

PROJECT DEVELOPMENT PHASE

SPRINT-3

| | |
|--------------|---|
| Date | 9 November 2022 |
| Team ID | PNT2022TMID00340 |
| Project Name | Hazardous Area Monitoring for Industrial Plant powered by IoT |

CODE:

```
#include "DHT.h" // Library for dht22
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
#define GAS_SENSOR 2 //GAS SENSOR MQ-02

DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and typr of dht
connected
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-----credentials of IBM Accounts-----
#define ORG "zfc7n" //IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_Controller" //Device type mentioned in
ibm watson IOT Platform
#define DEVICE_ID "ibmA-4" //Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "GRTl4P*Zvcm4PBiOft" //Token
String data3;
float h, t;
int val;

//----- 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
const int DHT_PIN = 15;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()
{
  val = digitalRead(GAS_SENSOR);
  //val = random(1) ;
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("Temparature:");
  Serial.println(t);
  Serial.print("Humidity:");
  Serial.println(h);
  Serial.print("Gas Sensor:");
  Serial.println(val);
  PublishData(t, h, val);
  delay(1000);
  if (!client.loop())
  {
    mqttconnect();
  }
  delay(1000);
}

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

void PublishData(float temp, float humid, int vol)
{
  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 += "," " \"Humidity\":";
  payload += humid;
  payload += "," "\"Gas Sensor\":";
  payload += val;
  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);
    }
    else
    {
        Serial.println(data3);
    }
    data3 = "";
}
```

WOKIWI SIMULATION:

WOKWI SAVE SHARE IBM_SPRINT-3 Docs

sketch.ino diagram.json libraries.txt Library Manager Simulation

00:21.662 99%

Editing DHT22
Temperature: -18.1°C
Humidity: 37.5%

PNT2022TMID00340

```

122 if (Client.subscribe(subscribetopic))
123 {
124   Serial.println((subscribetopic));
125   Serial.println("subscribe to cmd OK");
126 }
127 else
128 {
129   Serial.println("subscribe to cmd FAILED");
130 }
131
132
133 void callback(char* subscribetopic, byte* payload)
134 {
135   Serial.print("callback invoked for topic: ");
136   Serial.println(subscribetopic);
137   for (int i = 0; i < payloadLength; i++)
138   {
139     //Serial.print((char)payload[i]);
140     data3 += (char)payload[i];
141   }
142   Serial.println("data: " + data3);
143   if (data3 == "lighton")
144   {
145     Serial.println(data3);
146   }
147   else
148   {
149     Serial.println(data3);
150   }
151   data3 = "";
152 }
153
154

```

Sending payload: {"Temperature":27.80,"Humidity":68.50,"Gas Sensor":1}
Publish ok
Temperature:-18.10
Humidity:37.50
Gas Sensor:1
Sending payload: {"Temperature":-18.10,"Humidity":37.50,"Gas Sensor":1}
Publish ok

WOKWI SAVE SHARE IBM_SPRINT-3 Docs

sketch.ino diagram.json libraries.txt Library Manager Simulation

00:21.662 99%

Connecting to ..
Wifi connected
IP address:
10.10.0.2
Reconnecting client to zfc7n.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK

PNT2022TMID00340

```

122 if (Client.subscribe(subscribetopic))
123 {
124   Serial.println((subscribetopic));
125   Serial.println("subscribe to cmd OK");
126 }
127 else
128 {
129   Serial.println("subscribe to cmd FAILED");
130 }
131
132
133 void callback(char* subscribetopic, byte* payload)
134 {
135   Serial.print("callback invoked for topic: ");
136   Serial.println(subscribetopic);
137   for (int i = 0; i < payloadLength; i++)
138   {
139     //Serial.print((char)payload[i]);
140     data3 += (char)payload[i];
141   }
142   Serial.println("data: " + data3);
143   if (data3 == "lighton")
144   {
145     Serial.println(data3);
146   }
147   else
148   {
149     Serial.println(data3);
150   }
151   data3 = "";
152 }
153
154

```

Temperature:56.50
Humidity:70.00
Gas Sensor:1
Sending payload: {"Temperature":56.50,"Humidity":70.00,"Gas Sensor":1}
Publish ok
Temperature:56.50
Humidity:70.00
Gas Sensor:1
Sending payload: {"Temperature":56.50,"Humidity":70.00,"Gas Sensor":1}
Publish ok
Temperature:22.70
Humidity:70.00
Gas Sensor:1
Sending payload: {"Temperature":22.70,"Humidity":70.00,"Gas Sensor":1}
Publish ok
Temperature:22.70
Humidity:70.00
Gas Sensor:1

