

### Sprint -3

Date	19 November 2022
Team ID	PNT2022TMID06157
Project Name	IOT BASED CROP PROTECTION SYSTEM FOR AGRICULTURE

**Description:**

Implementation of simulation using wokwi online simulator for Detecting temperature and humidity of soil using DHT22 sensor and sending it the IBM IoT Watson Platform

**Code:**

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
#include "DHTesp.h"

const int DHT_PIN = 15;

DHTesp dhtSensor;

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

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

#define ORG "vow9v2"//IBM ORGANITION ID
#define DEVICE_TYPE "SmartCropProtection"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1234567890" //Token
String data3;
float h, 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() {
```

```

Serial.begin(115200);
dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
delay(10);
Serial.println();
wificonnect();
mqttconnect();
}

void loop() {
  TempAndHumidity data = dhtSensor.getTempAndHumidity();
  Serial.println("Temp: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");
  Serial.println("----");
  PublishData(data.temperature,data.humidity);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
  delay(1000);
}

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

void PublishData(float temp, float hum) {
  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 += ",";
  payload += "\"Humidity\":\"";
  payload += hum;
  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 = "";
}

```

## Python Script Output:

The screenshot displays the Wokwi IoT Platform interface. On the left, the 'esp32-dht22.ino' file is open, showing C++ code for an ESP32 microcontroller connected to a DHT22 temperature and humidity sensor. The code includes necessary libraries, defines pins and credentials, and sets up an MQTT client to send sensor data to the IBM Watson IoT Platform. On the right, the 'Simulation' window shows a visual representation of the hardware. The ESP32 is connected to the DHT22 sensor. A control panel for the DHT22 sensor is visible, showing a temperature of 8.9°C and a humidity of 61.5%. Below the simulation, the console output shows the payload being sent: {"Temperature":8.90,"Humidity":61.50}.

```

1  #include <WiFi.h> //library for wifi
2  #include <PubSubClient.h> //library for MQTT
3  #include "DHTesp.h"
4
5  const int DHT_PIN = 15;
6
7  DHTesp dhtSensor;
8
9  void callback(char* topic, byte* payload, unsigned int payloadLength);
10
11 //----- credentials of IBM Accounts -----
12
13 #define ORG "vow9v2" //IBM ORGANITION ID
14 #define DEVICE_TYPE "SmartCropProtection" //Device type mentioned in ibm watson IoT Platform
15 #define DEVICE_ID "12345" //Device ID mentioned in ibm watson IoT Platform
16 #define TOKEN "1234567890" //Token
17 String data3;
18 float h, t;
19
20 //----- customise the above values -----
21 char server[] = ORG".messaging.internetofthings.ibmcloud.com"; // Server Name
22 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform a
23 char subscribTopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type AND
24 char authMethod[] = "use-token-auth"; // authentication method
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
27
28 //-----
29 WiFiClient wificlient; // creating the instance for wificlient
30 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
31
32 void setup() {
33     Serial.begin(115200);
34     dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
35     delay(10);
36     Serial.println();

```

Simulation

Editing DHT22

Temperature: 8.9°C

Humidity: 61.5%

Sending payload: {"Temperature":8.90,"Humidity":61.50}

Publish ok

Temp: 8.90°C

Humidity: 61.5%

---

Sending payload: {"Temperature":8.90,"Humidity":61.50}

Publish ok

IBM Watson IoT Platform

Node-RED : smart-crop-protectiNode-RED DashboardSmartCropProtection - Wokwi A+

vow9v2.internetofthings.ibmcloud.com/dashboard/devices/browse

Graphic Designer M...Meet - Bp-njwz-kebAir-writng TinyML...ADXL3xx Accelerom...Alkha: An Arduino...Circuit design Char...Applications | Clart...IBM CloudSmartCropProtectio...

91761914024@smartinternz.comID: vow9v2

Browse

Action

Device Types

Interfaces

Add Device

>

123

Disconnected

Motor

Device

Nov 18, 2022 4:51 AM

12345

Connected

SmartCropProtection

Device

Nov 12, 2022 4:26 PM

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":8.9,"Humidity":61.5}	json	a few seconds ago
Data	{"Temperature":8.9,"Humidity":61.5}	json	a few seconds ago
Data	{"Temperature":8.9,"Humidity":61.5}	json	a few seconds ago
Data	{"Temperature":8.9,"Humidity":61.5}	json	a few seconds ago
Data	{"Temperature":8.9,"Humidity":61.5}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

0 Simulations running