DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

IBM NALAIYA THIRAN PROJECT

Project Development Phase

Date	26 October 2022
Team ID	PNT2022TMID03479
Project Name	IoT Based Smart Crop Protection System for Agriculture
Maximum Marks	8 Marks

Sprint 2:

Display the image and pre-process the level of the Node-RED web UI and display the temperature, humidity, and soil moisture levels. Integrate the buttons in the UI to control the Motors.

Solution Coding C++(Approach):

```
#include <ESP8266WiFi.h>

const char *wifissid = "SSID";

const char *wifipass = "Password";

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

}

void loop() {

// put your main code here, to run repeatedly:

delay(1000);

reconnectWiFi();

}

void reconnectWiFi(){

WiFi.mode(WIFI_STA);

delay(200);

WiFi.begin(wifissid,wifipass);

        while(WiFi.status()!= WL_CONNECTED){

        Serial.print(".");

        delay(500);

}

Serial.println("Connected to: \t");

Serial.println(WiFi.localIP());

#include <Wire.h>

#define Addr 0x40

Wire.beginTransmission(Addr);

// Send humidity measurement command, NO HOLD master

Wire.write(0xF5);
```

```
// Stop I2C transmission
Wire.endTransmission();
delay(500);
// Request 2 bytes of data
Wire.requestFrom(Addr, 2);
// Read 2 bytes of data
// humidity msb, humidity lsb
if(Wire.available() == 2)
{
  data[0] = Wire.read();
  data[1] = Wire.read();
}
float humidity = (((data[0] * 256.0 + data[1]) * 125.0) / 65536.0) - 6;
float cTemp = (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0) - 46.85;
float fTemp = (cTemp * 1.8) + 32;
void reconnect()
{
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("Attempting MQTT connection...");
if (client.connect("ESP8266Client")) {
Serial.println("connected");
}
else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
delay(5000);
    }
  }
}
void loop()
{

if (!client.connected()) {
reconnect();
}

client.publish(Topic to publish, Payload(message to publish), Return value (true or false));
client.loop();
}
```

Console Output (Based on Approach):