ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ────────

**BÀI TẬP LỚN**

CÁC CÔNG NGHỆ TRUYỀN THÔNG CHO IOT

**XÂY DỰNG HỆ THỐNG   
TƯỚI CÂY TỰ ĐỘNG & GIÁM SÁT MỰC NƯỚC**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Mã sinh viên** |
| Nguyễn Trọng Hải | 20183730 |
| Nguyễn Minh Đức | 20183713 |
| Ngô Đình Sáng | 20183819 |
|  |  |
| Lớp | : **Kỹ thuật máy tính – Khóa 63** |
| Giảng viên hướng dẫn | : TS. Nguyễn Đức Toàn |

*Hà Nội, tháng 2 năm 202**2*  
MỤC LỤC

[DANH SÁCH HÌNH ẢNH 4](#_Toc95978601)

[DANH SÁCH BẢNG BIỂU 4](#_Toc95978602)

[CHƯƠNG 1. MÔ TẢ ĐỀ TÀI 6](#_Toc95978603)

[I. Lý do chọn đề tài 6](#_Toc95978604)

[II. Mô tả bài toán 6](#_Toc95978605)

[1. Mục đích sử dụng 6](#_Toc95978606)

[2. Kịch bản sử dụng 6](#_Toc95978607)

[3. Các đặc điểm quan trọng 7](#_Toc95978608)

[4. Các yêu cầu cần đạt được 7](#_Toc95978609)

[5. Các tác nhân sử dụng hệ thống 7](#_Toc95978610)

[III. Các chức năng và dịch vụ 7](#_Toc95978611)

[IV. Kế hoạch thực hiện 7](#_Toc95978612)

[V. Thành viên nhóm 8](#_Toc95978613)

[CHƯƠNG 2. MÔ TẢ HỆ THỐNG 9](#_Toc95978614)

[I. Mô hình hóa hệ thống 9](#_Toc95978615)

[1. Các giao thức lõi (Core protocols) 9](#_Toc95978616)

[2. Các cảm biến (Sensors) 9](#_Toc95978617)

[3. Các vi xử lý (MCUs) 10](#_Toc95978618)

[4. Các thiết bị thực thi (Execution Devices) 10](#_Toc95978619)

[5. Máy chủ và Cơ sở dữ liệu (Server & Database) 10](#_Toc95978620)

[6. Người dùng (Clients) 10](#_Toc95978621)

[II. Phân tích thiết thế 10](#_Toc95978622)

[1. Thiết kế Use cases 10](#_Toc95978623)

[2. Thiết kế Cơ sở dữ liệu 10](#_Toc95978624)

[3. Thiết kế Giao diện 12](#_Toc95978625)

[CHƯƠNG 3. GIẢI PHÁP CÔNG NGHỆ 13](#_Toc95978626)

[I. Mô hình hóa các công nghệ 13](#_Toc95978627)

[II. Chi tiết hóa hệ thống 14](#_Toc95978628)

[1. Giao thức MQTT và xây dựng MQTT broker 14](#_Toc95978629)

[2. Các cảm biến 14](#_Toc95978630)

[3. Hệ thống van và rơ-le 16](#_Toc95978631)

[4. Vi xử lý ESP32 18](#_Toc95978632)

[5. Website giám sát 22](#_Toc95978633)

[6. Cơ sở dữ liệu MongoDB 24](#_Toc95978634)

[CHƯƠNG 4. KẾT QUẢ THỰC HIỆN 25](#_Toc95978635)

[I. Hệ thống nhúng 25](#_Toc95978636)

[1. Đọc kết quả từ cảm biến 25](#_Toc95978637)

[2. Điều khiển van bơm 27](#_Toc95978638)

[II. Website 28](#_Toc95978639)

[1. Chức năng xem cây/thùng nước 28](#_Toc95978640)

[2. Chức năng bật/tắt van 30](#_Toc95978641)

[CHƯƠNG 5. KẾT LUẬN 31](#_Toc95978642)

[I. Khó khăn và bài học 31](#_Toc95978643)

[II. Kết luận 31](#_Toc95978644)

[III. Hướng phát triển 31](#_Toc95978645)

[TÀI LIỆU THAM KHẢO 32](#_Toc95978646)

**DANH SÁCH HÌNH ẢNH**

[*Hình 1: Ảnh vườn bưởi của một hộ dân* 6](#_Toc95978535)

[*Hình 2: Mô hình hóa hệ thống* 9](#_Toc95978536)

[*Hình 3: Biểu đồ Use cases* 10](#_Toc95978537)

[*Hình 4: Thiết kế giao diện mẫu* 12](#_Toc95978538)

[*Hình 5: Mô hình hóa các công nghệ* 13](#_Toc95978539)

[*Hình 6: Cảm biến đô khoảng cách HY-SRF 05* 15](#_Toc95978540)

[*Hình 7: Cảm biến điện dung đo độ ẩm* 16](#_Toc95978541)

[*Hình 8: Van điện từ* 17](#_Toc95978542)

[*Hình 9: Mô-đun rơ-le 2 kênh* 18](#_Toc95978543)

[*Hình 10: Vi điều khiển ESP32* 19](#_Toc95978544)

[*Hình 11: Các chân kết nối của ESP32* 20](#_Toc95978545)

[*Hình 12: Sơ đồ đấu nối mạch Nhúng* 20](#_Toc95978546)

[*Hình 13: Hệ thống nhúng triển khai thực tế* 25](#_Toc95978547)

[*Hình 14: Giao diện Web* 28](#_Toc95978548)

[*Hình 15: Giao diện danh sách các cây* 28](#_Toc95978549)

[*Hình 16: Giao diện một cây* 29](#_Toc95978550)

[*Hình 17: Giao diện danh sách các thùng nước* 29](#_Toc95978551)

