Department of Engineering

The Hong Kong Institute of Vocational Education (IVE)

MBS4522 IoT Applications

Mini-project: Mini observatory and automatic watering system (v1.1)

Instructions:

Please follow the detailed manual and complete all of them. You need to submit a report that shows the screenshot results of major steps and answer questions if any. Please also include your copyable source code (e.g. C, C++, Arduino) at the end of the report. Here are some instructions:

1. Write down all your answers/screenshot to a PDF file. (Do not copy the problems/descriptions) Please ensure the answers/screenshot is readable and clear enough.
2. Only one report is required for a group. Although discussion is permitted, the solution must be written by yourself. Plagiarism will not be tolerated. Otherwise, you will get zero mark.
3. Due at 17:30pm, May 12, 2023. Late submission is capped at 70% of the original marks. It will be zero mark if it lates 48 hours or above.

Note:

1. If you are facing a problem, please try to deal it by yourself first. Debugging, problem-solving skill and self-learning are the most important learning points. Take the chance, do not give up easily. Otherwise, you learn nothing!
2. Do not forget that Google, Stack overflow, GitHub are your friends. Try it before you give up or ask someone.

Software tools:

1. Visual Studio Code (VS code)
2. PlatformIO IDE (an extension of VS code)
3. Node-RED
4. Mosquitto (MQTT)

Hardware tools:

1. ESP32 x1
2. DHT22 sensor x1
3. Resistor x2
4. Variable resistor x1
5. LDR (photoresistor) x1
6. 0.96" OLED monitor x1
7. LED x1
8. Router x1

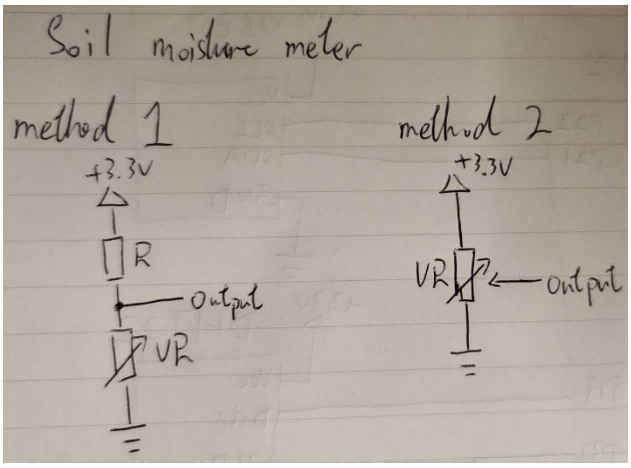
Group: D

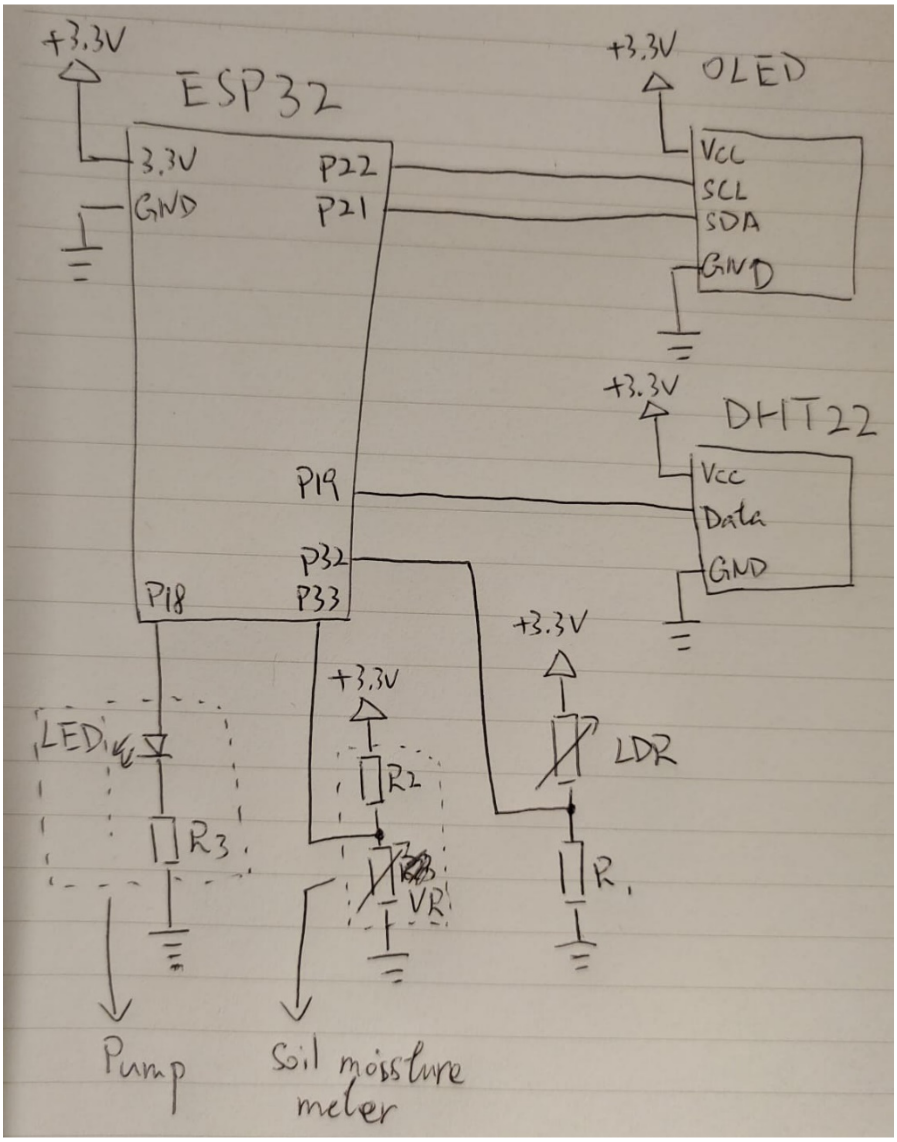
Name: Yu Hoi Lam, Cheung Tsz Chun Noddy, Lai Ho, Lai Ka Ming