Note: The slide switch input is given as dummy to Gas sensor

IBM WATSON PLATFORM →

DEVICE EVENT LOG:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays a table of devices. The selected device is 'ibmA-4', which is 'Connected' and has a status of 'ESP32_Controller'. The 'Recent Events' tab is active, showing a live stream of data. The device ID is 'PNT2022TMID00340'.

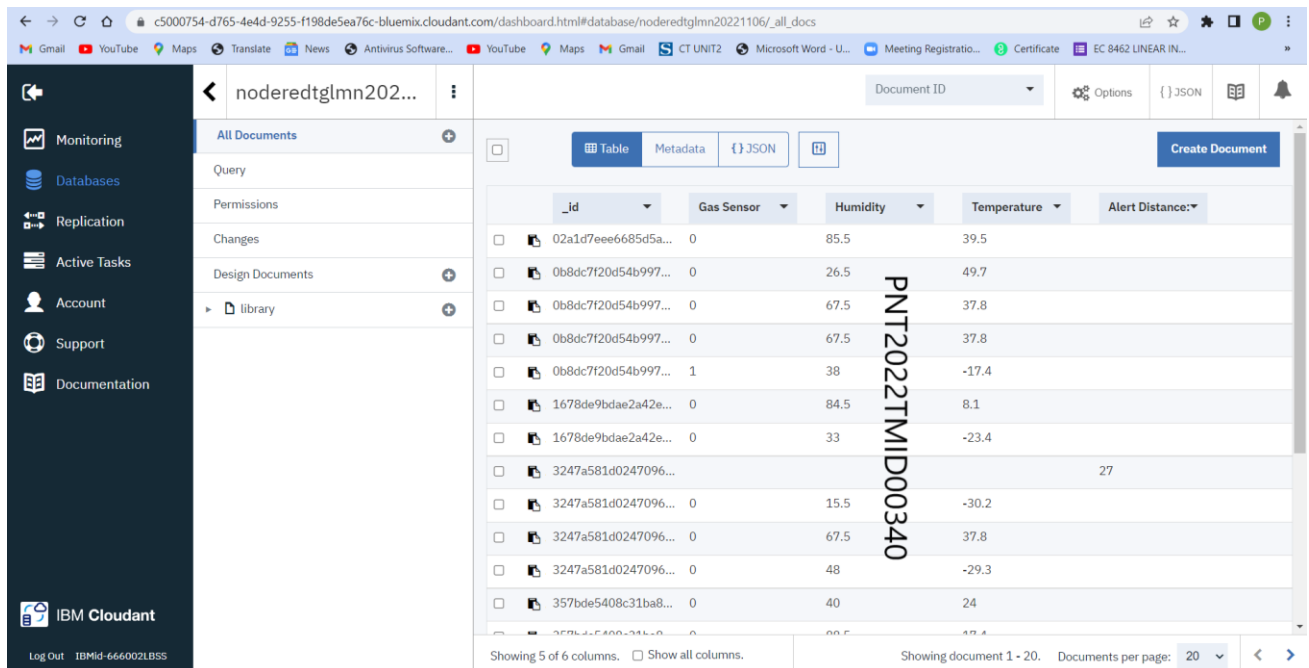
| Event | Value | Format | Last Received |
|-------|---|--------|-------------------|
| Data | {"Temperature":56.5,"Humidity":22.5,"Gas Sens... | json | a few seconds ago |
| Data | {"Temperature":-16.6,"Humidity":52,"Gas Senso... | json | a few seconds ago |
| Data | {"Temperature":30.2,"Humidity":52,"Gas Sensor... | json | a few seconds ago |
| Data | {"Temperature":30.2,"Humidity":55,"Gas Sensor... | json | a few seconds ago |
| Data | {"Temperature":-8.1,"Humidity":38,"Gas Sensor"... | json | a few seconds ago |

DEVICE EVENT PAYLOAD:

The screenshot shows the IBM Watson IoT Platform interface with an 'Event Payload' modal window open. The modal displays the event name 'Data', the time received 'Nov 9, 2022 12:01 AM', and the device ID 'PNT2022TMID00340'. The event payload is shown as a JSON object:

```
1 - | {
2   |   "Temperature": 56.5,
3   |   "Humidity": 36.5,
4   |   "Gas Sensor": 1
5   | }
```

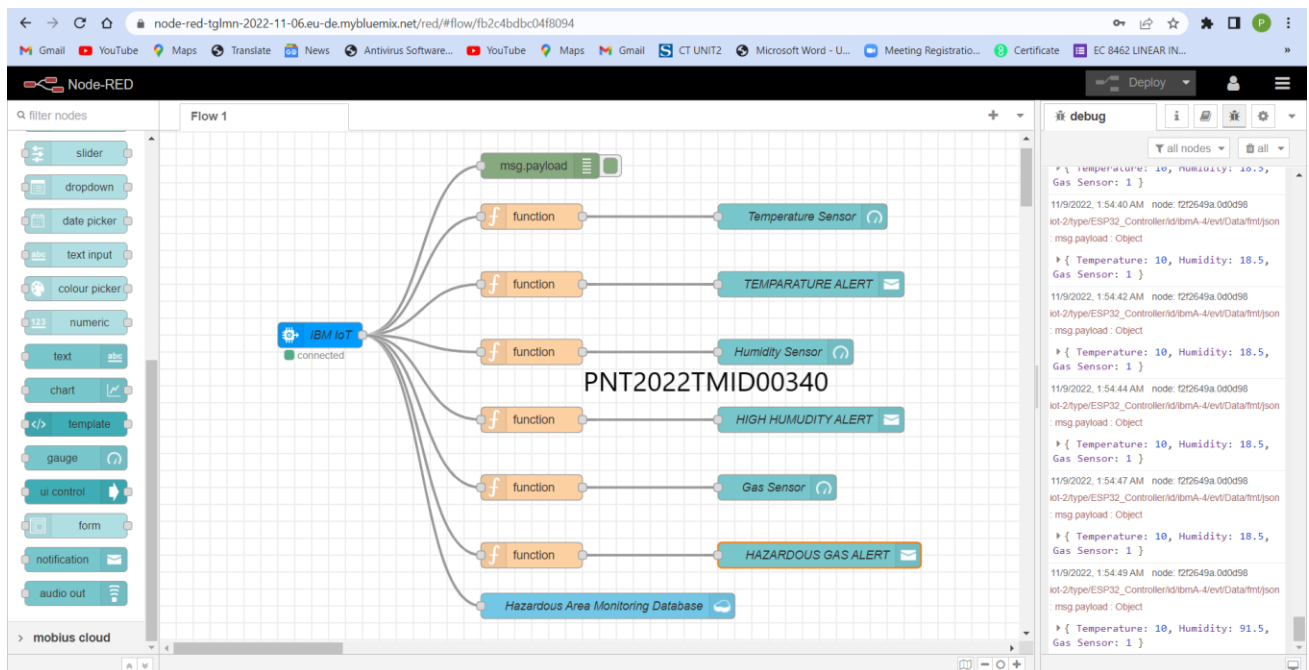
IBM CLOUDANT DB LOG:



| _id | Gas Sensor | Humidity | Temperature | Alert Distance |
|---------------------|------------|----------|-------------|----------------|
| 02a1d7eee6685d5a... | 0 | 85.5 | 39.5 | |
| 0b8dc7f20d54b997... | 0 | 26.5 | 49.7 | |
| 0b8dc7f20d54b997... | 0 | 67.5 | 37.8 | |
| 0b8dc7f20d54b997... | 0 | 67.5 | 37.8 | |
| 0b8dc7f20d54b997... | 1 | 38 | -17.4 | |
| 1678de9bdae2a42e... | 0 | 84.5 | 8.1 | |
| 1678de9bdae2a42e... | 0 | 33 | -23.4 | |
| 3247a581d0247096... | | | | 27 |
| 3247a581d0247096... | 0 | 15.5 | -30.2 | |
| 3247a581d0247096... | 0 | 67.5 | 37.8 | |
| 3247a581d0247096... | 0 | 48 | -29.3 | |
| 357bde5408c31ba8... | 0 | 40 | 24 | |

NODE-RED APP →

DESIGN FLOW:



WEB UI:

