ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

-----📖-----

BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐIỀU KHIỂN TỰ ĐỘNG CE212.M21

Giảng viên hướng dẫn: Nguyễn Hoài Nhân

Sinh viên thực hiện:

Phạm Quốc Đăng - 19520036

Phạm Trung Quốc - 19520887

Phạm Trọng Huỳnh - 19521651

Thành phố Hồ Chí Minh, tháng 4 năm 2022

Danh mục hình

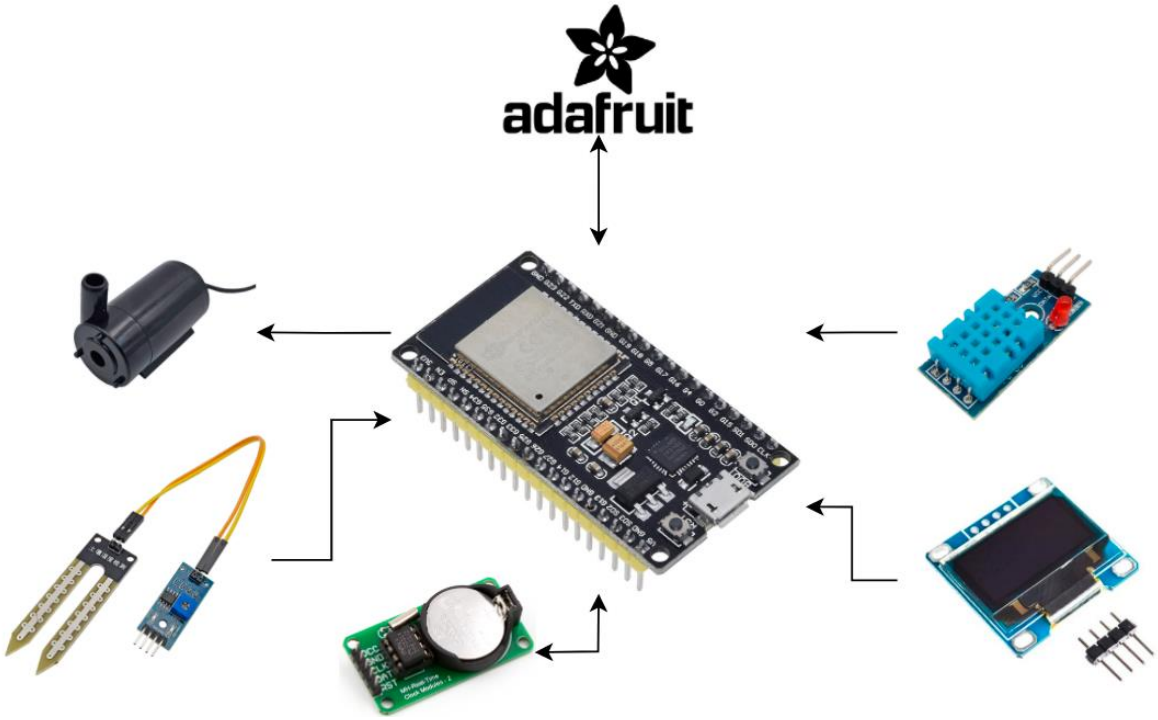
Mục lục

I. TỔNG QUAN ĐỀ TÀI

1. Mục tiêu và sơ lược về cách hoạt động của đề tài

Nhận thấy đề tài hệ thống tưới tiêu thông minh là một đề tài phù hợp và có thể thực hiện được nên nhóm đã quyết định chọn đề tài là thiết kế một hệ thống tưới tiêu thông minh với các chức năng cơ bản như hiển thị nhiệt độ, độ ẩm đất, thời gian thực, xử lý dữ liệu và đưa ra quyết định tưới phù hợp (hoặc có thể điều khiển thủ công thông qua adafruit), cập nhật dữ liệu lên server Adafruit phục vụ cho việc theo dõi.

Sơ đồ khối của đề tài:



Hình 1 Sơ đồ khối của hệ thống tưới tiêu thông minh

2. Phương pháp chung để thực hiện đề tài

Cần phải xác định rõ nội dung đề tài hướng đến là gì, xác định đối tượng mà đề tài hướng đến, nhiệm vụ nghiên cứu và phân tích cũng như phạm vi nghiên cứu.

Chia nhỏ công việc thành các giai đoạn và các nội dung cần nghiên cứu, như giai đoạn chuẩn bị, giai đoạn tìm hiểu nghiên cứu, giai đoạn hiện thực cơ bản, ghép nối và chạy thử, giai đoạn hoàn thiện, đóng gói.

Để đạt được mục tiêu trên, ta cần chia nhỏ nội dung nghiên cứu thực hiện thành các phần bao gồm phần cứng, chương trình (phần lập trình), phần dữ liệu, và phần

tương tác với hệ thống, mỗi phần sẽ là một giai đoạn trong việc làm đề tài và đòi hỏi phải nghiêm túc nghiên cứu và thực hiện.

Nghiên cứu đề tài từ các nguồn tài liệu trực tuyến, các trang tìm kiếm lớn, có thể tìm kiếm thêm sự trợ giúp của giảng viên môn học, giảng viên hướng dẫn đồ án, sách,… thêm nữa là sự hỗ trợ và trao đổi giữa các thành viên trong nhóm với nhau.

Chuẩn bị sẵn các phần cứng cũng như các thiết bị đáp ứng cho nhu cầu làm việc và thực hiện đề tài. Đối với các sản phẩm phần cứng cần tìm hiểu kĩ datasheet, nắm rõ cách hoạt động để dễ dàng hơn trong việc kết nối và thực hiện đề tài.

3. Phân chia thời gian thực hiện các nội dung

Thời Gian	Công Việc
Tuần 1 1/5 – 7/5	Chọn đề tài và xây dựng kế hoạch thực hiện
Tuần 2 8/5 – 14/5	Nắm vững lý thuyết và thực hiện giao tiếp với từng ngoại vi
Tuần 3 15/5 – 21/5	Tổng hợp và xử lý dữ liệu, cập nhật dữ liệu lên adafruit
Tuần 4 22/5 – 27/5	Kiểm tra và hoàn thiện sản phẩm

II. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. Xác định yêu cầu đề tài

No.	I want to	So that
1	Hiển thị độ ẩm đất, nhiệt độ môi trường	Theo dõi được độ ẩm đất, nhiệt độ độ ẩm môi trường và trạng thái bơm thông qua Oled
2	Tự động tưới/ngắt khi độ ẩm đất thấp/cao	Từ dữ liệu thu thập được giúp kịp thời tưới tiêu cho hệ thống cây trồng
3	Theo dõi quá trình tưới cây	Thông qua giao diện Adafruit, quan sát được quá trình tưới tiêu một cách trực quan hơn
4	Hiển thị thời gian thực	Theo dõi thời gian

2. Đặc tả sản phẩm

- Chức năng: Hệ thống theo dõi và tưới tiêu tự động, hiển thị nhiệt độ độ ẩm và thời gian thực.
- Input: Cảm biến nhiệt độ, độ ẩm đất, clock thời gian, các câu lệnh từ server
- Output: Thời gian, nhiệt độ, độ ẩm, tín hiệu điều khiển tưới tiêu.
- Giao diện có người dùng: Adafruit, oled
- Chuẩn giao tiếp: I2C, TTL, MQTT.

