

**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**





**BÁO CÁO ĐỒ ÁN**

**HỌC KỲ I, NĂM HỌC 2022-2023**

**Học phần:**

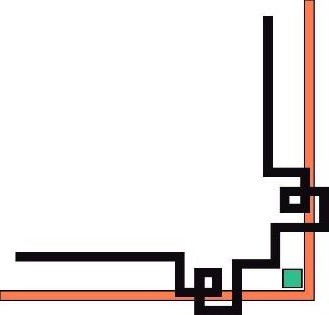
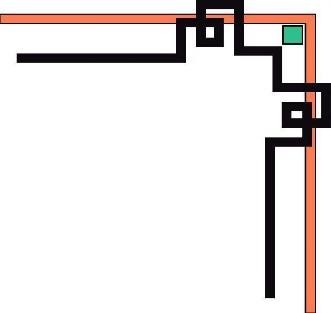
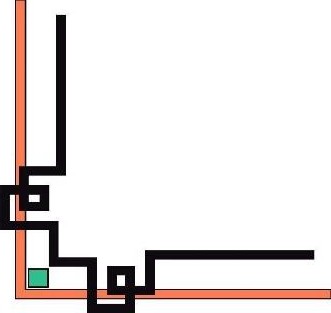
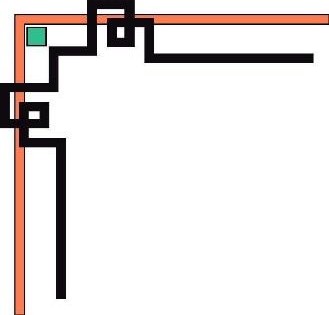
**Internet vạn vật**

***Đề tài:  
 Đọc và xử lý dữ liệu cảm biến thời gian thực với ESP8266  
 Triển khai giám sát và tương tác dữ liệu đám mây qua   
nền tảng đám mây ThingSpeak.***

**Số phách**

*(Do hội đồng chấm thi ghi)*

**Thừa Thiên Huế, ngày 30 tháng 12 năm 2022**



**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**





**ĐỒ ÁN**

**HỌC KỲ I, NĂM HỌC 2022-2023**

**Học phần:**

**Internet vạn vật**

***Đề tài:  
 Đọc và xử lý dữ liệu cảm biến thời gian thực với ESP8266  
 Triển khai giám sát và tương tác dữ liệu đám mây qua   
nền tảng đám mây ThingSpeak.***

**Giảng viên hướng dẫn: Ngô Xuân Cường**

**Lớp: KHDL & TTNT K1**

**Nhóm thực hiện: Nhóm 1**

**Số phách**

*(Do hội đồng chấm thi ghi)*

**Thừa Thiên Huế, ngày 30 tháng 12 năm 2022**

**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**PHIẾU ĐÁNH GIÁ ĐỒ ÁN**

**Học kỳ I, năm học 2022 - 2023**

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1** | **Cán bộ chấm thi 2** |
| **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  **Điểm đánh giá của CBCT1:**  Bằng số: .........................................  Bằng chữ: ....................................... | **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  **Điểm đánh giá của CBCT2:**  Bằng số: .........................................  Bằng chữ: ....................................... |

Điểm kết luận: ...........................................................................................................

Bằng số:.....................................................................................................................

Bằng chữ: .................................................................................................................

*Thừa Thiên Huế, ngày 30 tháng 12 năm 2022*

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ và tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ và tên)* |

**DANH SÁCH THÀNH VIÊN NHÓM 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **Mã số SV** | **Lớp** |
| 1 | Sử Thành Công | 20E1020045 | KHDL-TTNT K1 |
| 2 | Trần Đoàn Ngọc Duy | 20E1020014 | KHDL-TTNT K1 |
| 3 | Hồ Thị Thanh Ngân | 20E1020075 | KHDL-TTNT K1 |
| 4 | Đặng Văn Thắng | 20E1020065 | KHDL-TTNT K1 |
| 5 | Mai Thị Phúc Trâm | 20E1020080 | KHDL-TTNT K1 |
| 6 | Trần Cao Thục Uyên | 20E1020059 | KHDL-TTNT K1 |

**MỤC LỤC**

[Phần 1: ĐỀ TÀI 1](#_Toc123241781)

[Phần 2: THỰC HÀNH 1](#_Toc123241782)

[**1.** **Linh kiện và thiết bị** 1](#_Toc123241783)

[**2.** **Thư viện sử dụng** 2](#_Toc123241784)

[**3.** **Phương pháp** 3](#_Toc123241785)

[**3.1.** **Moving Average filter – Bộ lọc trung bình trượt** 3](#_Toc123241786)

[**3.2.** **Nền tảng đám mây ThingSpeak** 3](#_Toc123241787)

[**4.** **Thực hành** 4](#_Toc123241788)

[**4.1.** **Sơ đồ mạch** 4](#_Toc123241789)

[**4.2.** **Mạch thực tế** 4](#_Toc123241790)

[**4.3.** **Thiết lập ThingSpeak** 5](#_Toc123241791)

[**4.3.1.** **ThingSpeak Channel** 5](#_Toc123241792)

[**4.3.2.** **MATLAB Analysis** 5](#_Toc123241793)

[**4.3.3.** **TimeControl** 5](#_Toc123241794)

[**4.3.4.** **MATLAB Visualizations** 6](#_Toc123241795)

[**4.4.** **Trình bày Code** 6](#_Toc123241796)

[**5.** **Kết quả thực hành** 10](#_Toc123241797)

[**a)** **Kết quả hiển thị trên màn hình LCD** 10](#_Toc123241798)

[**b)** **Kết quả hiển thị trên màn hình Serial Monitor** 11](#_Toc123241799)

[**c)** **Kết quả hiện thị trên cửa sổ Serial Ploter** 11](#_Toc123241800)

[**d)** **Kết quả khi tính giá trị trung bình cảm biến bằng MATLAB Analysis** 12](#_Toc123241801)

[**e)** **Kết quả thu được khi vẽ đồ thị bằng bằng MATLAB Visualizations** 12](#_Toc123241802)

[**f)** **Kết quả ghi lại trên ThingSpeak** 13](#_Toc123241803)

[**g)** **Kết quả ghi lại trên ThingView (ThingSpeak App)** 14](#_Toc123241804)

[**h)** **Xuất giá trị và lưu trữ vào file CSV** 14](#_Toc123241805)

[**6.** **Kết luận** 15](#_Toc123241806)

[TÀI LIỆU THAM KHẢO 16](#_Toc123241807)

# Phần 1: ĐỀ TÀI

Sử dụng **ESP8266** đọc, sử dụng bộ lọc tín hiệu và hiển thị các thông số cảm biến trên màn hình LCD, gửi dữ liệu cảm biến thời gian thực lên nền tảng đám mây **ThingSpeak**, xử lý số liệu và trực quan hóa dữ liệu trên đám mây (hoặc app).

Thời gian làm việc của hệ thống trong 1 tuần. Tính giá trị trung bình cảm biến trong từng ngày và hiển thị giá trị đó trên màn hình màn hình LCD.

