SPRINT 1

Date	19 October 2022
Team ID	PNT2022TMID38164
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

TEAM MEMBERS:-

BAVADHARANI K	411819106002
SATHISH M	411819106005
ESAKKIRAJAN M	411819106305
ARUNKUMAR A	411819106001

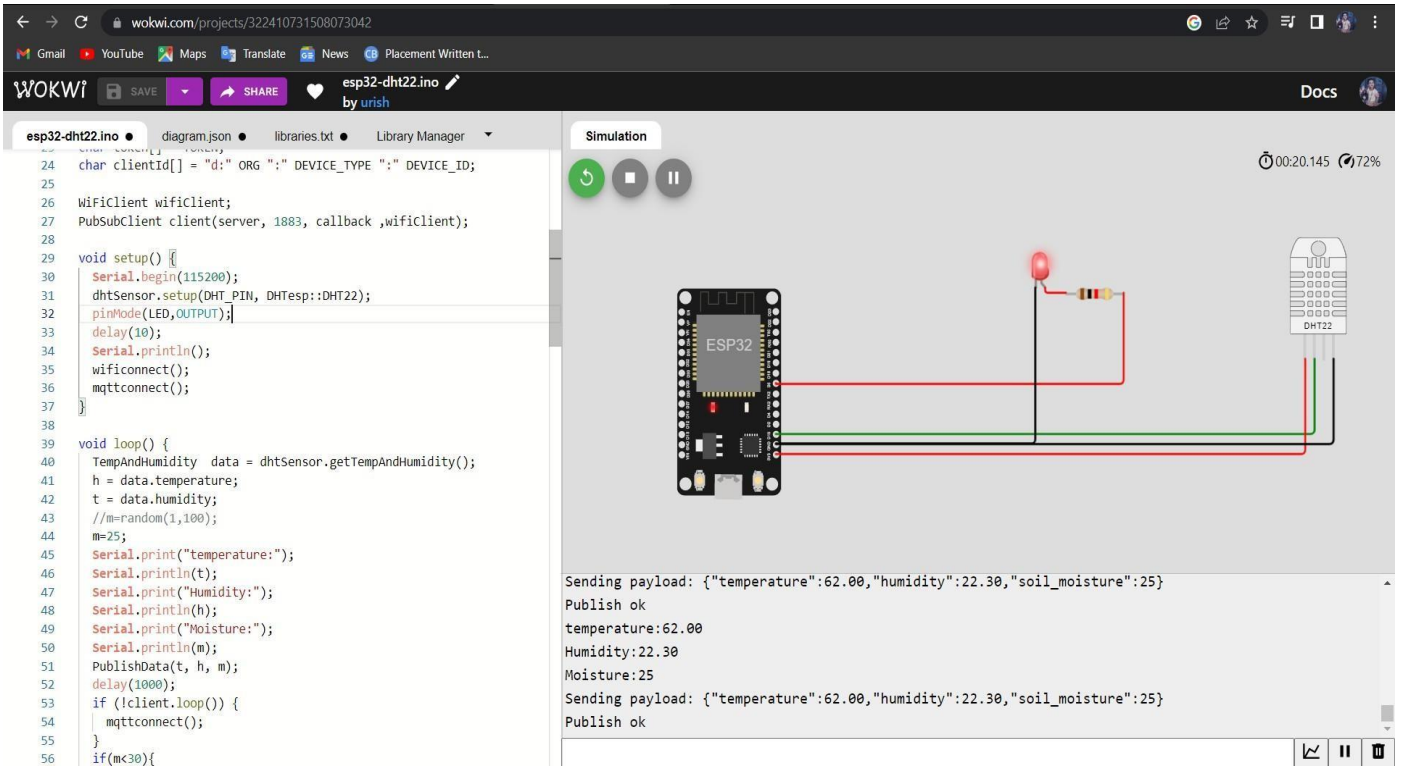
SENSOR WITH ESP32 WITH C++ CODE:-

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

const int DHT_PIN = 15;
#define DHTTYPE DHT11
#define LED 5
DHTesp dhtSensor;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
#define ORG "zxnybt"
#define DEVICE_TYPE "dominators"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
String data3;
float h, t;
int m;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);

