

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**NGHIÊN CỨU VÀ THI CÔNG TỦ TRỒNG RAU
NGẮN NGÀY TRONG NHÀ**

**GVHD: TRƯƠNG NGỌC ANH
SVTH: TRẦN TẤN SANG
SVTH: HUỲNH HỒNG SƠN**



Tp. Hồ Chí Minh, tháng 08/2020

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH**



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

**NGHIÊN CỨU VÀ THI CÔNG
TỦ TRỒNG RAU NGẮN NGÀY TRONG NHÀ**

GVHD: Th.s Trương Ngọc Anh

SVTH:

MSSV:

Trần Tấn Sang

16141256

Huỳnh Hồng Sơn

16141259

Tp. Hồ Chí Minh – 8/2020

Tp. HCM, ngày 20 tháng 7 năm

2020

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên:	Trần Tấn Sang	MSSV: 16141256
	Huỳnh Hồng Sơn	MSSV: 16141259
Chuyên ngành:	Điện tử công nghiệp	Mã ngành: 41
Hệ đào tạo:	Đại học chính quy	Mã hệ: 1
Khóa:	2016	Lớp: 16141DT

I. TÊN ĐỀ TÀI: **NGHIÊN CỨU VÀ THI CÔNG TỦ TRỒNG RAU NGẮN NGÀY TRONG NHÀ**

II. NHIỆM VỤ

1. Các số liệu ban đầu:

- Kit Arduino Nano, Node MCU và ngôn ngữ lập trình.
- Tài liệu về Arduino Nano, Node MCU, XAMPP.
- Thư viện về LCD, ESP 8266, XAMPP.

2. Nội dung thực hiện:

- Nội dung 1: Giao tiếp giữa Arduino với các cảm biến, hiển thị các giá trị cảm biến đo được và trạng thái các thiết bị chấp hành lên LCD 16x2.
- Nội dung 2: Thiết kế website dựa trên nền tảng Node.js, thực hiện quá trình truyền - nhận dữ liệu cho hệ thống thông qua module thu - phát WiFi Esp8266 Node MCU.
- Nội dung 3: Thiết kế ứng dụng chạy trên điện thoại có hệ điều hành Android. Gửi giá trị của các cảm biến và trạng thái các thiết bị chấp hành lên ứng dụng.
- Nội dung 4: Liên kết và đồng bộ hoá dữ liệu giữa ứng dụng và web.
- Nội dung 5: Thiết kế phân cài đặt loại cây trồng trên web.
- Nội dung 6: Thi công mạch và mô hình.
- Nội dung 7: Nhận xét - đánh giá kết quả thực hiện.
- Nội dung 8: Hoàn thành luận văn.

III. NGÀY GIAO NHIỆM VỤ: 08/03/2020

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 20/07/2020

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. Trương Ngọc Anh

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

Tp. Hồ Chí Minh, ngày 14 tháng 3 năm 2020

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP (Bản lịch trình này được đóng vào đồ án)

Họ tên sinh viên: Trần Tấn Sang

MSSV: 16141256

Họ tên sinh viên: Huỳnh Hồng Sơn

MSSV: 16141259

Tên đề tài: **Nghiên cứu và thi công tủ trồng rau ngắn ngày trong nhà .**

<i>Tuần/ngày</i>	<i>Nội dung chính cần thực hiện</i>	<i>Xác nhận GVHD</i>
Tuần 1 (8/3 – 14/3)	Gặp GV bộ môn để nghe phổ biến yêu cầu làm đồ án, nhận giấy giới thiệu làm đồ án. Gặp GVHD để chọn đề tài.	
Tuần 2 (15/3 – 21/3)	GVHD tiến hành xét duyệt đề tài và viết đề cương nộp lại.	
Tuần 3 (22/3 – 28/3)	- Tìm hiểu về các loại rau trồng ngắn ngày và các phương pháp nuôi trồng trong nhà. - Tìm hiểu về NodeMCU ESP8266, Arduino Nano.	
Tuần 4 (29/3 – 4/4)	- Tìm hiểu về cảm biến đo độ ẩm đất. - Tìm hiểu về cảm biến nhiệt độ, độ ẩm không khí DHT22. - Tìm hiểu động cơ bơm 12VDC, module Relay, màn hình LCD 16x2, module giảm áp LM2596, module I2C.	
Tuần 5 (5/4 – 11/4)	- Xây dựng sơ đồ khối, giải thích chức năng cho từng khối. - Tính toán lựa chọn linh kiện cho từng khối. - Thiết kế sơ đồ nguyên lí cho mạch.	
Tuần 6 (12/4 – 18/4)	- Thi công phần cứng, chạy thử nghiệm. - Viết chương trình Android điều khiển phần cứng.	

	- Kết nối ứng dụng Android với phần cứng và điều khiển xem đã hoạt động như yêu cầu chưa.	
Tuần 7 (19/4 – 25/4)	- Thi công phần cứng, chạy thử nghiệm. -Viết chương trình Android điều khiển phần cứng. - Kết nối ứng dụng Android với phần cứng và điều khiển xem đã hoạt động như yêu cầu chưa.	
Tuần 8 (26/4 – 2/5)	- Thi công phần cứng, chạy thử nghiệm. - Viết chương trình Android điều khiển phần cứng. - Kết nối ứng dụng Android với phần cứng và điều khiển xem đã hoạt động như yêu cầu chưa.	
Tuần 9 (3/5 – 9/5)	- Thi công phần cứng, chạy thử nghiệm. - Viết chương trình Android điều khiển phần cứng. - Kết nối ứng dụng Android với phần cứng và điều khiển xem đã hoạt động như yêu cầu chưa.	
Tuần 10 (10/5 – 16/5)	- Thiết kế Webserrver và kết nối Webserver với phần cứng.	
Tuần 11 (17/5 – 23/5)	- Thiết kế Webserver và kết nối Webserver với phần cứng.	
Tuần 12 (24/5 – 30/5)	- Thiết kế Webserver và kết nối Webserver với phần cứng.	
Tuần 13 (31/5 – 6/6)	- Thiết kế Webserver và kết nối Webserver với phần cứng.	
Tuần 14 (7/6 –13/6)	- Kiểm tra hoạt động của mô hình, web, ứng dụng Android.	

	- Demo cho GVHD và sửa sai nếu có.	
Tuần 15 (14/6 – 20/6)	- Viết báo cáo, làm slide.	
Tuần 16 (21/6 – 27/6)	Hoàn thiện quyền ĐATN và gửi cho GVHD để xem xét góp ý lần cuối trước khi in và báo cáo.	
Tuần 17 (28/6 – 4/7)	- Nộp quyền ĐATN vào ngày 04/08/2020 - Chuẩn bị cho ngày bảo vệ - In các giấy tờ, phiếu điểm để bảo vệ	
Tuần 18 (5/7– 11/7)	Bảo vệ ĐATN.	

GV HƯỚNG DẪN
(Ký và ghi rõ họ và tên)

LỜI CAM ĐOAN

Đề tài này là do chúng em tự thực hiện, tham khảo một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài

Trần Tấn Sang

Huỳnh Hồng Sơn

LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn sâu sắc đến Thầy Trương Ngọc Anh - Giảng viên bộ môn Điện tử công nghiệp đã trực tiếp hướng dẫn và tận tình giúp đỡ tạo điều kiện để hoàn thành tốt đề tài.

Đồng thời nhóm em xin gửi lời chân thành cảm ơn các Thầy Cô trong Khoa Điện-Điện Tử đã tạo những điều kiện tốt nhất cho nhóm hoàn thành đề tài.

Nhóm em cũng gửi lời cảm ơn đến các bạn lớp 16141DT đã chia sẻ trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian thực hiện đề tài.

Cảm ơn đến cha mẹ, gia đình đã luôn động viên bọn em.

Xin chân thành cảm ơn!

Người thực hiện đề tài

Trần Tấn Sang

Huỳnh Hồng Sơn

DANH MỤC HÌNH

Hình 2-1: Arduino Nano.	7
Hình 2-2: Module NodeMCU 8266	11
Hình 2-3: Sơ đồ chân và sơ đồ kết nối của ESP8266.	12
Hình 2-4: Giao tiếp UART.....	13
Hình 2-5: Giao tiếp I ² C	14
Hình 2-6: Giao tiếp One-Wire.....	15
Hình 2-7: Giao tiếp One-Wire.....	15
Hình 3-1: Sơ đồ khối hệ thống	19
Hình 3-2: Arduino Nano.	21
Hình 3-3: Cảm biến độ ẩm đất.	21
Hình 3-4: Cảm biến DHT22.....	22
Hình 3-5: Sơ đồ nguyên lý của Relay.	23
Hình 3-6: Hình ảnh thực tế module relay 5V.....	24
Hình 3-7: Màn hình LCD 16x2.....	25
Hình 3-8: Sơ đồ nguyên lý của LCD 16x2.	25
Hình 3-9: Sơ đồ nguyên lý của ESP 8266.....	27
Hình 3-10: Module NodeMCU 8266.....	28
Hình 3-11: Bơm mini 12VDC.....	28
Hình 3-12: Đèn led chiếu sáng cho cây trong nhà.	30
Hình 3-13: Quạt 12 VDC.	31
Hình 3-14: Sơ đồ nguyên lý của LM2596.	31
Hình 3-15: Module giảm áp LM2596.	32
Hình 3-16: Sơ đồ nguyên lý toàn mạch.	33
Hình 4-1: Mạch vẽ PCB.....	35
Hình 4-2: Mô hình 3D của mạch mặt trên.	36
Hình 4-3: Mô hình 3D của mạch mặt dưới.	37
Hình 4-4: Ảnh mặt bên hông của mô hình với bảng điều khiển.....	38
Hình 4-5: Mô hình tử nhìn từ trên xuống.....	38
Hình 4-6: Mô hình tử nhìn từ một bên hông, với cánh quạt thông gió.....	39
Hình 4-7: Mô hình tổng quát.....	39
Hình 4-8: Lưu đồ giải thuật Arduino Nano.....	40
Hình 4-9: Lưu đồ giải thuật NODE MCU ESP8266.	41
Hình 4-10: Lưu đồ giải thuật cho ứng dụng Android và web.....	42
Hình 4-11: Giao diện viết chương trình cho Arduino.....	43
Hình 4-12: Chọn Board Arduino sử dụng.....	43
Hình 4-13: Giao diện của phần mềm Visual Studio Code.....	44
Hình 4-14: Giao diện phần mềm XAMPP.	45

Hình 4-15: Giao diện trang chủ của Git.....	46
Hình 4-16: Giao diện khi khởi chạy Git Bash.....	47
Hình 4-17: Viết chương trình trên Visual Studio Code.....	48
Hình 4-18: Giao diện của XAMPP khi khởi chạy.	48
Hình 4-19: Khởi chạy server bằng XAMPP.	49
Hình 4-20: Giao diện của trang XAMPP.....	49
Hình 4-21: Giao diện quản lí database của phpMyAdmin.	50
Hình 4-22: Tạo cơ sở dữ liệu đầu tiên.....	50
Hình 4-23: Giao diện cơ sở dữ liệu khi mới khởi tạo xong.	51
Hình 4-24: Tạo các cơ sở dữ liệu con cho cơ sở dữ liệu vừa tạo.....	52
Hình 4-25: Giao diện quản lí dữ liệu cho “device”	52
Hình 4-26: Giao diện quản lí dữ liệu cho “plant”	53
Hình 4-27: Giao diện quản lí dữ liệu cho “sensor”.....	54
Hình 4-28: Giao diện quản lí dữ liệu cho “wateringchecktime”	55
Hình 4-29: Nhập thử thời gian tưới cho cây trên database.	56
Hình 4-30: Điều chỉnh giờ để kiểm tra hệ thống tưới.....	56
Hình 4-31: Giao diện sau khi đã nhập giờ để kiểm tra tưới.....	57
Hình 4-32: Viết chương trình ứng dụng Android trên Visual Studio Code.	57
Hình 4-33: Truy cập vào thư mục chứa dữ liệu chương trình.	58
Hình 4-34: Mở Git Bash Here.....	58
Hình 4-35: Cửa sổ Git Bash Here hiện ra.	59
Hình 4-36: Mã QR hiện ra để chạy thử ứng dụng.	59
Hình 4-37: Tải và cài đặt ứng dụng EXPO trên Google Play.....	60
Hình 4-38: Khởi chạy EXPO.	60
Hình 4-39: Giao diện tab “ Bảng điều khiển”.....	61
Hình 4-40: Giao diện tab “Trồng cây”.....	62
Hình 4-41: Khởi chạy Git Bash Here.....	63
Hình 4-42: Chạy dòng lệnh để tạo file apk.	63
Hình 4-43: Đường dẫn tải về xuất hiện.....	64
Hình 4-44: Ứng dụng đang được tạo trên server của EXPO.	64
Hình 4-45: Tập tin cài đặt đã được tạo thành công.....	65
Hình 4-46: Khởi chạy Git Bash.....	65
Hình 4-47: Giao diện của Git Bash.	66
Hình 4-48: Khởi chạy server.....	66
Hình 4-49: Thông báo đã khởi chạy thành công server.....	67
Hình 4-50: Giao diện của web điều khiển.....	68
Hình 4-51: Bắt đầu nhập thông tin cây trồng.....	68
Hình 4-52: Nhập các thông tin cơ bản cho cây trồng.	69
Hình 4-53: Giao diện chọn ngày trồng cây.	69
Hình 4-54: Kết quả sau khi đã nhập.....	70

Hình 5-1: Giao diện web điều khiển khi hoạt động	72
Hình 5-2: Giao diện ứng dụng điều khiển khi hoạt động.	73
Hình 5-3: Mô hình đang hoạt động với đèn được bật.....	75
Hình 5-4: Trạng thái các thiết bị và thông tin cảm biến được `hiển thị trên.	76

DANH MỤC BẢNG

Bảng 2-1: Đặc điểm kỹ thuật Arduino Nano.	7
Bảng 2-2: Thông tin các chân của Arduino Nano.....	9
Bảng 2-3: Một số chân đặc biệt của Arduino Nano.....	9
Bảng 3-1: Chức năng và thông số hoạt động của LCD 16x2.	27
Bảng 4-1: Danh sách linh kiện.	34
Bảng 5-1: Kết quả chạy trên ứng dụng điều khiển qua Wifi.	73
Bảng 5-2: Kết quả chạy trên web điều khiển qua Wifi.....	74
Bảng 5-3: Kết quả thử nghiệm trên mô hình.	77

TÓM TẮT

Nhằm đáp ứng nhu cầu trồng rau tại nhà đối với những hộ gia đình có không gian, diện tích hạn chế hoặc không có nhiều thời gian chăm sóc mảnh vườn của mình; nhóm em xin đưa ra đề tài: **NGHIÊN CỨU VÀ THI CÔNG TỬ TRỒNG RAU NGẪN NGÀY TRONG NHÀ.**

Hệ thống có các chức năng sau:

- Điều khiển hoạt động của mô hình thông qua ứng dụng Android cũng như web. Ngoài ra còn có thể điều khiển trực tiếp trên mô hình.

- Các thông số của cảm biến nhiệt độ và độ ẩm không khí, cảm biến độ ẩm đất được hiển thị trực quan trên màn hình LCD, đồng thời hiển thị trên ứng dụng điện thoại cũng như web để tạo điều kiện thuận lợi cho người dùng có thể giám sát, cũng như thay đổi thời gian tưới sao cho phù hợp.

- Dữ liệu của các cảm biến cũng như trạng thái của các thiết bị như máy bơm, đèn, quạt được đồng bộ một cách nhanh chóng, dễ dàng.

Với đề tài này, nhóm em hi vọng có thể làm tiền đề phát triển cho các mô hình trang trại thực tế với quy mô lớn, nâng cao và phát triển thêm nhiều tính năng hơn nữa. Đồng thời hi vọng đây sẽ là cơ sở nghiên cứu cho các nhóm khác có thể mở rộng, xây dựng thêm .

MỤC LỤC

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	i
LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP	ii
LỜI CAM ĐOAN	v
LỜI CẢM ƠN	vi
DANH MỤC HÌNH	vii
DANH MỤC BẢNG	x
TÓM TẮT	xi
MỤC LỤC	xii
CHƯƠNG 1: TỔNG QUAN	1
1.1 ĐẶT VẤN ĐỀ	1
1.2 MỤC TIÊU	1
1.3 NỘI DUNG NGHIÊN CỨU	1
1.4 GIỚI HẠN	2
1.5 BỐ CỤC.....	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 GIỚI THIỆU VỀ CÁC LOẠI RAU NGẮN NGÀY VÀ CÁC PHƯƠNG THỨC TRỒNG RAU TRONG NHÀ.....	3
2.1.1 Các loại rau ngắn ngày và lợi ích mang lại.....	3
2.1.2 Các phương thức trồng rau ngắn ngày tại nhà.....	3
2.2 QUY TRÌNH HOẠT ĐỘNG CỦA HỆ THỐNG.....	6
2.3 GIỚI THIỆU PHẦN CỨNG	6
2.3.1 Tổng quan Arduino Nano	6
2.3.2 Tổng quan Module ESP8266	10
2.4 TỔNG QUAN VỀ CÁC CHUẨN KẾT NỐI.....	12
2.4.1 Chuẩn giao tiếp UART.....	12
2.4.2 Chuẩn giao tiếp I2C	13
2.4.3 Chuẩn giao tiếp ONE-WIRE.....	14
2.5 SƠ LƯỢC VỀ MẠNG LOCALHOST.....	16
2.6 GIAO THỨC SOCKETIO	16

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ	18
3.1 GIỚI THIỆU	18
3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	18
3.2.1 Thiết kế sơ đồ khối hệ thống.....	18
3.2.2 Tính toán và thiết kế mạch.....	20
3.2.3 Sơ đồ nguyên lí toàn mạch.....	33
CHƯƠNG 4: THI CÔNG HỆ THỐNG	34
4.1 GIỚI THIỆU	34
4.2 THI CÔNG MẠCH ĐIỀU KHIỂN	34
4.2.1 Thi công mạch điều khiển.....	34
4.3 THI CÔNG MÔ HÌNH.....	37
4.4 LƯU ĐỒ GIẢI THUẬT	39
4.4.1 Lưu đồ giải thuật Nano	39
4.4.2 Lưu đồ giải thuật Node MCU	40
4.4.3 Lưu đồ giải thuật cho ứng dụng, web	42
4.5 XÂY DỰNG WEB VÀ ỨNG DỤNG ANDROID.....	42
4.5.1 Phần mềm lập trình cho vi điều khiển	42
4.5.2 Visuall Studio Code	44
4.5.3 XAMPP	44
4.5.4 Git Bash.....	46
4.5.5 NodeJS	47
4.5.6 Các bước thực thi	47
4.6 QUY TRÌNH TRỒNG RAU MẦM CƠ BẢN	70
4.6.1 Dụng cụ trồng rau mầm.....	70
4.6.2 Nguyên liệu	70
4.6.3 Cách trồng	71
4.6.4 Thu hoạch.....	71
4.6.5 Chế biến.....	71
CHƯƠNG 5: KẾT QUẢ - ĐÁNH GIÁ – NHẬN XÉT	72
5.1 KẾT QUẢ ĐẠT ĐƯỢC	72
5.2 KẾT QUẢ THỬ NGHIỆM HỆ THỐNG.....	72
5.2.1 Quá trình chạy ứng dụng Android và trên Web.....	72

5.3 NHẬN XÉT- ĐÁNH GIÁ.....	77
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	78
6.1 KẾT LUẬN.....	78
6.2 HƯỚNG PHÁT TRIỂN.....	78
TÀI LIỆU THAM KHẢO.....	79
Sách tham khảo.....	79

CHƯƠNG 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Người Việt Nam có thói quen trong bữa cơm hàng ngày của gia đình không thể thiếu các món liên quan đến rau củ. Với sự phát triển của khoa học kỹ thuật trong nông nghiệp, những loại thuốc kích thích tăng trưởng cho cây trồng ra đời cho năng suất thu hoạch cao và thời gian thu hoạch ngắn lại. Tuy nhiên người tiêu dùng không thể biết được liệu mình có mua phải bó rau đã được sử dụng thuốc kích thích / bảo vệ thực vật quá hàm lượng cho phép hay không. Vì vậy trào lưu trồng rau tại nhà đang rất phổ biến, để các hộ gia đình có thể đảm bảo được nguồn gốc an toàn thực phẩm.

Hiện nay, việc áp dụng các thành tựu của khoa học kỹ thuật, đặc biệt là kỹ thuật điện tử - lập trình tự động đã giúp cho cuộc sống của con người trở nên dễ dàng hơn bao giờ hết. Ý tưởng về một hệ thống trồng rau tại nhà nhỏ gọn, tiết kiệm không gian, được thiết lập tự động từ đó đã ra đời. Bên cạnh đó, với việc được thiết kế nhỏ gọn và sử dụng trong phạm vi gia đình, đáp ứng bữa cơm hàng ngày và có thể thu hoạch được sản phẩm trong thời gian ngắn; nhóm em quyết định chọn đề tài: **“Nghiên cứu và thi công tủ trồng rau ngắn ngày trong nhà”** làm đồ án tốt nghiệp dưới sự hướng dẫn của Thầy Trương Ngọc Anh.

1.2 MỤC TIÊU

Nghiên cứu và thi công tủ trồng rau mầm trong nhà với vi điều khiển ESP 8266 và xây dựng hệ thống quản lí, điều khiển trên Webserver. Đồng thời viết ứng dụng điều khiển trên Android.

1.3 NỘI DUNG NGHIÊN CỨU

- NỘI DUNG 1: Giao tiếp giữa Arduino với các cảm biến, hiển thị giá trị cảm biến và trạng thái các thiết bị ngoại vi lên LCD 16x2.
- NỘI DUNG 2: Thiết kế web dựa trên nền tảng Node.js, để thực hiện quá trình truyền - nhận dữ liệu cho hệ thống thông qua module thu - phát WiFi Esp8266 Node MCU.
- NỘI DUNG 3: Thiết kế ứng dụng chạy trên điện thoại có hệ điều hành Android. Gửi giá trị của cảm biến và trạng thái các thiết bị lên ứng dụng.
- NỘI DUNG 4: Liên kết và đồng bộ hoá dữ liệu giữa ứng dụng và web.

CHƯƠNG 1: TỔNG QUAN

- NỘI DUNG 5: Thiết kế phần cài đặt loại cây trồng trên web.
- NỘI DUNG 6: Thi công mạch và mô hình.
- NỘI DUNG 7: Nhận xét - đánh giá kết quả thực hiện.
- NỘI DUNG 8: Hoàn thành luận văn.

1.4 GIỚI HẠN

Các thông số giới hạn của đề tài bao gồm:

- Kích thước của tủ trồng rau trong nhà: Dài 60 cm, rộng 40 cm, cao 40 cm.
- Ứng dụng điện thoại chạy trên hệ điều hành Android.
- Web và ứng dụng Android kết nối với mạng localhost.
- Một Arduino Nano đóng vai trò làm bộ điều khiển trung tâm.
- Dữ liệu nhiệt độ, độ ẩm, trạng thái tắt mở của các thiết bị được hiện trên LCD 16x2.
- Sử dụng động cơ 12V để bơm nước lên cung cấp nước cho cây trồng.
- Trồng rau mầm củ cải trắng trong tủ.

1.5 BỐ CỤC

- **Chương 1: Tổng Quan**
Trình bày, đặt lí do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án.
- **Chương 2: Cơ Sở Lý Thuyết**
Trình bày các lý thuyết liên quan đến vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài.
- **Chương 3: Thiết Kế và Tính Toán**
Trình bày tổng quan các yêu cầu của đề tài về thiết kế và các tính toán hệ thống bao gồm sơ đồ nguyên lý toàn mạch và của từng phần của hệ thống.
- **Chương 4: Kết Quả, Nhận Xét và Đánh Giá**
Trình bày kết quả thi công phần cứng và kết quả hình ảnh trên màn hình hay mô phỏng tín hiệu, kết quả thống kê.
- **Chương 5: Kết Luận và Hướng Phát Triển**
Trình bày kết quả của cả quá trình nghiên cứu làm đề tài bao gồm thời gian nghiên cứu, kết quả đạt được, nhận xét, đánh giá về đề tài và tính ứng dụng của đề tài trong thực tiễn.
- **Chương 6: Kết luận và hướng phát triển**
Trình bày kết quả đạt được so với mục tiêu đề ra ban đầu, nhận xét và đánh giá kết quả đạt được của đề tài nghiên cứu. Hướng phát triển của đề tài sau này trong quá trình nghiên cứu.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 GIỚI THIỆU VỀ CÁC LOẠI RAU NGẮN NGÀY VÀ CÁC PHƯƠNG THỨC TRỒNG RAU TRONG NHÀ

2.1.1 Các loại rau ngắn ngày và lợi ích mang lại

Rau ngắn ngày là những loại rau có thời gian thu hoạch ngắn, dễ chăm sóc và không tốn nhiều công sức. Đây là những loại rau quen thuộc trong bữa ăn hằng ngày của nhiều gia đình và cực kì bổ dưỡng.

Một số loại rau ngắn ngày phổ biến có thể kể đến như rau muống (thời gian thu hoạch là 3 tuần), rau dền (3-4 tuần), rau cải xanh (khoảng 10 ngày)... Đặc biệt là rau mầm, thời gian thu hoạch còn nhanh hơn rất nhiều so với các loại rau vừa kể trên, mà hàm lượng dinh dưỡng cũng rất cao.

Rau mầm là một loại rau ngắn ngày. Đây là loại thực phẩm dễ chế biến và dễ sử dụng, là món ăn khoái khẩu của nhiều người. Rau mầm được đánh giá là có giá trị dinh dưỡng cao gấp 5 lần so với các loại rau bình thường và là nguồn chất xơ tự nhiên, giúp cho hệ thống tiêu hóa khỏe mạnh. Không những thế rau mầm còn chứa nhiều loại vitamin, axit amin và các chất dinh dưỡng cần thiết cho cơ thể với hàm lượng cao, cứ 50 gram rau mầm tương đương với lượng dinh dưỡng trong 200 gram rau bình thường. Rau mầm được sử dụng trong việc chế biến thành các món ăn đa dạng như xào, lẩu, súp, các món cuộn, trộn salad hay ăn kèm cùng các loại bánh, thịt, hải sản. Hơn nữa thời gian sinh trưởng của rau mầm cũng rất ngắn. Rau mầm rất đa dạng với các loại rau mầm họ cải. Rau mầm họ cải rất đa dạng, bao gồm nhiều loại: rau mầm củ cải trắng, rau mầm cải xanh, rau mầm cải ngọt, rau mầm cải thìa, rau mầm súp lơ, rau mầm cải xoong....

2.1.2 Các phương thức trồng rau ngắn ngày tại nhà

Hiện nay, thực phẩm nhiễm hóa chất ngày càng trở thành vấn đề báo động và đáng lo ngại. Không chỉ có thịt cá được sử dụng những chất bảo quản, chất hóa học làm cho thịt tươi lâu hơn mà ngay cả rau củ quả cũng vậy. Bên cạnh đó, trong quá trình sản xuất, các loại rau quả còn được bơm nhiều loại thuốc trừ sâu, phân hóa học và chất kích thích tăng trưởng làm cho những thực phẩm này trở nên đáng sợ và nguy hiểm với người sử dụng. Nhằm đảm bảo an toàn vệ sinh thực phẩm cho các thành viên trong gia đình thì hầu hết mọi người đều chọn cho mình phương pháp tự trồng rau sạch tại nhà.

Dưới đây sẽ là các bước trong quy trình trồng rau tại nhà chuẩn nhất hiện nay.

Bước 1: Thiết kế lắp đặt vườn rau sạch tại nhà bởi các giá đỡ, ống nước, hệ thống tưới nhỏ giọt

Khay trồng rau: bạn có thể sử dụng đồ tái chế như vỏ chai, thùng xốp, bao tải, túi vải, ống nhựa. Bạn hãy nhớ đục lỗ bên dưới để tránh ngập úng cho cây để có thể tiết

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

kiệm chi phí cho gia đình mình. Theo kinh nghiệm trồng rau sạch đây là cách làm vườn rau tại nhà nhanh gọn mà đơn giản nhất.

Thiết kế lắp đặt vườn rau sạch tại nhà: Giá đỡ, đựng, treo.... Bạn có thể để các khay trồng rau bằng các khung sắt thép được ghép nối với nhau. Những hàng rào cũng là một hình thức lựa chọn để đỡ các khay, ống, treo lên rất tiện ích. Bạn có thể để trên sân thượng.

Bước 2: Chọn đất trồng rau sạch tại nhà

Đất trồng rau sạch tại nhà cũng là một nguyên liệu rất quan trọng. Đất phải sạch, giàu dinh dưỡng. Tùy thuộc vào cách bạn chọn khay trồng thì lượng đất sẽ được theo đó mà cho vào sao cho phù hợp.

Đôi khi người trồng rau chỉ mua hạt giống và tận dụng đất sẵn có tại nhà để trồng rau. Thật ra đất đã sử dụng lâu ngày thường bị chai cứng, ít dưỡng chất cần thiết cho rau. Mặt khác rau là loài cây có bộ rễ ăn cạn trên lớp mặt từ 5-12 cm. Nếu đất không giữ ẩm tốt thì rau rất khó phát triển, cây rau cũng sẽ lên nhưng hay bị còi cọc, lá nhỏ dần.

Trường hợp trồng rau không dùng phân vô cơ như Urê, Lân, NPK thì rau cũng chậm lớn, lá có hiện tượng nhạt màu do thiếu dinh dưỡng. Nên bổ sung thêm đất dinh dưỡng hay phân hữu cơ.

Bước 3: Chọn hạt giống trồng rau sạch tại nhà

Hạt giống là nguyên liệu có thể cho là quan trọng nhất trong quá trình làm vườn rau tại nhà. Mỗi mùa có những loại cây trồng phù hợp với thời tiết. Giống cây có thể mua ở nhiều nơi, các trung tâm cây trồng hay các hàng tạp hóa phân bón cũng có bán. Giống cây thường mua tại các trại cây giống, đối với loại gieo hạt có thể mua tại các cửa hàng bán phân bón. Bạn có thể làm vườn rau tại nhà với nhiều loại rau của đa dạng.

Bước 4: Cần phải ngâm ủ hạt giống trước khi gieo vào chậu đất

Để giúp hạt giống rau đủ độ ẩm nảy mầm thì rất cần công đoạn ngâm hạt trong thời gian 6-10 giờ (thời gian còn phụ thuộc vào loại cây bạn muốn gieo trồng). Sau đó đem ủ lại trong lớp khăn ướt trong thời gian 1-2 ngày. Khi thấy hạt vừa nứt vỏ thì mới bắt đầu trồng vào chậu đất.

Bước 5: Tưới nước cho rau:

- Nước là một trong những yếu tố quan trọng không thể thiếu khi trồng rau sạch. Tuy nhiên, để có được những sản phẩm rau sạch và an toàn, các bạn cần phải sử dụng những loại nước có nguồn gốc tự nhiên như: nước mưa, nước giếng khoan hay nước ao hồ.

- Phân bón và thuốc bảo vệ thực vật cần dùng nước sạch để pha.

- Tuyệt đối không được dùng nước thải công nghiệp, nước thải bệnh viện hay nước từ các kênh mương bị ô nhiễm để tưới rau.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Bước 6: Bón phân hỗ trợ phát triển cho rau

- Nếu người trồng rau bán trên thị trường thì họ lạm dụng phân hóa học để rau mau lớn. Còn khi trồng rau tại nhà chúng ta đã khống chế liều lượng dưới ngưỡng cho phép. Do mình chủ động thì khi thu hoạch rau tại nhà vẫn đảm bảo sạch an toàn.

- Phân hóa học như NPK có giá thành khá rẻ dễ tìm, lại dễ sử dụng. Tốt nhất nên pha loãng vào nước sạch tưới cho rau là yên tâm.

- Việc bón phân cho rau phải được ngưng trước khi thu hoạch từ 15 - 20 ngày.

- Không dùng phân tươi để hòa nước tưới.

Bước 7: Phòng trừ sâu bệnh

Khi trồng rau, có thể sẽ xuất hiện những loại sâu bệnh gây hại cho cây trồng. Tuy nhiên, có thể sử dụng nhiều cách khác nhau để có thể hạn chế và tiêu diệt sâu bệnh hại. Không nên sử dụng các loại thuốc trừ sâu quá liều và bừa bãi để đảm bảo an toàn khi sử dụng.

- Áp dụng luân canh cây trồng một cách hợp lý.

- Sử dụng giống tốt để gieo trồng, không sâu bệnh.

- Lựa chọn cách chăm sóc cho phù hợp với từng loại cây.

- Dọn vệ sinh vườn sạch sẽ.

- Dùng chế phẩm sinh học để trừ sâu bệnh.

- Khi thật cần thiết mới sử dụng thuốc bảo vệ thực vật nhưng vẫn phải tuân thủ những nguyên tắc sau:

+ Không sử dụng thuốc cấm cho rau.

+ Chọn thuốc ít hoạt chất độc hại, hạn chế ảnh hưởng đến thiên địch và con người.

+ Nên sử dụng loại thuốc sinh học.

Bước 8: Thu hoạch và sử dụng:

Khi rau sắp thu hoạch, phải ngưng việc phun thuốc trừ sâu hay bón phân cho rau vì có thể lượng chất hóa học tác động lên rau sẽ không được phân giải hết và tích tụ trong rau gây hại đến sức khỏe con người.

Thu hoạch theo đúng vụ mùa, đúng yêu cầu của mỗi loại rau. Loại bỏ cây còi cọc và lá già héo.

Dùng bao túi sạch để đựng sau khi đã rửa rau bằng nước sạch. Khi sử dụng, cũng cần phải rửa kỹ và chế biến đúng cách để đảm bảo an toàn.

Rau sau khi bỏ vào túi nên bảo quản ở nhiệt độ mát của tủ lạnh. Nên sử dụng rau không quá 3 ngày sau khi thu hoạch. Bạn không cần ngâm nước muối hay chất làm sạch rau khi sử dụng loại rau này.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trồng rau sạch tại nhà hiện đang là xu hướng của người dân đô thị để có được rau sạch ăn hàng ngày. Với diện tích hạn chế thì một số cách sau được người dân ưa chuộng để trồng tại nhà:

- Trồng trong các chậu nhỏ và để ở ban công, sân thượng.
- Trồng trong các hộp xốp.
- Làm 1 giàn để cho các loại cây thân leo phát triển như mướp, bầu, bí....
-

Nhưng các phương pháp trên đều vẫn yêu cầu một khoảng không gian khá nhiều, và thời gian chăm sóc thủ công cho cây trồng. Vì vậy ở đề tài này, nhóm xin đưa ra một phương pháp mới. Đó là hệ thống tủ trồng rau với thiết kế nhỏ gọn và người sử dụng hoàn toàn có thể thiết lập chế độ tự động chăm sóc cho cây trồng.

2.2 QUY TRÌNH HOẠT ĐỘNG CỦA HỆ THỐNG

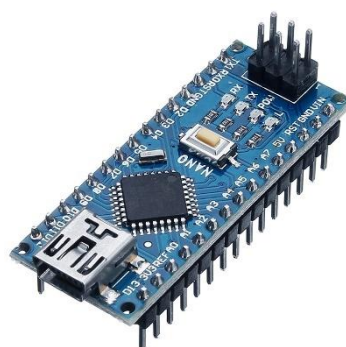
Mạch được điều khiển bởi module Arduino Nano đóng vai trò điều khiển trung tâm. Arduino điều khiển nhận giao tiếp với các module trong đề tài như cảm biến nhiệt độ - độ ẩm không khí, LCD 16x2, động cơ bơm, cảm biến độ ẩm đất. Người dùng thao tác điều khiển, thiết lập các chế độ tự động, theo dõi và điều khiển trực tiếp trên mô hình, hoặc qua web cũng như trên ứng dụng Android.

2.3 GIỚI THIỆU PHẦN CỨNG

2.3.1 Tổng quan Arduino Nano

a. Giới thiệu Arduino Nano:

- Arduino Nano là Kit phát triển dựa trên nền tảng của vi điều khiển ATmega328 thuộc dòng vi điều khiển AVR 8 bit. Có thể sử dụng trình biên dịch của Arduino để lập trình và nạp code.
- Trên board mạch có tích hợp đèn báo TX, RX của board và đèn kết nối pin D13, nút reset Kit.
- Cổng nạp mini USB dễ dàng, thuận tiện cho việc cung cấp nguồn và nạp code cho chip cũng như giao tiếp UART với máy tính. Ngoài ra ta có thể sử dụng cách nạp khác từ các chân giao tiếp khác của kit.



Hình 2-1: Arduino Nano.

b. Thông số kỹ thuật

Vi điều khiển chính: Atmega328 (8 bit).

- Tần số hoạt động: 16 MHz.
- Nguồn cung cấp: 5-12 VDC (cấp bằng cổng Mini USB hoặc chân Vin của chip).
- Mức điện áp giao tiếp của chân GPIO: 5 VDC.

Arduino Nano	Thông số kỹ thuật
Số chân analog I/O	8
Cấu trúc	AVR
Tốc độ xung	16 MHz
Dòng tiêu thụ I/O	40mA
Số chân Digital I/O	22
Bộ nhớ EEPROM	1 KB
Điện áp ngõ vào	(7-12) Volts
Vi điều khiển	ATmega328P
Điện áp hoạt động	5V
Kích thước bo mạch	18 x 45 mm
Nguồn tiêu thụ	19mA
Ngõ ra PWM	6
SRAM	2KB
Cân nặng	7 gms
Bộ nhớ Flash	32 KB of which 2 KB used by Bootloader

Bảng 2-1: Đặc điểm kỹ thuật Arduino Nano.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thứ tự chân	Tên chân	Kiểu	Chức năng
1	D1 / TX	I / O	Ngõ vào/ra số Chân TX-truyền dữ liệu
2	D0 / RX	I / O	Ngõ vào/ra số Chân Rx-nhận dữ liệu
3	RESET	Đầu vào	Chân reset, hoạt động ở mức thấp
4	GND	Nguồn	Chân nối mass
5	D2	I / O	Ngõ vào/ra digital
6	D3	I / O	Ngõ vào/ra digital
7	D4	I / O	Ngõ vào/ra digital
8	D5	I / O	Ngõ vào/ra digital
9	D6	I / O	Ngõ vào/ra digital
10	D7	I / O	Ngõ vào/ra digital
11	D8	I / O	Ngõ vào/ra digital
12	D9	I / O	Ngõ vào/ra digital
13	D10	I / O	Ngõ vào/ra digital
14	D11	I / O	Ngõ vào/ra digital
15	D12	I / O	Ngõ vào/ra digital
16	D13	I / O	Ngõ vào/ra digital
17	3V3	Đầu ra	Đầu ra 3.3V (từ FTDI)
18	AREF	Đầu vào	Tham chiếu ADC
19	A0	Đầu vào	Kênh đầu vào tương tự kênh 0
20	A1	Đầu vào	Kênh đầu vào tương tự kênh 1
21	A2	Đầu vào	Kênh đầu vào tương tự kênh 2
22	A3	Đầu vào	Kênh đầu vào tương tự kênh 3
23	A4	Đầu vào	Kênh đầu vào tương tự kênh 4
24	A5	Đầu vào	Kênh đầu vào tương tự kênh 5
25	A6	Đầu vào	Kênh đầu vào tương tự kênh 6
26	A7	Đầu vào	Kênh đầu vào tương tự kênh 7
27	+ 5V	Đầu ra hoặc đầu vào	+ Đầu ra 5V (từ bộ điều chỉnh On-board) hoặc

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

			+ 5V (đầu vào từ nguồn điện bên ngoài)
28	RESET	Đầu vào	Chân đặt lại, hoạt động ở mức thấp
29	GND	Nguồn	Chân nối mass
30	VIN	Nguồn	Chân nối với nguồn vào

Bảng 2-2: Thông tin các chân của Arduino Nano.

Tên pin Arduino Nano ICSP	Kiểu	Chức năng
MISO	Đầu vào hoặc đầu ra	Master In Slave Out
Vcc	Đầu ra	Cấp nguồn
SCK	Đầu ra	Tạo xung cho
MOSI	Đầu ra hoặc đầu vào	Master Out Slave In
RST	Đầu vào	Đặt lại, Hoạt động ở mức thấp
GND	Nguồn	Chân nối đất

Bảng 2-3: Một số chân đặc biệt của Arduino Nano.

Các chân: 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 và 16.

Như đã đề cập trước đó, Arduino Nano có 14 ngõ vào/ra digital. Các chân làm việc với điện áp tối đa là 5 V. Mỗi chân có thể cung cấp hoặc nhận dòng điện 40 mA và có điện trở kéo lên khoảng 20-50 k Ω . Các chân có thể được sử dụng làm đầu vào hoặc đầu ra, sử dụng các hàm pinMode (), digitalWrite () và digitalRead ().

Ngoài các chức năng đầu vào và đầu ra số, các chân này cũng có một số chức năng bổ sung.

- Chân 1, 2: Chân nối tiếp.

Hai chân nhận RX và truyền TX này được sử dụng để truyền dữ liệu nối tiếp TTL. Các chân RX và TX được kết nối với các chân tương ứng của chip nối tiếp USB tới TTL.

- Chân 6, 8, 9, 12, 13 và 14: Chân PWM.

Mỗi chân số này cung cấp tín hiệu điều chế độ rộng xung 8 bit. Tín hiệu PWM có thể được tạo ra bằng cách sử dụng hàm analogWrite ().

- Chân 5, 6: Ngắt.

Khi chúng ta cần cung cấp một ngắt ngoài cho bộ xử lý hoặc bộ điều khiển khác, chúng ta có thể sử dụng các chân này. Các chân này có thể được sử dụng để cho phép ngắt INT0 và INT1 tương ứng bằng cách sử dụng hàm attachInterrupt (). Các chân có thể được sử dụng để kích hoạt ba loại ngắt như ngắt trên giá trị thấp, tăng hoặc giảm mức ngắt và thay đổi giá trị ngắt.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Chân 13, 14, 15 và 16: Giao tiếp SPI.

Khi không muốn dữ liệu được truyền đi không đồng bộ, có thể sử dụng các chân ngoại vi nối tiếp này. Các chân này hỗ trợ giao tiếp đồng bộ với SCK. Mặc dù phần cứng có tính năng này nhưng phần mềm Arduino lại không có. Vì vậy, phải sử dụng thư viện SPI để sử dụng tính năng này.

- Chân 16: Led

Khi sử dụng chân 16, đèn led trên bo mạch sẽ sáng.

- Chân 18, 19, 20, 21, 22, 23, 24, 25 và 26: Ngõ vào/ra tương tự.

Như đã đề cập trước đó UNO có 6 chân đầu vào tương tự nhưng Arduino Nano có 8 đầu vào tương tự (19 đến 26), được đánh dấu A0 đến A7. Điều này có nghĩa là có thể kết nối 8 kênh đầu vào tương tự để xử lý. Mỗi chân tương tự này có một ADC có độ phân giải 1024 bit (do đó nó sẽ cho giá trị 1024). Theo mặc định, các chân được đo từ mặt đất đến 5V. Nếu muốn điện áp tham chiếu là 0V đến 3.3V, có thể nối với nguồn 3.3V cho chân AREF (pin thứ 18) bằng cách sử dụng chức năng analogReference (). Tương tự như các chân digital trong Nano, các chân analog cũng có một số chức năng khác.

- Chân 23, 24 như A4 và A5: chuẩn giao tiếp I²C.

Khi giao tiếp SPI cũng có những nhược điểm của nó như cần 4 chân và giới hạn trong một thiết bị. Đối với truyền thông đường dài, cần sử dụng giao thức I²C. I²C hỗ trợ chỉ với hai dây. Một cho xung (SCL) và một cho dữ liệu (SDA). Để sử dụng tính năng I²C này, cần phải nhập một thư viện có tên là Wire.

- Chân 18: AREF.

Điện áp tham chiếu cho đầu vào dùng cho việc chuyển đổi ADC.

- Chân 28: RESET.

Đây là chân reset mạch khi nhấn nút trên bo. Thường được sử dụng để được kết nối với thiết bị chuyển mạch để sử dụng làm nút reset.

2.3.2 Tổng quan Module ESP8266

a. Giới thiệu Module ESP8266

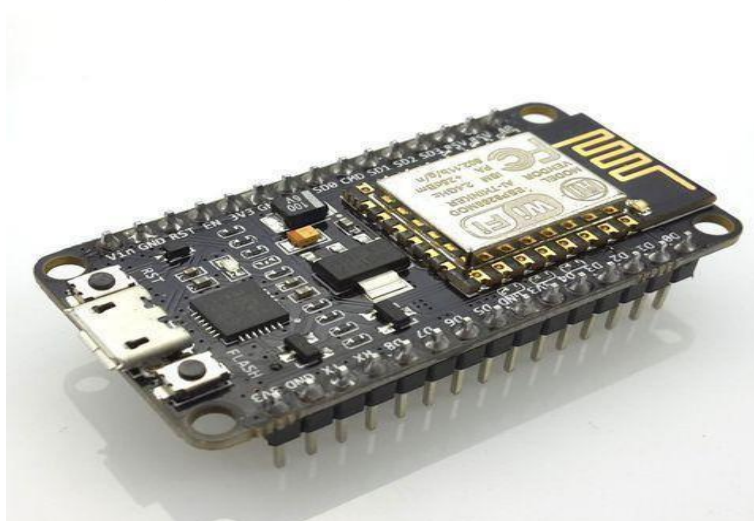
Khởi gửi-nhận dữ liệu dùng để gửi giá trị lên Database để lưu trữ và nhận giá trị từ Database. Ngày nay có nhiều module có thể thực hiện việc này như Esp8266 v1, Esp8266 v12, Esp32, Esp8266 NodeMCU.... Esp8266v1, Esp8266 v12, Esp8266 Node MCU thì tương tự nhau về mặt lập trình. Điểm khác giữa Esp8266 v12, Esp8266 Node MCU và Esp8266 v1 là Esp8266 v12, Esp8266 Node MCU có thêm nhiều chân GPIO để mở rộng việc điều khiển.

b. Các thông số kỹ thuật

- Wifi 802.11 b/g/n.
- Wi-Fi Direct (P2P), soft-AP.
- Tích hợp giao thức TCP / IP stack.

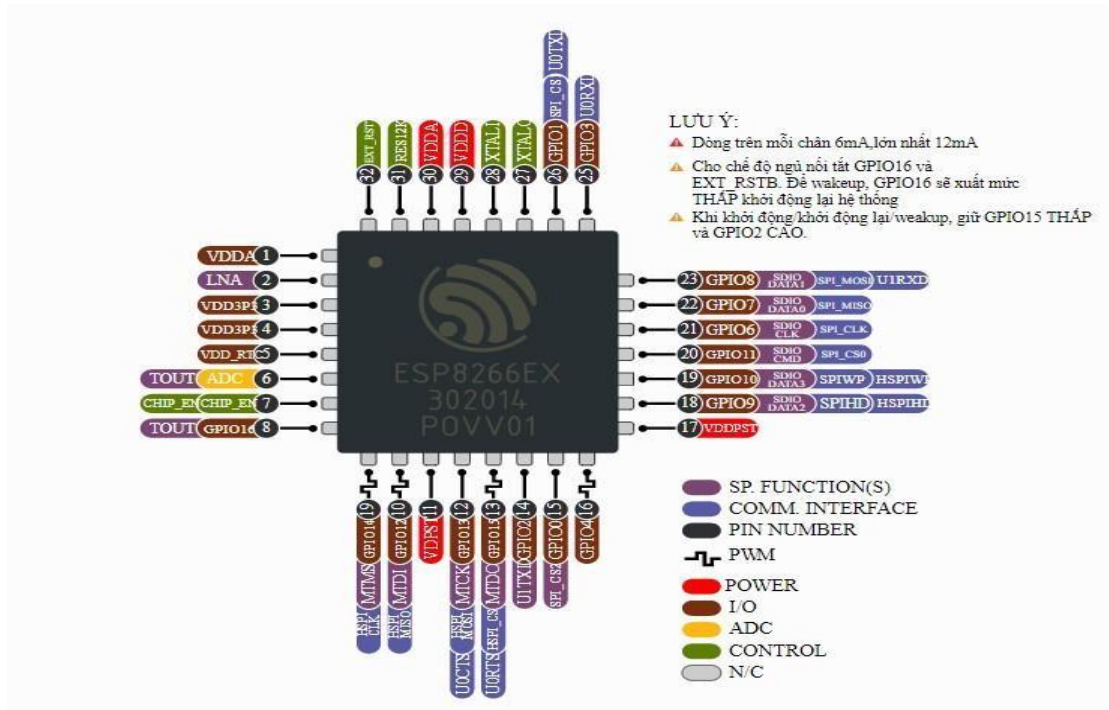
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Tích hợp TR chuyển đổi, balun, LNA, bộ khuếch đại và phù hợp với mạng.
- PLLs tích hợp, quản lý, DCXO và các đơn vị quản lý điện năng.
- Công suất đầu ra 19.5 dBm ở chế độ 802.11b.
- Tích hợp công suất thấp 32-bit CPU có thể được sử dụng như là bộ vi xử lý ứng dụng.
- SDIO 1.1 / 2.0, SPI, UART.
- Điện áp sử dụng: 3.3-3.6V.
- Dòng làm việc: 80 mA (170 mA max).



Hình 2-2: Module NodeMCU 8266.

c. Sơ đồ chân và sơ đồ kết nối



Hình 2-3: Sơ đồ chân và sơ đồ kết nối của ESP8266.

Thông số phần cứng:

- 32-bit RISC CPU: Tensilica Xtensa LX106 running at 80 MHz.
- Hỗ trợ Flash ngoài từ 512 KiB đến 4 MiB.
- 64 KBytes RAM thực thi lệnh.
- 96 KBytes RAM dữ liệu.
- 64 KBytes boot ROM.
- Chuẩn wifi IEEE 802.11 b/g/n, Wi-Fi 2.4 GHz. Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network. Hỗ trợ WEB, WPA/WPA2, open network.
- Tích hợp giao thức TCP/IP.
- Hỗ trợ nhiều loại anten.
- 16 chân GPIO-ADC 10-bit.
- Hỗ trợ SDIO 2.0, UART, SPI, I²C, PWM, I²S với DMA.

2.4 TỔNG QUAN VỀ CÁC CHUẨN KẾT NỐI

2.4.1 Chuẩn giao tiếp UART

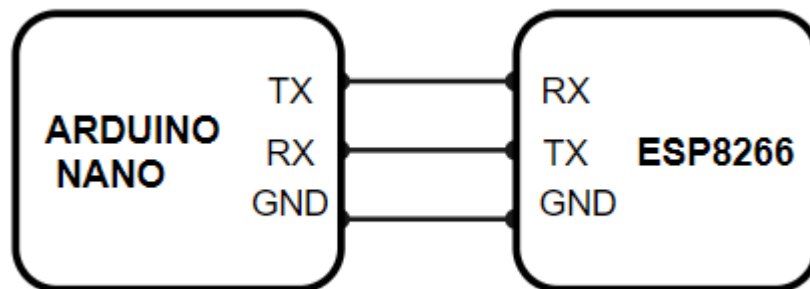
UART là viết tắt của Universal Asynchronous Receiver – Transmitter có nghĩa là truyền dữ liệu nối tiếp bất đồng bộ. UART chuyển đổi giữa dữ liệu kết nối tiếp và song song. Một chiều, UART chuyển đổi dữ liệu song song bus hệ thống ra dữ liệu nối tiếp để truyền đi. Một chiều khác, UART chuyển dữ liệu nhận được dạng dữ liệu nối

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

tiếp thành dạng dữ liệu song song cho CPU có thể đọc và bus hệ thống. Để truyền được dữ liệu thì cả hai bên phát và bên nhận phải tự tạo xung clock có cùng tần số và thường được gọi là tốc độ baud, ví dụ như 2400 baud, 4800 baud, 9600 baud....

UART của PC hỗ trợ cả hai kiểu giao tiếp là đồng thời và giao tiếp không đồng thời. Giao tiếp đồng thời tức là UART có thể gửi và nhận dữ liệu vào cùng một thời điểm. Còn giao tiếp không đồng thời là chỉ có một thiết bị có thể chuyển dữ liệu vào một thời điểm, với tín hiệu điều khiển hoặc mã sẽ quyết định bên nào có thể truyền dữ liệu. Giao tiếp không đồng thời được thực hiện khi mà cả 2 chiều chia sẻ một đường dẫn hoặc nếu có cả hai đường nhưng cả hai thiết bị chỉ giao tiếp qua một đường ở cùng một thời điểm.

Thêm vào đường dữ liệu, UART hỗ trợ bắt tay chuẩn RS232 và tín hiệu điều khiển như RST, CTS, DTR, DCR, RT và CD.

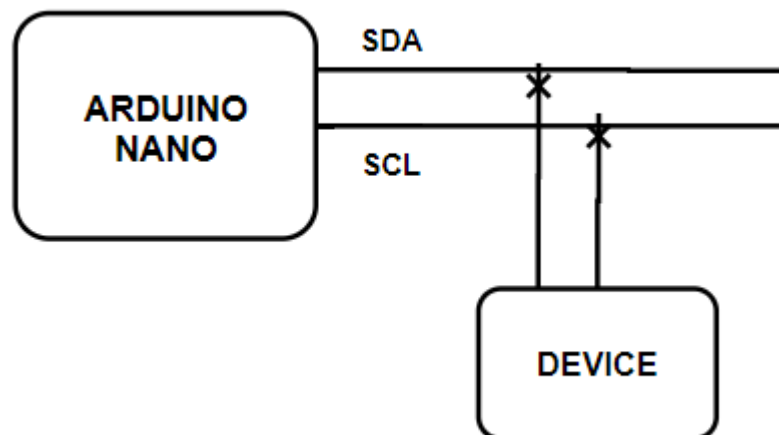


Hình 2-4: Giao tiếp UART

Để giao tiếp giữa hai thiết bị thông qua chuẩn giao tiếp UART, ta tiến hành nối dây Tx (chân gửi tín hiệu) của bên phát với chân Rx (chân nhận tín hiệu) của bên thu và ngược lại nối chân Rx (chân nhận tín hiệu) của bên phát với chân Tx (chân gửi tín hiệu) của bên thu. Cách nối dây này được gọi là nối chéo dây. Bên cạnh đó, cần phải nối chung GND cho cả 2 bên nhận và phát với nhau và muốn truyền nhận được, 2 bên phải có cùng tốc độ baud.

2.4.2 Chuẩn giao tiếp I2C

I2C là một chuẩn truyền thông theo mô hình chủ-tớ. Một thiết bị chủ có thể giao tiếp với nhiều thiết bị tớ. Muốn giao tiếp với các thiết bị nào, thiết bị chủ phải gửi đúng địa chỉ để kích hoạt thiết bị đó rồi mới được phép ghi hoặc đọc dữ liệu.



Hình 2-5: Giao tiếp I²C

Một giao tiếp I2C gồm có 2 dây: Serial Data(SDA) và Serial Clock (SCL).

SDA thường là đường truyền dữ liệu hai hướng, còn SCL là đường truyền xung đồng hồ và chỉ theo một hướng. Như hình vẽ trên, khi một thiết bị ngoại vi kết nối vào đường I2C thì chân SDA của nó sẽ nối với chân SDA của bus, chân SCL sẽ nối với dây SCL.

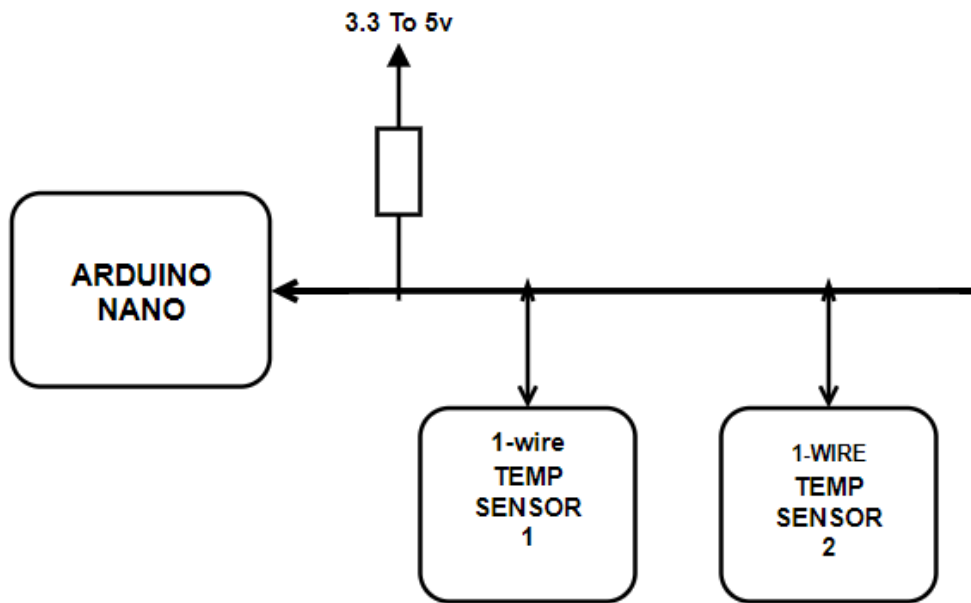
Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một dây điện trở kéo lên. Sự cần thiết của điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là cực máng hở. Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường là giao động trong khoảng 1k đến 4.7k.

Như hình trên, ta thấy nhiều thiết bị cùng được nối vào một bus I2C, tuy nhiên sẽ không xảy ra chuyện nhầm lẫn giữa các thiết bị, bởi mỗi thiết bị sẽ được nhận ra bởi một địa chỉ duy nhất với một quan hệ chủ tớ tồn tại trong suốt thời gian kết nối.

Mỗi thiết bị có thể hoạt động như là thiết bị nhận hoặc truyền dữ liệu hay có thể vừa truyền vừa nhận. Hoạt động truyền hay nhận còn phụ thuộc vào thiết bị đó là chủ (master) hay tớ (slave). Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ duy nhất để phân biệt thì nó còn được cấu hình là thiết bị chủ hay tớ.

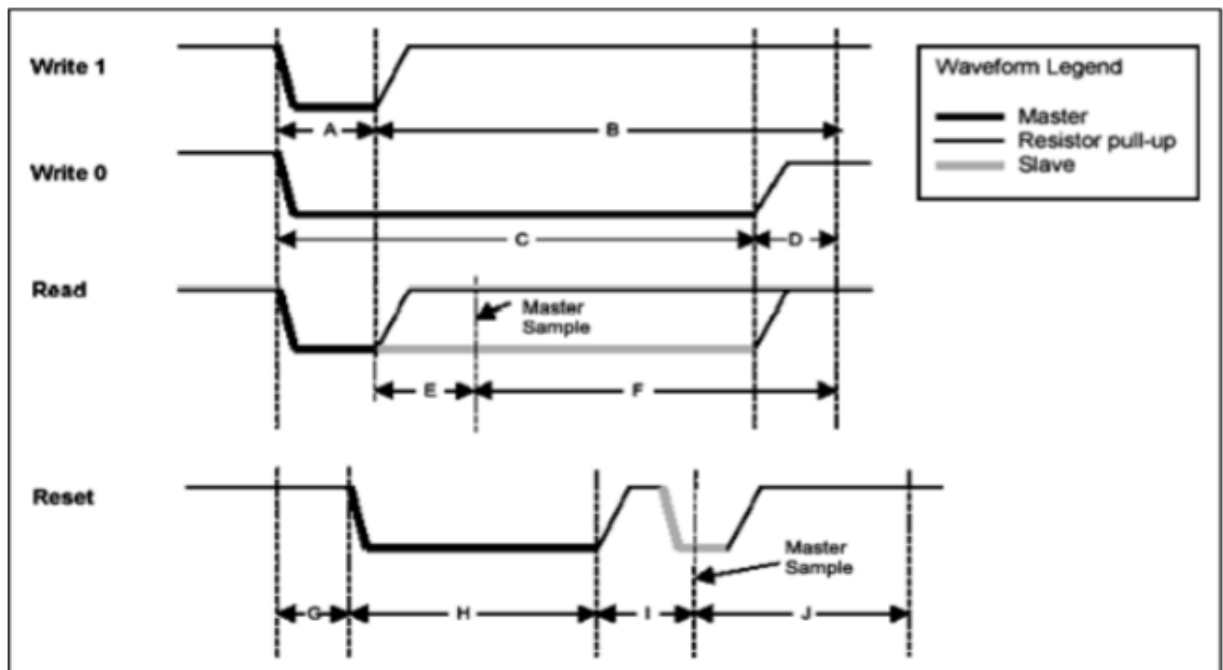
2.4.3 Chuẩn giao tiếp ONE-WIRE

Chuẩn giao tiếp 1 dây (1 wire) do hãng Dallas giới thiệu. Trong giao tiếp này chỉ cần 1 dây để truyền tín hiệu và làm nguồn nuôi (nếu không tín dây mass). Là chuẩn giao tiếp không đồng bộ và bán song công (half-duplex). Trong giao tiếp này tuân theo mối quan hệ chủ tớ một cách chặt chẽ. Trên một bus có thể gắn 1 hoặc nhiều thiết bị slave nhưng chỉ có 1 master có thể kết nối đến bus này. Bus dữ liệu khi ở trạng thái rảnh (khi không có dữ liệu trên đường truyền) phải ở mức cao do vậy bus dữ liệu phải được kéo lên kéo lên nguồn thông qua 1 điện trở. Giá trị điện trở này có thể tham khảo trong datasheet của thiết bị/các thiết bị slave.



Hình 2-6: Giao tiếp One-Wire

Để giao tiếp được với vi điều khiển, tín hiệu trên bus 1 wire chia thành các khe thời gian $60\mu s$. Một bit dữ liệu truyền trên bus dựa trên khe thời gian (time slots). Các thiết bị slave khác nhau cho phép có thời gian quy định khác nhau. Nhưng quan trọng nhất trong chuẩn giao tiếp này là cần chính xác về thời gian. Vì vậy để tối ưu đường truyền thì cần một bộ định thời để delay chính xác nhất.



Hình 2-7: Giao tiếp One-Wire.

Bốn thao tác hoạt động cơ bản của bus 1 wire là Reset/Presence, gửi bit 0, và đọc bit cụ thể là:

- Wire 1 (gửi bit 1): Master kéo xuống 0 một khoảng A (us) rồi về mức 1 khoảng B (us).
- Wire 0 (gửi bit 0): Master kéo xuống 0 một khoảng C (us) rồi thả về mức 1 khoảng B (us).
- Wire 1 (gửi bit 1): Master kéo xuống 0 một khoảng A (us) rồi về mức 1 khoảng D.
- Read (đọc một bit): Master kéo xuống 0 một khoảng A (us) rồi trả về 1 delay khoảng E rồi đọc giá trị slave gửi về delay F(us).
- Restart: Master kéo xuống 0 một khoảng H rồi thả về mức 1 sau đó cấu hình Master là chân in delay I(us) rồi đọc giá trị slave trả về. Nếu bằng 0 thì cho phép giao tiếp, nếu bằng 1 thì đường truyền lỗi hoặc slave đang bận.

2.5 Sơ lược về mạng Localhost

Localhost là khái niệm được ghép từ hai chữ “local” (máy tính của bạn) và “host” (máy chủ). Đây là thuật ngữ chỉ máy chủ chạy trên máy tính cá nhân. Localhost cơ bản như một webserver bao gồm các thành phần: Apache, MySQL, PHP và phpMyAdmin. Localhost dùng chính ổ cứng máy tính làm không gian lưu trữ và cài đặt trang web.

2.6 Giao thức Socketio

Để xây dựng một ứng dụng realtime cần sử dụng socketio. Socketio sẽ giúp các bên ở những địa điểm khác nhau kết nối với nhau, truyền dữ liệu ngay lập tức thông qua server trung gian. Socketio có thể sử dụng trong nhiều ứng dụng như chat, game online, cập nhật kết quả của một trận đấu đang diễn ra,....

Socketio không phải một ngôn ngữ, mà chỉ là một công cụ giúp thực hiện những ứng dụng realtime. Vì thế, không thể sử dụng socketio để thay thế hoàn toàn cho một ngôn ngữ, mà phải sử dụng kết hợp với một ngôn ngữ khác. Ngôn ngữ đó có thể là PHP, ASP.net, NodeJs,....

Cấu trúc một ứng dụng realtime sử dụng socketio bao gồm 2 phần: phía server và phía client.

Phía server: đây là nơi sẽ cài đặt socket.io. Ngôn ngữ để dựng server có thể là PHP, ASP.net, Nodejs,.... Tuy nhiên tùy vào ngôn ngữ lựa chọn mà cấu trúc server khác nhau. Ở đây nếu được thì khuyến khích dùng Nodejs để dựng server, vì như vậy có thể cài trực tiếp socketio vào cùng một server. Nếu sử dụng PHP thì phải cài thêm những package khác, hoặc phải chuẩn bị riêng server để chạy socketio.

Phía client: ở phía client sẽ xây dựng giao diện người dùng. Ở đây có thể sử dụng Js, hoặc các thư viện của Js như JQuery,.... Nói chung là ngôn ngữ gì cũng được.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Các lớp Socket được sử dụng để tiến hành kết nối giữa server và client. Nó được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.

Khi cần trao đổi dữ liệu cho nhau thì 2 ứng dụng cần phải biết thông tin IP và port bao nhiêu của ứng dụng kia. Việc sử dụng socket.io rất đơn giản và giống nhau ở cả client lẫn server, gồm 3 phần chính:

- Khởi tạo kết nối.
- Lắng nghe sự kiện (event).
- Gửi sự kiện (event).

Riêng ở server thì sẽ không có phần khởi tạo kết nối vì chỉ có client mới cần khởi tạo kết nối đến server. Việc sử dụng socket.io đồng bộ ở cả client lẫn server, cú pháp cũng khá đơn giản.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

3.1 GIỚI THIỆU

Đề tài “NGHIÊN CỨU VÀ THI CÔNG TỬ TRỒNG RAU NGẮN NGÀY TRONG NHÀ” bao gồm:

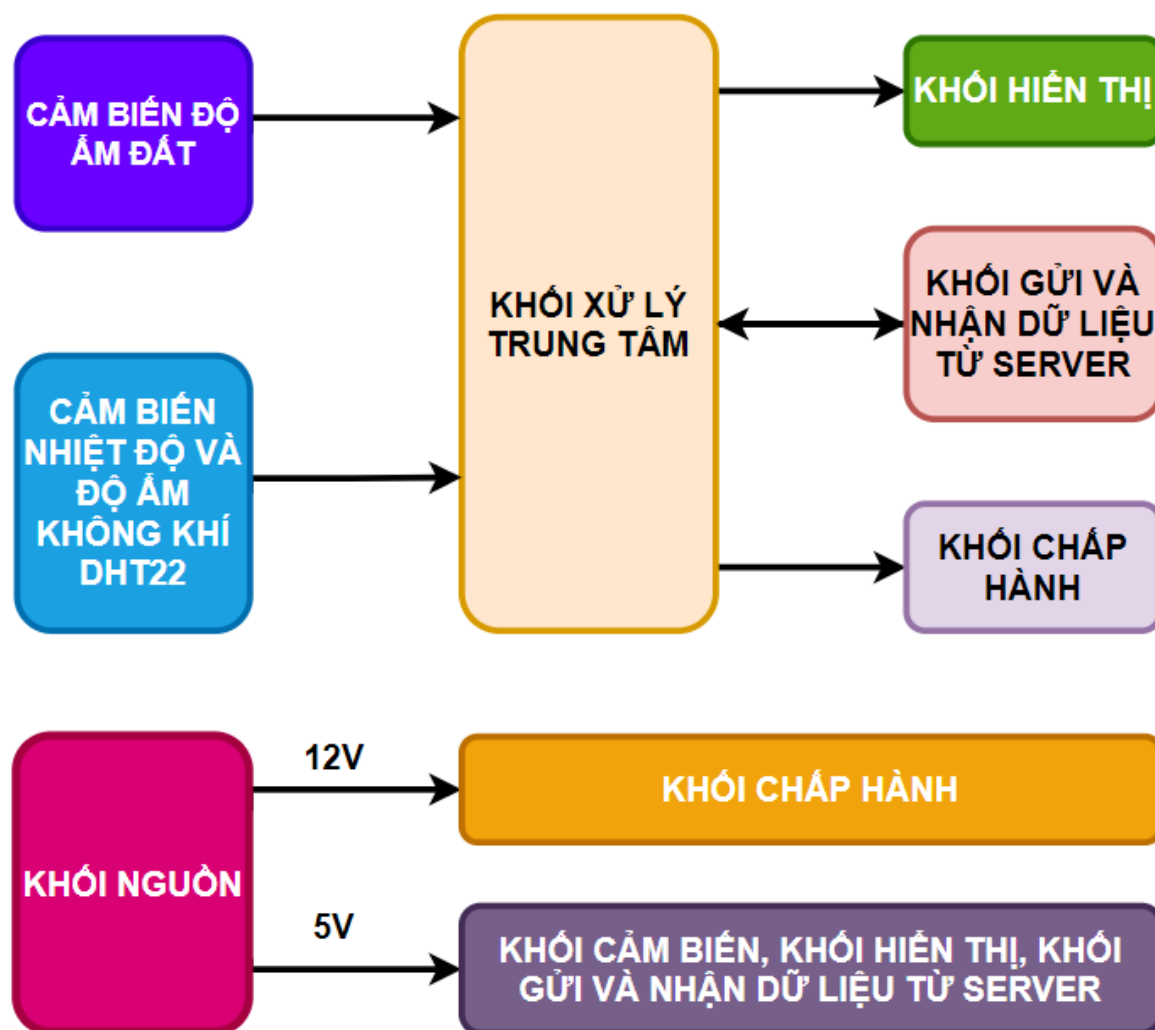
- Hệ thống điều khiển máy bơm nước cho cây trồng qua nút nhấn gắn trên mô hình cũng như web và ứng dụng Android. Dữ liệu của các cảm biến, trạng thái hoạt động của các thiết bị sẽ hiển thị trên màn hình LCD được gắn trên mô hình và web/ứng dụng.
- Hệ thống mở rộng bao gồm: có nguồn điện dự phòng hoạt động khi có sự cố mất điện. Hệ thống tự động bật/tắt đèn dựa vào cảm biến ánh sáng. Ngoài ra việc tưới nước sẽ dựa vào cảm biến độ ẩm đất, nhiệt độ môi trường. Việc kết nối và điều khiển qua web, ứng dụng Android sẽ dễ dàng hơn khi mua một tên miền truy cập riêng thay vì sử dụng mạng localhost.

3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2.1 Thiết kế sơ đồ khối hệ thống

a. Sơ đồ khối

Để biết được hệ thống có những khối cơ bản nào để phục vụ cho quá trình hoạt động. Hãy quan sát hình 3-1:



Hình 3-1: Sơ đồ khối hệ thống.

b. Giải thích chức năng các khối

Khối xử lý trung tâm:

- Nhận dữ liệu từ các khối cảm biến, ESP8266 để điều khiển các thiết bị qua module Relay và LM2596.
- Truyền dữ liệu cho khối hiển thị.
- Truyền dữ liệu cho Module ESP8266 để gửi thông tin hiển thị lên server và server sẽ gửi thông tin hiển thị về ứng dụng android và web.

Khối Module ESP8266:

- Dùng để kết nối wifi, đưa thông tin lên server.
- Giao tiếp với Arduino Nano qua các chân TX, RX để lấy thông tin gửi lên server.

Khối hiển thị:

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

- Hiển thị các giá trị mà các cảm biến đo được và trạng thái hoạt động của các thiết bị chấp hành.

Khối Chấp hành:

- Gồm 1 đèn, 2 quạt, 1 máy bơm.

Khối cảm biến độ ẩm đất:

- Đo độ ẩm đất trồng.

Khối cảm biến nhiệt độ và độ ẩm không khí DHT22:

- Đo nhiệt độ, độ ẩm không khí của môi trường để người dùng dựa vào đó mà thay đổi thời gian tưới sao cho phù hợp.

Khối nguồn:

- Cung cấp nguồn cho các khối ở trên để các khối hoạt động.

3.2.2 Tính toán và thiết kế mạch

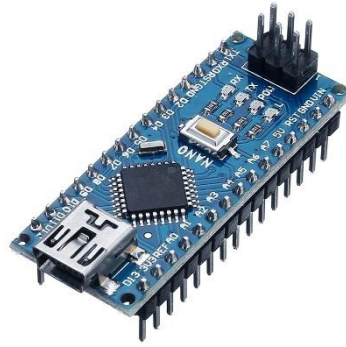
a. Khối xử lý trung tâm

Khối xử lý trung tâm: được xem là phần quan trọng nhất của hệ thống. Khối có chức năng tiếp nhận xử lý mọi tín hiệu ngõ vào thu được từ các cảm biến, các tín hiệu điều khiển và truyền nhận dữ liệu từ web để xử lý điều khiển thiết bị của khối chấp hành thông qua khối delay và đưa ra khối điều khiển trực tiếp và hiển thị để người dùng theo dõi. Toàn bộ hoạt động điều khiển của hệ thống được thông qua khối xử lý này.

Với các yêu cầu trên, hiện nay có rất nhiều sự lựa chọn ở nhiều phân khúc khác nhau như các dòng PLC của Siemens, Panasonic,... hay các dòng vi điều khiển họ Pic, các dòng vi điều khiển ARM, các dòng kit Arduino,... Tuy nhiên với yêu cầu tiện dụng dễ sử dụng, và phù hợp với mạch nên nhóm em quyết định sử dụng Arduino Nano cho khối xử lý trung tâm.

Vì Arduino Nano sử dụng chip Atmega328 (họ 8 bit) nên nó có 8 chân analog I/O và 14 chân Digital I/O, một thạch anh dao động 16MHz, kết nối USB, một jack cắm điện, bộ nhớ flash 32KB, SRAM 2KB, EEPROM 1 KB. Nó chứa tất cả mọi thứ cần thiết để tạo thành khối xử lý trung tâm với các port.

Sử dụng Arduino Nano với vi điều khiển chính Atmega328 (8 bit). Nguồn cung cấp: 5-12VDC (cấp bằng cổng Mini USB hoặc chân Vin của chip). Mức điện áp giao tiếp của chân GPIO: 5VDC.

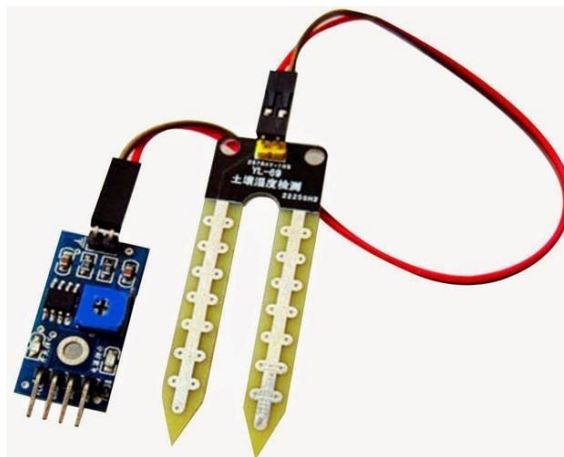


Hình 3-2: Arduino Nano.

b. Cảm biến đất

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. Phần đầu đo được cắm vào đất để phát hiện độ ẩm của đất, khi độ ẩm đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao. Nhờ thế, người dùng có thể sử dụng Analog hoặc Digital của Arduino để đọc giá trị từ cảm biến.

- Module cảm biến độ ẩm đất có thể được sử dụng cho các ứng dụng nông nghiệp, tưới nước tự động cho các vườn cây khi đất khô, hoặc dùng trong các ứng dụng của hệ thống nhà thông minh. Module cảm biến độ ẩm đất gồm hai phần:
 - + Đầu dò: hai đầu đo của đầu dò được cắm vào đất để phát hiện độ ẩm. Dùng dây nối giữa cảm biến và module chuyển đổi, khi độ ẩm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao. Thông tin về độ ẩm đất sẽ được đọc về và gửi tới module chuyển đổi.
 - + Module chuyển đổi: module chuyển đổi có cấu tạo chính gồm một IC so sánh LM393, một biến trở, 4 điện trở dán 100 ohm và 2 tụ dán. Biến trở có chức năng định ngưỡng so sánh với tín hiệu độ ẩm đất đọc về từ cảm biến. Ngưỡng so sánh và tín hiệu cảm biến sẽ là 2 đầu vào của IC so sánh LM393. Khi độ ẩm thấp hơn ngưỡng định trước, ngõ ra của IC là mức cao (1), ngược lại là mức thấp (0).



Hình 3-3: Cảm biến độ ẩm đất.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Thông số kỹ thuật:

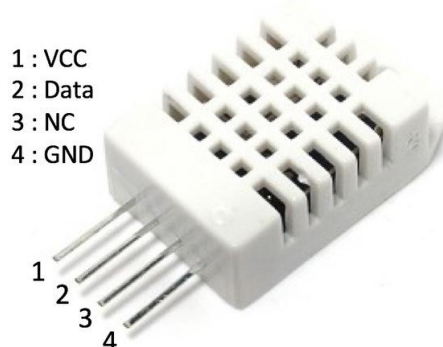
Điện áp hoạt động: 3.3-5 VDC.

- Tín hiệu đầu ra:
 - + Analog: theo điện áp cấp nguồn tương ứng.
 - + Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trở thông qua mạch so sánh LM393 tích hợp.

Sự hấp thụ độ ẩm (hơi nước) làm biến đổi thành phần cảm nhận trong cảm biến (ở đây là các chất hóa học như LiCl, P2O5) làm thay đổi điện trở của cảm biến qua đó xác định được độ ẩm. Khi thay đổi độ ẩm điện trở trên cảm biến thay đổi dẫn đến điện áp đầu ra đưa vào cổng so sánh trên Opam thay đổi, điện áp này được so sánh với điện áp đặt được đặt bằng biến trở, nếu điện áp đọc về từ cảm biến chưa vượt qua ngưỡng đặt thì đầu ra D0 là mức thấp và led báo trạng thái không sáng, khi điện áp đầu vào vượt qua ngưỡng đặt thì đầu ra D0 là mức cao và led báo trạng thái sẽ sáng lên.

c. Cảm biến nhiệt độ và độ ẩm DHT22

Chọn loại DHT22 cho đề tài vì nó có thể đo được cả nhiệt độ và độ ẩm. Cảm biến này có thể đo tốt ở nhiệt độ từ -40°C đến 80°C với sai số $\pm 0.5\%$, độ ẩm đo trong khoảng từ 20-95%, sai số $\pm 5\%$ có thể đáp ứng yêu cầu thực tế. Cảm biến DHT22 được bán phổ biến, giá hợp lý.



Hình 3-4: Cảm biến DHT22.

Cảm biến nhiệt độ, độ ẩm DHT22 sử dụng chuẩn giao tiếp one wire (1 dây) để giao tiếp với vi điều khiển thực hiện đo nhiệt độ, độ ẩm không khí.

Thông số kỹ thuật:

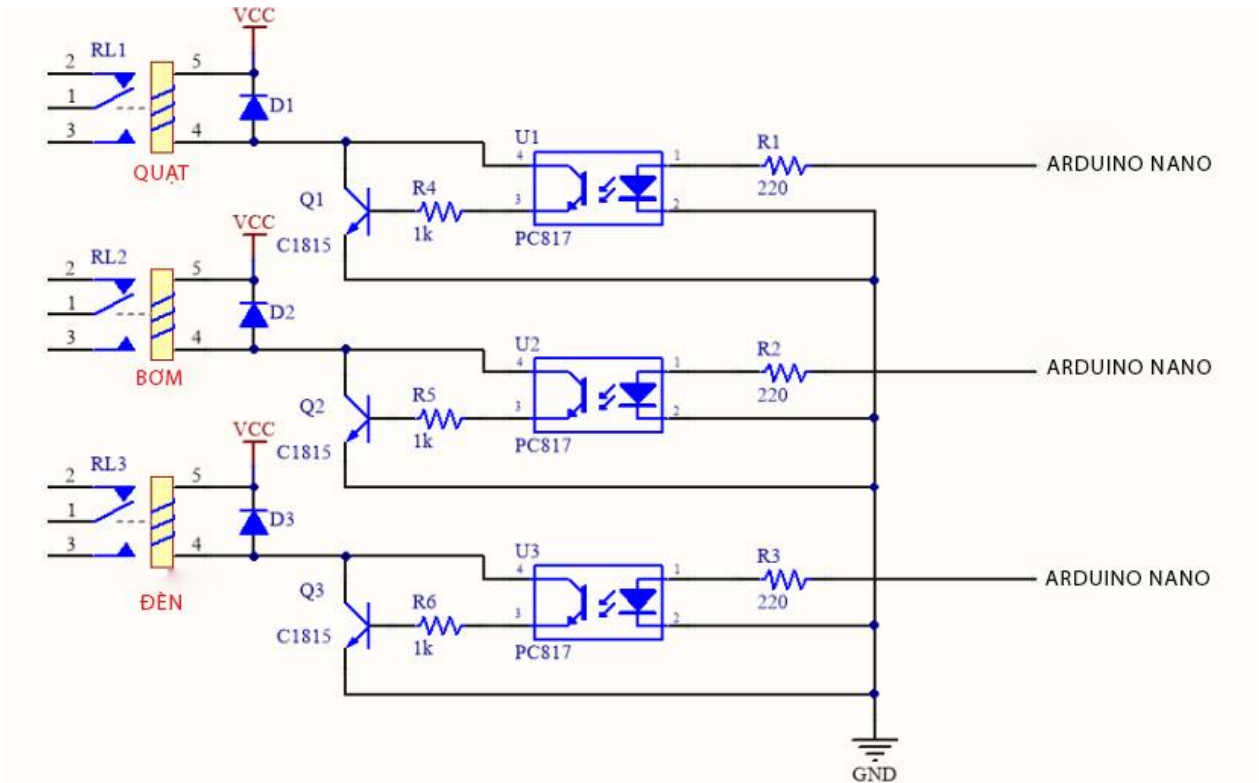
- Nguồn cung cấp: 3-5 VDC.
- Tần số lấy mẫu tối đa: 0.5 Hz (khoảng 2 giây 1 lần).
- Chuẩn giao tiếp: one wire.

d. Module Relay

- Cách ly hoàn toàn giữa mạch xử lý có điện áp thấp và thiết bị có điện áp cao và công suất tải lớn. An toàn cho bộ xử lý và mạch điều khiển.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

- Trở kháng giữ hai trạng thái ON có trở kháng rất nhỏ, trạng thái OFF có trở kháng rất lớn và xem như lý tưởng. Ổn định vận hành cho thiết bị có công suất nhỏ, điện áp nhỏ của mô hình đề tài.
- Mô hình của đề tài không có khí dễ gây cháy nổ: ga, xăng,... Do đó có thể bỏ qua yếu tố cháy nổ gây ra do tia lửa điện trong thời gian chuyển trạng thái. Việc sử dụng relay để đóng, ngắt bơm, đèn và quạt là giải pháp khả thi và ưu tiên vì đơn giản và ổn định.



Hình 3-5: Sơ đồ nguyên lý của Relay.

- Arduino nano có $V_{OH} = 5\text{ V}$, $V_{OL} = 0\text{ V}$
- Relay: SRD-05VDC-SL-C relay 5V DC 10A
 - + Coil voltage: DC 5V
 - + Rated load: 10A 250V AC/ 10A 125V AC/ 10A 30V DC/ 10A 28V DC.
 - + operating current: 43mA~46mA
 - + Release current: 15mA~18mA
- Transistor có dòng bão hòa I_C ra: $I_{Csat} = 100\text{mA}$ và $\beta_{\text{bão hòa}}=10$, chọn $I_{\text{relay}} = 45\text{mA}$
 - Ta có: $I_R = I_{\text{relay}}/\beta = 45/10 = 4,5\text{mA}$
 - $\Rightarrow R = (V_{OH} - 0,7)/I_R = (5 - 0,7)/4,5 = 0,9556\text{k}\Omega$
- Nên ta chọn giá trị điện trở gần đó là $1\text{k}\Omega$.
- Để bảo vệ cho transistor không bị hư hỏng ta sử dụng diode 1N4001 vì có dòng phân cực thuận là 1A đủ lớn để dập dòng điện do cuộn dây phát ra trong quá trình transistor ngưng dẫn đột ngột.

Giới thiệu relay 5V:



Hình 3-6: Hình ảnh thực tế module relay 5V.

- Module relay 5V có tích hợp opto cách ly kích mức logic High hoặc Low.
- Mạch được sử dụng để đóng ngắt nguồn điện AC hoặc DC, có thể chọn cách kích mức High hoặc Low bằng cách Jumper.
- Có 3 tiếp điểm đóng ngắt là tiếp điểm NC (thường đóng), tiếp điểm NO (thường mở) và COM (chân chung). Khi ở trạng thái bình thường chưa được kích thì NC sẽ được nối với COM và khi được kích thì COM sẽ chuyển sang kết nối với NO.
- Có tích hợp đèn báo hiệu đóng ngắt relay trên module.

Thông số kỹ thuật:

- Nguồn điện cung cấp: 5 VDC.
- Mỗi relay tiêu thụ dòng khoảng 80mA.
- Điện thế đóng ngắt tối đa: 250V và 10A (đối với nguồn AC), 30V và 10A (đối với nguồn DC).

e. Khối hiển thị

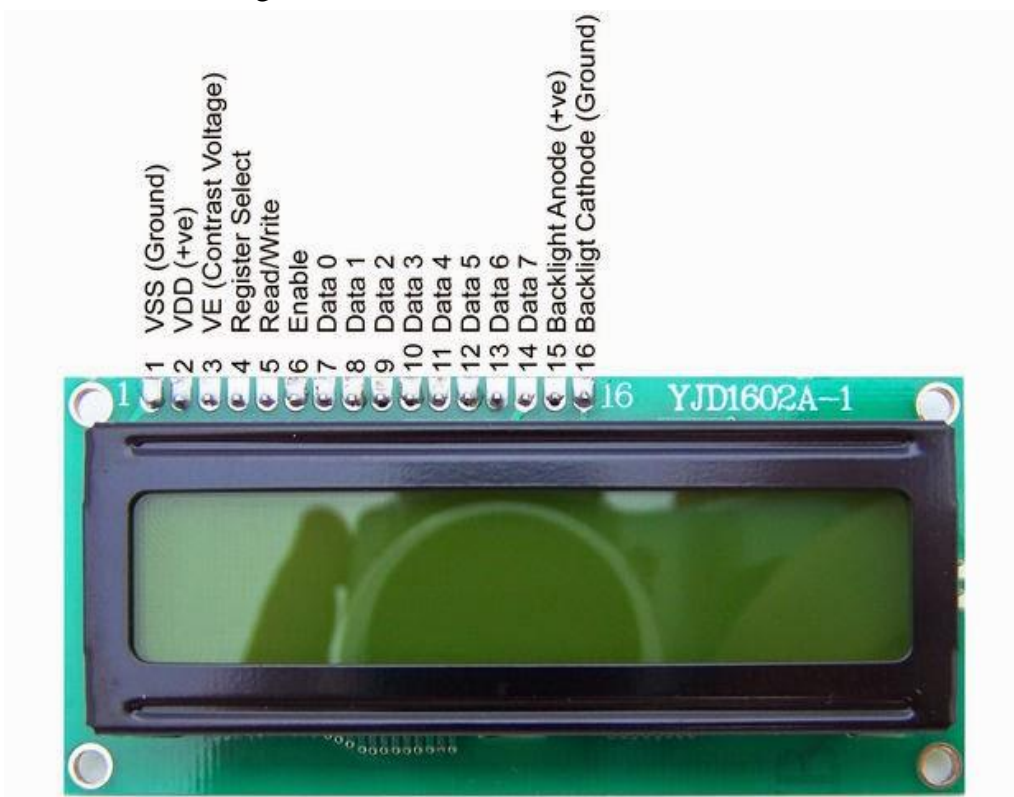
LCD text 16x2 có khả năng hiển thị 2 hàng, mỗi hàng 16 ký tự, đủ để hiển thị các thông tin cần thiết cho mô hình đề tài. Có độ bền cao cũng như giá thành hợp lý, đặc biệt được cộng đồng Arduino hỗ trợ nhiều nên được sử dụng phổ biến. Ngoài ra còn đồng bộ điện áp với Arduino.

Thông số kỹ thuật của LCD 16x2:

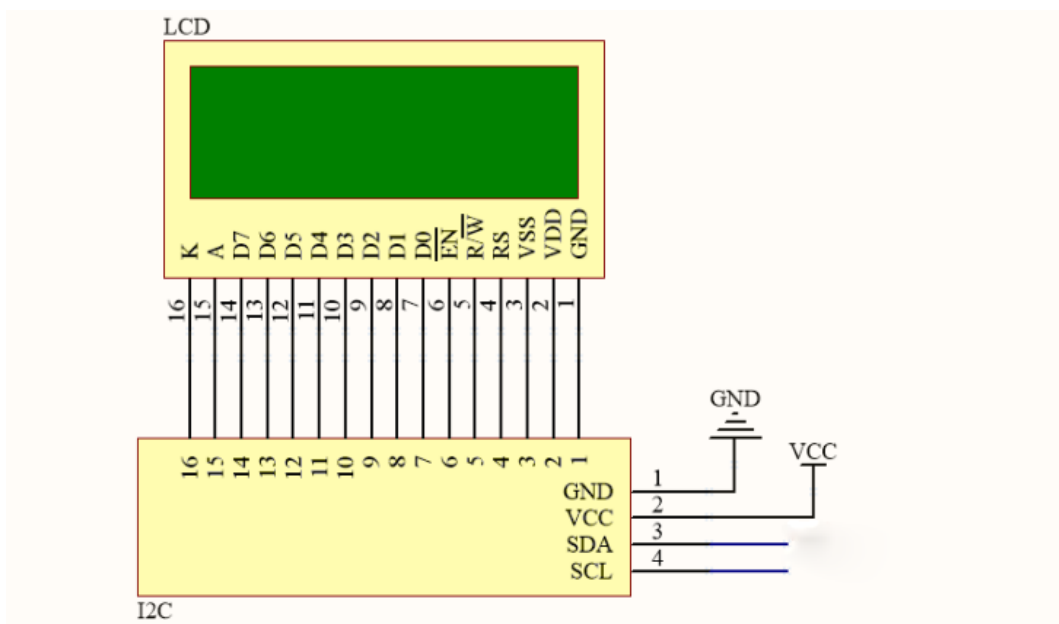
- Điện áp MAX: 7V
- Điện áp MIN: - 0,3V
- Hoạt động ổn định: 2.7-5.5V
- Điện áp ra mức cao: > 2.4V

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

- Điện áp ra mức thấp: $<0.4V$
- Dòng điện cấp nguồn: $350\mu A-600\mu A$
- Nhiệt độ hoạt động: -30 đến 75 độ C



Hình 3-7: Màn hình LCD 16x2.



Hình 3-8: Sơ đồ nguyên lí của LCD 16x2.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Chân	Kí hiệu	Chức năng
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển.
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC = 5V của mạch điều khiển.
3	VEE	Điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0 - DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read). + Logic “1”: Bus DB0 - DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7- 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.
15	A	Nguồn dương cho đèn nền.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

16	K	GND cho đèn nền.
----	---	------------------

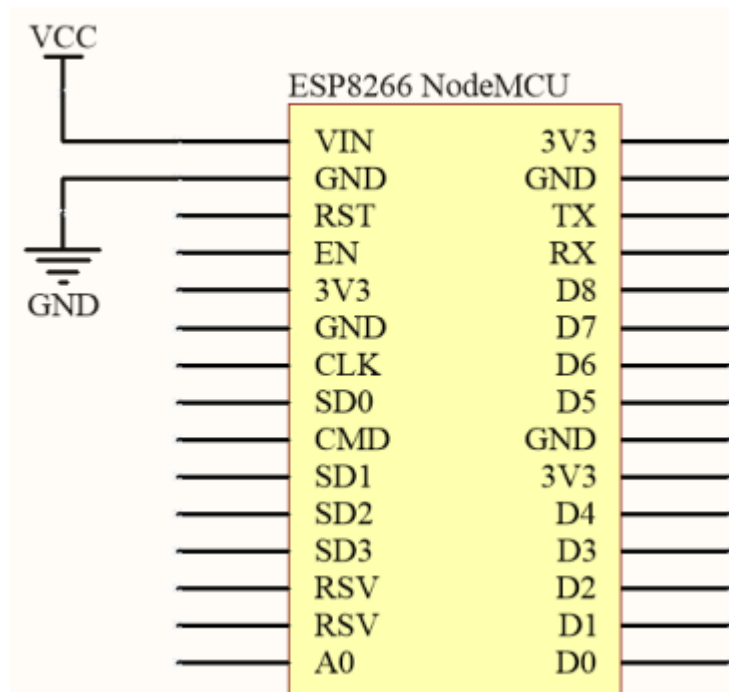
Bảng 3-1: Chức năng và thông số hoạt động của LCD 16x2.

f. ESP8266

Node MCU V1.0 được phát triển dựa trên Chip Wifi ESP8266EX bên trong Module ESP-12E dễ dàng kết nối Wifi với một vài thao tác. Board còn tích hợp IC CP2102, giúp dễ dàng thao tác với máy tính thông qua micro USB để thao tác với board. Và có sẵn các chân giao tiếp để thuận tiện trong việc kết nối, truyền nhận dữ liệu với board điều khiển, có nút reset, đèn led.

Kích thước nhỏ gọn, linh hoạt, dễ dàng kết nối với Arduino để tạo thành sản phẩm một cách nhanh chóng.

Vì vậy nhóm em sử dụng ESP8266 Node MCU để truyền dữ liệu lên server.



Hình 3-9: Sơ đồ nguyên lý của ESP 8266.

- Điện áp sử dụng: 3.3V-3.6V.
- Dòng làm việc: 80mA (170mA max).



Hình 3-10: Module NodeMCU 8266.

g. Các thiết bị chấp hành

Bơm: Dùng cho hệ thống tưới nước điều khiển qua module relay.

Thông số kỹ thuật:

- Lực hút tối đa 2m.
- Đường kính đầu vào và ra: 8mm .
- Trọng lượng: 120g .
- Lưu lượng tối đa: 2-3 lít/phút .
- Áp suất tối đa: 1-2,5 kg.
- Lực nâng tối đa: 1-2,5 mét.
- Điện áp 12VDC, dòng 3A. Công suất 36W.



Hình 3-11: Bơm mini 12VDC.

Đèn:

Cây xanh cần ánh sáng mặt trời, hơi nước và CO₂ có trong không khí để có thể quang hợp. Quá trình này cần lượng ánh sáng phù hợp để tạo ra chất hữu cơ giàu năng lượng giúp cây phát triển.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Dù ánh sáng mặt trời có tầm quan trọng như vậy nhưng ngày nay con người đã có thể tự trồng cây lương thực, rau quả hay cây cảnh trong nhà mà không cần ánh sáng tự nhiên. Thay vào đó, đèn LED chiếu sáng cây trồng, đèn chiếu sáng chuyên dụng trong nông nghiệp đã được con người sử dụng một cách hiệu quả, mang lại năng suất cao.

Đối với mỗi loại cây lại có nhu cầu ánh sáng khác nhau, chính vì thế khi lựa chọn đèn chiếu sáng cho cây trong nhà, cần phải xem chúng thuộc loại cây gì, yêu cầu cường độ và bước sóng ánh sáng ra sao, màu ánh sáng nào là phù hợp. Thông thường cây không thể hấp thụ toàn bộ ánh sáng mặt trời mà chỉ hấp thụ ánh sáng màu xanh và đỏ. Chính vì thế nên quang phổ ánh sáng dao động trong khoảng màu này là ổn nhất.

Thực vật ở những khu vực khô, nắng gắt yêu cầu một nguồn sáng có cường độ mạnh. Những giống cây ở khu vực ôn đới lại cần nguồn sáng yếu hơn. Không chỉ vậy, mỗi giai đoạn của cây trồng khác nhau cũng yêu cầu một quang phổ khác nhau. Chẳng hạn như:

Ánh sáng xanh có nhiệt độ màu từ 5000K đến 6000K với bước sóng 400-500 nanomet, giúp cho quá trình sinh dưỡng của chu kỳ phát triển hoặc giéo hạt tốt nhất.

Ánh sáng xanh lá thường được sử dụng trong chu kỳ tối của cây trồng vì nó không kích hoạt quá trình quang hợp. Do vậy màu ánh sáng này phù hợp cho ban đêm hơn cả.

Ánh sáng đỏ với màu 2000-3000k và bước sóng 640-680 nanomet cực kì quan trọng trong quá trình ra hoa. Chính vì thế ánh sáng đỏ được mọi người sử dụng để thúc đẩy quá trình phân hóa mầm hoa, nâng cao thời gian chiếu sáng cho cây phát triển nhanh hơn. Do đó ánh sáng này thích hợp cho những loại cây ngắn ngày, rau màu.

Với vườn cây không có ánh sáng mặt trời chiếu đến thì cây trồng chỉ hấp thụ hai dải ánh sáng chính là dải ánh sáng màu xanh dương (425nm-475nm) và dải ánh sáng đỏ (620nm-730nm) cho việc quang hợp. Chính vì thế, khi chọn loại đèn sử dụng để thay thế ánh sáng mặt trời, cần chọn loại có quang phổ từ 350-800nm.

Chúng ta thường thấy có 2 loại đèn chuyên dụng trồng cây là đèn chưa tách quang phổ và đèn đã tách quang phổ. Thực tế, hai loại đèn này công dụng không chênh lệch nhau quá nhiều nhưng giá thành loại đèn đã tách quang phổ lại cao hơn rất nhiều. Chính vì thế, khi quyết định chọn lựa loại đèn led, nên cân nhắc kỹ về kinh tế và hiệu quả đạt được để lựa chọn loại đèn cho thích hợp.



Hình 3-12: Đèn led chiếu sáng cho cây trong nhà.

Thông số kỹ thuật:

- LED Công suất: 8W
- Loại bóng đèn: LED
- Số lượng đèn LED: 15 bóng
- Chống nước
- Kích thước: Dài 30cm x Rộng 0.8cm x Dày 0.20cm
- Điện áp: 12V

Quạt:

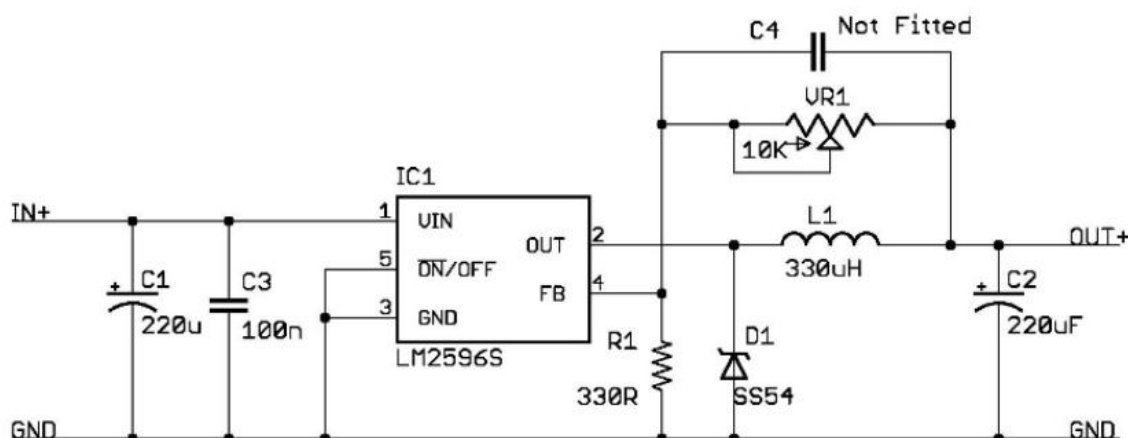
- Quạt định hướng 12V: Khác với quạt thường thổi gió theo phương dọc trục, quạt định hướng gom gió lại và thổi theo phương bán kính của cánh quạt. Nhờ cấu tạo như vậy nên quạt định hướng có tốc độ gió cao, dễ dàng bố trí ống dẫn khí tới lỗ đèn nhựa trên máy in 3D.
- Dòng: 0.18A
- Điện áp: 12VDC



Hình 3-13: Quạt 12 VDC.

h. Module hạ áp:

Nguồn cấp chính của mạch là nguồn 12V nên nếu muốn sử dụng cho các thiết bị 5V cần phải hạ áp mạch nguồn. Ở đây nhóm em sử dụng module LM2596. Chính điện áp đầu ra bằng cách vận biến trở trên module rồi lấy đồng hồ đo điện áp đầu ra, hiệu chỉnh sao cho phù hợp.



Hình 3-14: Sơ đồ nguyên lý của LM2596.

Giới thiệu module giảm áp LM2596:



Hình 3-15: Module giảm áp LM2596.

- Module LM2596 là module giảm áp đầu vào giúp nguồn ra ở mức ổn định, có tích hợp biến trở tinh chỉnh dùng cho việc điều khiển điện áp ngõ ra.
- Điện áp ngõ vào phải đảm bảo lớn hơn điện áp ngõ ra ở mức tối thiểu là 1.5V để tạo sự ổn định cho nguồn.

Thông số kỹ thuật:

- Điện áp đầu vào: 4.5-53 VDC.
- Điện áp ngõ ra: 3V-40 VDC.
- Dòng ngõ ra tối đa: 3A.

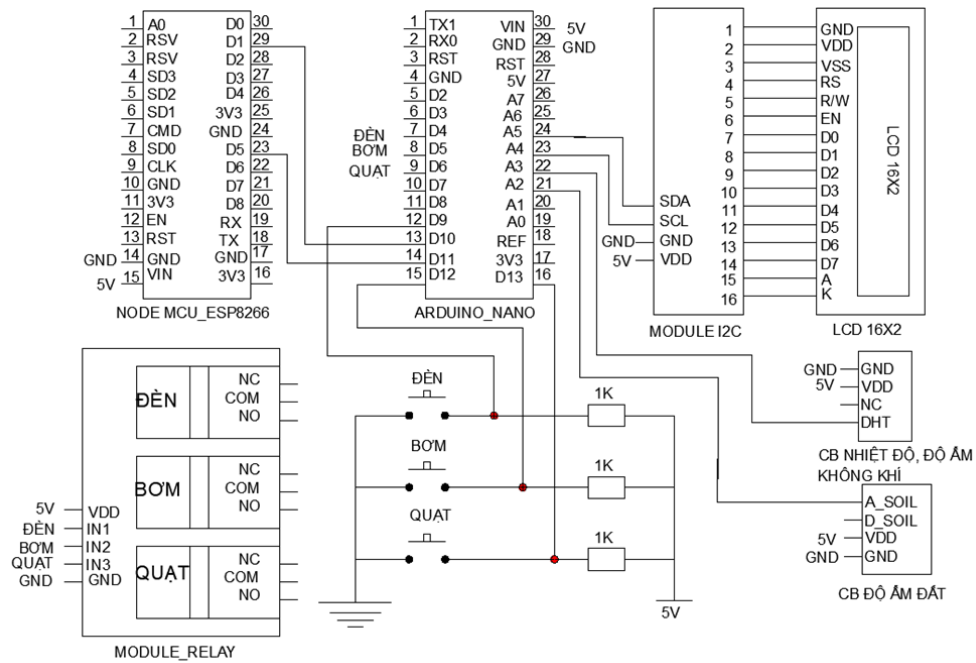
i. Khối nguồn:

Đây là khối cung cấp nguồn cho toàn bộ mô hình.

- Công suất của máy bơm: $12V \times 3A = 36 W$.
=> Sử dụng Adapter 12V-5A để cung cấp nguồn hoạt động cho máy bơm, thông qua module hạ áp.
- Công suất của 2 quạt: $2 \times (12V \times 0.18A) = 4.32W$
- Công suất của đèn chiếu sáng cho cây: 8W.
- Công suất Node MCU ESP8266 sử dụng $5 \times 10 \times 0.08 = 4W$
- Công suất của cảm biến DHT22: $5V \times 0.0025A = 0.0125W$
- Công suất của 3 khối Relay: $3 \times (5V \times 0.2A) = 3W$
- Công suất của LCD 16x2: $5V \times 0.016A = 0.08W$
- Công suất của Arduino Nano với 13 chân I/O, mỗi chân tiêu thụ 0.04A là: $5V \times 13 \times 0.04 = 2.6W$
- Tổng công suất là: 22.0125 W

=> Sử dụng Adapter 12V-2A để cung cấp nguồn hoạt động cho máy bơm, thông qua module hạ áp.

3.2.3 Sơ đồ nguyên lý toàn mạch



Hình 3-16: Sơ đồ nguyên lý toàn mạch.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.1 GIỚI THIỆU

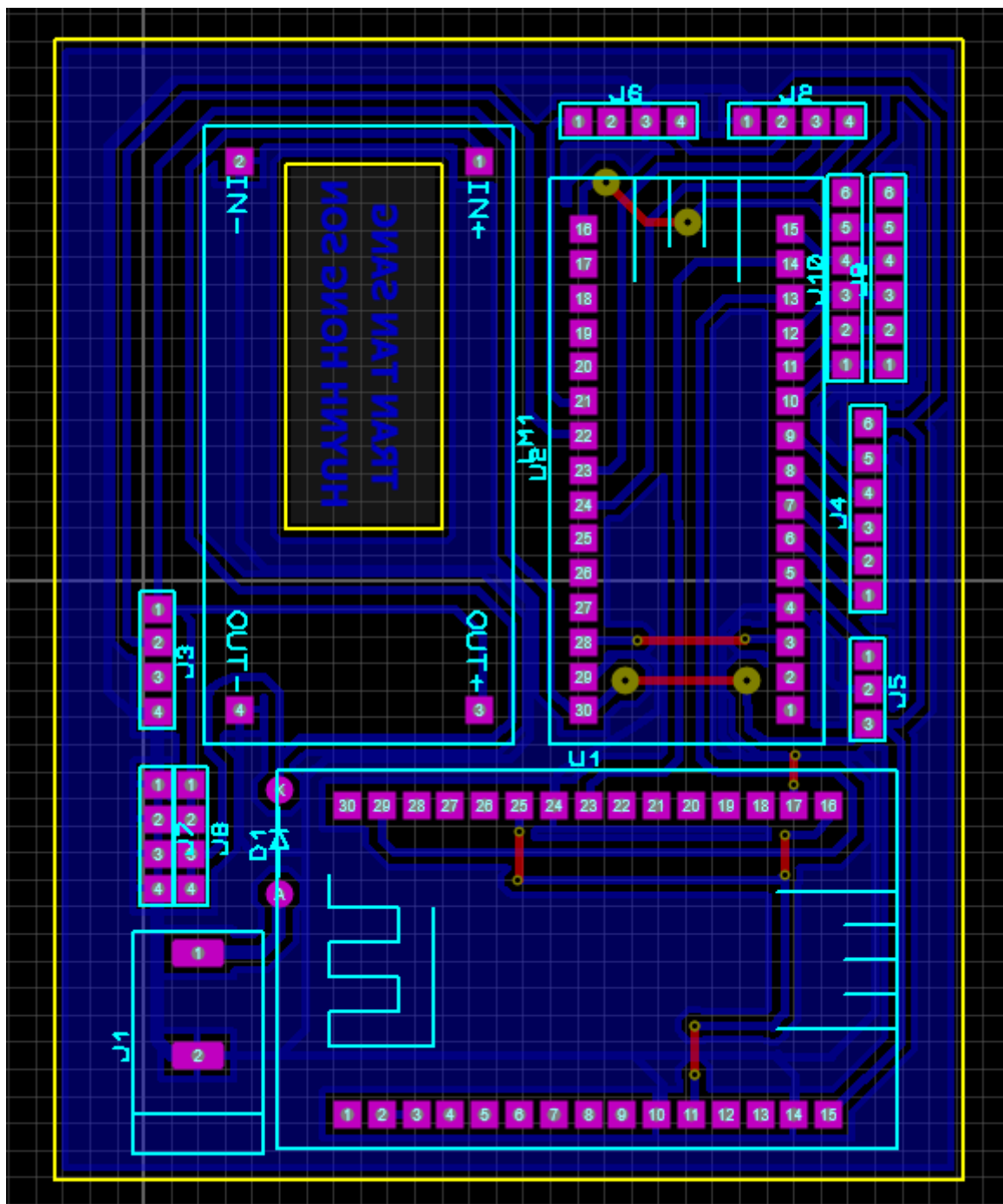
Chương này nhóm em tiến hành thi công mạch như đã tính toán và thiết kế ở các chương trước, bao gồm vẽ mạch layout, hàn linh kiện, lắp ráp cũng như thi công mô hình. Sau đó kiểm tra mạch, viết chương trình điều khiển cũng như thiết kế web điều khiển và ứng dụng Android. Tiếp tục kiểm tra mạch lần nữa, kiểm tra hoạt động của web và ứng dụng đã hoạt động, kết nối đúng theo ý muốn chưa và tinh chỉnh lại sao cho phù hợp.

4.2 THI CÔNG MẠCH ĐIỀU KHIỂN

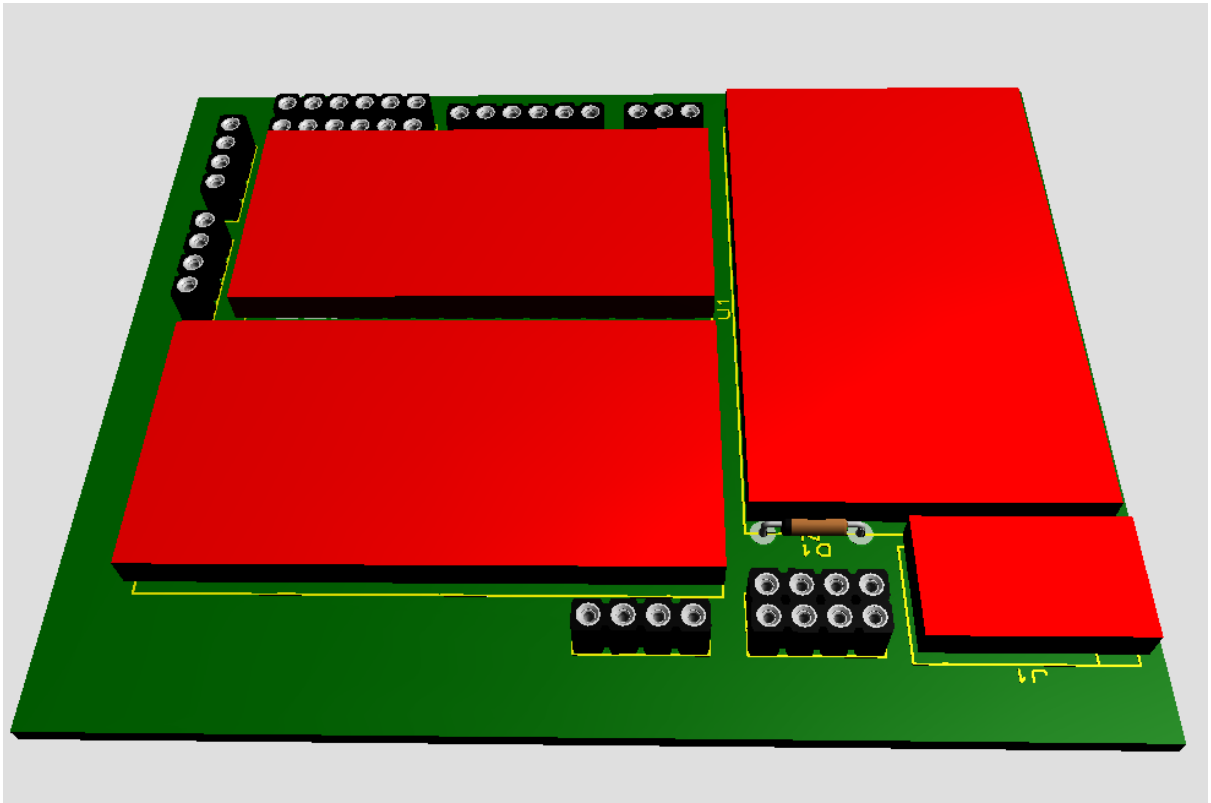
4.2.1 Thi công mạch điều khiển

STT	Tên linh kiện	Thông số	Số lượng
1	Arduino Nano	5 VDC	1
2	ESP 8266	5 VDC	1
3	Module 2596	Vào 12 VDC, ra 5 VDC	1
4	Cảm biến đất	3.3 VDC-5 VDC	1
5	Cảm biến DHT22	5 VDC	1
6	Bơm	12 VDC, 3A	1
7	Đèn	5 VDC	1
8	Quạt	5 VDC	2
9	Adapter 12V-2A	12 VDC, 2A	1
10	Adapter 12V -5A	12 VDC, 5A	1
11	Điện trở	330 Ω	3
12	Module I2C	2.5-6 VDC	1
13	Nút nhấn	5 VDC	3
14	Module Relay	5 VDC	3

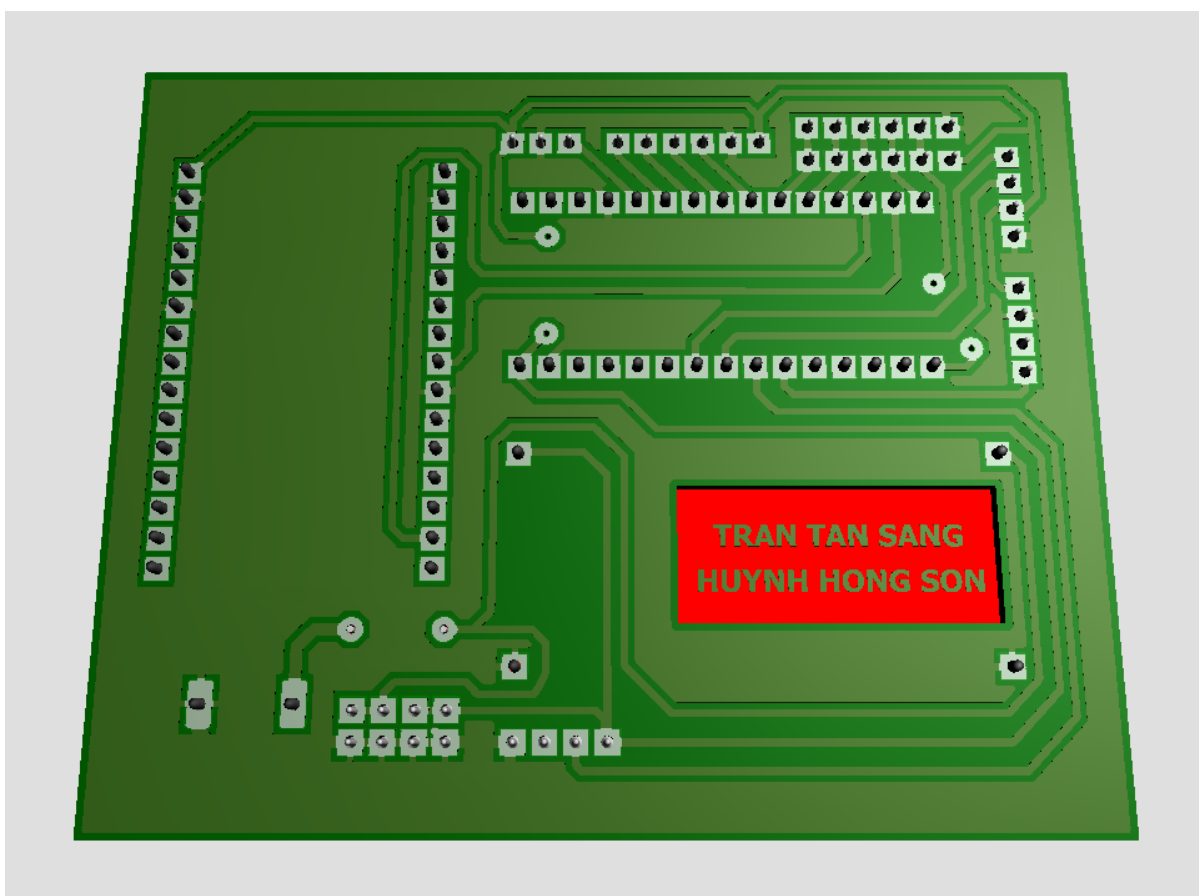
Bảng 4-1: Danh sách linh kiện.



Hình 4-1: Mạch vẽ PCB.



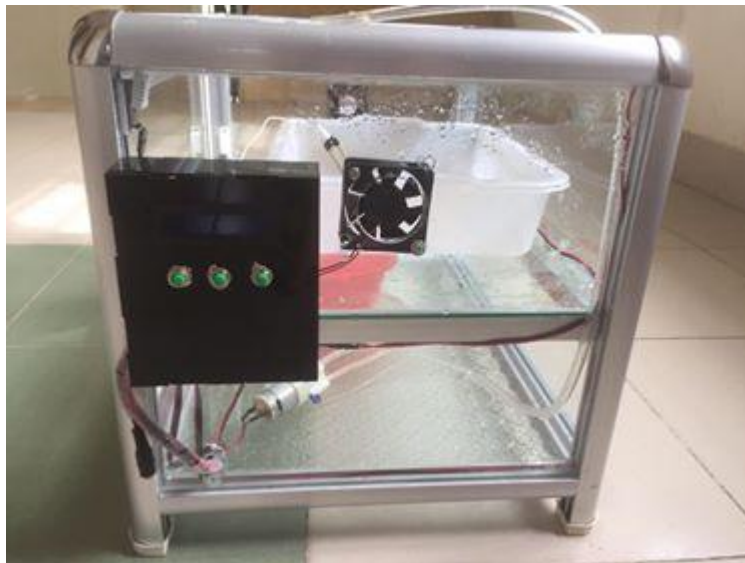
Hình 4-2: Mô hình 3D của mạch mặt trên.



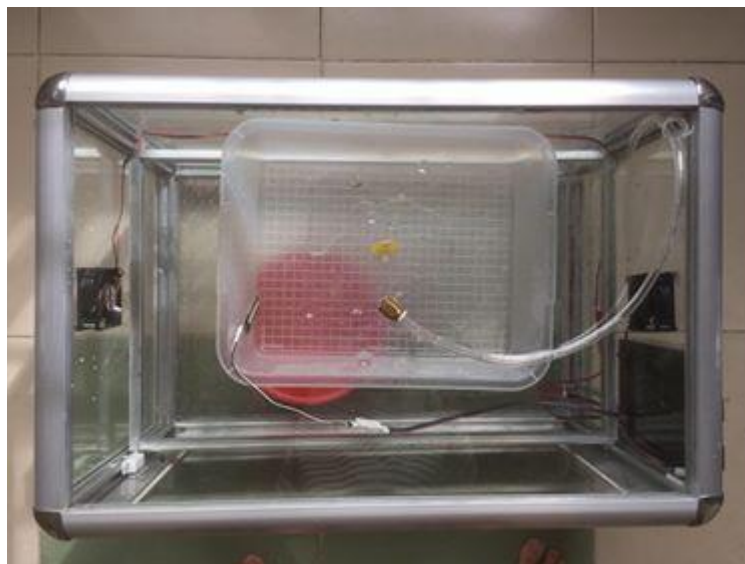
Hình 4-3: Mô hình 3D của mạch mặt dưới.

4.3 THI CÔNG MÔ HÌNH

- Tủ trồng rau có kích thước dài 60 cm, cao 40 cm, rộng 40 cm. Vật liệu nhôm kính để dễ quan sát từ bên ngoài.
- Tủ được chia thành 2 tầng, tầng dưới cao 20 cm là nơi để chứa máy bơm và khay đựng nước, cung cấp nước tưới cho cây trồng.
- Tầng trên cao 20 cm, rau mầm phát triển đến 15 cm là có thể thu hoạch, cộng thêm 3 cm đất trồng, tổng là 18 cm. Còn 2 cm còn lại trong tổng số 20 cm chiều cao của ngăn trồng cây thì để bố trí vòi phun sương.



Hình 4-4: Ảnh mặt bên hông của mô hình với bảng điều khiển.



Hình 4-5: Mô hình từ nhìn từ trên xuống.



Hình 4-6: Mô hình tủ nhìn từ một bên hông, với cánh quạt thông gió.



Hình 4-7: Mô hình tổng quát.

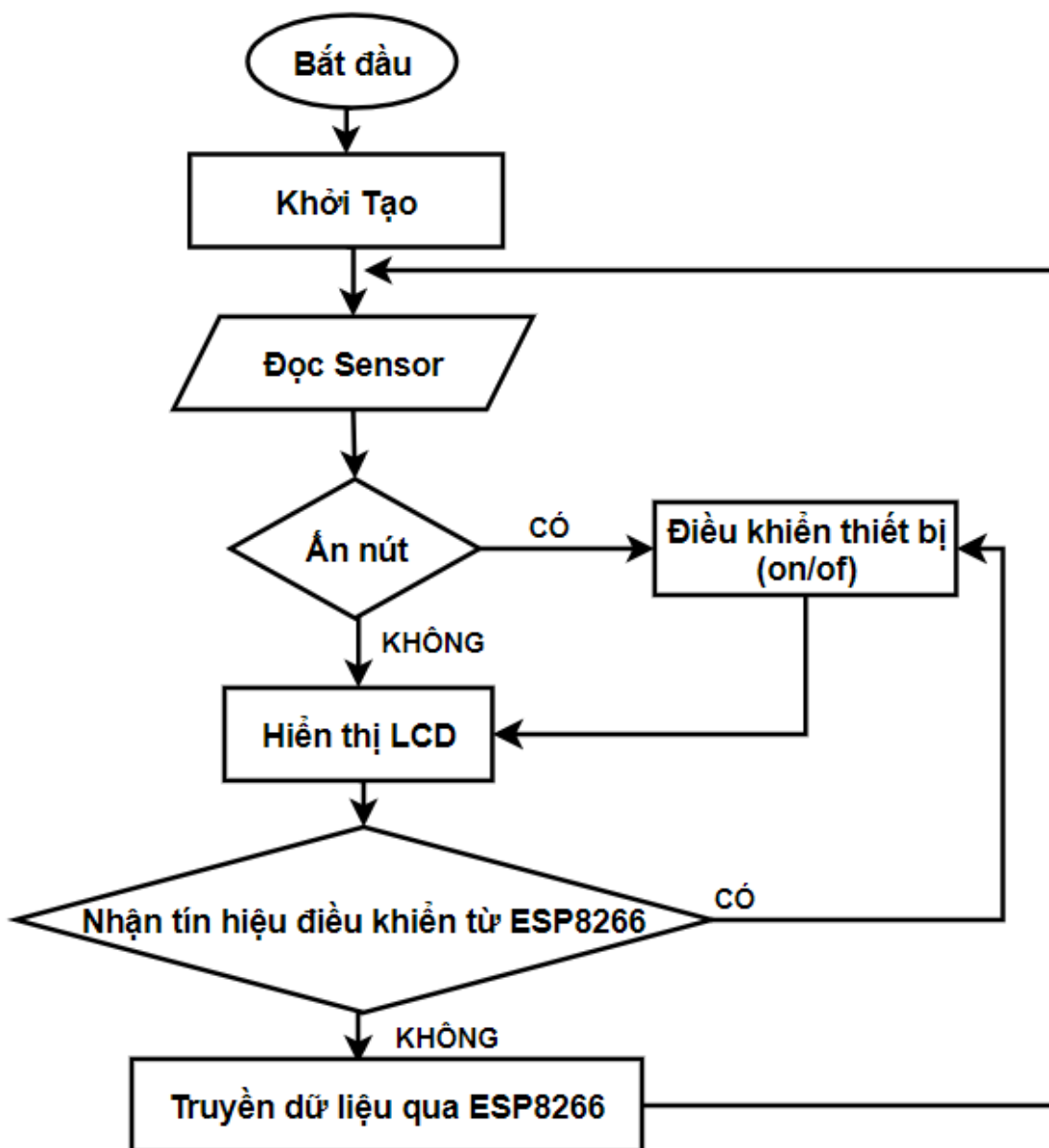
4.4 LƯU ĐỒ GIẢI THUẬT

4.4.1 Lưu đồ giải thuật Nano

Để hệ thống có được những thông số như nhiệt độ, độ ẩm không khí, độ ẩm đất và trạng thái hoạt động của các thiết bị chấp hành. Ta cần phải có khối xử lý trung tâm để xử lý và đọc các giá trị đó. Đầu tiên khối xử lý trung tâm sẽ đọc dữ liệu từ khối cảm biến sau đó sẽ kiểm tra nút nhấn. Nếu nút nhấn có nhấn sẽ điều khiển thiết bị chấp hành hoạt động và hiển thị dữ liệu ra màn hình LCD đồng

CHƯƠNG 4: THI CÔNG HỆ THỐNG

thời gửi dữ liệu cho ESP8266 để ESP8266 gửi dữ liệu lên server. Và server sẽ gửi dữ liệu hiển thị ra app và web. Và được thể hiện ở hình 4-8:



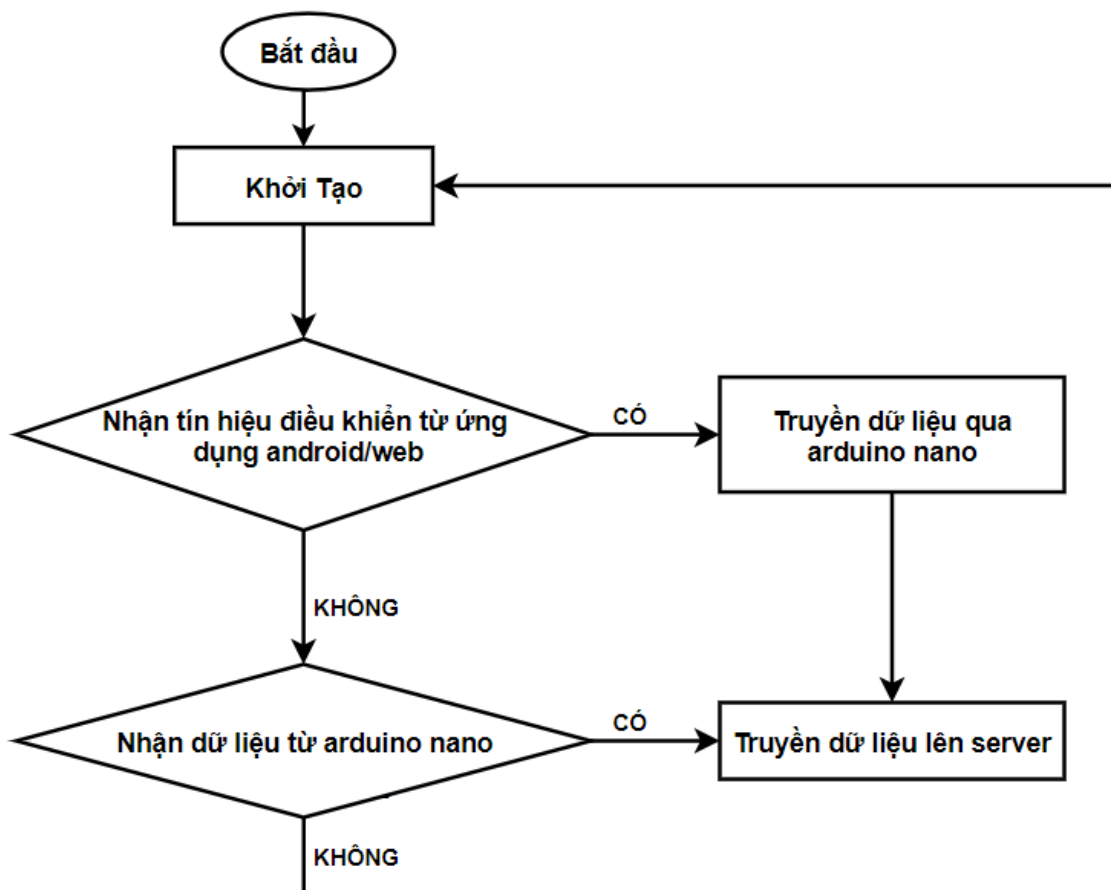
Hình 4-8: Lưu đồ giải thuật Arduino Nano.

4.4.2 Lưu đồ giải thuật Node MCU

Để gửi dữ liệu lên server và điều khiển các thiết bị trên app/web thì ta cần phải có Node MCU ESP8266. Khi ESP8266 nhận tín hiệu điều khiển từ app/web thì nó sẽ truyền dữ liệu cho arduino nano xử lý và điều khiển các thiết bị chấp hành sau đó hiển thị dữ liệu ra màn hình LCD đồng thời cũng cập nhật trạng thái trên app/web. Nếu ESP8266 nhận tín hiệu điều khiển từ arduino nano thì sẽ gửi

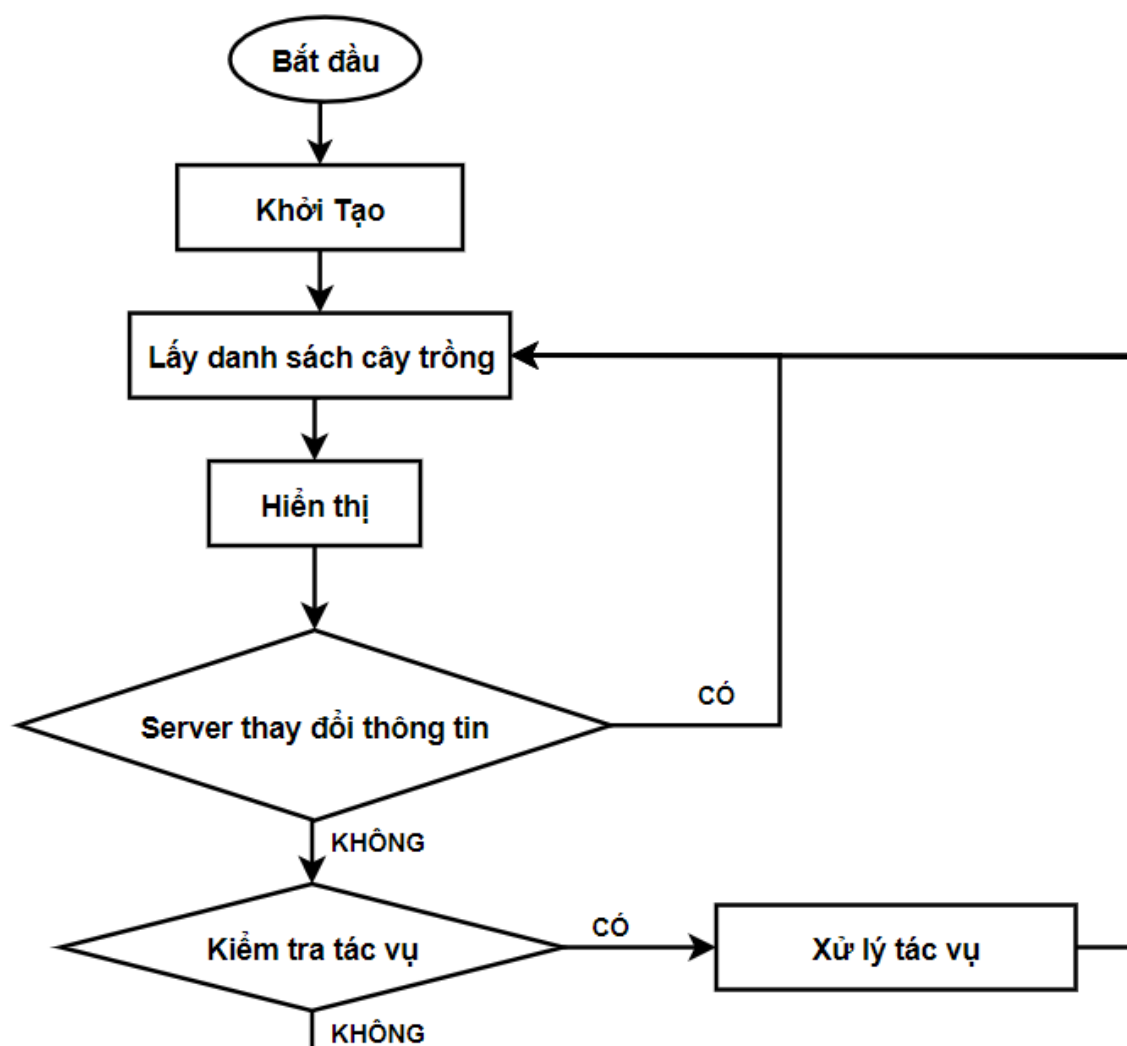
CHƯƠNG 4: THI CÔNG HỆ THỐNG

dữ liệu lên server sau đó server đưa thông tin hiển thị ra app/web đồng thời LCD cũng cập nhật trạng thái. Và được thể hiện ở hình 4-9:



Hình 4-9: Lưu đồ giải thuật NODE MCU ESP8266.

4.4.3 Lưu đồ giải thuật cho ứng dụng, web



Hình 4-10: Lưu đồ giải thuật cho ứng dụng Android và web.

Khi người dùng cập nhật thông tin ở tab “trồng cây” thì server sẽ cập nhật thông tin và phản hồi lại, gửi dữ liệu về web cũng như ứng dụng Android. Khi có thay đổi bất kỳ thông tin nào (server sẽ kiểm tra tác vụ), nếu có thì xử lý tác vụ đó, và lặp lại thao tác lấy danh sách cây trồng, nếu không thì thôi, chỉ hiển thị những gì đã thao tác trước đó.

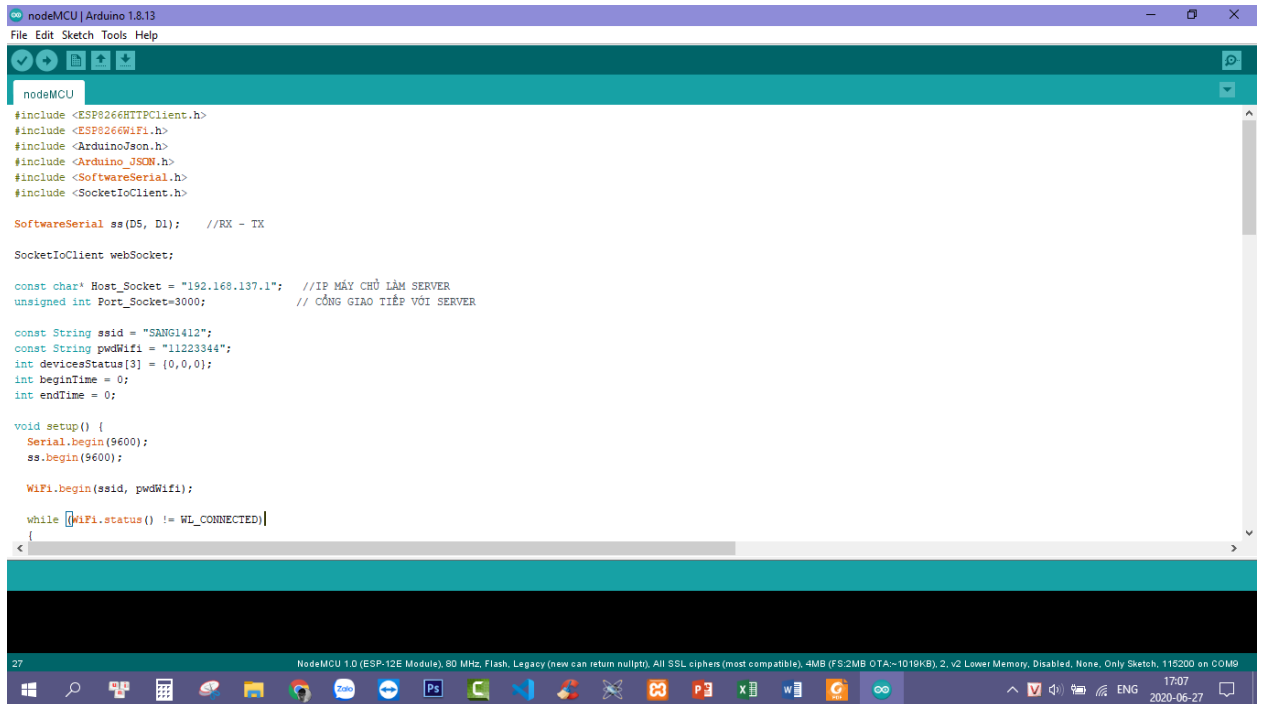
4.5 XÂY DỰNG WEB VÀ ỨNG DỤNG ANDROID

4.5.1 Phần mềm lập trình cho vi điều khiển

Một trong những phần mềm lập trình tốt nhất cho Arduino hiện nay đó chính là phần mềm ARDUINO IDE. Phần mềm này có thể sử dụng được cho bất kỳ bo mạch Arduino nào.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Ngôn ngữ phổ biến để lập trình trên Arduino là C và C++. Phần mềm có rất nhiều thư viện, và người dùng cũng có thể tự tạo thư viện cho riêng mình.



```
nodeMCU
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <Arduino_JSON.h>
#include <SoftwareSerial.h>
#include <SocketIOClient.h>

SoftwareSerial ss(D5, D1); //RX - TX

SocketIOClient webSocket;

const char* Host_Socket = "192.168.137.1"; //IP MÁY CHỦ LÀM SERVER
unsigned int Port_Socket=3000; // Cổng GIAO TIẾP VỚI SERVER

const String ssid = "SANG1412";
const String pwdWifi = "11223344";
int devicesStatus[3] = {0,0,0};
int beginTime = 0;
int endTime = 0;

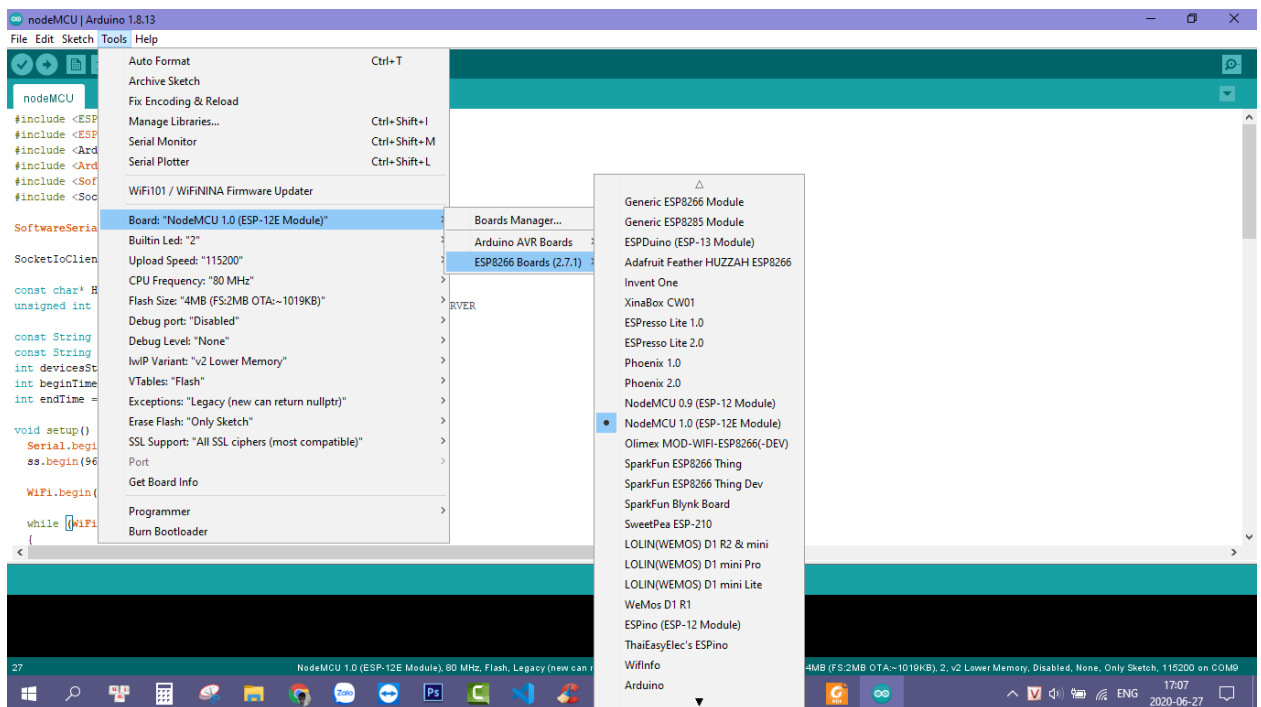
void setup() {
  Serial.begin(9600);
  ss.begin(9600);

  WiFi.begin(ssid, pwdWifi);

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

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:2MB OTA~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9
27
Windows taskbar: 17:07, 2020-06-27
```

Hình 4-11: Giao diện viết chương trình cho Arduino.



Hình 4-12: Chọn Board Arduino sử dụng.

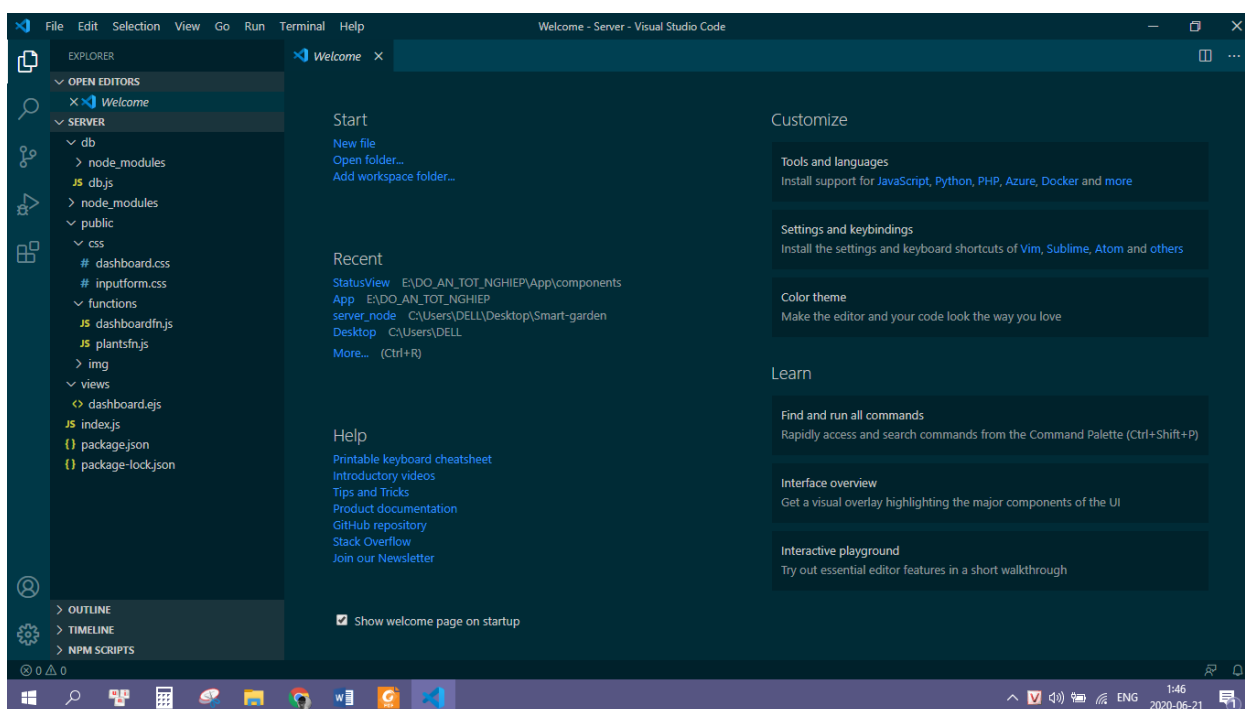
CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.5.2 Visual Studio Code

Visual Studio Code (VS Code hay VCS) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, HTML, CSS, JavaScript,.... Vì vậy nó dễ dàng phát hiện và đưa ra thông báo nếu chương trình có lỗi.

Visual Studio hỗ trợ nhiều ứng dụng web. Ngoài ra nó cũng có trình soạn thảo và thiết kế website.

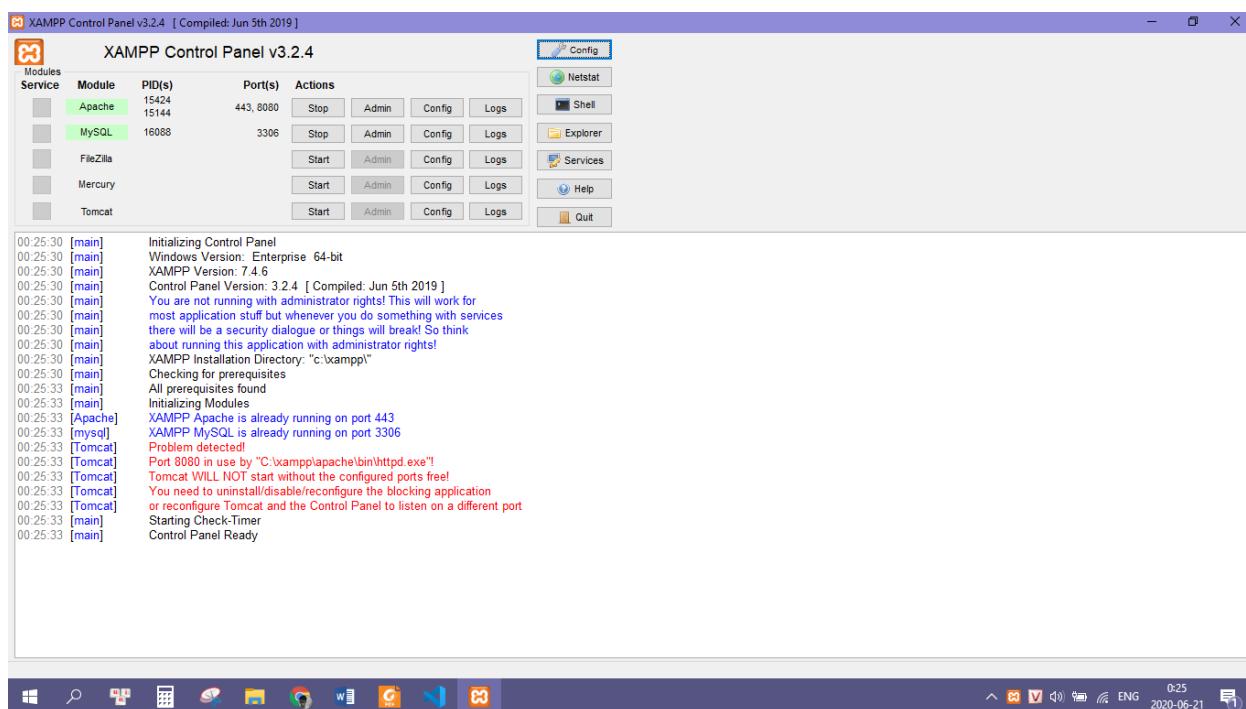


Hình 4-13: Giao diện của phần mềm Visual Studio Code.

4.5.3 XAMPP

XAMPP là chương trình tạo webserver được ứng dụng trên các hệ điều hành như Windows, MacOS, Linux,.... XAMPP là một chương trình tạo máy chủ web được sử dụng phổ biến nhất hiện nay. Ưu điểm của XAMPP là không phải trả phí bản quyền và sử dụng mã nguồn mở, bên cạnh đó cấu hình của web server này tương đối đơn giản, gọn nhẹ, nên được sử dụng ngày càng phổ biến hiện nay.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-14: Giao diện phần mềm XAMPP.

Hiện nay phiên bản mới nhất 7.4.6 tích hợp rất nhiều công cụ hữu ích như:

- Apache.
- MySQL.
- Tomcat.
- Mercury.
- FileZilla.

XAMPP được nghiên cứu, phát triển website qua localhost của máy tính cá nhân. Localhost là khái niệm được ghép từ hai chữ “local” (máy tính của bạn) và “host” (máy chủ). Đây là thuật ngữ chỉ máy chủ chạy trên máy tính cá nhân. Localhost cơ bản như một webserver bao gồm các thành phần: Apache, MySQL, PHP và phpMyAdmin. Localhost dùng chính ổ cứng máy tính làm không gian lưu trữ và cài đặt trang web.

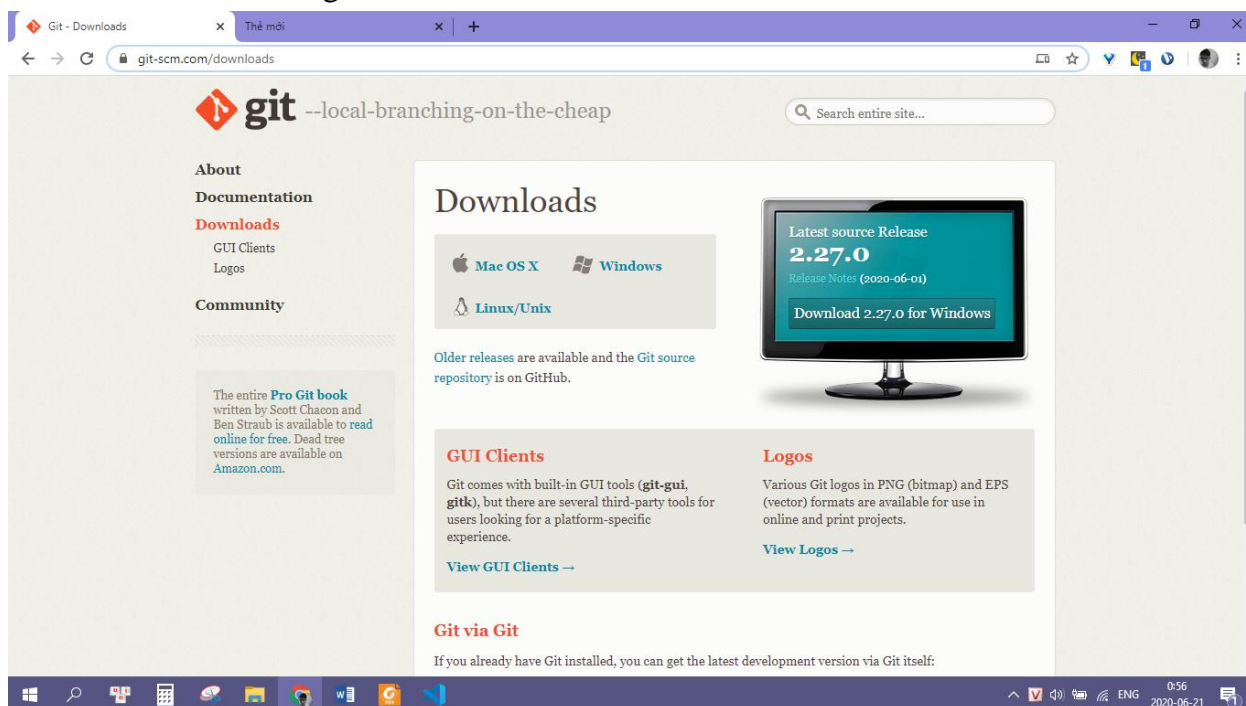
Các ưu điểm của XAMPP bao gồm:

- XAMPP có thể chạy được nhiều hệ điều hành như MacOS, Linux, Windows,....
- XAMPP tích hợp nhiều thành phần với các tính năng:
 - + Apache.
 - + PHP (tạo môi trường chạy các tập tin script*.php).
 - + MySQL (hệ quản trị dữ liệu mysql).
 - + Mã nguồn mở: người dùng có thể chủ động bật, tắt hoặc khởi động lại các dịch vụ máy chủ bất cứ lúc nào.

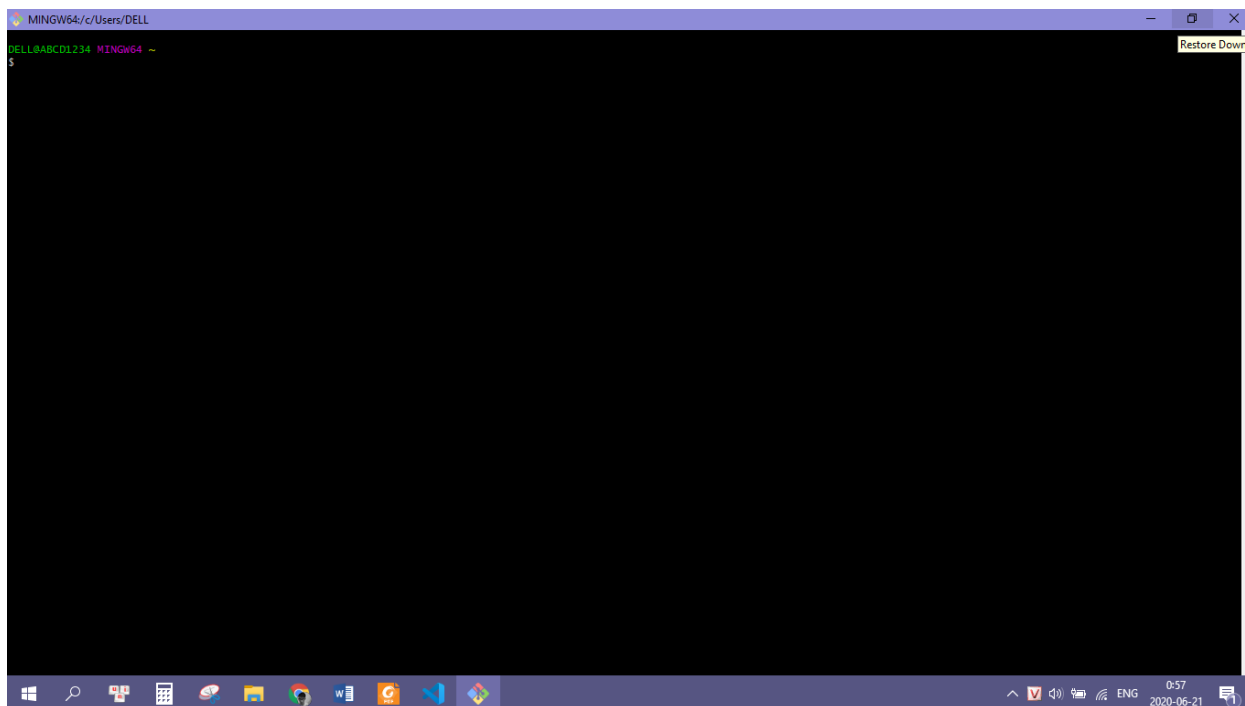
4.5.4 Git Bash

Git Bash là một ứng dụng chạy trên môi trường Windows, cung cấp lớp mô phỏng cho trải nghiệm dòng lệnh Git. Bash là từ viết tắt của Bourne Again Shell. Shell là một ứng dụng đầu cuối được sử dụng để giao tiếp với hệ điều hành thông qua các lệnh văn bản. Git Bash là gói cài đặt Bash, một số tiện ích bash phổ biến và Git trên hệ điều hành Windows.

Để cài đặt Git, truy cập vào trang chủ <https://git-scm.com/downloads> và tải về cài đặt như bình thường. Phiên bản mới nhất hiện tại là 2.27.0:



Hình 4-15: Giao diện trang chủ của Git.



Hình 4-16: Giao diện khi khởi chạy Git Bash.

Đây cũng là nơi lập trình viên gõ các dòng lệnh.

4.5.5 NodeJS

Node.js là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript. Trước khi NodeJS ra đời, JavaScript chủ yếu được sử dụng để nhúng vào HTML sử dụng trên các website cho phép người dùng tương tác với trình duyệt. Với sự ra đời của NodeJS, lập trình viên có thể dùng JavaScript để phát triển các ứng dụng trên máy chủ hoặc các ứng dụng về network.

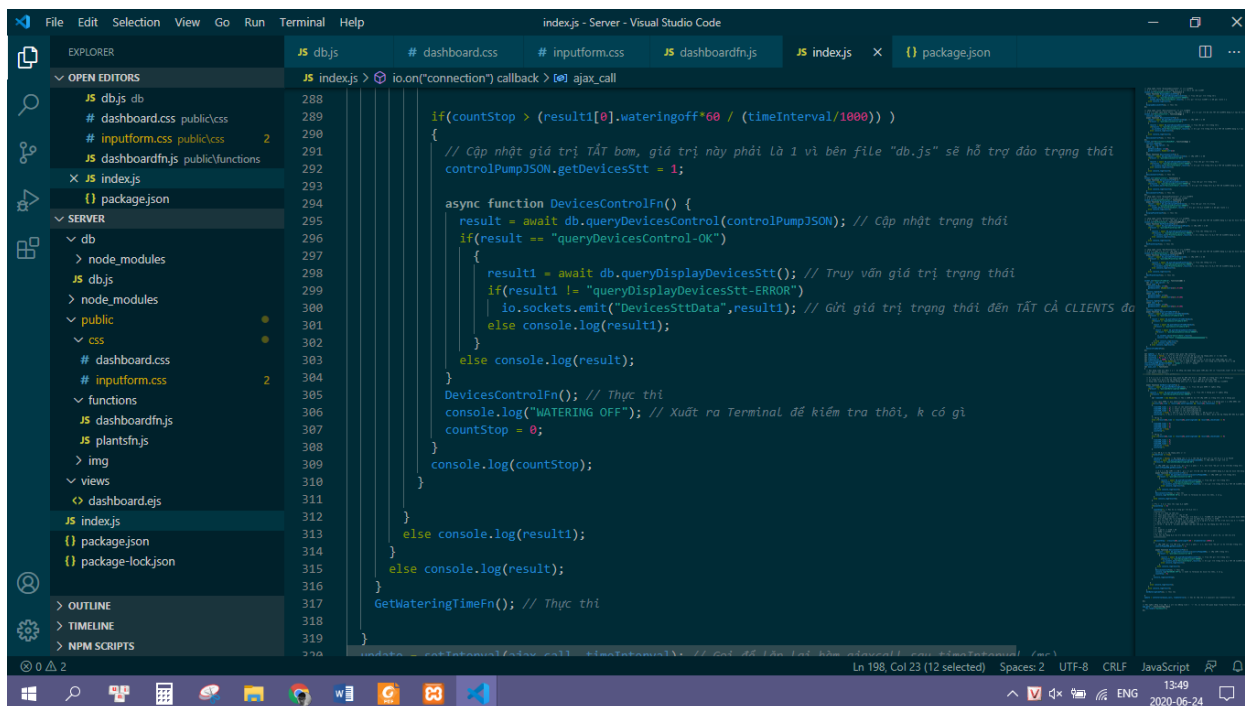
NodeJS không phải là một ngôn ngữ lập trình mới, thay vào đó nó chỉ là một nền tảng máy chủ được viết dựa trên ngôn ngữ JavaScript. Cú pháp khi viết chương trình sử dụng NodeJS giống với cú pháp ngôn ngữ JavaScript.

4.5.6 Các bước thực thi

Thiết kế web và server:

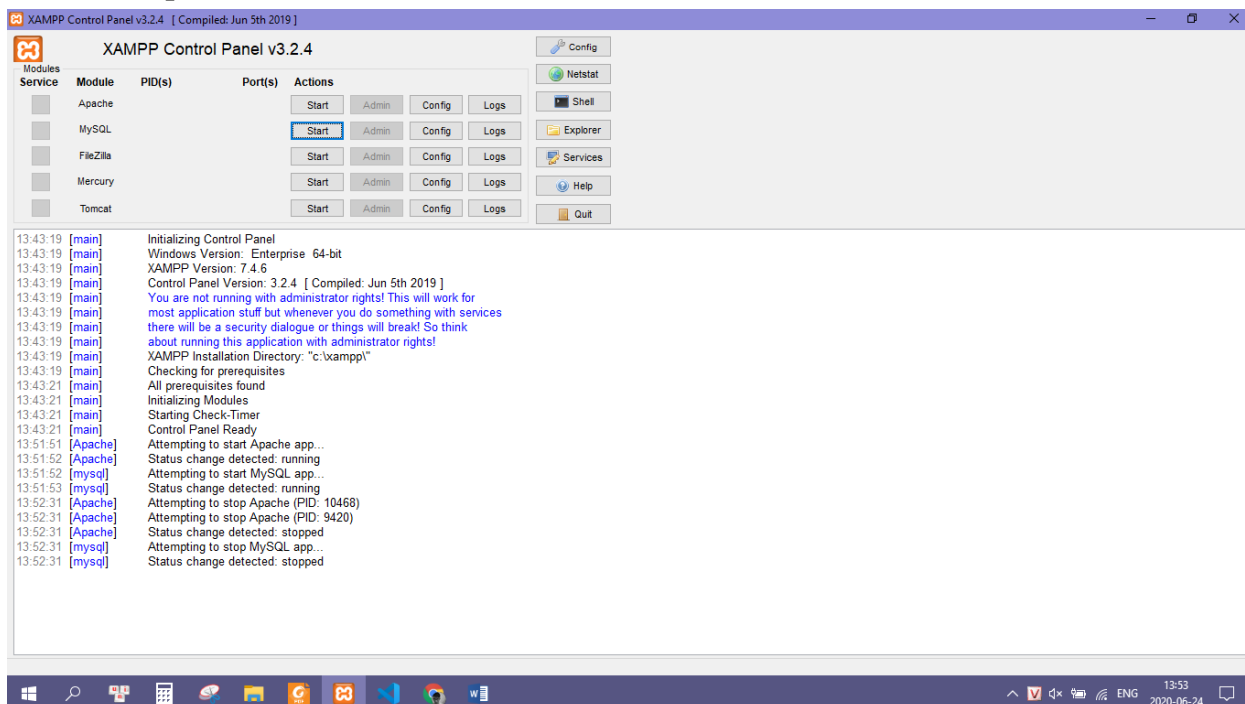
- Chương trình được viết trên Visual Studio Code:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-17: Viết chương trình trên Visual Studio Code.