# Phần 2: THỰC HÀNH

1. **Linh kiện và thiết bị**

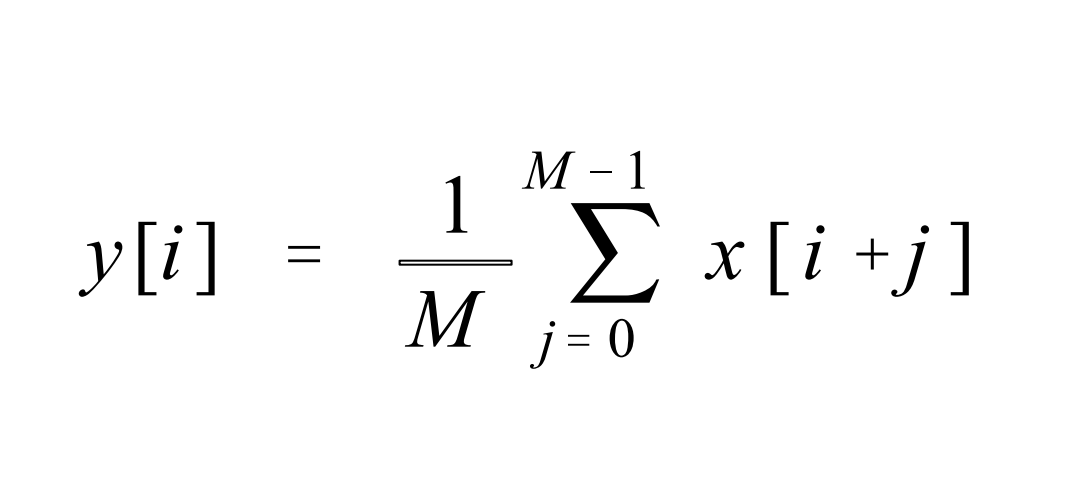
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Số lượng** | **Chức năng** | **Hình ảnh** |
| 1 | Kit RF thu phát Wifi ESP8266 NodeMCU Ai-Thinker | 1 | Vi điều khiển chính |  |
| 2 | Màn hình LCD I2C 1602 | 1 | Hiển thị |  |
| 3 | Cáp micro USB | 1 | Nối máy tính |  |
| 4 | Module Cảm biến nhiệt độ và độ ẩm DHT 22 | 1 | Cảm biến | https://lh6.googleusercontent.com/bY7_KD7To19qGsPQ5urlqiBCDkfH-aMngpJIbUejN1J__tsp9DLvGgK0w0lbVFB9eBUP1NgTKI560VT-lylM6miYaYF1cUR18EO3-tjDW50tIJw49GOssOK5eWYZGl7w7c5ntPrA-4WH37gbGzebnJ5PWqLQ7n4Z4859A8NjvNFbh61uZcM4S9JwivlomA |
| 5 | Module Cảm biến độ ẩm đất | 1 | Cảm biến | Soil Moisture Sensor Module Pinout, Features, Specs & Circuit |
| 6 | Dây cắm cái-cái | 10 | Nối mạch |  |

1. **Thư viện sử dụng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thư viện** | **Chức năng** | **Link** |
| 1 | DHT sensor library | Thư viện Arduino cho cảm biến nhiệt độ/độ ẩm DHT | [**https://github.com/adafruit/DHT-sensor-library**](https://github.com/adafruit/DHT-sensor-library) |
| 2 | LiquidCrystal\_I2C | Thư viện Arduino cho màn hình LCD I2C | [**https://github.com/johnrickman/LiquidCrystal\_I2C**](https://github.com/johnrickman/LiquidCrystal_I2C) |
| 3 | ESP8266WiFi | Thư viện Arduino cho ESP8266 kết nối Wifi | [**https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi**](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi) |
| 4 | Thingspeak-arduino | Thư viện giao tiếp ThingSpeak cho Arduino, ESP8266 và ESP32 | [**https://github.com/mathworks/thingspeak-arduino**](https://github.com/mathworks/thingspeak-arduino)**?** |

1. **Phương pháp**
2. **Moving Average filter – Bộ lọc trung bình trượt**

Bộ lọc trung bình động lấy giá trị trung bình của M giá trị đầu vào cuối cùng trong tín hiệu và tính trung bình chúng để tạo ra đầu ra.

 (1)

Tham số thực duy nhất có thể được kiểm soát trong bộ lọc trung bình động là kích thước cửa sổ [window size], là số lượng các mục tín hiệu trước đó có thể được tính trung bình cùng nhau.

Nếu cửa sổ quá nhỏ, tín hiệu vẫn có thể bị nhiễu. Tuy nhiên, nếu cửa sổ quá lớn, thông tin quan trọng trong tín hiệu có thể bị mất.

Chọn kích thước cửa sổ phù hợp là một vấn đề cần nhiều thử nghiệm. Trong bài thực hành này nhóm chúng em chọn kích thước của sổ

**SMOOTHING\_WINDOW\_SIZE = 10**

1. **Nền tảng đám mây ThingSpeak**

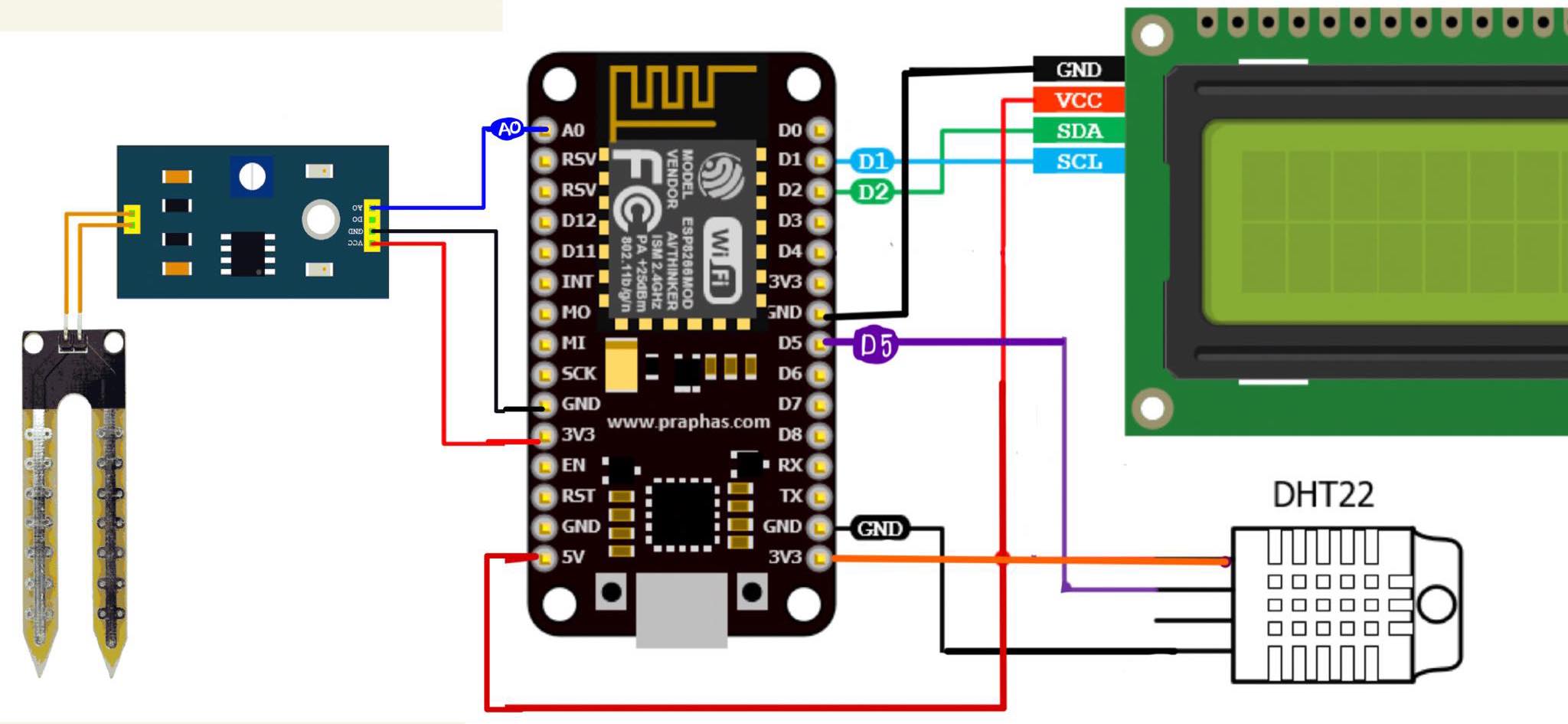
|  |  |
| --- | --- |
| ThingSpeak là một dịch vụ nền tảng phân tích IoT của MathWorks®, nhà sản xuất MATLAB® và Simulink®. (2)  ThingSpeak cho phép tổng hợp, trực quan hóa và phân tích các luồng dữ liệu trực tiếp trên đám mây. ThingSpeak cung cấp hình ảnh trực quan tức thì về dữ liệu do thiết bị đăng tải. | C:\Users\ACER\Desktop\iot\download.png |