[*Hình 18: Giao diện một thùng nước* 29](#_Toc95978552)

[*Hình 19: Giao diện khi dang bật van nước* 30](#_Toc95978553)

[*Hình 20: Nước đang được bơm vào cả 2 cây* 30](#_Toc95978554)

**DANH SÁCH BẢNG BIỂU**

[*Bảng 1: Bảng User* 10](#_Toc95978526)

[*Bảng 2: Bảng MCU ESP32* 11](#_Toc95978527)

[*Bảng 3: Bảng cây* 11](#_Toc95978528)

[*Bảng 4: Bảng thùng nước* 11](#_Toc95978529)

[*Bảng 5: Bảng đấu nối Module cảm biến đo khoảng cách HY-SRF05* 20](#_Toc95978530)

[*Bảng 6: Bảng đấu nối đo độ ẩm số 1* 21](#_Toc95978531)

[*Bảng 7: Bảng đấu nối đo độ ẩm số 2* 21](#_Toc95978532)

[*Bảng 8: Bảng đấu nối điều khiển van* 21](#_Toc95978533)

[*Bảng 9: Bảng đấu nối nguồn điện điều khiển van* 21](#_Toc95978534)

**CHƯƠNG 1. MÔ TẢ ĐỀ TÀI**

## I. Lý do chọn đề tài

Một vườn cây nhỏ có thể tự tay người nông dân chăm sóc hằng ngày mà không gặp nhiều khó khăn, nhưng vườn cây lớn là cả một thách thức không nhỏ. Đơn giản nhất là việc tưới nước đầy đủ cho cây cũng đã tốn kém thời gian cho người nông dân.



*Hình 1: Ảnh vườn bưởi của một hộ dân*

Nếu một hệ thống đường ống nước bình thường sẽ giúp người nông dân không phải di chuyển quá nhiều nhưng rất khó để kiểm soát lượng nước, đôi khi là gây lãng phí hoặc không đủ độ ẩm cho đất.

Một hệ thống tưới cây tự động sẽ giúp đảm bảo độ ẩm đất luôn ở trong khoảng thích hợp cho từng cây. Người chủ sẽ bớt bận tâm về việc phải tưới nước đầy đủ cho từng cây. Cùng với đó, hệ thống cũng giám sát mực nước trong các thùng chứa để kiểm soát lượng nước được sử dụng.

## II. Mô tả bài toán

### 1. Mục đích sử dụng

Tưới nước đầy đủ, giữ độ ẩm từng cây trong khoảng thích hợp và kiểm soát lượng nước được sử dụng để tưới từng cây.

### 2. Kịch bản sử dụng

Kịch bản 1: Nước tự động được bơm nếu dưới mức cho phép và ngắt khi đến ngưỡng cho phép.

* Kịch bản 2: Người dùng muốn cho phép tưới cây theo ý mình.
* Kịch bản 3: Xem trạng thái của từng cây, lượng nước còn lại trong thùng.

### 3. Các đặc điểm quan trọng

Các cảm biến cho ra kết quả chính xác.

Lượng nước được tính toán chính xác.

Điều khiển ổn định, trễ ít.

### 4. Các yêu cầu cần đạt được

Độ ẩm của cây luôn được duy trì trong mức cho phép.

Lượng nước luôn ở trạng thái sẵn sàng tưới.

Người dùng có thể điều khiển tưới nước.

### 5. Các tác nhân sử dụng hệ thống

Người dùng sử dụng.

Người quản lý hệ thống.

## III. Các chức năng và dịch vụ

Xem, sửa, xóa thông tin cây và thùng nước trong hệ thống.

Quản lý các thiết bị,

Chủ động điều khiển tưới nước.

Xem lịch sử lưu lượng nước tưới từng cây.

## IV. Kế hoạch thực hiện

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Người tham gia** | **Thời điểm**  **bắt đầu** | **Thời điểm**  **kết thúc** |
| Mô tả chi tiết bài toán | Nguyễn Trọng Hải | 8/11 | 20/11 |
| ESP32 | Nguyễn Trọng Hải | 22/11 | 27/11 |
| Relay điều khiển + Van điện từ | Nguyễn Trọng Hải | 27/11 | 8/12 |
| Cảm biến độ ẩm đất + Cảm biến siêu âm | Ngô Đình Sáng | 27/11 | 8/12 |
| Hoàn thiện hệ thống Nhúng | Nguyễn Trọng Hải  Ngô Đình Sáng | 13/12 | 31/12 |
| Lập trình Nhúng | Nguyễn Trọng Hải  Ngô Đình Sáng | 20/12 | 7/1 |
| Use case | Nguyễn Trọng Hải  Nguyễn Minh Đức | 27/11 | 18/12 |
| Thiết kế Database | Nguyễn Minh Đức | 27/11 | 18/12 |
| Thiết kế Front-end | Ngô Đình Sáng | 27/11 | 18/12 |
| Thiết kế Back-end | Nguyễn Minh Đức | 27/11 | 18/12 |
| Lập trình Web | Ngô Đình Sáng  Nguyễn Minh Đức | 20/12 | 14/1 |
| Làm Báo cáo | Nguyễn Trọng Hải  Ngô Đình Sáng | 27/11 | 21/1 |
| Làm Slides | Nguyễn Trọng Hải  Nguyễn Minh Đức | 27/11 | 21/1 |

## V. Thành viên nhóm

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Email** |
| Nguyễn Trọng Hải | 20183730 | [hai.nt183730@sis.hust.edu.vn](mailto:hai.nt183730@sis.hust.edu.vn) |
| Nguyễn Minh Đức | 20183713 | [duc.nm183713@sis.hust.edu.vn](mailto:duc.nm183713@sis.hust.edu.vn) |
| Ngô Đình Sáng | 20183819 | [sang.nd183819@sis.hust.edu.vn](mailto:sang.nd183819@sis.hust.edu.vn) |

**CHƯƠNG 2. MÔ TẢ HỆ THỐNG**

## I. Mô hình hóa hệ thống

Hệ thống gồm các thành phần chính như hình bên dưới:

+ Các giao thức lõi (Core protocols);

+ Các cảm biến (Sensors);

+ Các vi xử lý (MCUs);

+ Các thiết bị thực thi (Execution Devices);

+ Máy chủ và Cơ sở dữ liệu (Server & Database).