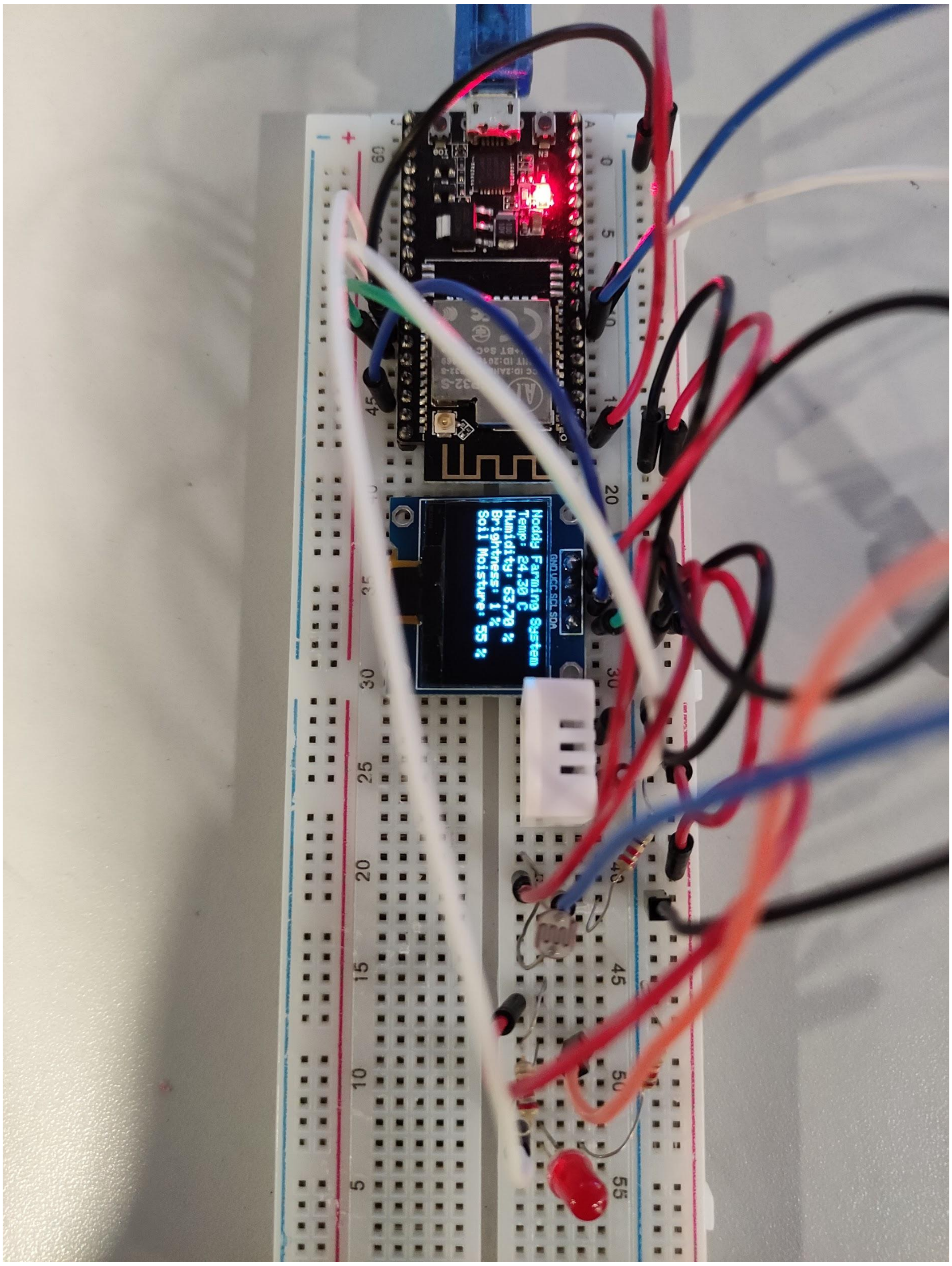
ID: 220135612, 220171174, 220668951, 220140474

Task:

a) Build the circuit and take photo. (15%)
   i. Variable resistance representative soil moisture meter
   ii. LED representative the pump
   iii. You have a choice to use method 1 or 2 for soil moisture meter. The rest follows the pinout and circuit exactly.



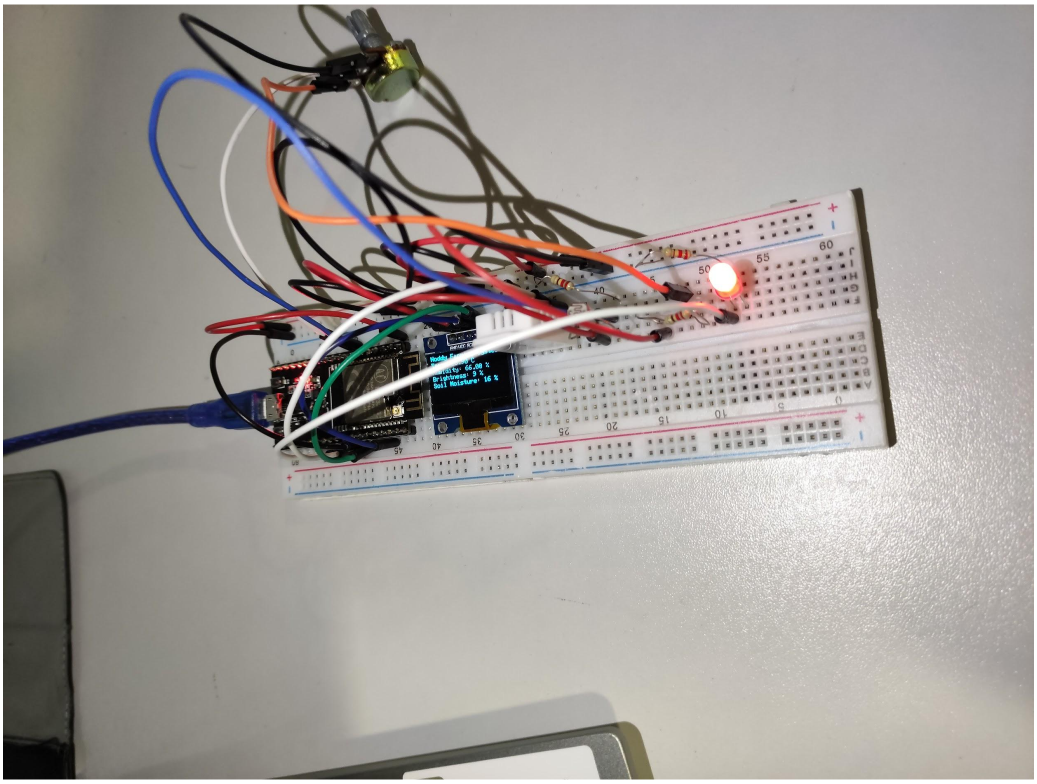




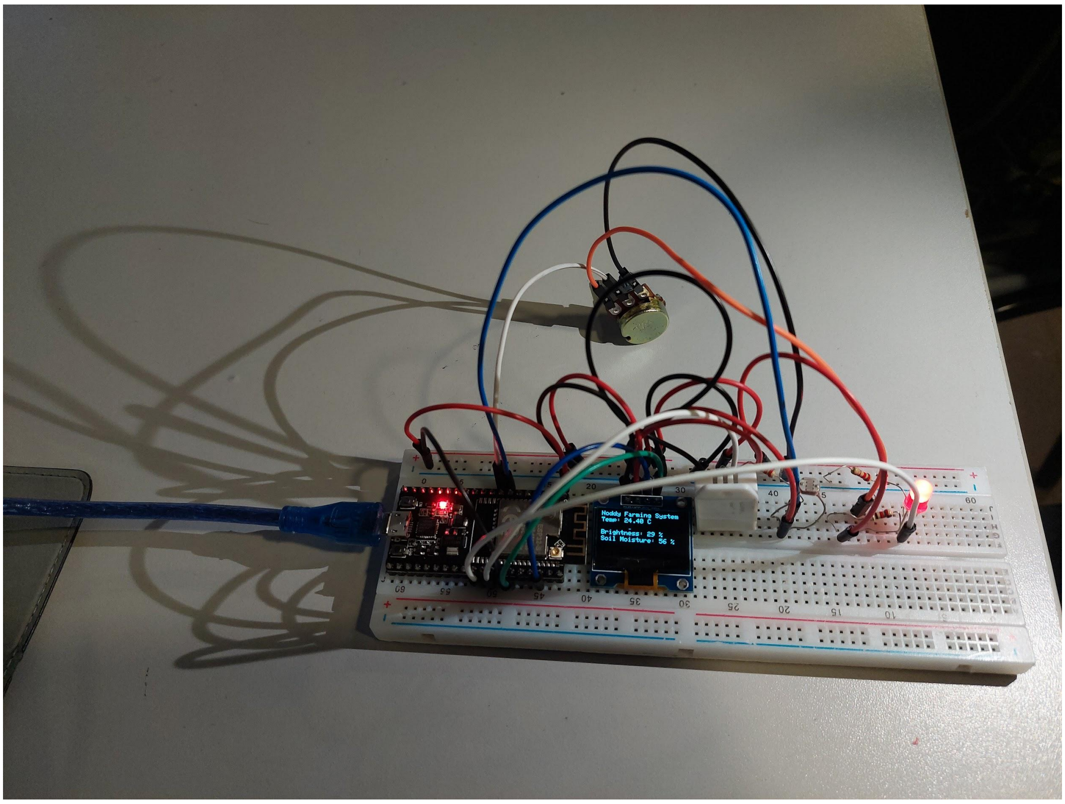
```
Noddy Farming System
Temp: 24.30 C
Humidity: 63.70 %
Brightness: 1 %
Soil Moisture: 55 %
```

b) Code a program on ESP32 and fulfil below requirements. (Total: 20%)
   i. Display the real time temperature, humidity, brightness and soil moisture on OLED monitor. (10%)
   ii. When the soil moisture is less than 30%, turn on the pump and water the plant. Turn it off when the soil moisture level exceeds 65%. (10%)

LED ON:

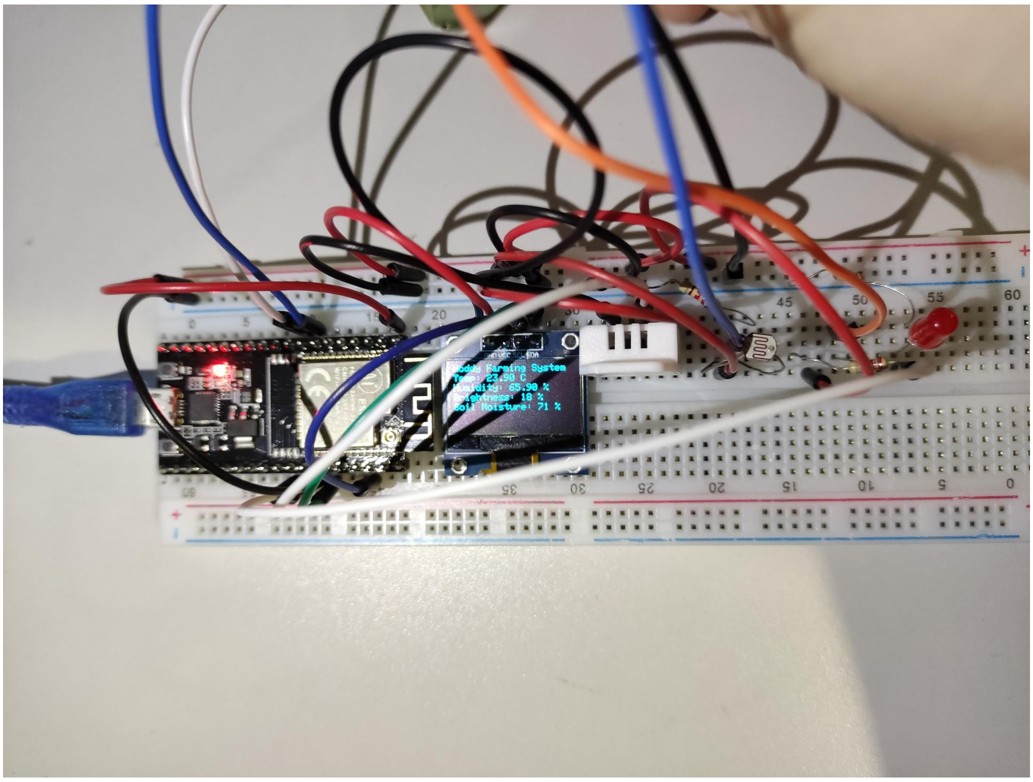


```
Noddy Farm
      .90 C
Humidity: 66.00 %
Brightness: 9 %
Soil Moisture: 16 %
```



```
Noddy Farming System
Temp: 24.40 C

Brightness: 29 %
Soil Moisture: 56 %
```

LED OFF:



Might need to replace the resistor currently using to a smaller resistor to function normally.

Github link:

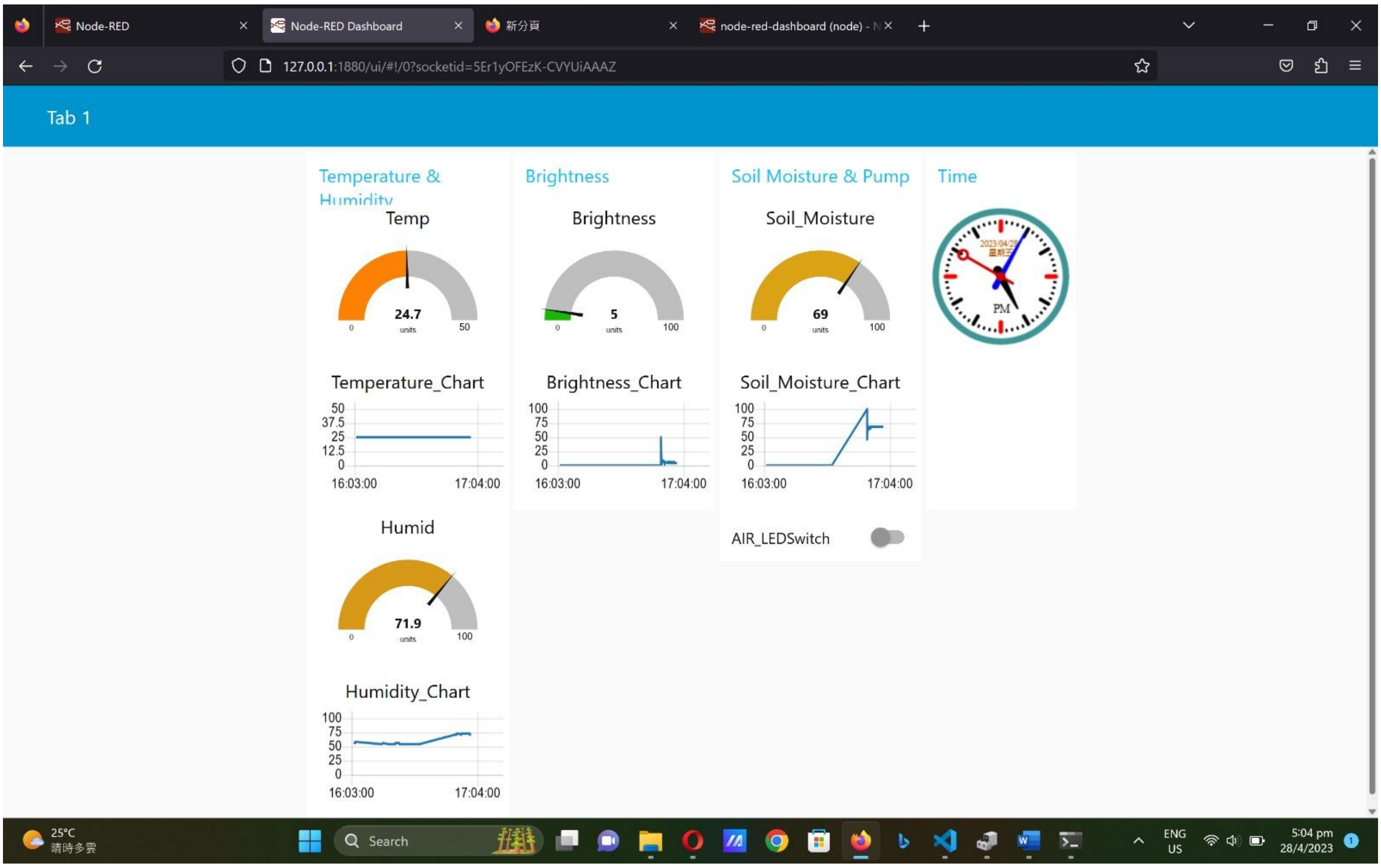
https://github.com/noddycheung/IoT-lesson-2223/blob/571d5e1663805ce398f81921d28a3d142679992b/mini%20project%20part%20A