Với khả năng thực thi mã MATLAB® trong ThingSpeak, điều này giúp phân tích trực tuyến và xử lý khi có dữ liệu. ThingSpeak thường được sử dụng để tạo nguyên mẫu (prototype) và chứng minh khái niệm về các hệ thống IoT yêu cầu phân tích.

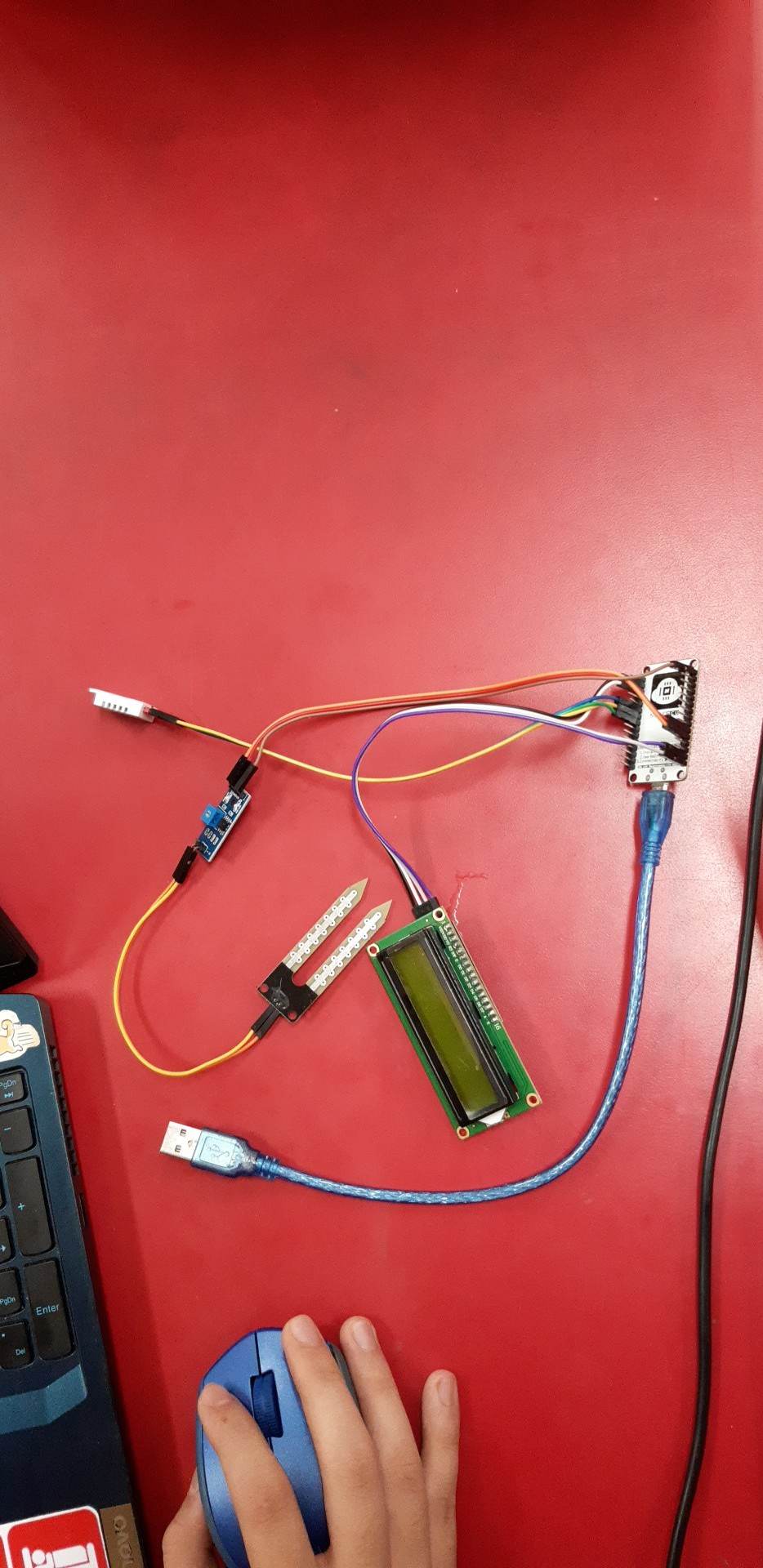
* Ngoài ra, còn hỗ trợ người dùng xây dựng một hệ thống IoT mà không cần thiết lập máy chủ hoặc phát triển phần mềm web.
* Đối với các hệ thống IoT có quy mô vừa và nhỏ, ThingSpeak cung cấp một giải pháp lưu trữ có thể được sử dụng trong sản xuất.

1. **Thực hành**
2. **Sơ đồ mạch**

|  |  |  |
| --- | --- | --- |
| **Module** | **Module PIN** | **ESP8266 PIN** |
| DHT22 | Data out | D5 |
| LCD I2C 1602 | SDA  SCL | D2  D1 |
| Cảm biến độ ẩm đất | A0 | A0 |