3. Đặc tả kỹ thuật

Phần cứng: ESP32, cảm biến nhiệt độ độ ẩm DHT11, OLED, motor 12(V), cảm biến độ ẩm đất, DS1302 module, mạch hạ áp.

Yêu cầu phần cứng và phần mềm:

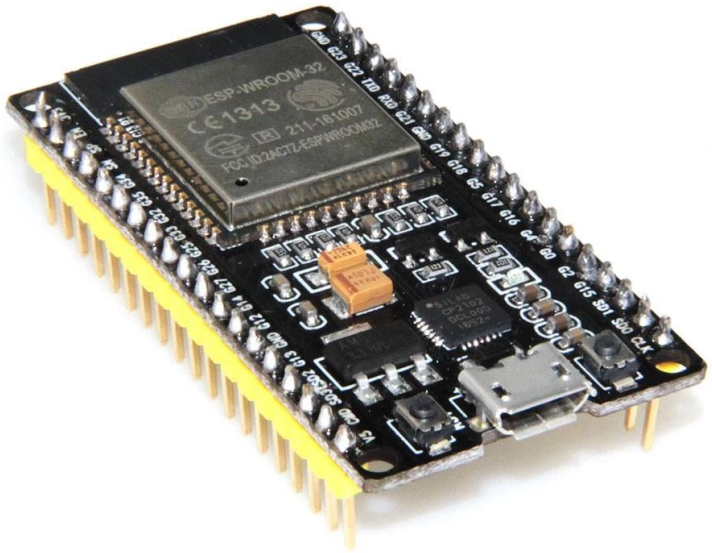
- Xử lý thời gian trên DS1302
- Hiển thị nhiệt độ, độ ẩm đất, thời gian
- Theo dõi tình trạng nhiệt độ độ ẩm trên server
- Tự động tưới tiêu khi độ ẩm thấp
- Điều khiểu motor thông qua server

4. Hệ thống phần cứng

a. Cơ sở lý thuyết

❖ Module ESP32

Tổng quan



Hình 2 Kit ESP32

ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.

Điểm tốt về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại nhận tiếng ồn thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.

Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng.

Thông số kỹ thuật

Hình 3 Sơ đồ các GPIO cơ bản trên ESP8266 NodeMCU

Thành phần chính	**Miêu tả**
ESP32-WROOM-32	Một mô-đun với ESP32 là lõi. Để biết thêm thông tin, hãy xem Datasheet ESP32-WROOM-32.
EN	Nút Reset
Boot	Nhấn giữ nút Boot và sau đó nhấn EN bắt đầu chế độ Tải xuống chương trình cơ sở để tải xuống chương trình cơ sở thông qua cổng nối tiếp
USB-to-UART Bridge	USB-UART cung cấp tốc độ truyền lên đến 3 Mbps.
Micro USB Port	USB. Nguồn cung cấp cho bo mạch cũng như giao diện giao tiếp giữa máy tính và mô-đun ESP32-WROOM-32.

Thành phần chính	Miêu tả
5V Power On LED	Sáng lên khi kết nối USB hoặc nguồn điện 5V bên ngoài được kết nối với bo mạch. Để biết chi tiết, hãy xem các sơ đồ trong Tài liệu Liên quan.
I/O	Hầu hết các chân trên mô-đun ESP bị đứt ra khỏi các đầu ghim trên bảng. Bạn có thể lập trình ESP32 để kích hoạt nhiều chức năng như PWM, ADC, DAC, I2C, I2S, SPI, v.v.

❖ **OLED SSD1306**

Tổng quan

Màn hình Oled 0.91 inch giao tiếp I2C cho khả năng hiển thị đẹp, sang trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình LCD sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO.

Hình 4 OLED SSD1306

Thông số kỹ thuật

- Điện áp sử dụng: 2.2~5.5VDC.
- Công suất tiêu thụ: 0.03w
- Góc hiển thị: lớn hơn 160 độ
- Số điểm hiển thị: 128×32 điểm.

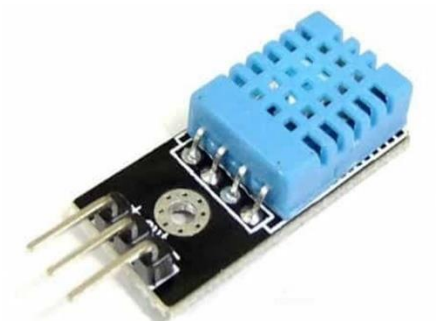
- Độ rộng màn hình: 0.96 inch
- Màu hiển thị: Trắng / Xanh Dương.
- Giao tiếp: I2C
- Driver: SSD1306

❖ **Cảm biến nhiệt độ độ ẩm DHT11**

Tổng quan

Cảm biến độ ẩm nhiệt độ DHT11 ra chân được tích hợp sẵn điện trở 5,1k ohm giúp người dùng dễ dàng kết nối và sử dụng hơn so với cảm biến DHT11 chưa ra chân, module lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp 1 dây).

Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán nào. Module được thiết kế hoạt động ở mức điện áp 5VDC.



Hình 5 Cảm biến nhệt độ, độ ẩm DHT11

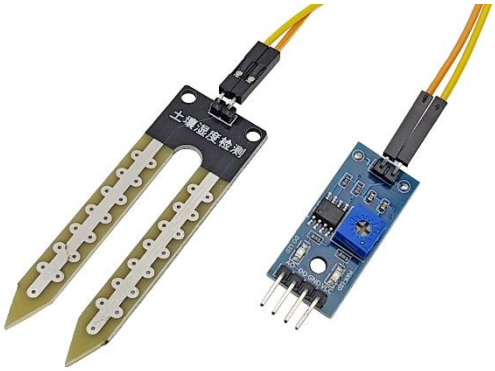
Thông số kỹ thuật

- Điện áp hoạt động: 5V / 3.3V DC
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số ± 5%RH
- Khoảng đo nhiệt độ: 0-50°C sai số ± 2°C
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
- Kích thước: 28mm x 12mm x10m

❖ Cảm biến độ ẩm đất

Tổng quan

Trạng thái đầu ra mức thấp (0V), khi đất thiếu nước đầu ra sẽ là mức cao (5V), độ nhạy cao chúng ta có thể điều chỉnh được bằng biến trở. Cảm biến độ ẩm đất có thể sử dung tưới hoa tự động khi không có người quản lý khu vườn của bạn hoặc dùng trong những ứng dụng tương tự như trồng cây.



Hình 6 Cảm biến độ ẩm đất

Độ nhạy của cảm biến độ ẩm đất có thể tùy chỉnh được (bằng cách điều chỉnh chiết áp màu xanh trên board mạch). Phần đầu DO được cắm vào đất để phát hiện độ ẩm của đất, khi độ ầm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao.