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>
 #include <Adafruit_Sensor.h>

#define DHTPIN 19
#define DHTTYPE DHT22
#define LDRPIN 32
#define VARPIN 33
```

```
#define LEDPIN 18

DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(128, 64, &Wire, -1);

void setup() {
  Serial.begin(115200);
  pinMode(LEDPIN, OUTPUT);
  dht.begin();

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
}

void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  int brightness = analogRead(LDRPIN);
  int brightnesslevel = map(brightness, 0, 1500, 0, 100);
  if (brightnesslevel > 100){
    brightnesslevel = 100;
  }
  int varValue = analogRead(VARPIN);

  int currentPercentage = map(varValue, 0, 4095, 0, 100);

  if (currentPercentage < 30) {
    digitalWrite(LEDPIN, HIGH);
  } else if (currentPercentage > 65) {
    digitalWrite(LEDPIN, LOW);
  }

  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("Noddy Farming System");
  display.setCursor(0, 10);
  display.print("Temp: ");
  display.print(temperature);
  display.print(" C");
  display.setCursor(0, 20);
  display.print("Humidity: ");
  display.print(humidity);
  display.print(" %");
  display.setCursor(0, 30);
  display.print("Brightness: ");
  display.print(brightnesslevel);
  display.print(" %");
  display.setCursor(0, 40);
  display.print("Soil Moisture: ");
  display.print(currentPercentage);
  display.print(" %");
  display.display();

  delay(1000);
}
```

c) Using Node-RED to build a web and fulfil below requirements. (Total: 65%)

i. Communication by MQTT
ii. Display the real time temperature and humidity (20%)
iii. Display the real time brightness and soil moisture (10%)
iv. Display the history record of temperature, humidity, brightness and soil moisture. (20%)
v. Have a switch to turn on/off the pump. Display the status of the pump also. (10%)
vi. Nice outlook of website (5%)



GitHub link:

https://github.com/noddycheung/IoT-lesson-2223/blob/789698adaa365f7f19f057e63bb2591d66268fb5/IOT%20Q2

```
#include <Arduino.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h>

#include <Adafruit_Sensor.h>
```

```
#include <PubSubClient.h>
#include <WiFi.h>

#define DHTPIN 19
#define DHTTYPE DHT22
#define LDRPIN 32
#define VARPIN 33
#define LEDPIN 18

// Change the credentials below, so your ESP8266 connects to your router
const char* ssid = "Azurix.exe";
const char* password = "CH3COOCH2CH3";


// Change the variable to your Raspberry Pi IP address, so it connects to your MQTT broker
const char* mqtt_server = "test.mosquitto.org";
//For example
//const char* mqtt_server = "192.168.1.106";