****

1. **Mạch thực tế**



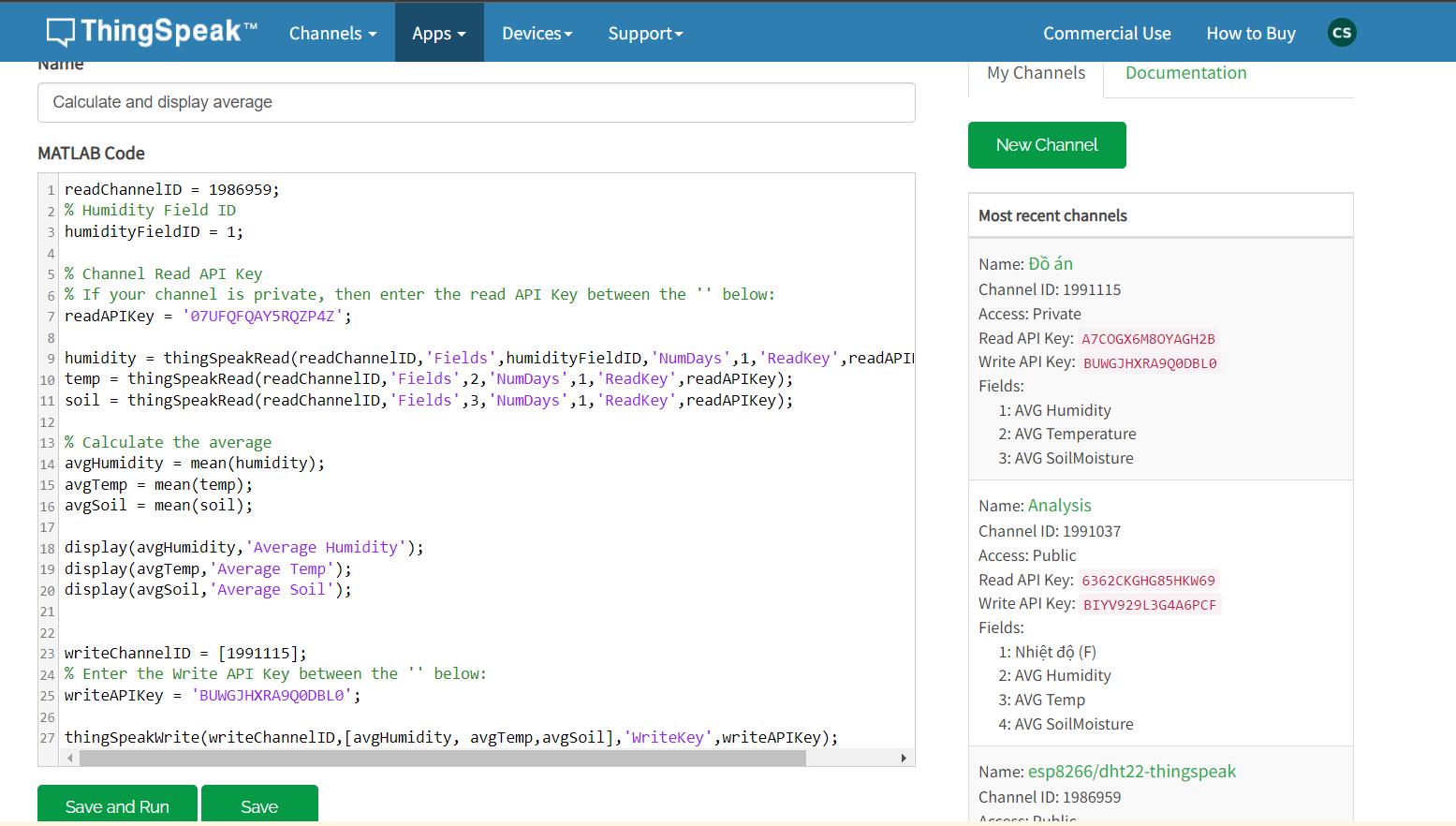
1. **Thiết lập ThingSpeak**
2. **ThingSpeak Channel**

* **Tạo channel**

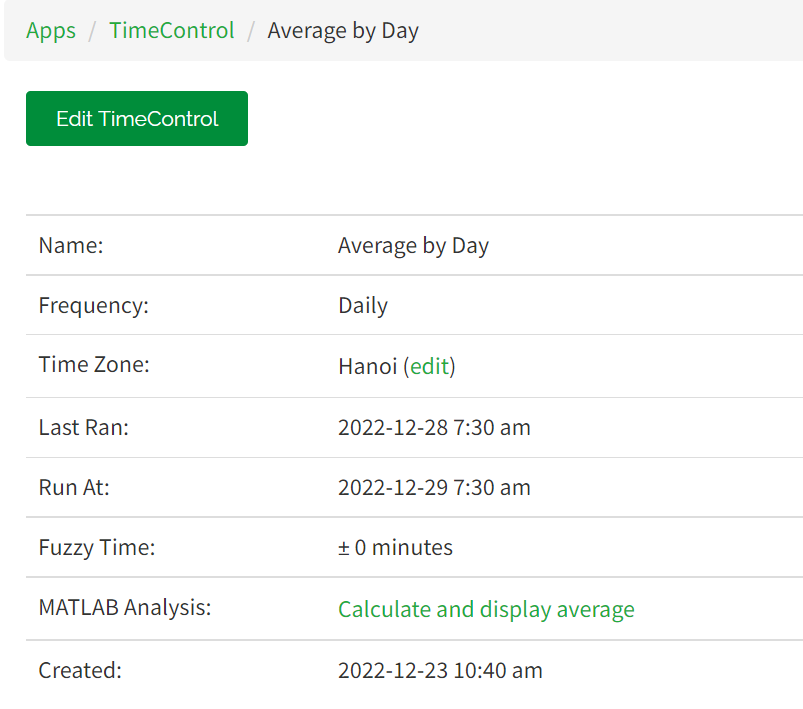
|  |  |
| --- | --- |
| Kênh để lưu trữ dữ liệu từ ESP8266 | Kênh để tổng hợp sau khi qua xử lý |
|  |  |

1. **MATLAB Analysis**

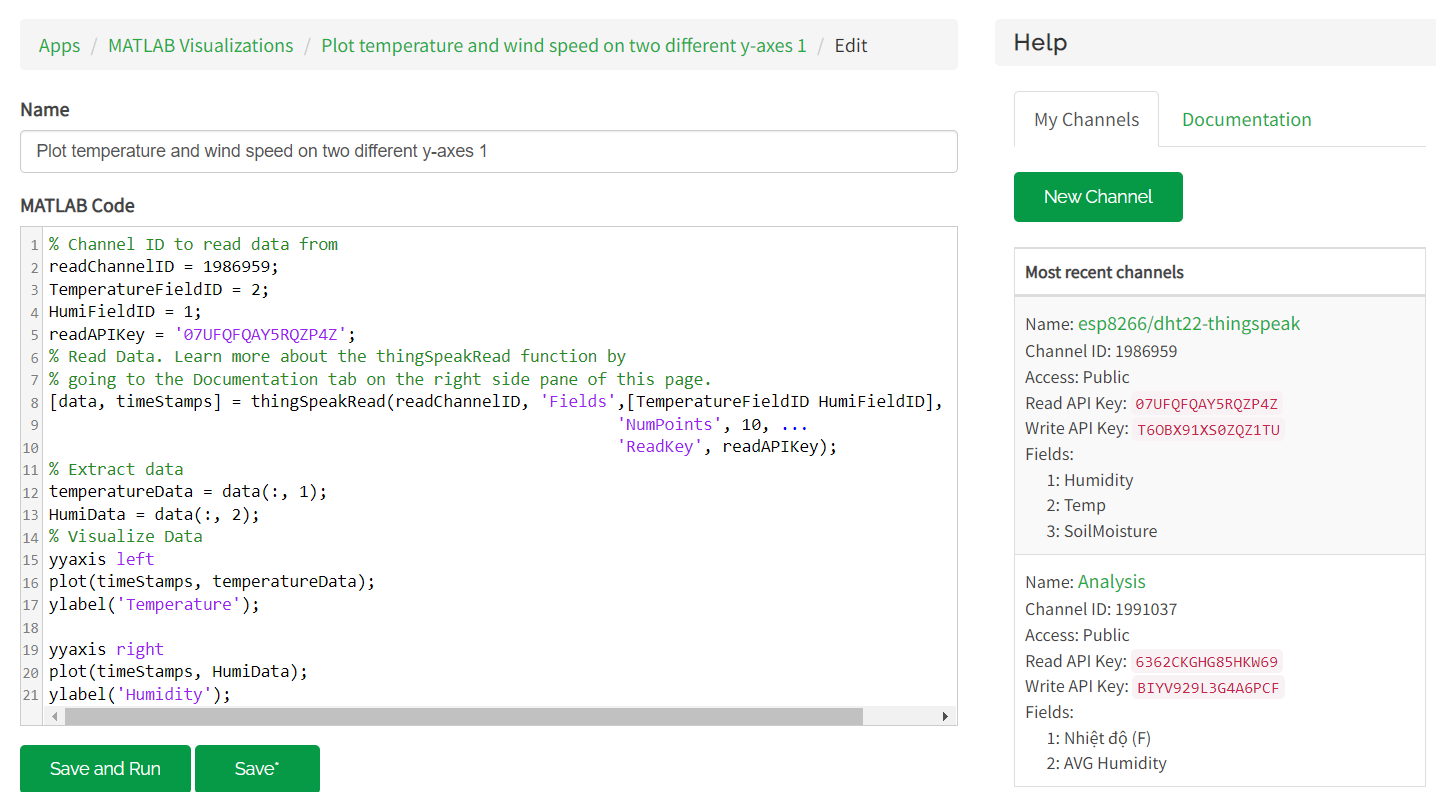
* **Tính giá trị trung bình từng ngày**

****

1. **TimeControl**



1. **MATLAB Visualizations**



1. **Trình bày Code**

///// Khai báo thư viện

#include <ESP8266WiFi.h>  /// Kết nối Wifi

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#include "ThingSpeak.h"

// The Arduino Uno ADC is 10 bits (thus, 0 - 1023 values)

#define MAX\_ANALOG\_INPUT\_VAL 1023

/// DHT config

#define DHTTYPE DHT22   // DHT 22

#define DHTPIN D5 //Chân Out của cảm biến nối chân số 5 ESP8266

const int SOIL\_MOIST\_PIN = A0;

/// WiFi Settings

const char\* ssid = "stc";    // tên wifi

const char\* password = "suthanhcon";  // mật khẩu WiFi

WiFiClient client;

// ThingSpeak Settings

const int channelID = 1986959; // ThingSpeak Channel ID

String writeAPIKey = "T6OBX91XS0ZQZ1TU"; // ThingSpeak write API key

const char\* server = "api.thingspeak.com";

// Khởi tạo các biến

float humi;

float temp;

float soil;

// Xác định số lượng mẫu quan sát (kích thước mảng).

// Con số này càng lớn, giá trị lọc ra sẽ càng mượt, nhưng sẽ phản hồi càng chậm.

// Sử sử dụng giá trị này để xác định kích thước của mảng đọc.

const int SMOOTHING\_WINDOW\_SIZE = 10; // 10 giá trị trong mảng

int curReadIndex = 0;                // vị trí ban đầu trong mảng

/// Humidity moving AVG config

float humi\_samples[SMOOTHING\_WINDOW\_SIZE];  // Giá trị Độ ẩm được lưu trong mảng humi\_samples

float humi\_sampleTotal = 0;                 // the running total

float humi\_sampleAvg = 0;                   // the average

/// Temperature moving AVG config

float temp\_samples[SMOOTHING\_WINDOW\_SIZE];  // Giá trị Nhiệt độ được lưu trong mảng temp\_samples

float temp\_sampleTotal = 0;                 // the running total

float temp\_sampleAvg = 0;                   // the average

/// Soil moisture moving AVG config

float soil\_samples[SMOOTHING\_WINDOW\_SIZE];  // Giá trị Độ ẩm đất được lưu trong mảng soil\_samples

float soil\_sampleTotal = 0;                 // the running total

float soil\_sampleAvg = 0;                   // the average

DHT dht(DHTPIN, DHTTYPE); //Khai báo thư viện chân cảm biến và kiểu cảm biến

void setup() {

  Serial.begin(9600);

  WiFiConnect(); // Kết nối với WiFi

  dht.begin(); //Khởi động cảm biến

  lcd.init(); //Khởi động LCD

  lcd.backlight(); //Mở đèn

  lcd.setCursor(0,0);

  lcd.print("Humi:");

  lcd.setCursor(0,1);

  lcd.print("Temp:");

  // Khởi tạo các giá trị trong mảng = 0:

  for (int i = 0; i < SMOOTHING\_WINDOW\_SIZE; i++) {

    humi\_samples[i] = 0;

    temp\_samples[i] = 0;

    soil\_samples[i] = 0;

  }

}

void loop() {

  // Đọc tín hiệu từ các cảm biến

  readSensors();

  // Tổng từng biến sẽ trừ đi giá trị cũ:

  humi\_sampleTotal = humi\_sampleTotal - humi\_samples[curReadIndex];

  temp\_sampleTotal = temp\_sampleTotal - temp\_samples[curReadIndex];

  soil\_sampleTotal = soil\_sampleTotal - soil\_samples[curReadIndex];

// Thêm mới giá trị đo được từ cảm biến theo vị trí hiện tại vào mảng

  humi\_samples[curReadIndex] = humi;

  temp\_samples[curReadIndex] = temp;

  soil\_samples[curReadIndex] = soil;

  // Tổng lại các giá trị trong mảng

  humi\_sampleTotal = humi\_sampleTotal + humi\_samples[curReadIndex];

  temp\_sampleTotal = temp\_sampleTotal + temp\_samples[curReadIndex];

  soil\_sampleTotal = soil\_sampleTotal + soil\_samples[curReadIndex];

  // Tính giá trị trung bình trượt (Moving AVG):

  humi\_sampleAvg = humi\_sampleTotal / SMOOTHING\_WINDOW\_SIZE;

  temp\_sampleAvg = temp\_sampleTotal / SMOOTHING\_WINDOW\_SIZE;

  soil\_sampleAvg = soil\_sampleTotal / SMOOTHING\_WINDOW\_SIZE;

  // Hiển thị giá trị raw values/smooth values trên màn hình Serial

  Serial.print("HumidityRaw:");  Serial.print(humi);  Serial.print(", ");  Serial.print("HumidotySmooth:");  Serial.print(humi\_sampleAvg);  Serial.print(", ");

  Serial.print("TemperatureRaw:");  Serial.print(temp);  Serial.print(", ");  Serial.print("TemperatureSmooth:");  Serial.print(temp\_sampleAvg);  Serial.print(", ");

  Serial.print("SoilMoistureRaw:");  Serial.print(soil);  Serial.print(", ");  Serial.print("SoilMoistureSmooth:");  Serial.println(soil\_sampleAvg);

  // Hiển thị giá trị trên màn hình LCD

  lcd.setCursor(5,0); lcd.print(humi\_sampleAvg); //con trỏ vị trí số 5 hàng 1, hiện giá trị độ ẩm ô số 6

  lcd.setCursor(5,1); lcd.print(temp\_sampleAvg); //con trỏ vị trí số 5 hàng 2, hiện giá trị nhiệt độ ô số 6

//  lcd.setCursor(5,0); lcd.print(humi\_sampleAvg); //con trỏ vị trí số 5 hàng 1, hiện ô số 6

  // Cập nhật vị trí mới cho lần lặp tiếp theo

  curReadIndex = curReadIndex + 1;

  // Sau khi đến vị trí cuối mảng...

  if (client.connect(server, 80) && curReadIndex >= SMOOTHING\_WINDOW\_SIZE ) {

    // ...Đặt biến vị trí về lại vị trí đầu mảng và cập nhật dữ liệu lên ThingSpeak:

    curReadIndex = 0;

    // Construct API request body

    String body = "field1=" + String(humi\_sampleAvg, 1) + "&field2=" + String(temp\_sampleAvg, 1) + "&field3=" + String(soil\_sampleAvg,1);

    client.print("POST /update HTTP/1.1\n");

    client.print("Host: api.thingspeak.com\n");

    client.print("Connection: close\n");

    client.print("X-THINGSPEAKAPIKEY: " + writeAPIKey + "\n");

    client.print("Content-Type: application/x-www-form-urlencoded\n");

    client.print("Content-Length: ");

    client.print(body.length());

    client.print("\n\n");

    client.print(body);

    client.print("\n\n");

  }

/// Nhận giá trị từ ThingSpeak sau khi xử lý và hiển thị Giá trị trung bình từng ngày trên LCD

statusCode = ThingSpeak.readMultipleFields(1991115,"A7COGX6M8OYAGH2B");

  /// Nhận giá trị từ ThingSpeak sau khi xử lý và hiển thị Giá trị trung bình từng ngày trên LCD

  if(statusCode == 200)

    {

      // Fetch the stored data

      float dailyHumi = ThingSpeak.getFieldAsFloat(field[0]); // Field 1

      float dailyTemp = ThingSpeak.getFieldAsFloat(field[1]); // Field 2

      float dailySoil = ThingSpeak.getFieldAsFloat(field[2]); // Field 3

      // Hiển thị giá trị trên màn hình LCD

      lcd.setCursor(5,0); lcd.print(dailyHumi); //con trỏ vị trí số 5 hàng 1, hiện giá trị độ ẩm ô số 6

      lcd.setCursor(5,1); lcd.print(dailyTemp); //con trỏ vị trí số 5 hàng 2, hiện giá trị nhiệt độ ô số 6

  //  lcd.setCursor(5,0); lcd.print(humi\_sampleAvg); //con trỏ vị trí số 5 hàng 1, hiện ô số 6

  }

  else{

      Serial.println("Problem reading channel. HTTP error code " + String(statusCode));

  }

  client.stop();

  delay(50); // read samples at ~20Hz (once every 50ms)

}

void WiFiConnect(){

  WiFi.begin(ssid, password);

  Serial.print("Connecting to SSID: ");

  Serial.println(ssid);

  while(WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("");

Serial.print("Connected to WiFi network with IP Address: ");

  Serial.println(WiFi.localIP());

}

int getSoilMoist()

{

  int anaValue = 0;

  anaValue = analogRead(SOIL\_MOIST\_PIN); //Đọc giá trị cảm biến độ ẩm đất

  anaValue = map(anaValue, 1023, 0, 0, 100); //Ít nước:0%  ==> Nhiều nước 100%

  return anaValue;

}

void readSensors(void)

{

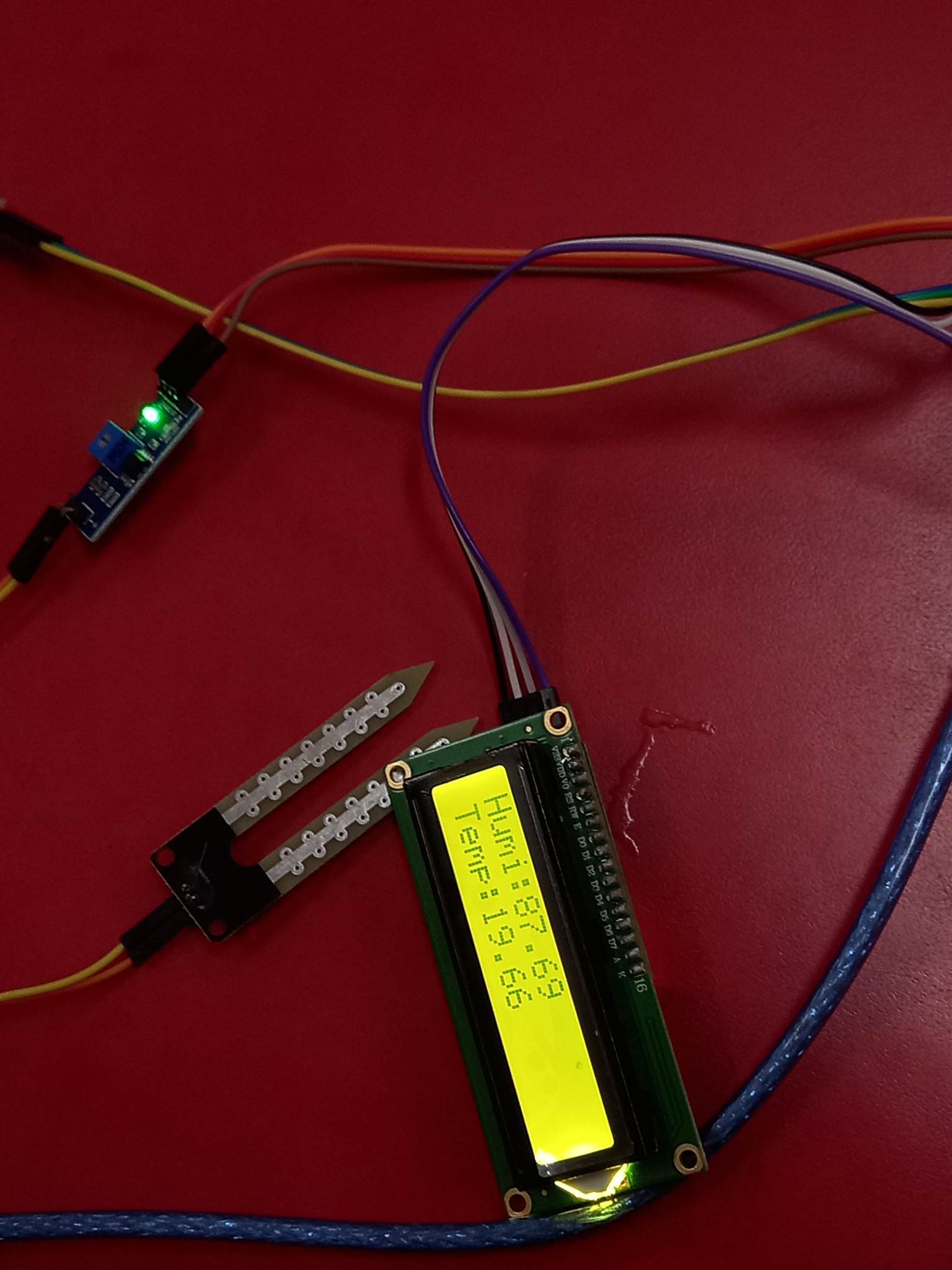
  temp = dht.readTemperature();   //Đọc nhiệt độ DHT22