void setup() {
```

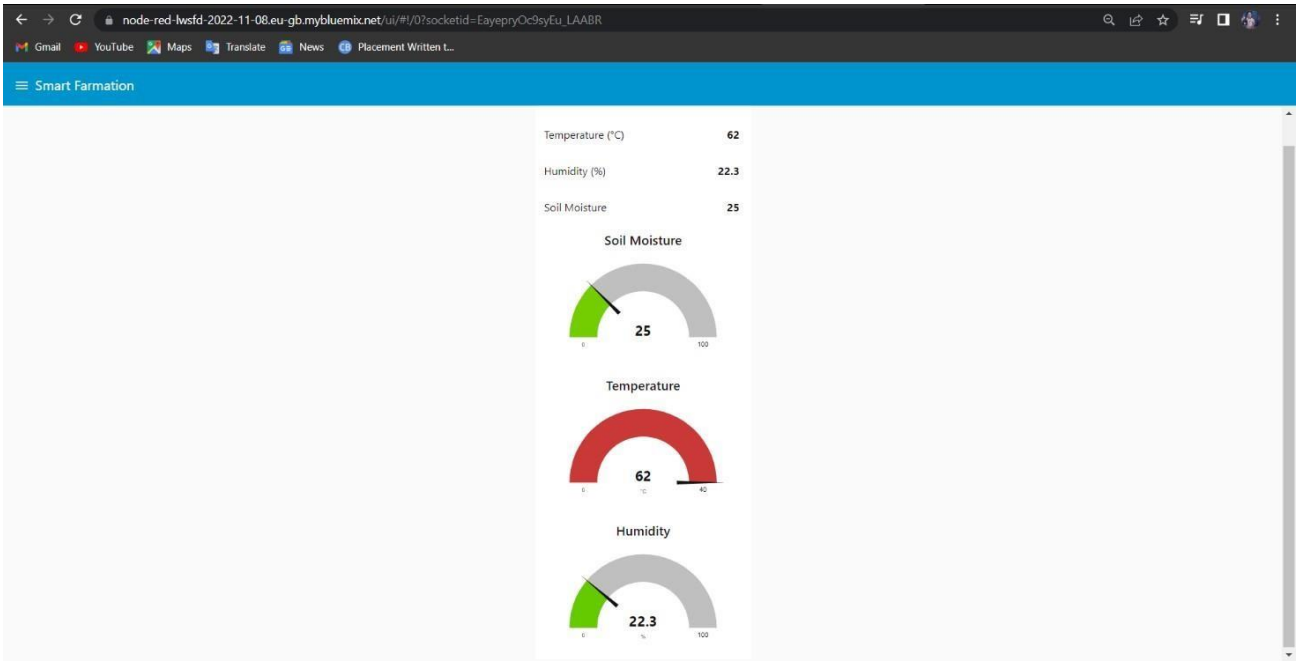
```
  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop() {
  TempAndHumidity data = dhtSensor.getTempAndHumidity();
  h = data.temperature;
  t = data.humidity;
  m=random(1,100);
  //m=25;
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Humidity:");
  Serial.println(h);
  Serial.print("Moisture:");
  Serial.println(m);
  PublishData(t, h, m);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
  if(m<30){
    digitalWrite(LED,HIGH);
  }
  else{
     digitalWrite(LED,LOW);
  }
}
void PublishData(float temp, float humid,int moist) {
  mqttconnect();
  String payload = "{\"temperature\":";
  payload += temp;
  payload += "," "\"humidity\":";
  payload += humid;
  payload += "," "\"soil_moisture\":";
  payload += moist;
  payload += "}";
 Serial.print("Sending payload: ");
 Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
```

```
  } 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()
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);
  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=="1")
{
    Serial.println(data3);
    digitalWrite(LED,HIGH);

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

}
data3="";
}
```



In this website there is no availability of motor also it doesn't have moisture sensor , so instead of motor we use LED and instead of moisture level we are using the random value .



PYTHON SCRIPT FOR CONTROLLING A MOTOR AND GENERATING THE RANDOM SENSOR DATA:-

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


organization = "zxnybt"
deviceType = "dominators"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    for key in cmd.data.keys():
      if key == 'motor':
         if cmd.data['motor'] == 'ON':
            print("MOTOR is turned ON")

         elif cmd.data['motor'] == 'OFF':
```

```
            print("MOTOR is turned OFF")
    try:
       deviceOptions = {"org": organization, "type": deviceType, "id":
    deviceId, "auth-method": authMethod, "auth-token": authToken}
       deviceCli = ibmiotf.device.Client(deviceOptions)


    except Exception as e:
       print("Caught exception connecting device: %s" % str(e))
       sys.exit()

    deviceCli.connect()

    while True:


        temp=random.randint(0,40)
        Humid=random.randint(0,100)
        moist=random.randint(0,40)
        data = { 'temperature' : temp, 'humidity': Humid, 'soil_moisture':moist
    }

        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s
     %%" % Humid, "soil moisture =%s" % moist,"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
    on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback


deviceCli.disconnect()
```