+ Người dùng + giám sát (Clients).

Diagram

Description automatically generated

*Hình 2: Mô hình hóa hệ thống*

### 1. Các giao thức lõi (Core protocols)

Giao thức nằm ở trung tâm của hệ thống, có vai trò truyền các thông điệp giữa các vi xử lý, vi điều khiển và các thành phần khác như server, clients…

### 2. Các cảm biến (Sensors)

Các cảm biến đo nhưng thông số về độ ẩm của từng cây, lượng nước của mỗi thùng. Dữ liệu đo được sẽ được đưa về các vi xư lý.

### 3. Các vi xử lý (MCUs)

Các vi xử lý nhận dữ liệu từ các cảm biến để gửi tới các thành phần khác thông qua giao thức lõi, đồng thời nhận tín hiệu điều khiển để truyền đến các thiết bị thực thi.

### 4. Các thiết bị thực thi (Execution Devices)

Các thiết bị thực thi nhận tín hiệu điều khiển từ các vi xử lý và thực hiện yêu cầu nhận được.

### 5. Máy chủ và Cơ sở dữ liệu (Server & Database)

Trạm quan sát nhận dữ liệu thời gian thực của tất cả các cảm biến, tính toán các thông số cần thiết.

Xử lý những yêu cầu truy vấn của trạm quan sát và người dùng khi cần thiết.

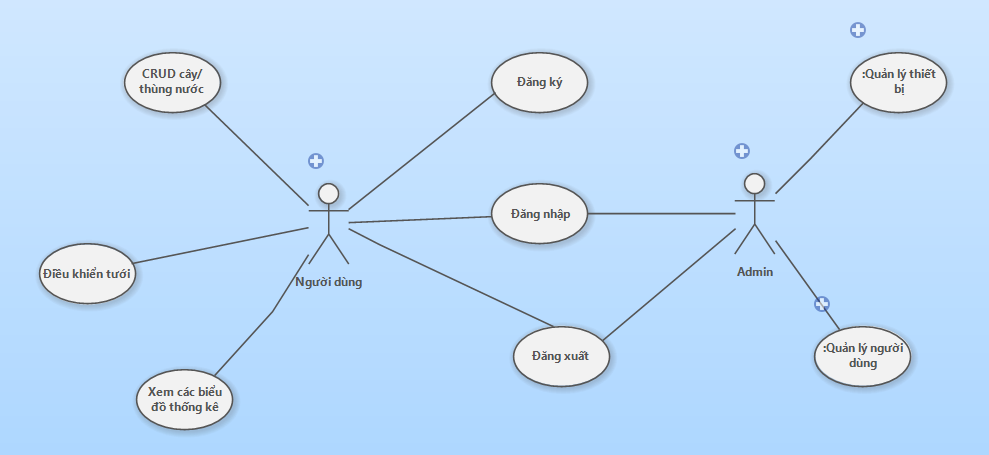
### 6. Người dùng (Clients)

Người dùng nhận dữ liệu thời gian thực về thông số của các cây, lượng nước trong các thùng chứa.

Người dùng chủ động điều khiển việc tưới nước cho cây.

## II. Phân tích thiết thế

### 1. Thiết kế Use cases



*Hình 3: Biểu đồ Use cases*

### 2. Thiết kế Cơ sở dữ liệu

*Bảng 1: Bảng User*

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| \_id | ObjectID | Id trong mongo |
| account | String |  |
| password | String |  |
| isAdmin | Boolean | Quyền Admin |

*Bảng 2: Bảng MCU ESP32*

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| \_id (MongoDB) | ObjectID | Id trong mongo |
| owner | ObjectID | Tham chiếu tới id của user |
| espid | String | Id phần cứng của esp32 |

*Bảng 3: Bảng cây*

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| \_id (MongoDB) | ObjectID | Id trong mongo |
| espid | String | Tham chiếu tới id của esp32 quản lý |
| index | Number | Số thứ tự cây |
| name | String | Tên cây |
| maxMoisture | Number | Độ ẩm tối đa của cây |
| minMoisture | Number | Độ ẩm tối thiểu |
| currentMoisture | Number | Độ ẩm hiện tại |
| isValveOpen | Boolean | true = mở, false = đóng |

*Bảng 4: Bảng thùng nước*

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| \_id (MongoDB) | ObjectID | Id trong mongo |
| espid | String | Tham chiếu tới id của esp32 quản lý |
| index | Number | Số thứ tự thùng nước |
| area | Number | Diện tích đáy thùng |
| height | Number | Chiều cao thùng |
| liquidLevel | Number | Mực nước hiện tại (tính từ đáy lên) |

### 3. Thiết kế Giao diện

Graphical user interface, application

Description automatically generated

*Hình 4: Thiết kế giao diện mẫu*

**CHƯƠNG 3.** **GIẢI PHÁP CÔNG NGHỆ**

## I. Mô hình hóa các công nghệ

Hệ thống gồm các thành phần như hình bên dưới:

+ Giao thức lõi là MQTT;

+ Vi xử lý ESP32;

+ Các cảm biến: đo khoảng cách, đo độ ẩm đất;

+ Hệ thống van và rơ-le;

+ Website;

+ Máy chủ NodeJS;

+ Cơ sở dữ liệu MongoDB.