  humi = dht.readHumidity();       //Đọc độ ẩm DHT22

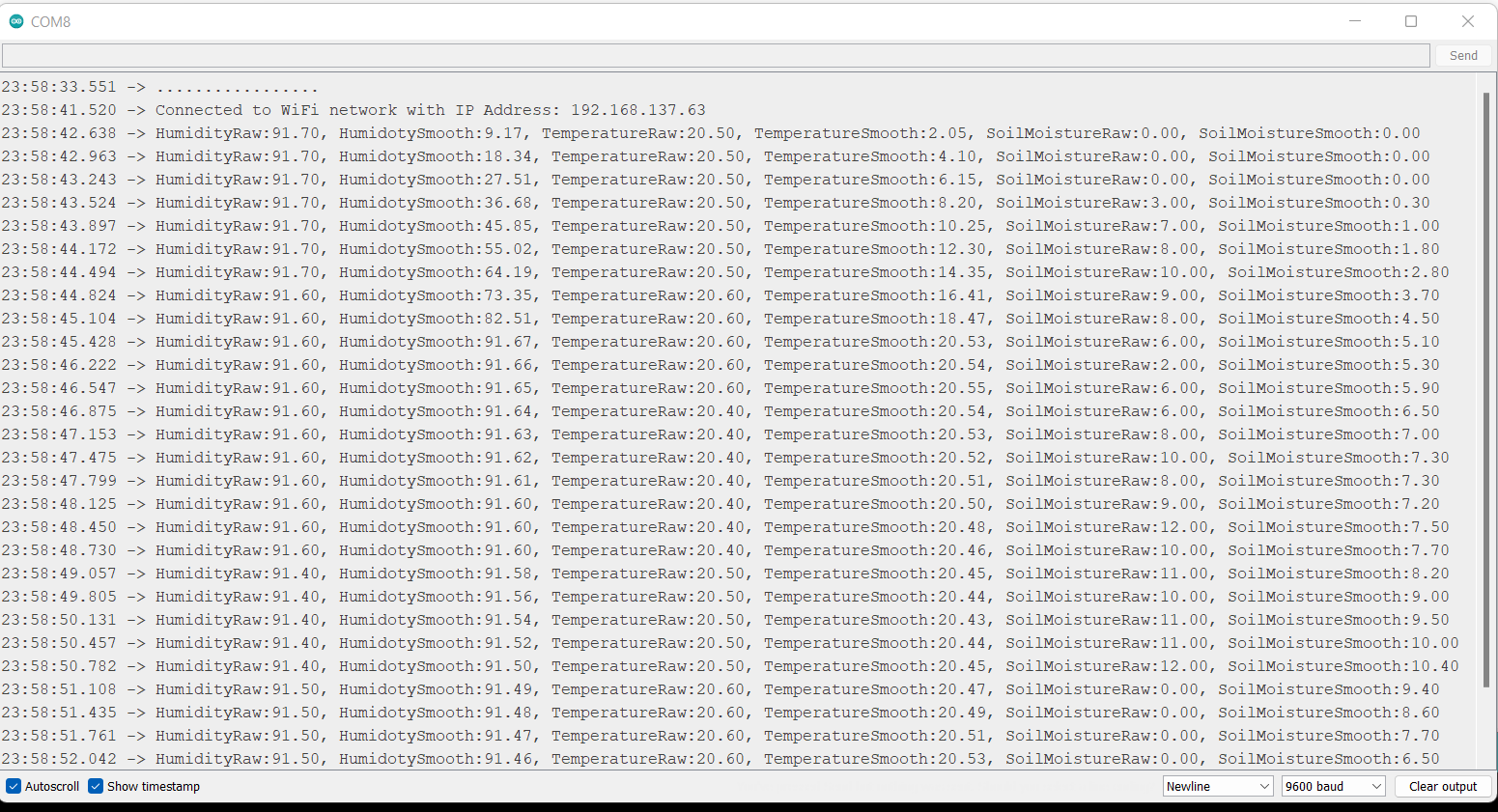
  soil = getSoilMoist();        //Đọc cảm biến độ ẩm đất

}

1. **Kết quả thực hành**
2. **Kết quả hiển thị trên màn hình LCD**



1. **Kết quả hiển thị trên màn hình Serial Monitor**



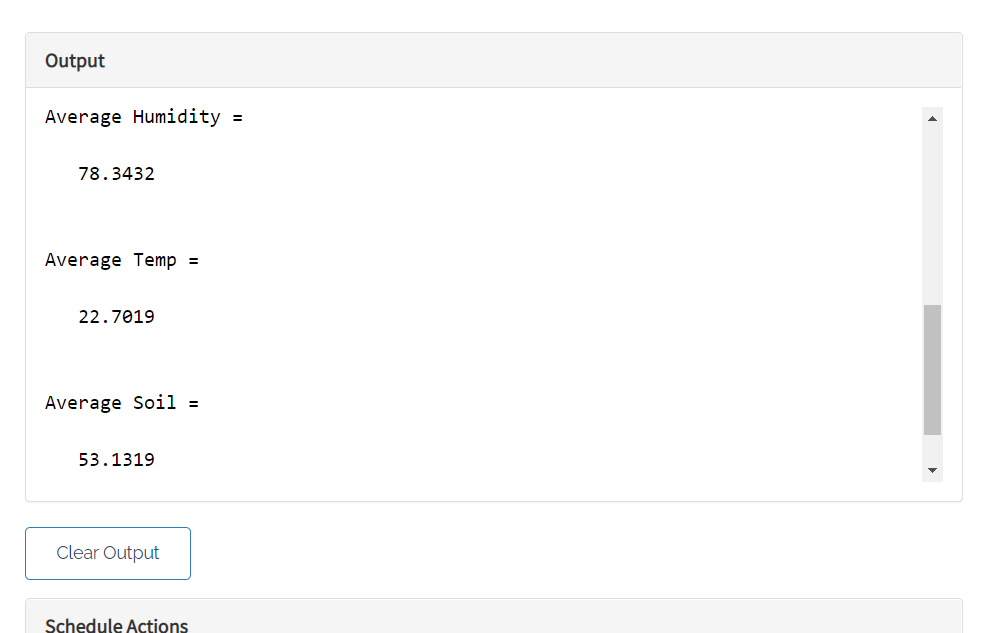
1. **Kết quả hiện thị trên cửa sổ Serial Ploter**



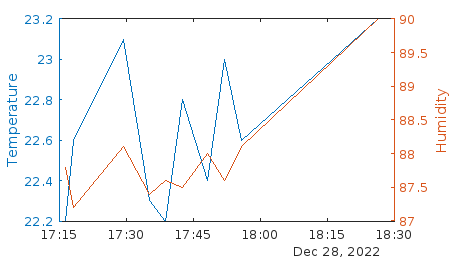
**Nhận xét:**

* Màn hình hiển thị các giá trị thô và đã lọc nhiễu của Nhiệt độ, Độ ẩm và Độ ẩm đất.
* So với Nhiệt độ và Độ ẩm (tin hiệu digital thu được từ cảm biến DHT22), dữ liệu thô của Cảm biến Độ ẩm đất (tín hiệu Analog) nhiễu hơn hơn rất nhiều và nhận thấy rõ qua đường màu hồng. Trong trường hợp này, việc sử dụng bộ lọc Trung bình chạy là cần thiết trước khi gửi/lưu trữ giá trị từ cảm biến.
* Không có sự khác biệt quá lớn khi sử dụng bộ lọc lên giá trị digital (Độ ẩm và Nhiệt độ); Đường thể hiện giá trị thô và giá trị đã lọc nhiễu gần như trùng khớp nhau (chỉ có sự chênh lệch nếu xảy ra sự cố đường truyền).

