--------	--------

01.	Gas:933 Temperature:59.30 Flame:207	Exhaust fan on:TRUE Sprinklers:OFF	Passed
02.	Gas:437 Temperature:59.30 Flame:693	Exhaust fan on:TRUE Sprinklers:OFF	Passed
03.	Gas:218 Temperature:59.30 Flame:369	Exhaust fan on:TRUE Sprinklers:ON	Passed
04.	Gas:2503 Temperature:59.30 Flame:531	Exhaust fan on:TRUE Sprinklers:ON	Passed
05.	Gas:437 Temperature:59.30 Flame:693	Exhaust fan on:TRUE Sprinklers:ON	Passed
06.	Gas:722 Temperature:59.30 Flame:855	Exhaust fan on:TRUE Sprinklers:ON	Passed
07.	Gas:7 Temperature:59.30 Flame:1017	Exhaust fan on:FALSE Sprinklers:ON	Passed
08.	Gas:941 Temperature:59.30 Flame:155	Exhaust fan on:TRUE Sprinklers:OFF	Passed
09.	Gas:226 Temperature: 59.30 Flame:317	Exhaust fan on:TRUE Sprinklers:OFF	Passed
10.	Gas:511 Temperature:59.30 Flame:479	Exhaust fan on:TRUE Sprinklers:ON	Passed
11.	Gas:444 Temperature:59.30 Flame:641	Exhaust fan on:TRUE Sprinklers:ON	Passed

```
ibm.py - C:/Python/Python3.11/ibm.py (3.11.0)
File Edit Format Run Options Window Help

#IBM Watson IOT platform
import wiotp.sdk.device
import time
import random
myConfig={
    "identity": {
        "orgId": "sg5c1o",
        "typeId": "xyz",
        "deviceId": "5678"
    },
    "auth": {
        "token": "567891011"
    }
}
def myCommandCallback(cmd):
    print("Message received from IBM IOT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
    client.connect()
    while True:
        temp=random.randint(-20,125)
        hum=random.randint(0,100)
        myData={'temperature':temp,'humidity':hum}
        client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
        print("Published data Successfully: %s", mydata)
        client.commandCallback=myCommandCallback
        time.sleep(2)
        client.disconnect()
```

Ln: 12 Col: 27

```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Python/Python311/2.py =====
2022-11-15 18:27:44,495 wiotp.sdk.device.client.DeviceClient INFO
successfully: d:s8ovlq:abcd:12345Published data Successfully: %s
({'temperature': 54, 'humidity': 51})
Published data Successfully: %s {'temperature': 34, 'humidity': 53}
Published data Successfully: %s {'temperature': 29, 'humidity': 53}
Published data Successfully: %s {'temperature': 102, 'humidity': 54}
Published data Successfully: %s {'temperature': -3, 'humidity': 62}
Published data Successfully: %s {'temperature': 85, 'humidity': 92}
Published data Successfully: %s {'temperature': 33, 'humidity': 7}
Published data Successfully: %s {'temperature': 20, 'humidity': 74}
Published data Successfully: %s {'temperature': -5, 'humidity': 5}
Published data Successfully: %s {'temperature': 112, 'humidity': 81}
Published data Successfully: %s {'temperature': 58, 'humidity': 5}
Published data Successfully: %s {'temperature': 53, 'humidity': 99}
Published data Successfully: %s {'temperature': 48, 'humidity': 40}
>>>
```


IBM Watson IoT Platform

1vqj0j.internetofthings.ibmcloud.com/dashboard/devices/browse

spurushothaman146@gmail.com
ID: 1vqj0j

Browse Action Device Types Interfaces

criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	node-red	Device	18 Nov 2022 09:26	

Identity Device Information Recent Events State Logs

Device ID: 12345
Device Type: node-red
Date Added: 18 Nov 2022 09:26
Added By: spurushothaman146@gmail.com
Connection Status: Connected
Connection Time: 18 Nov 2022 21:33
Client Address: 50.31.197.64 Insecure