Diagram

Description automatically generated

*Hình 5: Mô hình hóa các công nghệ*

## II. Chi tiết hóa hệ thống

### 1. Giao thức MQTT và xây dựng MQTT broker

#### 1.1 Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp/thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt [2].

Sử dụng giao thức MQTT cho bài toán này là bởi vì đây là một bài toán không đòi hỏi xử lý ở cường độ cao, những tính toán khá đơn giản và lượng dữ liệu không lớn.

#### 1.2 Xây dựng MQTT broker

Broker có sẵn: EMQX Broker [3]

Link tham khảo : <https://www.emqx.io/>

2. Các cảm biến

#### 2.1 Cảm biến siêu âm

- Thông số kỹ thuật [4]:

Trigger pin: xung cỡ 10us

Echo pin output: 0-Vcc

Khoảng cách đo: 2cm đến ~4.5m

Độ chính xác cỡ 3mm

Góc lệch tối đa: 15 độ

Tần số âm thanh: 40kHz

Tốc độ đo: 40Hz

Điện áp: 4.5V-5V

Dòng điện: 10 - 40mA

Kết nối: 5 chân (**Vcc, Trig, Echo, OUT, GND**)



*Hình 6: Cảm biến đô khoảng cách HY-SRF 05*

- Cách sử dụng:

Duy trì điện áp giữa Vcc và GND trong khoảng 4.5V – 5V.

Tạo xung khoảng 10us cho chân Trigger.

Module gửi 8 tín hiệu 40KHz và chờ nhận xung trả về và tính khoảng cách đo được bao gồm thời gian lan truyền đi và lan truyền về:

***Khoảng cách*** = ***thời gian lan truyền*** \* ***vận tốc ánh sáng / 2***

=> Cách tính khoảng cách theo mm.

Tốc độ âm thanh 343m/s ~ (343\*1000)/1000000 = 343/1000 (mm/uS).

***Khoảng cách*** = ***thời gian lan truyền*** **\*343/1000/2.**

#### 2.2 Cảm biến đo độ ẩm đất

- Thông số kỹ thuật [5]:

Điện áp 3.3 – 5.5 V

Output: 0 - 3.0 VDC

Dòng diện: 5mA

Giao diện: PH2.0-3P

Kết nối: 3 chân (**VCC,Analog, GND**)

Graphical user interface, text

Description automatically generated with medium confidence

*Hình 7: Cảm biến điện dung đo độ ẩm*

- Cách sử dụng:

Duy trì điện áp giữa Vcc và GND trong khoảng 3.3V – 5V.

Đo giá trị tương tự Analog trong không khí, gọi là “AirValue”.

Đo giá trị tương tự Analog trong nước, gọi là “WaterValue”.

Giá trị độ ẩm đất được đặt từ min 0% tại “AirValue” và max 100%

“WaterValue”.

Giá trị analog đo độ ẩm đất đo được sẽ được so sánh với thang đo 100 để ra được kết quả % độ ẩm đất.

3. Hệ thống van và rơ-le

#### 3.1 Van điện từ

- Thông số kỹ thuật [6]:

Điện áp 3.3 – 4.5 VDC

Dòng diện: ~ 200mA

A picture containing text

Description automatically generated

*Hình 8: Van điện từ*

- Cách sử dụng:

Van thường đóng, khi có điện áp 3.3 – 4.5V chạy qua thì van mở.

#### 3.2 Rơ-le 2 kênh

- Thông số kỹ thuật [7][8]:

Điện áp 3.375 – 6 V

Output: 5V, 10A/250VAC, 10A / 30 VDC

Trigger: 5mA

Dòng diện: 70mA (1 kênh), 140mA (2 kênh)

Kết nối: 10 chân (**VCC,COM,IN1,IN2, GND,NC1,COM1,NO1, NC2,COM2,NO2**)

IN1: điều khiển kênh 1

IN2: điều khiển kênh 2

COM: thường nối với chân lửa của điện xoay chiều hoặc cực dương của điện một chiều

NO: thường nối với chân lửa của điện xoay chiều hoặc cực dương của điện một chiều

NC: thường nối với chân lạnh của điện xoay chiều hoặc cực âm của điện một chiều

Graphical user interface

Description automatically generated

*Hình 9: Mô-đun rơ-le 2 kênh*

- Cách sử dụng:

Module cho phép chọn xung kích hoạt “HIGH” hoặc “LOW”, như hình trên là module được kích hoạt ở mức “HIGH”

Tín hiệu truyền đến IN1 là “HIGH” thì kênh 1 được bật, dòng diện sẽ được nối 2 trong 3 chân COM1, NO1, NC1 (tùy cách mắc chân vào van, cụ thể chọn 2 chân COM1 và NO1).

Tương tự, tín hiệu truyền đến IN2 là “HIGH” thì kênh 2 được bật, dòng diện sẽ được nối 2 trong 3 chân COM2, NO2, NC2 (tùy cách mắc chân vào van, cụ thể chọn 2 chân COM2 và NO2).

Ngược lại, tín hiệu đến IN1, IN2 là “LOW” sẽ tắt tương ứng từng kênh.

### 4. Vi xử lý ESP32

#### 4.1 Xi xử lý ESP32

ESP32 là thế hệ tiếp theo của ESP8266, có cải tiến là dual-core và khả năng kết nối Wi-fi, Bluetooth [9].

A close-up of a computer chip

Description automatically generated with medium confidence

*Hình 10: Vi điều khiển ESP32*

ESP32 là thế hệ tiếp theo của ESP8266, có cải tiến là dual-core và khả năng kết nối Wi-fi, Bluetooth.

Thông số ESP32 DEVKIT DOIT board

+ Core: dual-core

+ Wifi: 2.4GHz to 150Mbits/s

+ Bluetooth: BLE (Bluetooh Low Energy) và thừa kế đặc trưng Bluetooth

+ Kiến trúc: 32bit

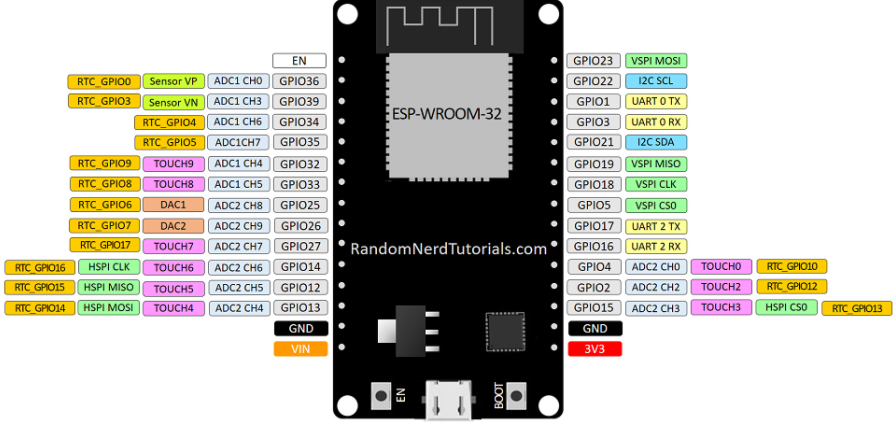
+ Xung lên tới 240 MHz

+ RAM: 512kB

+ Pins: 30 – 36 tùy mẫu

+ Mức logic là 3.3V

+ Nhiều kết nối ngoại vi



*Hình 11: Các chân kết nối của ESP32*

#### 4.2 Sơ đồ đấu nối mạch

Diagram

Description automatically generated

*Hình 12: Sơ đồ đấu nối mạch Nhúng*

#### 4.3 Bảng chi tiết thiết kế các đấu nối

*Bảng 5: Bảng đấu nối Module cảm biến đo khoảng cách HY-SRF05*

|  |  |
| --- | --- |
| Cảm biến đo khoảng cách | ESP32 |
| Vcc | Vin |
| Trig | D13 |
| Echo | D12 |
| OUTPUT |  |
| GND | GND |

*Bảng 6: Bảng đấu nối đo độ ẩm số 1*

|  |  |
| --- | --- |
| Cảm biến đo độ ẩm 1 | ESP32 |
| Vcc | 3V3 |
| A | D32 |
| GND | GND |

*Bảng 7: Bảng đấu nối đo độ ẩm số 2*

|  |  |
| --- | --- |
| Cảm biến đo độ ẩm 2 | ESP32 |
| Vcc | 3V3 |
| A | D33 |
| GND | GND |

*Bảng 8: Bảng đấu nối điều khiển van*

|  |  |  |  |
| --- | --- | --- | --- |
| Van | Relay 2 kênh | ESP32 | Relay 2 kênh |
| Vcc1 | COM1 | D2 | IN1 |
| GND1 | NO1 | D4 | IN2 |
|  | NC1 | Vin | Vcc |
| Vcc2 | COM2 | GND | GND |
| GND2 | NO2 |  |  |
|  | NC2 |  |  |

*Bảng 9: Bảng đấu nối nguồn điện điều khiển van*

|  |  |  |
| --- | --- | --- |
| ESP32 | Van | Relay 2 kênh |
| Vin | Vcc |  |
| 3V3 |  | COM |
| GND | GND | NO |

### 5. Website giám sát

#### 5.1. Server - NestJS

Icon

Description automatically generated

NodeJS là một Javascript Runtime Environment mã nguồn mở và đa nền tảng. Nó được viết dựa trên V8 Javascript Engine với cơ chế đơn luồng, không đồng bộ và non-blocking, vì vậy NodeJS chạy rất nhanh và rất thích hợp với hệ thống IoT cần các cơ chế real-time

Logo, company name

Description automatically generated

Hiện nay có rất nhiều framework backend cho NodeJS, phổ biến nhất là ExpressJS, SailJS, KoaJS,… Tuy nhiên các framework này đều sử dụng Javascript nên đều có một vấn đề chung đó là vấn đề về kiến trúc và rất khó kiểm soát kiểu dữ liệu do Javascript là ngôn ngữ dynamic type. NestJS [11] ra đời như một giải pháp cho vấn đề này, nó là một framework mạnh mẽ để xây dựng một ứng dụng phía máy chủ với kiến trúc module được lấy cảm hứng từ framework frontend Angular cùng với rất nhiều các module tiện ích như Security, Websocket, GraphQL, Microservice… Tương tự như Angular, NestJS sử dụng Typescript thay vì Javascript để giải quyết vấn đề về dynamic type.

A blue sign with white text

Description automatically generated with medium confidence

Typescript là một ngôn ngữ mã nguồn mở do Microsoft phát triển, ngôn ngữ này có thể coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript. Bản chất của Typescript là biên dịch ra mã Javascript vì vậy Typescript có thể chạy trên NodeJS, hiện nay các Javascript framework cũng được khuyến khích chuyển sang dùng Typescript.

#### 5.2. Client – ReactJS

**A blue logo with black text

Description automatically generated with low confidence**

Ngôn ngữ lập trình duy nhất được dùng để xây dựng ứng dụng phía client và chạy trên trình duyệt là Javascript (gần đây có WebAssembly, tuy nhiên ngôn ngữ này chưa hoàn thiện), vì vậy có rất nhiều Javascript libraries và frameworks để hỗ trợ xây dựng ứng dụng web phía client. Trong số đó, phổ biến nhất hiện nay chính là ReactJS [12], đây là một library mã nguồn mở do Facebook phát triển. ReactJS sử dụng cú pháp có tên gọi là JSX (Javascript + XML) để biểu thị UI (User Interface) components thay vì HTML truyền thống, ReactJS không bắt buộc dùng JSX tuy nhiên nếu sử dụng sẽ giúp quản lý từng thành phần của trang web một cách đơn giản và dễ đọc hơn.

#### 5.3. Một số công nghệ khác

***5.3.1.* *Json Web Token***

JWT [15] là một chuẩn mở định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA. Trong hệ thống này, JWT sử dụng để xác thực người dùng của các HTTP request – vì HTTP là stateless nên mỗi khi gửi một request mới nó cần gắn JWT để máy chủ có thể xác định là ai đang gửi request.

***5.3.2. Mqtt***

Thư viện cho phép subscribe vào broker để theo dõi dữ liệu một cách realtime phía client.

***5.3.3. Websocket – Socket.io***

Websocket được sử dụng để truyền dữ liệu real-time giữa client và server, khác với HTTP là server chỉ có thể gửi response khi nhận request thì ở đây server có thể gửi dữ liệu cho client bất cứ lúc nào (ví dụ khi nhận được update từ cảm biến và muốn update lên giao diện người dùng). Kích thước của những dữ liệu được truyền qua websocket nhỏ hơn rất nhiều lần so với kích thước của các gói tin HTTP vì vậy nên có tốc độ rất cao và gần như là real-time.

Socket.IO [13] là một thư viện rất nổi tiếng của NodeJS hỗ trợ cho Websocket và có thể được sử dụng cả ở phía server lẫn client, sử dụng cơ chế lắng nghe và phát sự kiện của NodeJS.

### 6. Cơ sở dữ liệu MongoDB

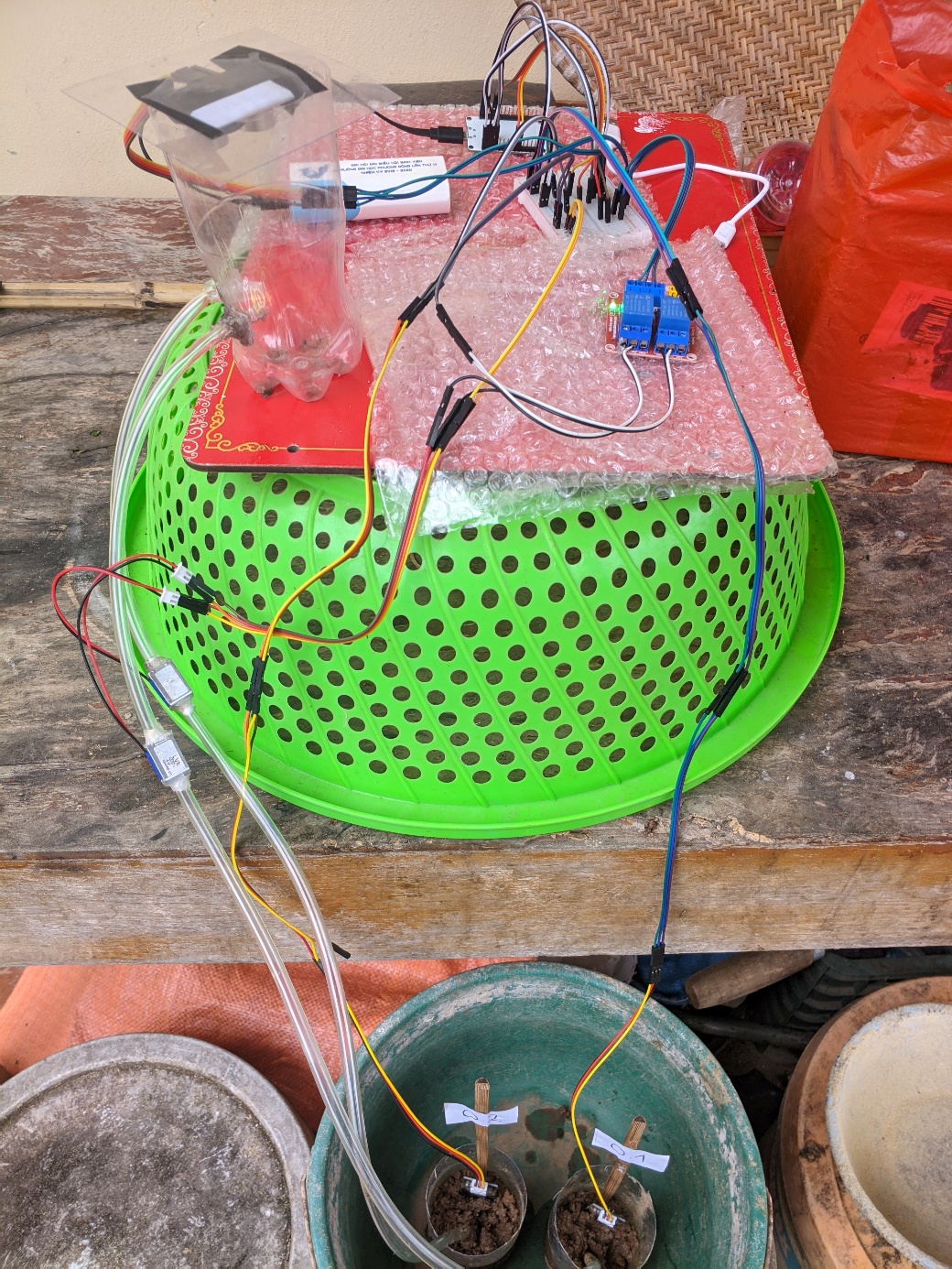
MongoDB [14] là một cơ sở dữ liệu NoSQL, có thể lưu trữ các JSON trong đó, và cấu trúc của các tài liệu này có thể thay đổi vì nó không bắt buộc như các cơ sở dữ liệu SQL. Đây là một trong những lợi thế của việc sử dụng NoSQL vì nó tăng tốc độ phát triển ứng dụng và giảm sự phức tạp của việc triển khai.

# 

# CHƯƠNG 4. KẾT QUẢ THỰC HIỆN

## I. Hệ thống nhúng

- Hệ thống nhúng được triển khai thực tế như hình **13** như đã thiết kế.



*Hình 13: Hệ thống nhúng triển khai thực tế*

### 1. Đọc kết quả từ cảm biến

- Khởi tạo các biến

Text

Description automatically generated

Text

Description automatically generated

#### 1.1 Đo khoảng cách

Graphical user interface, text, application

Description automatically generated with medium confidence

#### 1.2 Đo độ ẩm cây

Text

Description automatically generated with medium confidence

#### 1.3 Lấy trạng thái van bơm

Graphical user interface, text, application

Description automatically generated

### 2. Điều khiển van bơm

Graphical user interface, text

Description automatically generated

-

## II. Website

Graphical user interface, application, chat or text message

Description automatically generated

*Hình 14: Giao diện Web*

### 1. Chức năng xem cây/thùng nước

#### 1.1 Chức năng xem cây

Danh sách các loại cây sẽ được hiển thị như hình **15**. Nếu có nhiều cây thì sẽ trượt xuống để xem các cây khác.

Graphical user interface, text, application

Description automatically generated

*Hình 15: Giao diện danh sách các cây*

Thông tin mỗi cây sẽ hiển thị như hình **16**. Mỗi cây sẽ có một ID đánh số thự tự, một khoảng độ ẩm phù hợp cho cây và một thang độ ẩm hiện tại. Người dùng có thể chỉnh sửa thông tin của cây và bật/tắt chế độ tưới nước cho cây.

Graphical user interface, text, application, chat or text message

Description automatically generated

*Hình 16: Giao diện một cây*

#### 1.2 Chức năng xem thùng nước

Danh sách các loại cây sẽ được hiển thị như hình **17**. Nếu có nhiều thùng thì sẽ trượt xuống để xem các thùng nước khác.

Graphical user interface, text, application

Description automatically generated

*Hình 17: Giao diện danh sách các thùng nước*

Thông tin mỗi cây sẽ hiển thị như hình **18**. Mỗi cây sẽ có một ID đánh số thự tự, mực nước hiện tại và diện tích đáy thùng để tính thể tích, số phần trăm lượng nước như trên hình đang là 60%.

Graphical user interface, text, application

Description automatically generated

*Hình 18: Giao diện một thùng nước*

### 2. Chức năng bật/tắt van

#### 2.1 Chức năng bật/tắt van từ Web

Người dùng có thể chủ động bật/tắt van nước từ Web nếu lượng nước nằm trong khoảng mức độ thích hợp của cây.

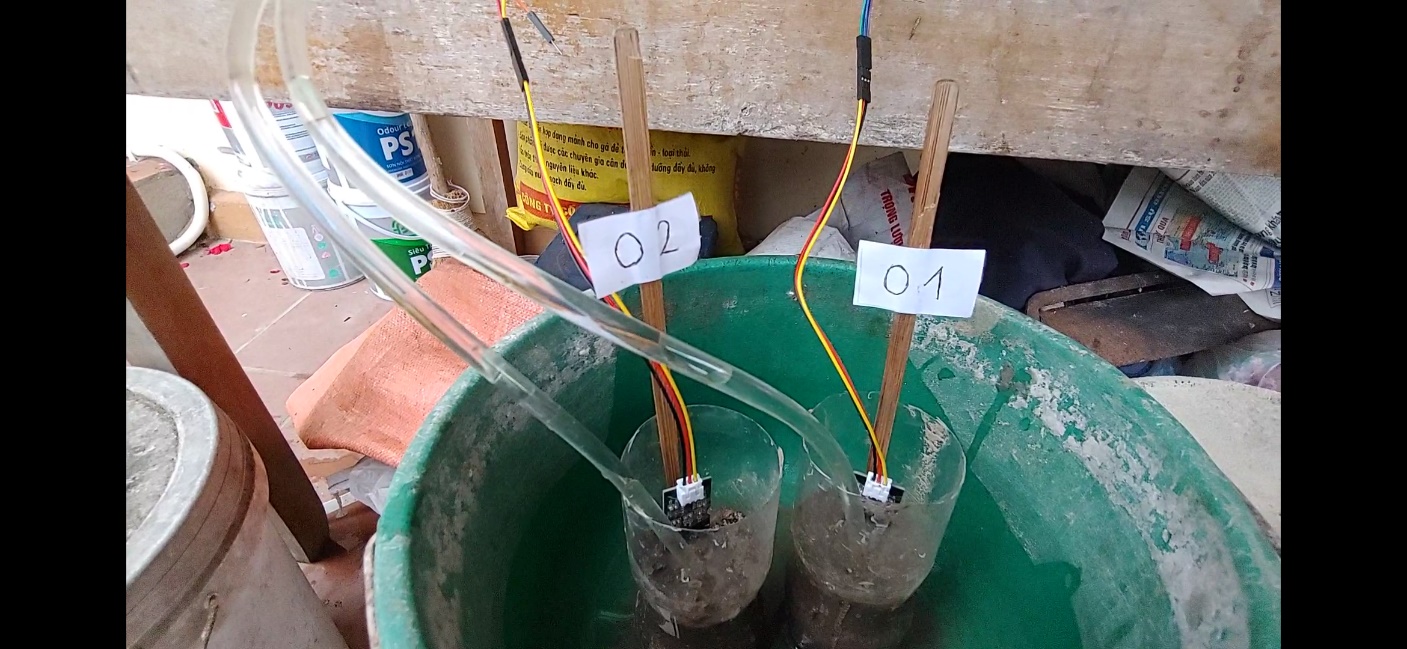
Graphical user interface, text, application

Description automatically generated

*Hình 19: Giao diện khi dang bật van nước*

#### 2.2 Chức năng tự động bật/tắt van

Nếu độ ẩm xuống dưới độ ẩm thích hợp, van sẽ tự động bật để bơm nước như hình **20**, đến khi độ ẩm vượt qua độ ẩm thích hợp thì van sẽ tự động tắt.