Thông số kỹ thuật

- Điện áp làm việc 3.3V ~ 5V
- Có lỗ cố định để lắp đặt thuận tiện
- PCB có kích thước nhỏ 3.2 x 1.4 cmỗ
- Sử dung chip LM393 để so sánh, ổn định làm việc

❖ **Động cơ bơm nước mini tự mồi YYP370-12C3 12VDC**

Tổng quan

Hình 7 Động cơ bơm nước mini tự mồi YYP370-12C3 12VDC

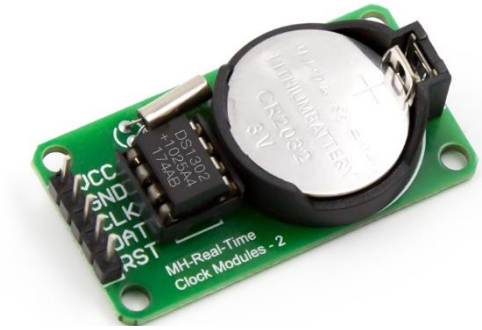
Động cơ bơm nước mini tự mồi YYP370-12C3 12VDC là loại máy bơm mini mà bạn không cần phải mồi nước cho nó sau mỗi lần khởi động lại sau khi đã lắp đặt và hoạt động ổn định ở lần thứ nhất, vì vậy sản phẩm cũng có thể được sử dụng trong máy uống nước, máy pha sữa, máy pha cafe, bình xịt thông minh, vòi uống nước tự động, phân biệt xà phòng… sản phẩm làm đẹp và y tế

Thông số kỹ thuật

- Model: YYP370-12C3
- Điện áp hoạt động: 12VDC
- Dòng điện làm việc không tải: 0.15 – 0.17A
- Dòng điện khi nén: ~0.7A
- Lưu lượng: 0.3 – 1.6L/MIN
- Độ ồn: < 60dB
- Phạm vi sử dụng: nước, không khí, chất lỏng
- Áp suất: 10 – 80 Psi
- Đầu ống: có thể dùng ống 6 hoặc 7mm
- Lực nâng: lên đến 3 mét (đã test thực tế ở 3 mét)
- Tuổi thọ: hoạt động trong 30s, dừng 20s, chu kỳ hơn 50.000 lần
- Nhiệt độ hoạt động: 5 – 100 độ C
- Trọng lượng: 62g

❖ **RTC DS1302 Đồng Hồ Thời Gian**

Tổng quan



Hình 8 RTC DS1302 Đồng Hồ Thời Gian

Mạch thời gian thực RTC DS1302 được sử dụng để cung cấp thông tin thời gian: ngày, tháng, năm, giờ, phút, giây,...cho vi điều khiển với chỉ 3 chân tín hiệu giao tiếp là Reset, Data và Clock, mạch có tích hợp pin CR2032 giúp duy trì thời gian khi không được cấp nguồn.

Thông số kỹ thuật

- IC chính: RTC DS1302
- Nguồn cung cấp: 3.3/5VDC (tuỳ thuộc vào mức điện áp giao tiếp của thiết bị cần kết nối).
- Giao tiếp: 3-Wires Interface (Reset, Data, Clock)
- Lưu trữ và cung cấp các thông tin thời gian thực: ngày, tháng, năm, giờ, phút, giây,...
- Có pin backup duy trì thời gian trong trường hợp không cấp nguồn.
- Kích thước: 44 x 23mm

❖ **Mạch giảm áp Buck DC-DC XL4015 5A**

Tổng quan

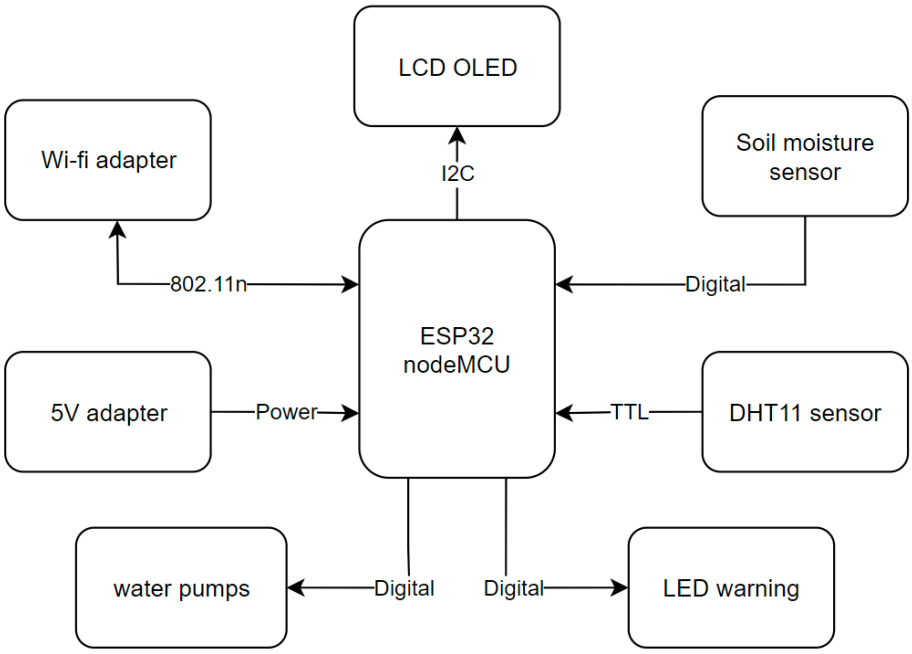
Hình 9 Mạch giảm áp

Mạch giảm áp XL4015 công suất ngõ ra 5A 75W, dùng để giảm điện áp, cho điện áp đầu vào từ 5 - 36V và đầu ra từ 1.15V - 32V. Ứng dụng cho hạ áp từ nguồn Pin, biến thế, đầu áp ra và dòng ra có thể chỉnh được dựa trên biến trở nhỏ màu xanh trên board. nhỏ gọn, cơ động dễ sử dụng, cho ra các mức điện áp mong muốn.

Thông số kỹ thuật

- Mạch giảm áp 5A XL4015E1 Có chỉnh dòng.
- Mạch giảm áp không cách ly
- Điện áp đầu vào: 4 - 36V
- Điện áp đầu ra: 1.25V - 32V
- Dòng chỉnh đầu ra : điều chỉnh tối đa ra 5A
- Dòng điện đầu ra: 0 - 5A.
- Công xuất đầu ra: 75W.
- Nhiệt độ hoạt động: -40 to +85 độ.
- Hiệu suất hoạt động : 96%.
- Đèn báo : có.
- Kích thước board: 51.2 * 26.2 * 15mm
- Cân nặng : 23g

b. Sơ đồ hoạt động phần cứng



Hình 10 Sơ đồ hoạt động của phần cứng

c. Chức năng cụ thể

- ESP 32 được kết nối với WiFi (nơi người dùng)
- ESP32 lấy được thời gian thực và hiển thị lên OLED
- ESP32 lấy được nhiệt độ, độ ẩm đất và hiển thị lên OLED
- ESP32 cập nhật giá trị nhiệt độ, độ ẩm đất lên server Adafruit
- ESP32 xử lý dữ liệu thu được và đưa ra quyết định tưới tiêu phù hợp (hoặc nhận được tín hiệu tưới tiêu từ server Adafruit)

5. Hệ thống phần mềm

a. Cở sở lý thuyết

❖ **Phần mềm arduino**

Tổng quan về Arduino IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm.

Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linhkiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.

Cách Arduino IDE hoạt động

Cách Arduino IDE hoạt động là: khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

Hình 11 Phần mềm Arduino IDE

Một số lý do nên sử dụng Arduino IDE

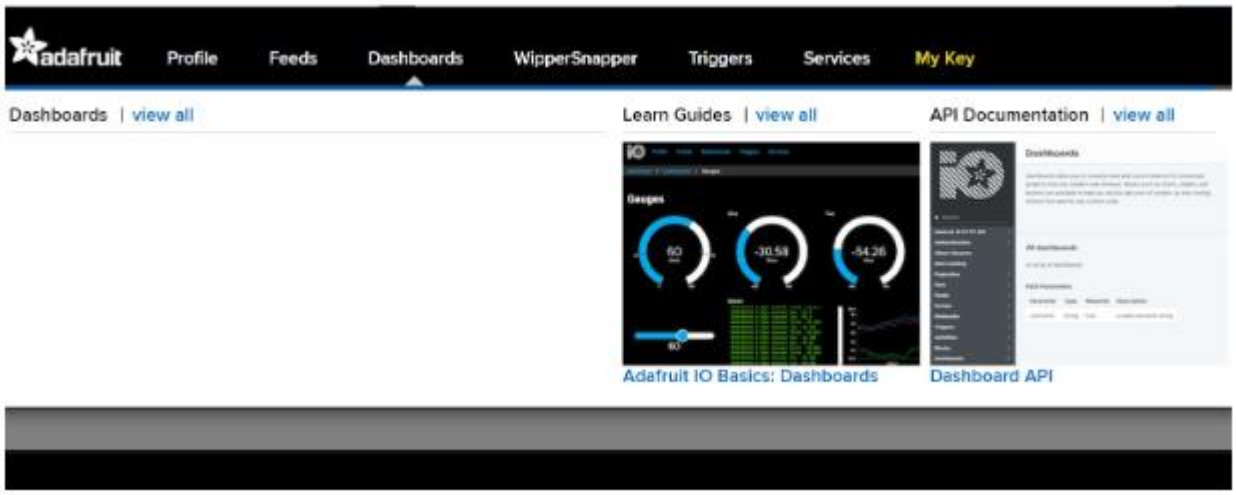
- Arduino IDE là phần mềm lập trình mã nguồn mở miễn phí
- Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên
- Hỗ trợ lập trình tốt cho bo mạch Arduino
- Thư viện hỗ trợ phong phú
- Giao diện đơn giản, dễ sử dụng
- Hỗ trợ đa nền tảng như Windows, MacOS, Linux

❖ **Adafruit.io**

Tổng quan về Adafruit IO

Adafruit đưa ra một MQTT miễn phí Cloud Service cho thí nghiệm IOT và người học gọi Adafruit IO trong năm 2015. Adafruit IO là một dịch vụ internet đơn giản dễ sử dụng, dễ dàng cho phép các thiết bị IoT nhận và gửi dữ liệu. Và Adafruit IO có thể làm các công việc sau:

- Hiển thị dữ liệu của bạn trong thời gian thực, trực tuyến
- Làm cho dự án của bạn kết nối internet: Điều khiển động cơ, đọc dữ liệu
- cảm biến và hơn thế nữa!
- Kết nối các dự án với các dịch vụ web như Twitter, nguồn cấp dữ liệu RSS,
- dịch vụ thời tiết, v.v.
- Kết nối dự án của bạn với các thiết bị hỗ trợ internet khác.
- Tất cả những điều trên có thể làm miễn phí với Adafruit IO.



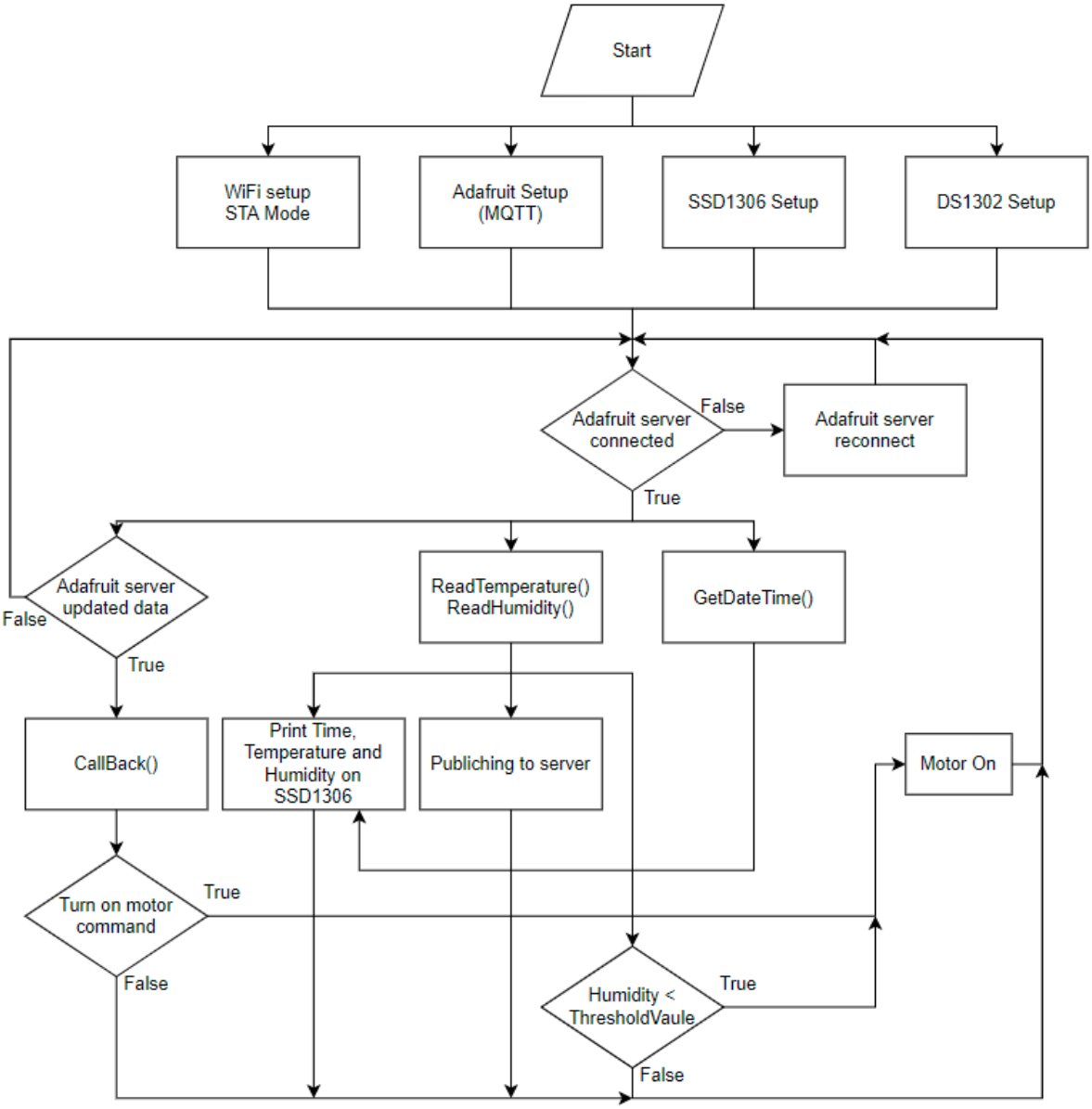
Hình 12 Giao diện chính của web Adafruit IO

Khái niệm cơ bản trong Adafruit IO

Dashboard: là một bảng điều khiển kỹ thuật số hoặc là giao diện số. Trong bảng này với rất nhiều thông tin đã được thu thập và tổng hợp từ nhiều nguồn dữ liệu khác nhau của tổ chức và doanh nghiệp lớn, nhỏ vào màn hình.

Trigger: Sử dụng kích hoạt trong Adafruit IO để kiểm soát và phản ứng với dữ liệu của bạn. Cấu hình các kích hoạt để gửi email cho bạn khi hệ thống của bạn ngoại tuyến, phản ứng với cảm biến nhiệt độ quá nóng và xuất bản tin nhắn đến nguồn cấp dữ liệu mới.

b. Sơ đồ hoạt động phần mềm



Hình 13 Sơ đồ hoạt động của phần mềm

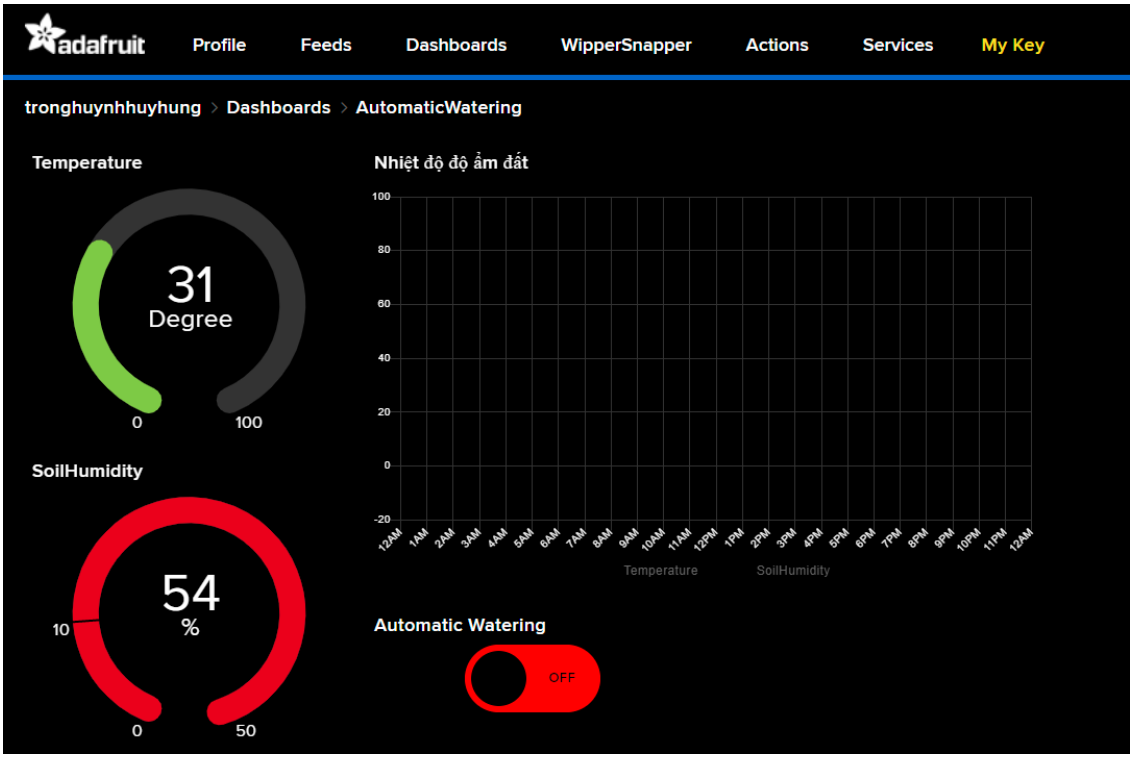
c. Chức năng cụ thể

ESP32 là bộ xử lý trung tâm, sau khi kết nối Wi-fi và kết nối với server Adafruit, các cảm biến gửi dữ liệu về ESP32 và tiến hành gửi lên server Adafruit để theo dõi

quá trình tưới tiêu của hệ thống, đồng thời dữ liệu được dùng để so sánh và thực hiện các tác vụ bật tắt motor thông qua các ngưỡng được cấu hình trong phần mềm.

Các dữ liệu từ các cảm biến và thời gian được lưu trong SSD1302 được hiển thị lên OLED, màn hình luân phiên hiển thị các thông tin về thời gian thực, nhiệt độ, độ ẩm đất

6. Thiết kế giao diện



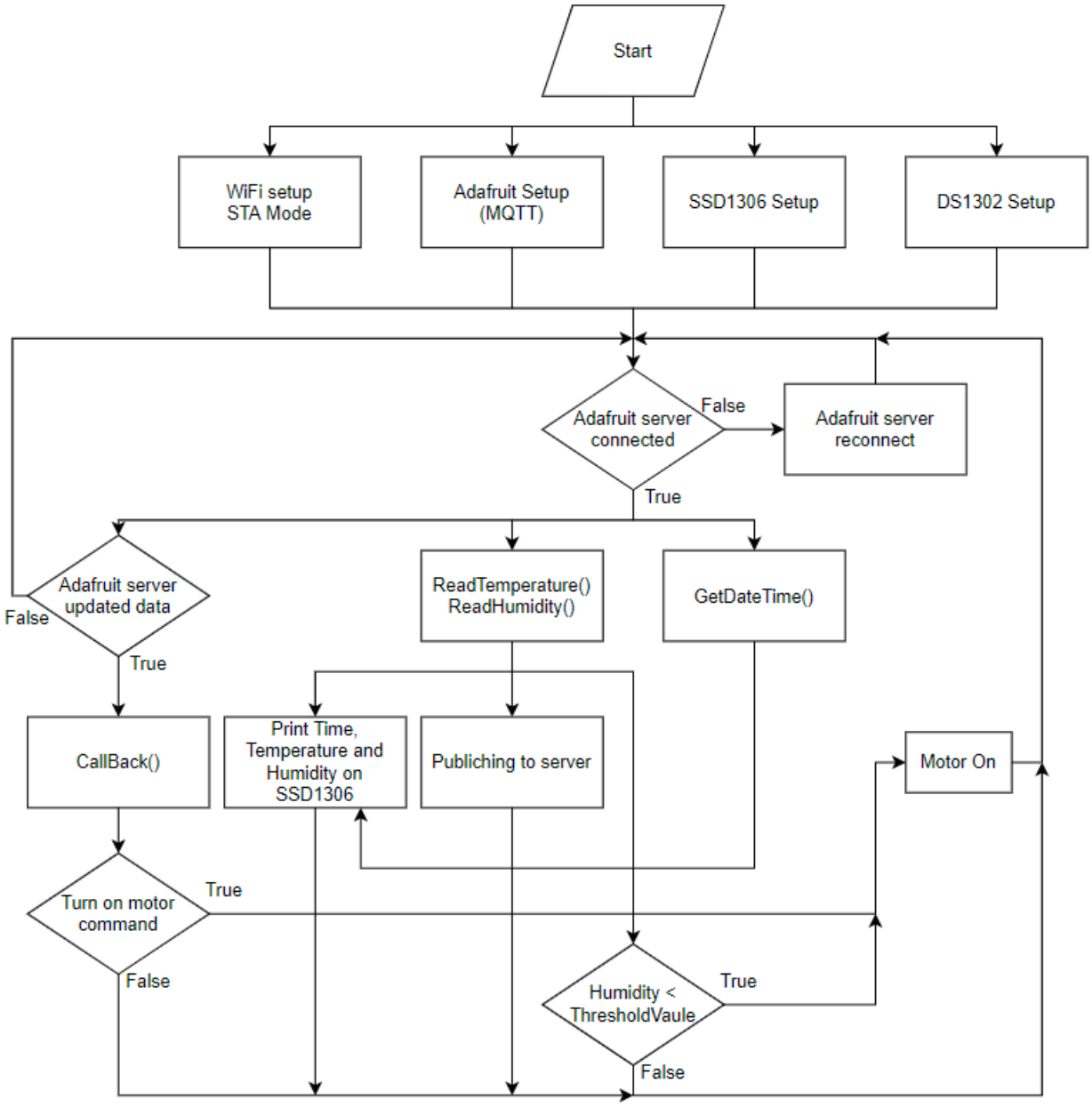
Hình 14 Giao diện theo dõi dữ liệu trên Adafruit

Đây là giao diện Adafruit được được thiết kế bao gồm các chức năng như sau:

- Theo dõi được nhiệt độ ở lần cập nhật gần nhất
- Thời dõi được độ ẩm đất ở lần cập nhật gần nhất
- Quá trình thay đổi của nhiệt độ và độ ẩm đất qua những lần cập nhật
- Nút tưới tiêu thủ công

III. LẬP TRÌNH HỆ THỐNG

1. Lưu đồ giải thuật



Hình 15 Lưu đồ giải thuật

2. Source Code

a. Source code

Link: https://drive.google.com/file/d/18ZIYw3oM0OWUgChNGItsLh

b. Thiết lập các chức năng

```
// WiFi connect set up
void WiFiConnect() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while(WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting");
  }
  Serial.println(WiFi.localIP()); // print to debug
}
```

Đây là phần thiết lập Wi-fi, với WIFI_SSID, WIFI_PASSWORD là Wi-fi nơi đặt thiết bị, cần kết nối để hệ thống kết nối được với mạng internet.

```
// Set up Adafruit Server
void Setup_Adafruit_Server() {
  Client.setServer("io.adafruit.com", 1883);
  Client.setCallback(CallBack);
  Client.connect("AutomaticWatering", IO_USERNAME,
IO_KEY);
  Client.subscribe("tronghuynhhuyhung/feeds/+");
  Serial.println("Adafruit Server has been set up");
}
```

Để kết nối ESP32 với server Adafruit chúng ta cần có phần thiết lập như trên bao gồm:

- setServer: đường dẫn tới trang chủ của Adafruit, và 1883 là port cho giao thức TCP là non-ssl.
- setCallback: chọn hàm để thực hiện chức năng call back từ server về hệ thống
- connect: kết nối với project AutomaticWatering, với tài khoản tương ứng (IO_USERNAME, IO_KEY)
- subcribe: dấu “+” là chọn tất cả các feed của tài khoản để đăng ký

```
// Try to reconnect when server broken
void Adafruit_Server_reconnect()
{
  while(!Client.connected()) {
```

```
        if(Client.connect("AutomaticWatering", IO_USERNAME,
IO_KEY)) {
          Serial.println("Connected (re)");
          Client.subscribe("tronghuynhhuyhung/feeds/+");
        }
        else {
          Serial.println("Waiting reconnected");
          delay(4000);
        }
      }
    }
```

Phần chức năng reconnect này được dùng khi ESP32 bị mất kết nối với internet và thiết bị sẽ cố gắng kết nối lại với Server

```
    void  Setup_RTC_DS1302()
    {
      Rtc.Begin();
      RtcDateTime compiled = RtcDateTime(__DATE__,
__TIME__);

     if (!Rtc.IsDateTimeValid())
        {
            // Common Causes:
            //    1) first time you ran and the device wasn't
running yet
            //    2) the battery on the device is low or even
missing

            Serial.println("RTC lost confidence in the
DateTime!");
            Rtc.SetDateTime(compiled);
        }

        if (Rtc.GetIsWriteProtected())
        {
            Serial.println("RTC was write protected, enabling
writing now");
            Rtc.SetIsWriteProtected(false);
        }

        if (!Rtc.GetIsRunning())
        {
            Serial.println("RTC was not actively running,
starting now");
            Rtc.SetIsRunning(true);
        }

        RtcDateTime now = Rtc.GetDateTime();
        if (now < compiled)
        {
```

```
            Serial.println("RTC is older than compile
time!  (Updating DateTime)");
            Rtc.SetDateTime(compiled);
        }
        else if (now > compiled)
        {
            Serial.println("RTC is newer than compile time.
(this is expected)");
        }
        else if (now == compiled)
        {
            Serial.println("RTC is the same as compile time!
(not expected but all is fine)");
        }
    }
```

Hàm Setup_RTC_DS1302 giúp setup một số chức năng cho DS1302 (mạch đếm thời gian thực)

- RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__): lấy thời gian từ laptop tại thời điểm nạp, và DS1302 sẽ bắt đầu lưu vào IC đếm của mình, dùng cho trường hợp mất điện vẫn đảm bảo thời gian luôn chính xác.
- Các kiểm tra còn lại để chắc rằng DS1302 đang hoạt động tốt bằng các lệnh so sánh và print để debug các lỗi.

```
void Setup_SSD1306() {
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
  delay(2000);

  display.clearDisplay();
  display.setTextSize(2); // size text
  display.setTextColor(WHITE); // color text
  display.setCursor(0, 10); // local pointer
  display.display();
}
```

Hàm Setup_SSD1306 dùng để kiểm tra việc kết nối màn hình OLED đã hoàn tất, cũng như thiết lập một số cài đặt ban đầu như kích thước, màu sắc, vị trí nội dung.

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

  // Setup Soil Humidity sensor
  pinMode(ANALOG_SENSOR_VALUE_PIN, INPUT);
```

```
    pinMode(MOTOR_PIN, OUTPUT);

    WiFiConnect();
    Setup_Adafruit_Server();
    DHT_Sensor.begin();
    Setup_RTC_DS1302();
    Setup_SSD1306();
}
```

Và đây là phần Setup của toàn bộ chương trình, có nhiệm vụ set tốc độ baud, cài đặt các chân pin cho phần cứng và gọi các hàm setup cho các phần cứng.

c. Chức năng bật tắt chức năng tự động tưới

```
void CallBack(char* topic, byte* payload, unsigned int
length) {
    Serial.println("Topic upadated");

    // Print message
    Serial.print("payload: ");
    for(int i = 0; i < length; i++) {
      Serial.print((char)payload[i]);
    }
    Serial.println();

    if ((char)payload[0] == 'o' || (char)payload[0] == 'f')
{
      Serial.print("Remote: ");
      Serial.println((char)payload[0]);
      Server_Remote_Motor(payload, length);
    }
}
```

Hàm Callback sẽ được gọi ngay lập tức nếu server có bất kỳ lệnh nào được cập nhật. Ở đây Callback được sử dụng để phát hiện việc bật tắt chức năng tưới tiêu tự động. Với giá trị “o” hay “f” được gửi về tương ứng với bật hay tắt chức năng tưới thủ công.

```
//   Server_Remote_Motor
void Server_Remote_Motor(byte* payload, unsigned int
length) {
    if ((char)payload[0] == 'o') {
      digitalWrite(MOTOR_PIN, HIGH);
      delay(3000);
    }
    else if ((char)payload[0] == 'f'){
      digitalWrite(MOTOR_PIN, LOW);
    }
}
```

Sau mỗi lần hàm Callback được gọi, nếu nhận về “o” thì motor máy bơm sẽ được bật trong thời gian 3 giây.

d. Gửi dữ liệu lên Server

```
    //Publishing Temperature to the server
    void Publishing_Temperature(float value) {
      char temp[4];
      itoa(value, temp, 10);
      char* message = temp; // convert value to char
      Client.publish("tronghuynhhuyhung/feeds/Temperature",
message, 4);
      delay(2000);
    }
```

Hàm Publishing_Temperature chức năng gửi dữ liệu lên server Adafruit. Với hàm Client.publish("tronghuynhhuyhung/feeds/Temperature", message, 4) dữ liệu sẽ được gửi lên server theo đường dẫn của Drashboard " tronghuynhhuyhung/ feeds/ Temperature "

```
    //Publishing SoilHumidity to the server
    void Publishing(int value) {
      char temp[4];
      itoa(value, temp, 10);
      char* message = temp;
      Client.publish("tronghuynhhuyhung/feeds/SoilHumidity",
message, 4);
      delay(2000);
    }
```

Hàm Publishing chức năng gửi dữ liệu lên server Adafruit. Với hàm Client.publish("tronghuynhhuyhung/feeds/SoilHumidity ", message, 4) dữ liệu sẽ được gửi lên server theo đường dẫn của Drashboard " tronghuynhhuyhung/ feeds/ SoilHumidity "

e. Hiển thị dữ liệu ra OLED

```
    // Print temp to OLED
    void SSD1306_Print_Temp(float temperature) {
      display.setCursor(0, 0);
      display.clearDisplay();
      display.display();
      display.println("TEMPERATUR");
      display.print(temperature);
      display.print(" oC");
      display.display();
    }
```

SSD1306_Print_Temp là hàm in nhiệt độ ra màn hình OLED

- display.setCursor: chọn vị trí ban đầu hiển thị cho nội dung
- display.clearDisplay(): đây là lệnh xóa màn hình
- display.display(): thông báo chuẩn bị truyền nội dung
- display.print(): nội dung được bỏ trong print(“”);
- display.display(): thông báo đã truyền xong nội dung

```
// Print Soil Humidity to OLED
void SSD1306_Print_Soil_Humidity(int Soil_Sensor_value) {
  display.setCursor(0, 0);
  display.clearDisplay();
  display.display();
  display.println("HUMIDITY");
  display.print(Soil_Sensor_value);
  display.print(" %");
  display.display();
}
```

SSD1306_Print_Soil_Humidity là hàm in nhiệt độ ra màn hình OLED

- display.setCursor: chọn vị trí ban đầu hiển thị cho nội dung
- display.clearDisplay(): đây là lệnh xóa màn hình
- display.display(): thông báo chuẩn bị truyền nội dung
- display.print(): nội dung được bỏ trong print(“”);
- display.display(): thông báo đã truyền xong nội dung

```
// Print real time to OLED
void SSD1306_Print_Realtime (const RtcDateTime& dt)
{
  display.setCursor(0, 0);
  display.clearDisplay();
  display.display();

  display.println("TIME:");
  display.print(dt.Hour());
  display.print(":");
  display.print(dt.Minute());
  display.print(":");
  display.println(dt.Second());
  display.display();
}
```

SSD1306_Print_Realtime là hàm in nhiệt độ ra màn hình OLED

- display.setCursor: chọn vị trí ban đầu hiển thị cho nội dung

- display.clearDisplay(): đây là lệnh xóa màn hình
- display.display(): thông báo chuẩn bị truyền nội dung
- display.print(): nội dung được bỏ trong print(“”);
- display.display(): thông báo đã truyền xong nội dung

f. Vòng lặp của chương trình

```
    Client.loop();
      RtcDateTime now = Rtc.GetDateTime();
```

Biến Rtc sẽ lấy thời gian từ mạch đếm thời gian thực DS1302

```
      if (!Client.connected()) {
        Adafruit_Server_reconnect();
      }
```

Nỗ lực kết nối với Server của ESP32

```
      Analog_Sensor_value =
analogRead(ANALOG_SENSOR_VALUE_PIN);
      DHT_Value_temperature = DHT_Sensor.readTemperature();
```

Đọc dữ liệu từ các cảm biến độ ẩm đất và DHT11 lưu vào biến Analog_Sensor_value và DHT_Value_temperature để phục vụ cho các mục đích sau.

```
      Percent = map(Analog_Sensor_value, map_low, map_high,
0, 100) ;
```

Định nghĩa lại giới hạn của cảm biến độ ẩm đất với giá trị sẽ từ 0 đến 100 tương ứng với 0% đến 100% độ ẩm đất.

```
      int HighHumi = 50; // set Soil Humidity threshold
      if (Percent >= HighHumi)
      {
         digitalWrite(MOTOR_PIN, LOW);
         Serial.println("off");
      }
      else
      {
         digitalWrite(MOTOR_PIN, HIGH);
         Serial.println("on");
         delay(3000);
         digitalWrite(MOTOR_PIN, LOW);
         Serial.println("off");
      }
```

Với việc chọn 50 là giá trị ngưỡng cho độ ẩm để quyết định motor có hoạt động hay không. Khi phát Sensor phát hiện độ ẩm dưới 50% motor sẽ bật trong 3s để tưới tiêu và ngược lại nếu phát hiện độ ẩm cao motor sẽ không hoạt động

```
      // use count variable so as not to affect other tasks
(Publish_count, Display_count)
      if(Publish_count == 20) {  // upload data to the server
after 20s
        if (!isnan(Digital_Sensor_value)) {
          Publishing(Percent);
          Publishing_Temperature(DHT_Value_temperature);
        }
        //printDateTime(now);
        Publish_count = 0;
      }
      Publish_count++;
```

Với việc biến Publish_count = 20 sẽ tương ứng với việc chúng ta đan cài đặt cho hệ thống cập nhật dữ liệu từ Sensor lên Server là 20 giây một lần. Việc sử dụng biến đếm thay cho việc dùng delay như thế sẽ không ảnh hưởng để các hoạt động khác, khi dùng delay mọi hoạt động đều chờ trong thời gian delay đó và biến đếm sẽ giải quyết được tình trạng trên.

```
      if (Display_count >= 0 && Display_count < 3) {  //
display time 3s
        //printDateTime();
        SSD1306_Print_Realtime(now);
        Display_count++;
      }
      else if (Display_count >= 3 && Display_count < 6){ //
display Soil Humidity 3s
        SSD1306_Print_Soil_Humidity(Percent);
        Display_count++;
      }
      else if (Display_count >= 6 && Display_count < 9){ //
display temperature 3s
        SSD1306_Print_Temp(DHT_Value_temperature);
        Serial.println(DHT_Value_temperature);
        Display_count++;
      }
      else {
        Display_count = 0;
      }
```

Cài đặt cho màn hình OLED sẽ hiển thị thời gian thực, độ ẩm đất và nhiệt độ với thời lượng là 3 giây.

```
      delay(1000);
```

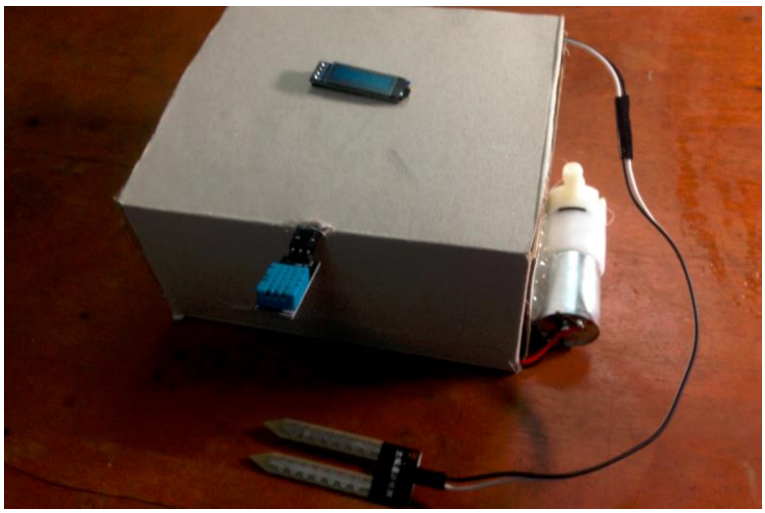
Với delay(1000) sẽ là đơn vị thời gian cho các biến đếm thời gian bên trên hoạt động đúng với yêu cầu đặt ra của hệ thống mà không ảnh hưởng các nhiệm vụ khác.

IV. KẾT QUẢ THỰC NGHIỆM

1. Các nội dung thực nghiệm

- Thực nghiệm thứ nhất: kết nối Wi-fi cho hệ thống
- Thực nghiệm thứ hai: Lấy nhiệt độ độ ẩm từ DHT11, cảm biến độ ẩm đất
- Thực nghiệm thứ ba: Lấy được thời gian thực từ DS1302
- Thực nghiệm thứ tư: Hiển thị các dữ liệu lấy được lên OLED
- Thực nghiệm thứ năm: Đưa dữ liệu lên adafruit và xử lý dữ liệu
- Thực nghiệm thứ sáu: Nhận tín hiệu từ adafruit điều khiển motor
- Thực nghiệm thứ bảy: ESP32 xử lý dữ liệu thu được và ra quyết định tưới tiêu

2. Kết quả thực nghiệm



Hình 16 Kết quả thực nghiệm

Video về sản phẩm, mô tả việc thực nghiệm theo trình tự và kết quả được đăng ở đường dẫn dưới đây: https://drive.google.com/drive/folders/18BbmNM0g

3. Đánh giá kết quả thực nghiệm

Đối với từng lần thử khác nhau, kết quả thực nghiệm đều cho ra đúng yêu cầu đã đặt ra của đề tài. Tuy nhiên nhóm chỉ mới kiểm tra một số lần nhất định, sau khi báo cáo kết thúc, nhóm sẽ tiếp tục kiểm tra để có thể phát hiện ngoại lệ hoặc sai xót nào đó khi sử dụng sản phẩm và có kế hoạch nâng cấp, sửa chữa cho sản phẩm một cách tốt nhất.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận đề tài

Tính đến thời điểm hiện tại, sản phẩm của nhóm đã được hoàn thiện cả về mẫu mã bên ngoài lẫn chức năng bên trong, đúng với mục đích ban đầu của nhóm đặt ra cho sản phẩm.

Đây là một đề tài không quá mới mẻ và khá dễ tiếp cận cũng như tìm hiểu để thực hiện, tuy nhiên đối với các bạn sinh viên thì đây là một đề tài cần thiết để các bạn làm quen với môn học hiện tại và các môn học phía sau nó.

Đề tài yêu cầu việc tìm hiểu nghiêm túc cùng khối lượng công việc và kiến thức bao quát phù hợp với các bạn sinh viên bắt đầu nghiên cứu về vi điều khiển và giao tiếp với các thiết bị phần cứng khác.

2. Ưu điểm và nhược điểm

a. Ưu điểm

- Tiết kiệm nước, thời gian và công sức tưới tiêu
- Theo dõi được quá trình tưới tiêu
- Giá thành sản phẩm thấp

b. Nhược điểm

- Chưa tối ưu được diện tích phần cứng
- Chưa tối ưu năng lượng tiêu thụ
- Độ chính xác mang tính tương đối

3. Hướng phát triển đề tài

- In mạch để tối ưu diện tích phần cứng
- Có thể điều chỉnh thời gian tưới tiêu bằng giao diện người dùng
- Có thể điều chỉnh mức nước tưới để phù hợp cho từng loại cây trồng
- Thêm các chức năng quản lý cây trồng khác như báo hiệu sâu bệnh, vàng lá,...

4. Mức độ đóng góp các thành viên

Bảng dưới đây được đánh giá dựa vào mức độ đóng góp cho đề tài của từng thành viên:

Tên thành viên	Mã số sinh viên	Mức độ đóng góp
Phạm Trung Quốc	19520887	33.3%
Phạm Quốc Đăng	19520036	33.3%
Phạm Trọng Huỳnh	19521651	33.3%

TÀI LIỆU THAM KHẢO

[1] "Project Hub," [Trực tuyến]. Available: https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239.

[2] "randomnerdtutorials," [Trực tuyến]. Available: https://randomnerdtutorials.com/getting-started-with-esp32/.

[3] Adafruit, "Adafruit," [Trực tuyến]. Available: https://learn.adafruit.com/.

[4] E. Systems, "espressif," 2022. [Trực tuyến]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[5] datasheetspdf, "datasheetspdf," [Trực tuyến]. Available: https://datasheetspdf.com/datasheet/DHT11.html.

[6] arduino.cc, "arduino.cc," [Trực tuyến]. Available: https://www.arduino.cc/en/Tutorial/HomePage.