12345 Disconnected raspberrypi Device 11 Nov 2022 11:50

Items per page 50 | 1-2 of 2 items

0 Simulations running

iot-2/type/iot_device/id/1234/evt/event_1/fmt/json : msg.payload : number 70

IBM Watson IoT Platform

1vqj0j.internetofthings.ibmcloud.com/dashboard/devices/browse

spurushothaman146@gmail.com
ID: 1vqj0j

Browse Action Device Types Interfaces

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago

12345 Disconnected raspberrypi Device 11 Nov 2022 11:50

Items per page 50 | 1-2 of 2 items

1 of 1 page

0 Simulations running

node-red-yugla-2022-11-10.eu-gb.mybluemix.net/red/#flow/f90d865089a6c48

Node-RED

Flow 1

Flow 2

debug

filter nodes

- function
- switch
- change
- range
- template
- delay
- trigger
- filter
- OpenWhisk

network

- mqtt in
- mqtt out
- http in
- http response
- http request
- websocket in
- websocket

connected

msg

temperature

temperature

```

11/18/2022, 9:27:22 PM node: a3af4b6ae23c7498
iot-2/type/node-red/id/12345/evl/Data/rml/json : msg : Object
> { topic: "iot-2/type/node-red/id/12345/e-",
  payload: object, deviceId: "12345", deviceType:
"node-red", eventType: "Data" - }

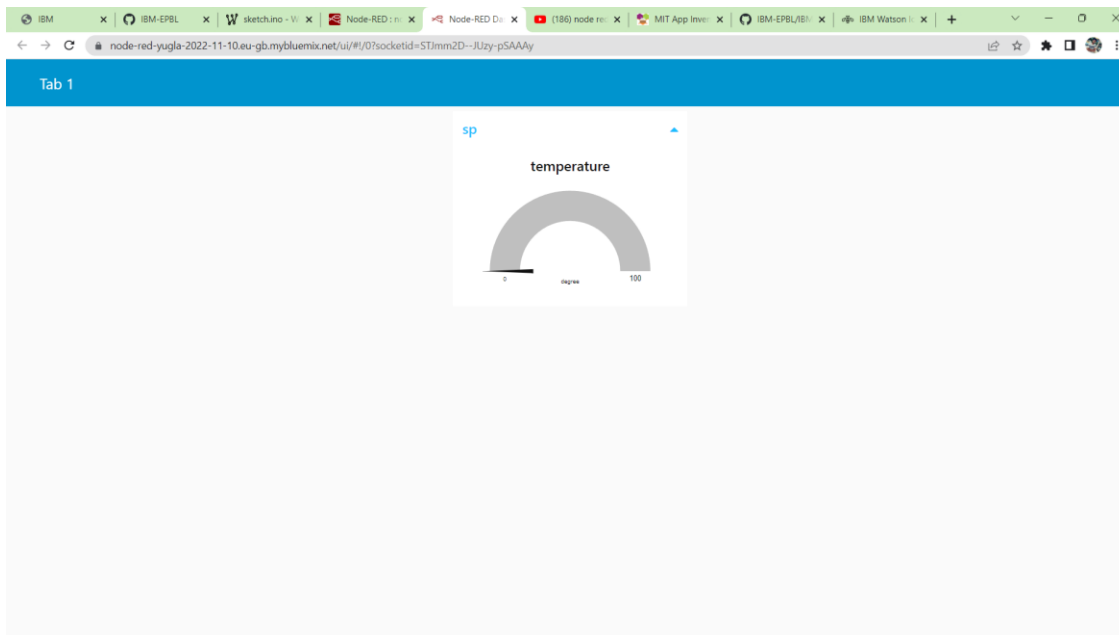
11/18/2022, 9:27:23 PM node: a3af4b6ae23c7498
iot-2/type/node-red/id/12345/evl/Data/rml/json : msg : Object
> { topic: "iot-2/type/node-red/id/12345/e-",
  payload: object, deviceId: "12345", deviceType:
"node-red", eventType: "Data" - }

11/18/2022, 9:31:20 PM node: a3af4b6ae23c7498
iot-2/type/node-red/id/12345/evl/Data/rml/json : msg : Object
> { topic: "iot-2/type/node-red/id/12345/e-",
  payload: object, deviceId: "12345", deviceType:
"node-red", eventType: "Data" - }

11/18/2022, 9:31:21 PM node: a3af4b6ae23c7498
iot-2/type/node-red/id/12345/evl/Data/rml/json : msg : Object
> { topic: "iot-2/type/node-red/id/12345/e-",
  payload: object, deviceId: "12345", deviceType:
"node-red", eventType: "Data" - }

11/18/2022, 9:35:27 PM node: a3af4b6ae23c7498
iot-2/type/node-red/id/12345/evl/Data/rml/json : msg : Object
> { topic: "iot-2/type/node-red/id/12345/e-",
  payload: object, deviceId: "12345", deviceType:
"node-red", eventType: "Data" - }

```



IBM Watson IoT Platform

1vqj0j.internetofthings.ibmcloud.com/dashboard/devices/browse

spurushothaman146@gmail.com
ID: 1vqj0j

Browse Action Device Types Interfaces Add Device

criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	node-red	Device	18 Nov 2022 09:26	

Identity Device Information Recent Events State Logs

Device ID 12345
Device Type node-red
Date Added 18 Nov 2022 09:26
Added By spurushothaman146@gmail.com
Connection Status Connected
Connection Time: 18 Nov 2022 21:33
Client Address: 50.31.197.64 Insecure

12345 Disconnected raspberrypi Device 11 Nov 2022 11:50

Items per page 50 | 1-2 of 2 items

0 Simulations running

IBM Watson IoT Platform

1vqj0j.internetofthings.ibmcloud.com/dashboard/devices/browse

spurushothaman146@gmail.com
ID: 1vqj0j

Browse Action Device Types Interfaces Add Device

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

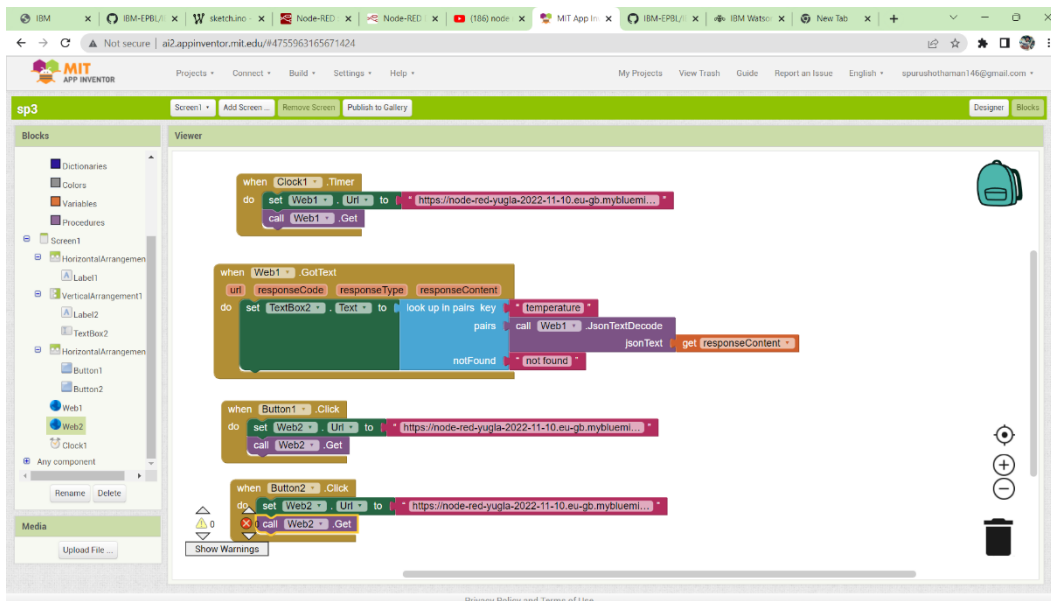
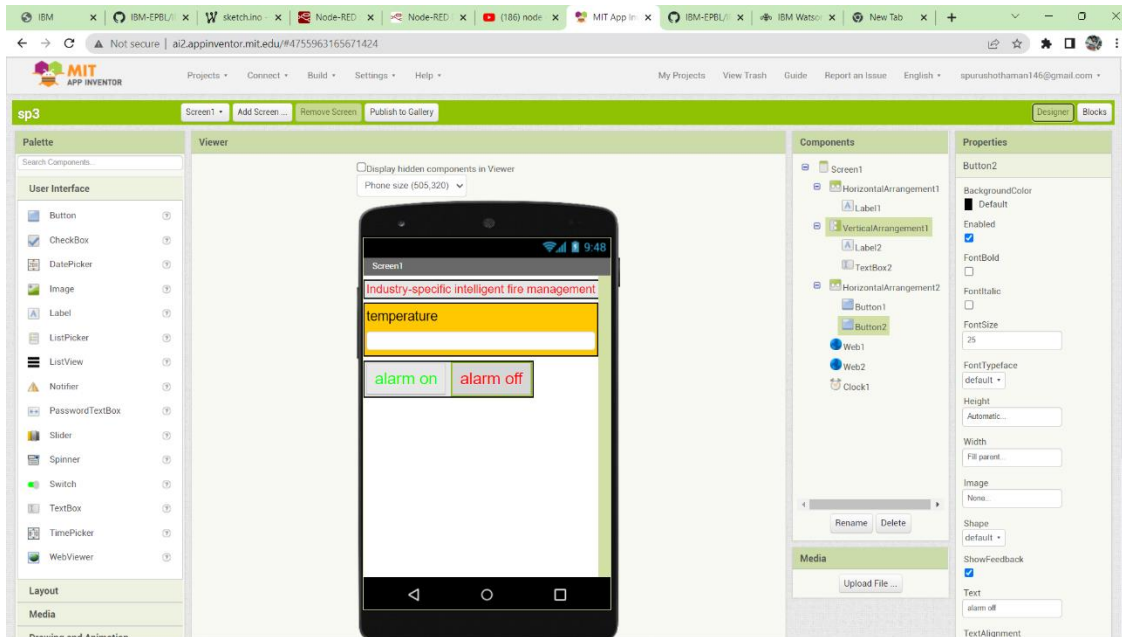
Event	Value	Format	Last Received
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago
Data	{"temperature":24}	json	a few seconds ago