*Hình 20: Nước đang được bơm vào cả 2 cây*

# CHƯƠNG 5. KẾT LUẬN

## I. Khó khăn và bài học

Cả một học kỳ học online nên việc triển khai thực hiện giữa các thành viên còn gặp nhiều khó khăn. Kết hợp triển khai giữa phần cứng và phần mềm tốn thời gian vì cần họp online và khi triển khai có sự cố không khớp vì không nắm được chi tiết các công việc của nhau.

Các cảm biến giá rẻ như cảm biến đo độ ẩm đất cho ra kết quả không chính xác nên rất khó khăn trong việc triển khai. Nhóm đã cố gắng khắc phục nhưng mọi thành phần khác đều hoạt động tốt, trừ cảm biến đo độ ẩm đất (một thành phần cực kỳ quan trọng trong bài toán).

Nhóm đã chia ra mỗi người làm một phần riêng nhưng phải có sự hỗ trợ lần nhau mới có thể hoàn thành, một người làm độc lập thực sự rất khó. Hệ thống gồm nhiều thành phần đòi hỏi tính liên kết và đồng bộ cao nên quá trình đồng bộ đòi hỏi các thành viên phải liên hệ với nhau rất nhiều để kiểm tra và sửa lỗi.

## II. Kết luận

Cả hệ thống Nhúng và Web đều hoạt động tốt với các chức năng đã được hoàn thiện, trao đổi giữa các thiết bị vật lý và phần mềm khá chính xác mặc dù trễ hơi lâu (1-2 giây) do đường truyền mạng và triển khai trên “**ngrok**”. Một phần thành công là do sử dụng giao thưc MQTT là một loại giao thức hướng thông điệp hoạt động nhanh, chính xác, ổn định và không gặp lỗi trong quá trình sử dụng.

Nông nghiệp thông minh là một hướng đi lâu dài và bền vững cho nông nghiệp, đây hứa hẹn là một vùng đầy tiềm năng trong công nghiệp công nghệ cao. Các bài toán như tưới nước cho cây, giám sát chất lượng dinh dưỡng ngày càng được chú trọng, có khả năng sẽ phát triển lớn trong tương lai.

## III. Hướng phát triển

Đầu tiên, nhóm cần hoàn thiện những chức năng còn thiếu, chưa hoàn thiện. Đây là công việc cần thiết nhất vì thời gian làm bài tập lớn không đủ nhiều để hoàn thiện tất cả các chức năng. Các chức năng đến hiện tại là tối giản để đảm bảo minh họa những dịnh vụ quan trọng nhất mà người dùng cần.

Hoàn thiện sản phẩm để khả thi hơn trong thực tế vói chi phí thấp vì chi phí cho các thiết bị khá tốn kém nhưng độ bền lại không cao, giá thành của hệ thống sẽ làm giá của nông sản và cây trồng tăng cao.

Triển khai App trên GooglePlay, Webside trên tên miền thật để đánh giá kiểm tra, đánh giá các lỗi để tiến hành sửa chữa.

**TÀI LIỆU THAM KHẢO**

[1] Bài giảng học phần Các công nghệ truyền thông cho IOT – TS. Nguyễn Đức Toàn.

[2] Giao thức MQTT:

<https://viblo.asia/p/mqtt-la-gi-vai-tro-cua-mqtt-trong-iot-V3m5WL3bKO7>

[3] MQTT Broker

<https://www.emqx.com/en/blog/esp32-connects-to-the-free-public-mqtt-broker>

[4] Cảm biến HY-SRF05

<http://www.datasheet-pdf.com/PDF/HY-SRF05-Datasheet-ETC-813041>

[5] Cảm biến đo độ ẩm

<https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf>

[6] Van điện từ

<https://vi.aliexpress.com/item/32941109543.html>

[7] Moldule relay điều khiển 2 kênh

<https://www.robodukkan.com/class/INNOVAEditor/assets/2_CHANNEL_5V_10A_RELAY_MODULE.pdf>

[8] Moldule relay điều khiển 2 kênh

<https://components101.com/switches/5v-dual-channel-relay-module-pinout-features-applications-working-datasheet>

[9] Vi xử lý ESP32

<https://randomnerdtutorials.com/getting-started-with-esp32/>

[10] Lấy địa chỉ MAC của ESP32

<https://arduino.stackexchange.com/questions/58677/get-esp32-chip-id-into-a-string-variable-arduino-c-newbie-here>

[11] NestJS: [NestJS - A progressive Node.js framework](https://nestjs.com/)

[12] ReactJS: [React – A JavaScript library for building user interfaces (reactjs.org)](https://reactjs.org/)

[13] Socket.IO: [Socket.IO](https://socket.io/)

[14] Mongoose: [Mongoose ODM v6.2.1 (mongoosejs.com)](https://mongoosejs.com/)

[15] JWT: [JWT authentication: Best practices and when to use it - LogRocket Blog](https://blog.logrocket.com/jwt-authentication-best-practices/)