// Initializes the espClient. You should change the espClient name if you have multiple ESPs running in your home automation system
WiFiClient espClient;
PubSubClient client(espClient);

DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(128, 64, &Wire, -1);

// This functions connects your ESP8266 to your router
void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
  Serial.println(WiFi.localIP());
}


// This function is executed when some device publishes a message to a topic that your ESP8266 is subscribed to
// Change the function below to add logic to your program, so when a device publishes a message to a topic that
// your ESP8266 is subscribed you can actually do something
void callback(String topic, byte* message, unsigned int length) {
  Serial.print("Message arrived on topic: ");
  Serial.print(topic);
  Serial.print(". Message: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
  Serial.println();

  // Feel free to add more if statements to control more GPIOs with MQTT

  Serial.println();
}

// This functions reconnects your ESP8266 to your MQTT broker
// Change the function below if you want to subscribe to more topics with your ESP8266
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    /*
     YOU MIGHT NEED TO CHANGE THIS LINE, IF YOU'RE HAVING PROBLEMS WITH MQTT MULTIPLE CONNECTIONS
     To change the ESP device ID, you will have to give a new name to the ESP8266.
     Here's how it looks:
       if (client.connect("ESP8266Client")) {
     You can do it like this:
       if (client.connect("ESP1_Office")) {
     Then, for the other ESP:
       if (client.connect("ESP2_Garage")) {
      That should solve your MQTT multiple connections problem
    */
    if (client.connect("ESP32Client", "", "")) {
      Serial.println("connected");
      // Subscribe or resubscribe to a topic
      // You can subscribe to more topics (to control more LEDs in this example)
      client.subscribe("AIR2223_Temp");
      client.subscribe("AIR2223_humid");
      client.subscribe("AIR2223_brightness");
      client.subscribe("AIR2223_soil_moisture");
      client.subscribe("AIR2223_LEDStatus");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}


void setup() {
  Serial.begin(9600);

  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  pinMode(LEDPIN, OUTPUT);
  dht.begin();

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
}


void loop() {
  if (!client.connected()) {
    reconnect();
  }
  if(!client.loop())
    client.connect("ESP32Client");

float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  int brightness = analogRead(LDRPIN);
  int brightnesslevel = map(brightness, 0, 1500, 0, 100);
  if (brightnesslevel > 100){
    brightnesslevel = 100;
  }
  int varValue = analogRead(VARPIN);

  int currentPercentage = map(varValue, 0, 4095, 0, 100);

  if (currentPercentage < 30) {
    digitalWrite(LEDPIN, HIGH);
  } else if (currentPercentage > 65) {
    digitalWrite(LEDPIN, LOW);
  }

  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("Noddy Farming System");
  display.setCursor(0, 10);

  display.print("Temp: ");

  display.print(temperature);

  display.print(" C");

  display.setCursor(0, 20);

  display.print("Humidity: ");

  display.print(humidity);

  display.print(" %");

  display.setCursor(0, 30);

  display.print("Brightness: ");

  display.print(brightnesslevel);

  display.print(" %");

  display.setCursor(0, 40);

  display.print("Soil Moisture: ");

  display.print(currentPercentage);

  display.print(" %");

  display.display();


  delay(1000);


      //Publishes Temperature and Humidity values

    client.publish("AIR2223_Temp", String(temperature).c_str());

    client.publish("AIR2223_humid", String(humidity).c_str());

    client.publish("AIR2223_brightness", String(brightnesslevel).c_str());

    client.publish("AIR2223_soil_moisture", String(currentPercentage).c_str());

    client.publish("AIR2223_LEDStatus", String().c_str());

}
```

- Complete and sign the form. All team members need to agree the weighting and sign.

Name	1. Yu Hoi Lam	2. Cheung Tsz Chun Noddy	3. Lai Ho	4. Lai Ka Ming
ID	1. 220135612	2. 220171174	3. 220668951	4. 220140474
Workload	Node-Red coding Node-Red Layout debugging Report Googling	Core Code Node-Red coding Circuit debugging Wiring OLED Report Googling	mental support(warm and friendly)	N/A
Weighting (Amount is 100%)	45	45	10	0
Signature		張子俊	Ho	銘
Remark				Quitted IVE
Adjustment (For faculty use)				

Tips/references:

- For task c (ii, iii, iv): https://www.tomshardware.com/how-to/raspberry-pi-pico-w-node-red

~~END~~