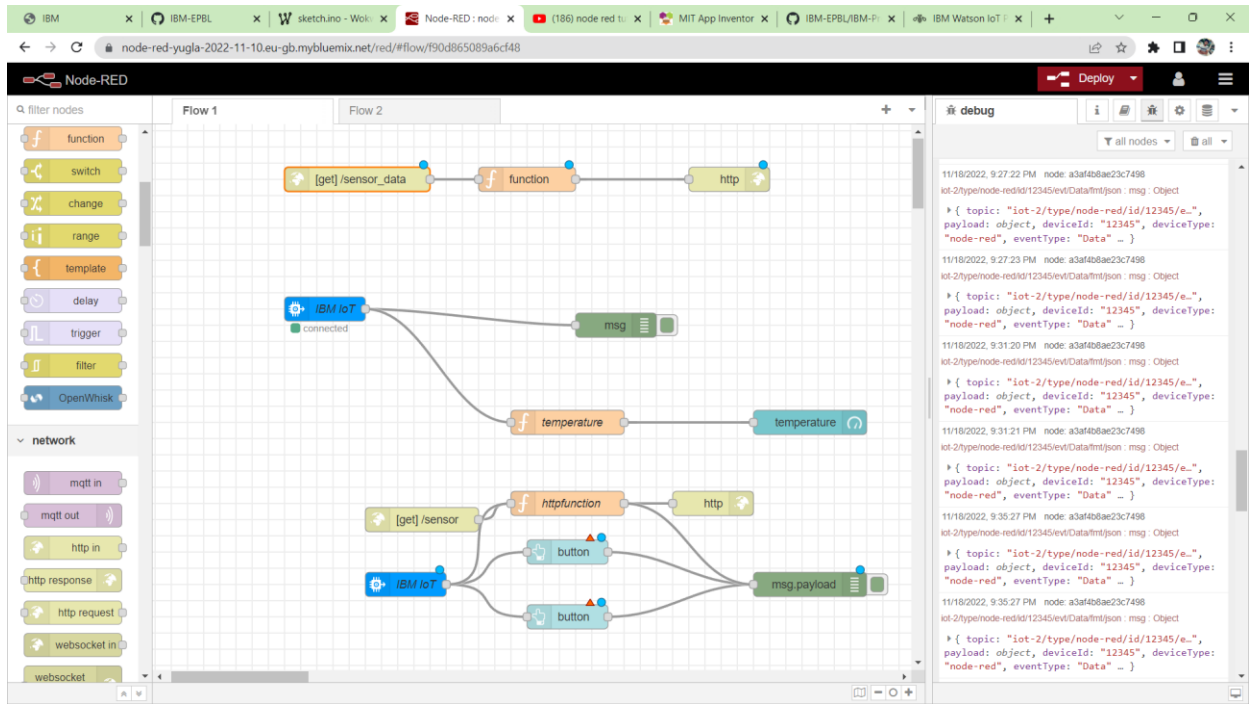
12345 Disconnected raspberrypi Device 11 Nov 2022 11:50

Items per page 50 | 1-2 of 2 items

1 of 1 page

0 Simulations running





IBM Watson IoT Platform

1vqi0j.internetofthings.ibmcloud.com/dashboard/devices/browse

criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator: ☒ ☐ ☐

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	node-red	Device	18 Nov 2022 09:26	

Identity Device Information Recent Events State Logs

Device ID	12345
Device Type	node-red
Date Added	18 Nov 2022 09:26
Added By	spurushothaman146@gmail.com
Connection Status	Connected
	Connection Time: 18 Nov 2022 21:33
	Client Address: 50.31.197.64 Insecure

12345 Disconnected raspberrypi Device 11 Nov 2022 11:50

Items per page 50 | 1-2 of 2 items

0 Simulations running