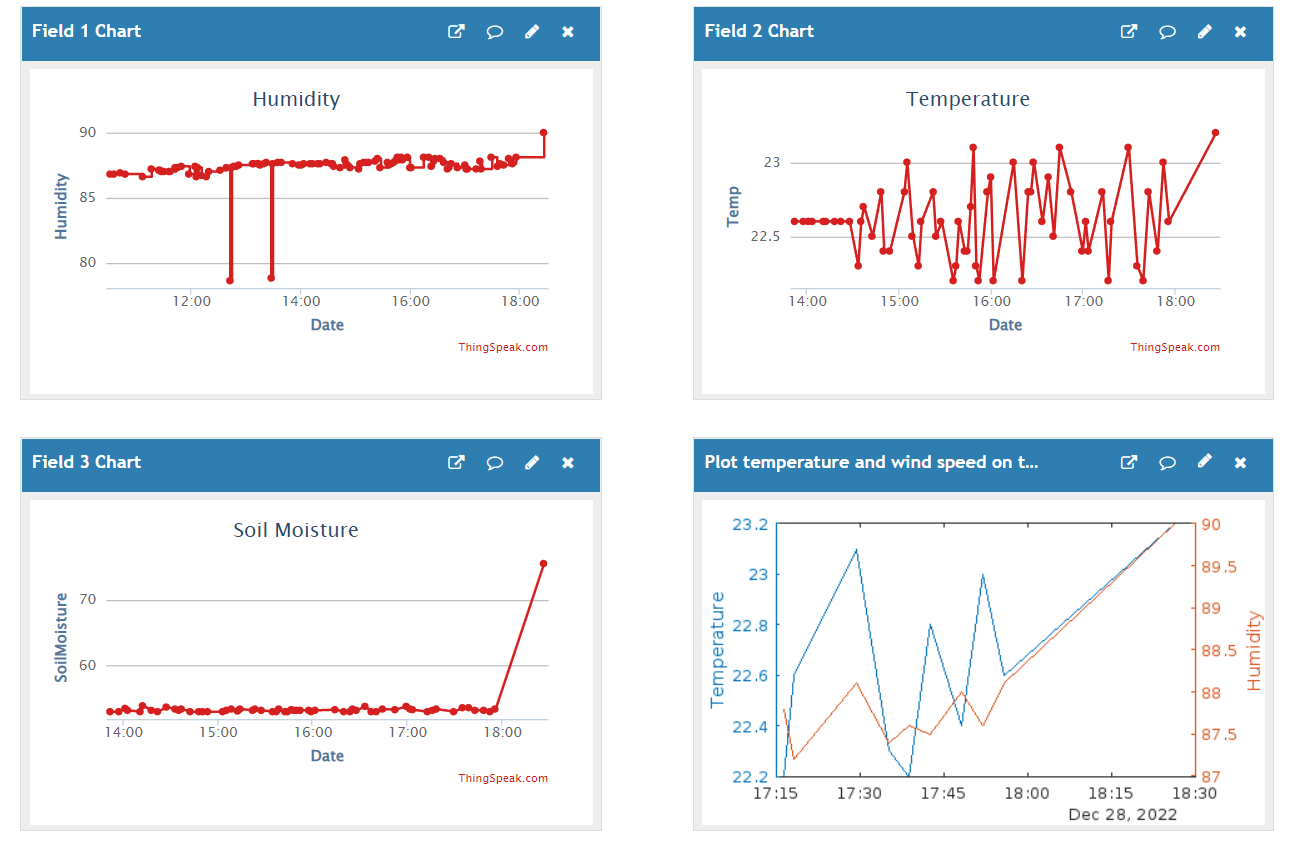
1. **Kết quả khi tính giá trị trung bình cảm biến bằng MATLAB Analysis**



1. **Kết quả thu được khi vẽ đồ thị bằng bằng MATLAB Visualizations**



1. **Kết quả ghi lại trên ThingSpeak**

**  
Channel esp8266/dht22-thinkspeak**

[**https://thingspeak.com/channels/1986959**](https://thingspeak.com/channels/1986959)

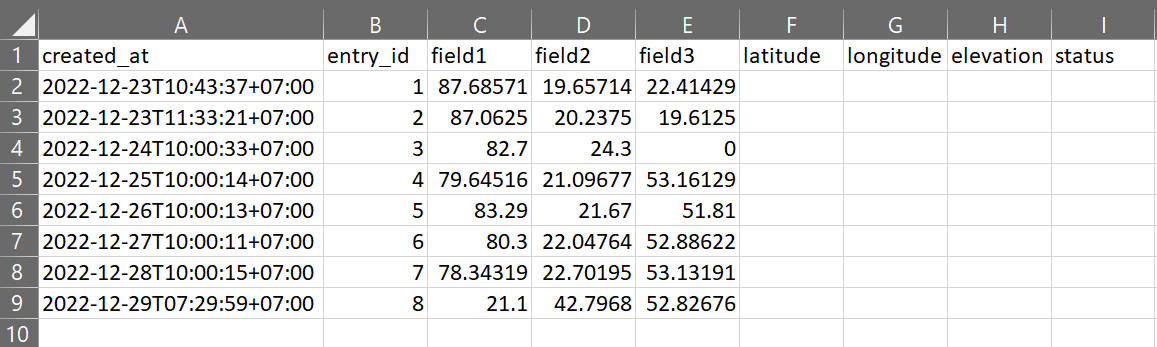
**  
Channel Đồ án**

[**https://thingspeak.com/channels/1995973**](https://thingspeak.com/channels/1995973) **v**

1. **Kết quả ghi lại trên ThingView (ThingSpeak App)**



1. **Xuất giá trị và lưu trữ vào file CSV**



1. **Kết luận**

**Sau khi thực hành và thu được kết quả nhóm chúng em đã đạt được:**

* **Đọc và ghi giá trị thời gian thực từ cảm biến DHT22 và cảm biến độ ẩm đất với esp8266.**
* Sử dụng thành công **bộ lọc tín hiệu (Moving Average)** để giảm nhiễu các giá trị thô thu được từ cảm biến.
* Hiển thị thành công các giá trị đo được trên **màn hình LCD**.
* Gửi thành công dữ liệu cảm biến thời gian thực lên nền tảng đám mây **ThingSpeak.**
* **Xử lý số liệu và trực quan hóa dữ liệu** thu được bằng **MATLAB Analysis và Visualizations.**
* Đã làm việc với hệ thống và **ghi lại dữ liệu trong 1 tuần**.
* Đưa ra được giá trị trung bình cảm biến trong từng ngày và **hiển thị giá trị đó trên màn hình màn hình LCD.**



# TÀI LIỆU THAM KHẢO

1. Lesson 1: Smoothing Input. *makeabilitylab.* [Online] [Cited: 12 22, 2022.] https://makeabilitylab.github.io/physcomp/advancedio/smoothing-input.html.

2. ThingSpeak. *thingspeak.com.* [Online] [Cited: 12 22, 2022.] https://thingspeak.com/pages/learn\_more.