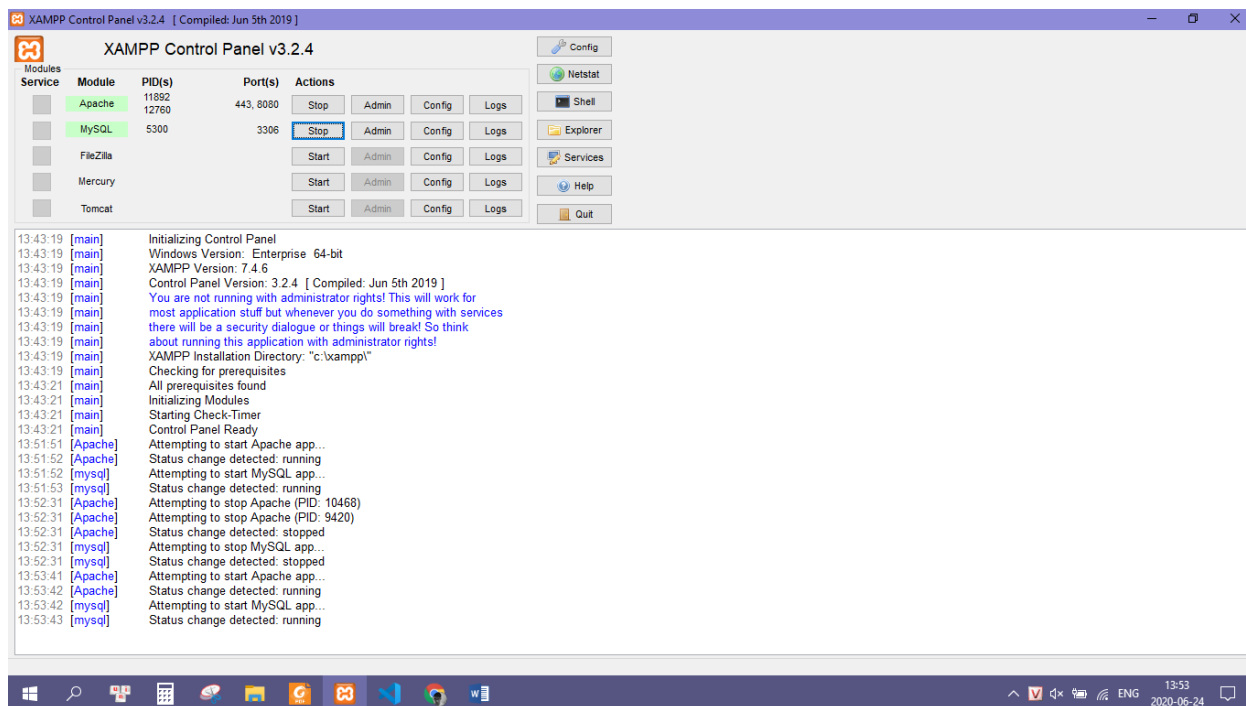
Sau đó mở phần mềm XAMPP:



Hình 4-18: Giao diện của XAMPP khi khởi chạy.

Chọn vào nút “**Start**” cho mục Apache và MySQL:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-19: Khởi chạy server bằng XAMPP.

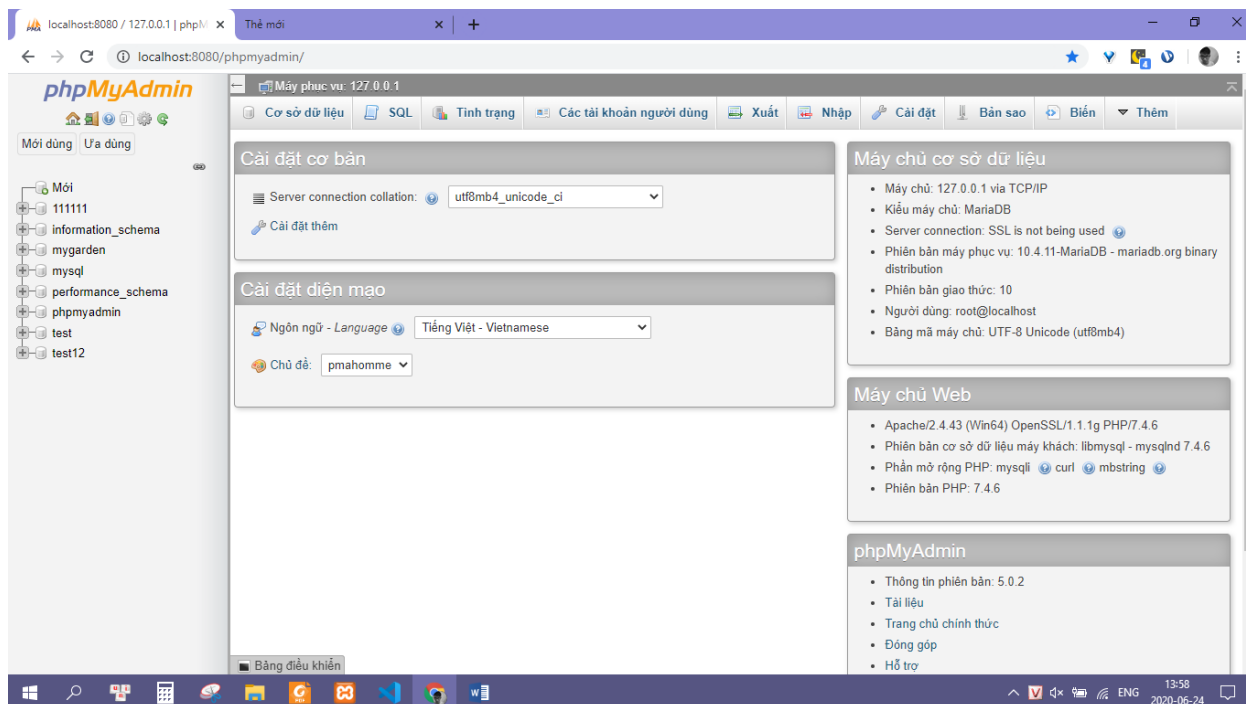
Con số “8080” chính là cổng port thứ 8080 để giao tiếp giữa máy tính và trang quản lý dữ liệu database. Có thể chọn cổng port giao tiếp tùy ý trong mục “Config” của Apache. Tiếp theo mở trình duyệt Google Chrome, truy cập vào trang web: <http://localhost:8080/>



Hình 4-20: Giao diện của trang XAMPP.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

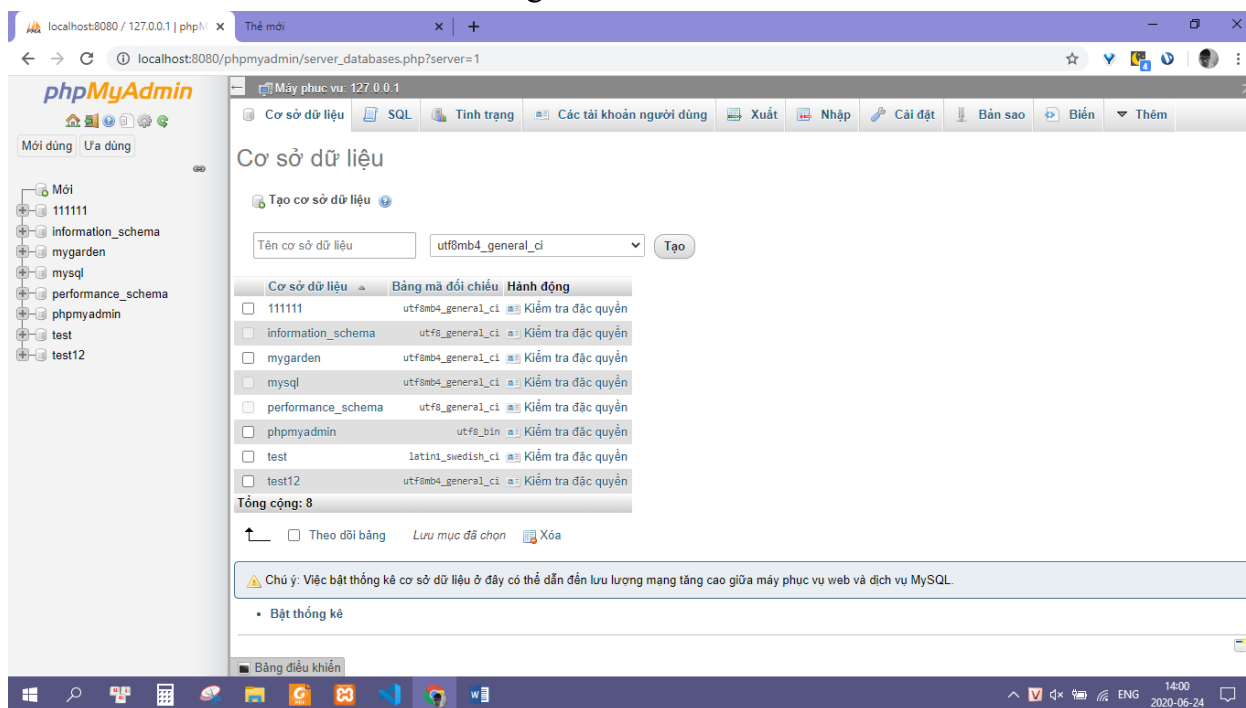
Truy cập vào localhost từ trình duyệt. Chọn vào mục “phpMyAdmin”:



Hình 4-21: Giao diện quản lý database của phpMyAdmin.

Đây là trang tạo ra cơ sở dữ liệu cho web và ứng dụng Android, quản lý các thông số cũng như trạng thái của các thiết bị.

Chọn vào mục “Mới” để tạo ra 1 vùng cơ sở dữ liệu mới:



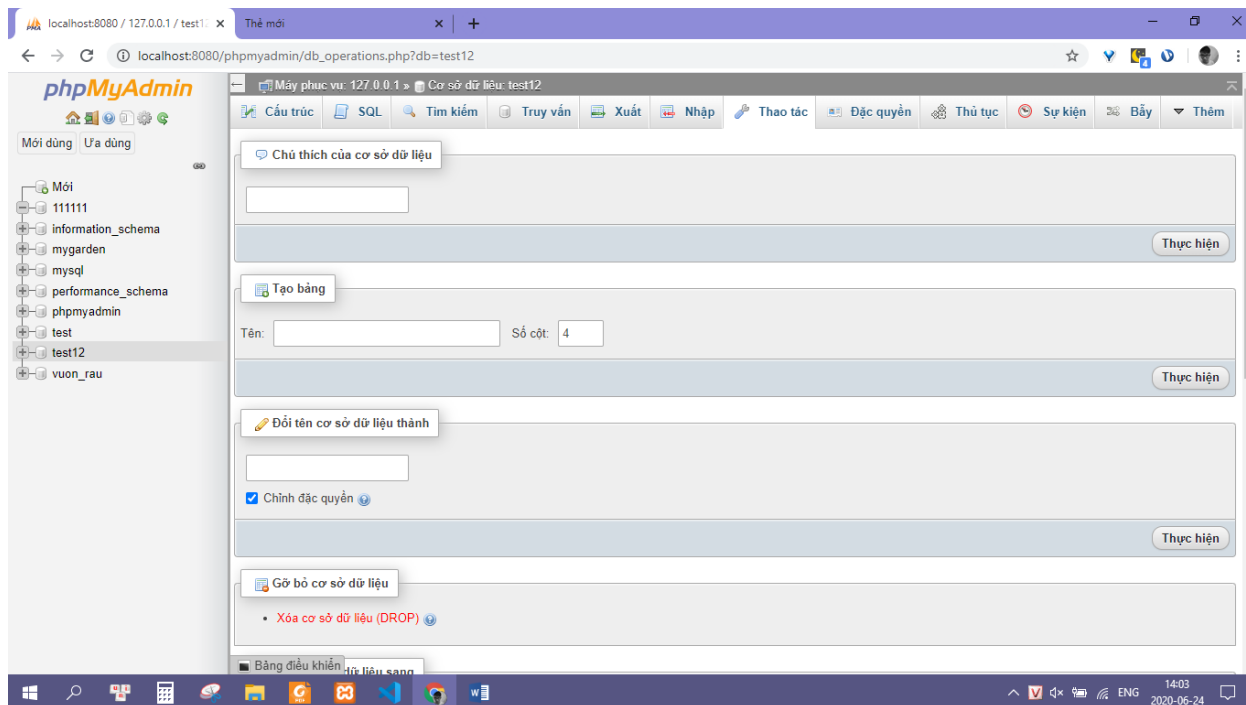
Hình 4-22: Tạo cơ sở dữ liệu đầu tiên.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Nhấn vào ô “**Tên cơ sở dữ liệu**” và nhập tên tùy ý, sau đó ấn nút “**Tạo**”.

Ở đây nhóm em tạo vùng cơ sở dữ liệu có tên là “**mygarden**” để quản lí.

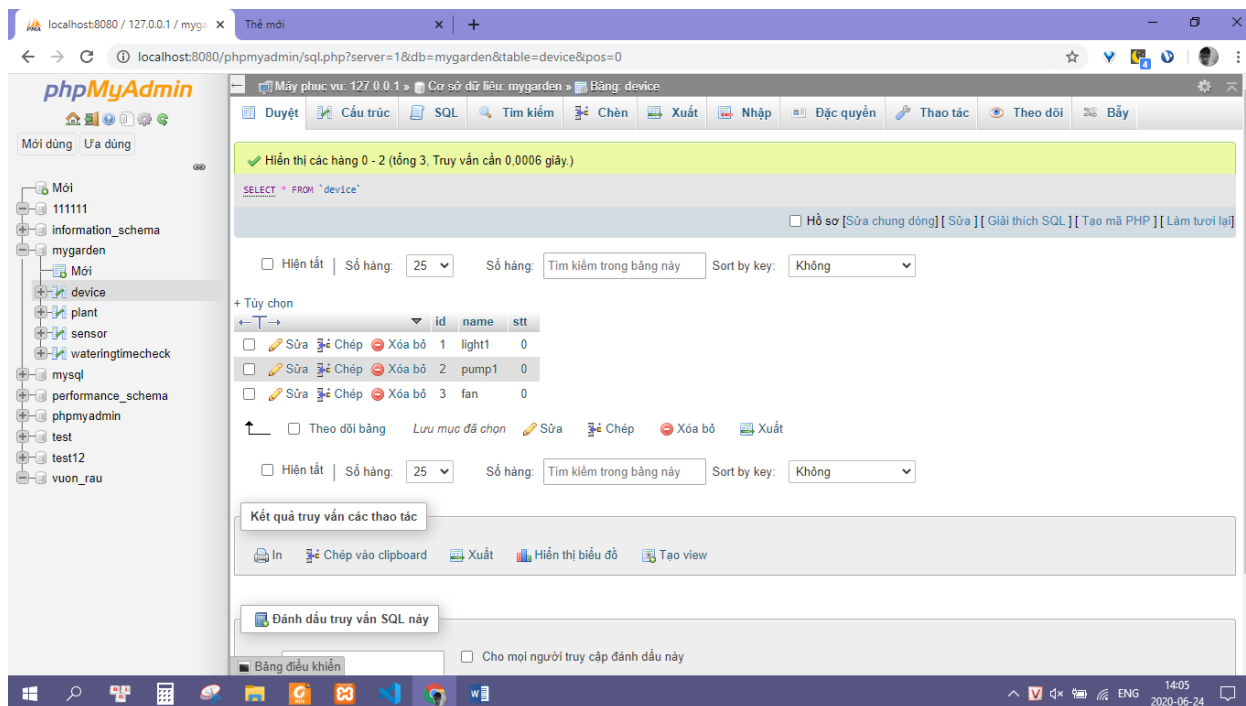
Sau khi tạo xong ta được như hình, vùng cơ sở dữ liệu “**mygarden**” đã xuất hiện:



Hình 4-23: Giao diện cơ sở dữ liệu khi mới khởi tạo xong.

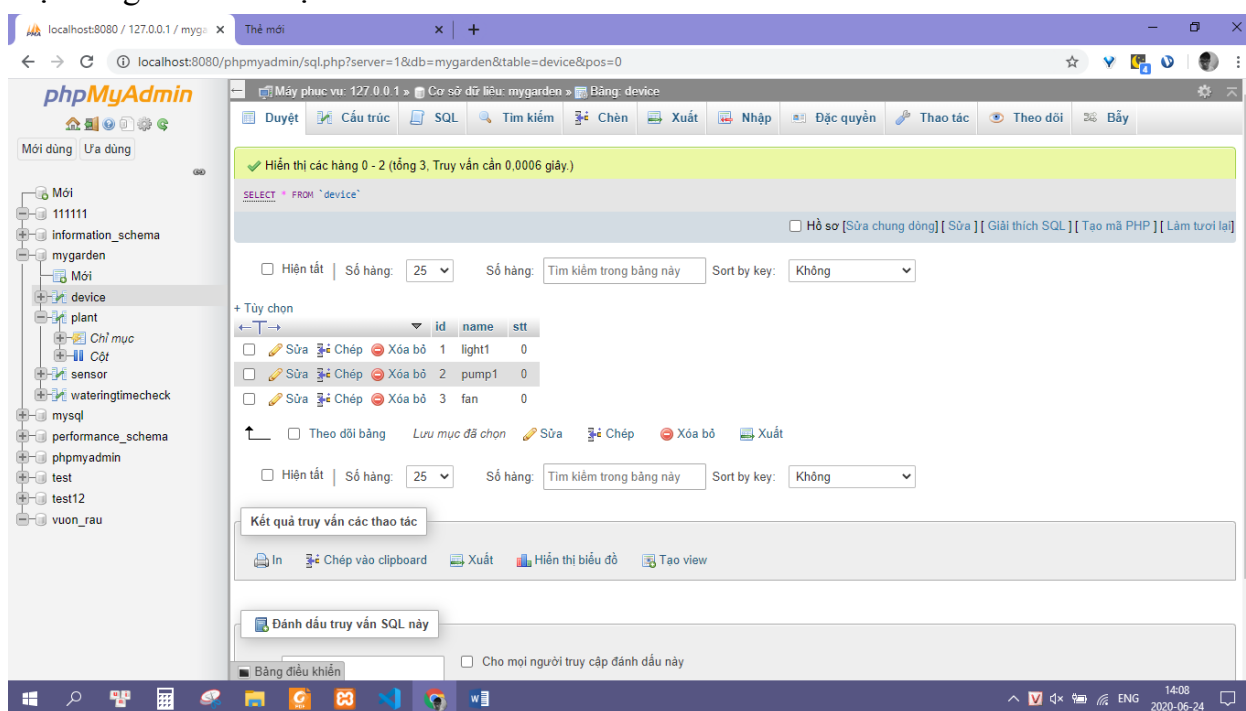
Chọn vào “**mygarden**”, sau đó chọn vào “**Mới**” để nhập các thông số liên quan:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-24: Tạo các cơ sở dữ liệu con cho cơ sở dữ liệu vừa tạo.

Đề tài sử dụng các thiết bị chấp hành như quạt, đèn, bơm. Các thiết bị này sẽ nằm trong một vùng cơ sở dữ liệu con có tên là "device".



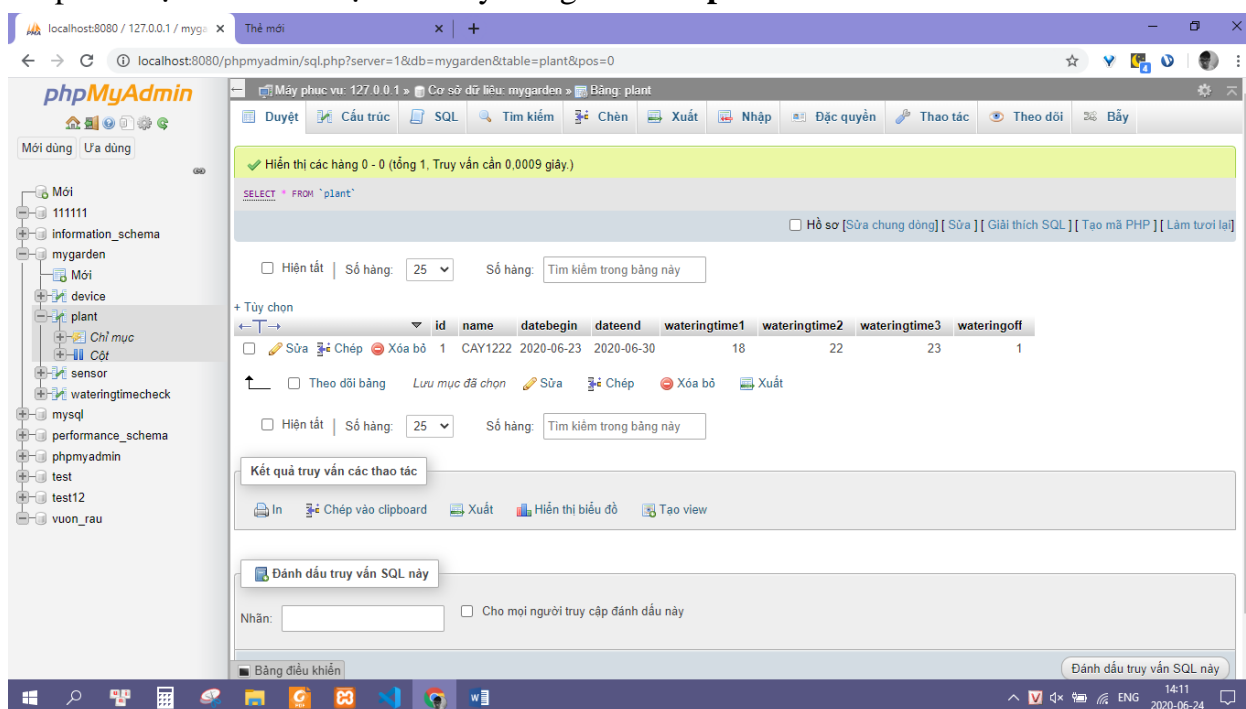
Hình 4-25: Giao diện quản lý dữ liệu cho "device".

Trong mục **device**, thiết lập 3 đối tượng cho 3 loại thiết bị đèn, bơm, quạt (2 quạt) với các tên (**name**) tương ứng là light1, pump1, fan với các **id** (mã xác định đối tượng cụ thể,

CHƯƠNG 4: THI CÔNG HỆ THỐNG

tên đối tượng có thể trùng nhau nhưng mã id thì không) tương ứng là 1, 2, 3. Trạng thái (**stt**) của các thiết bị được đặt ở 2 trạng thái là 0 (tắt) và 1 (bật).

Tiếp theo tạo cơ sở dữ liệu cho cây trồng với tên “**plant**”:



Hình 4-26: Giao diện quản lý dữ liệu cho “plant”.

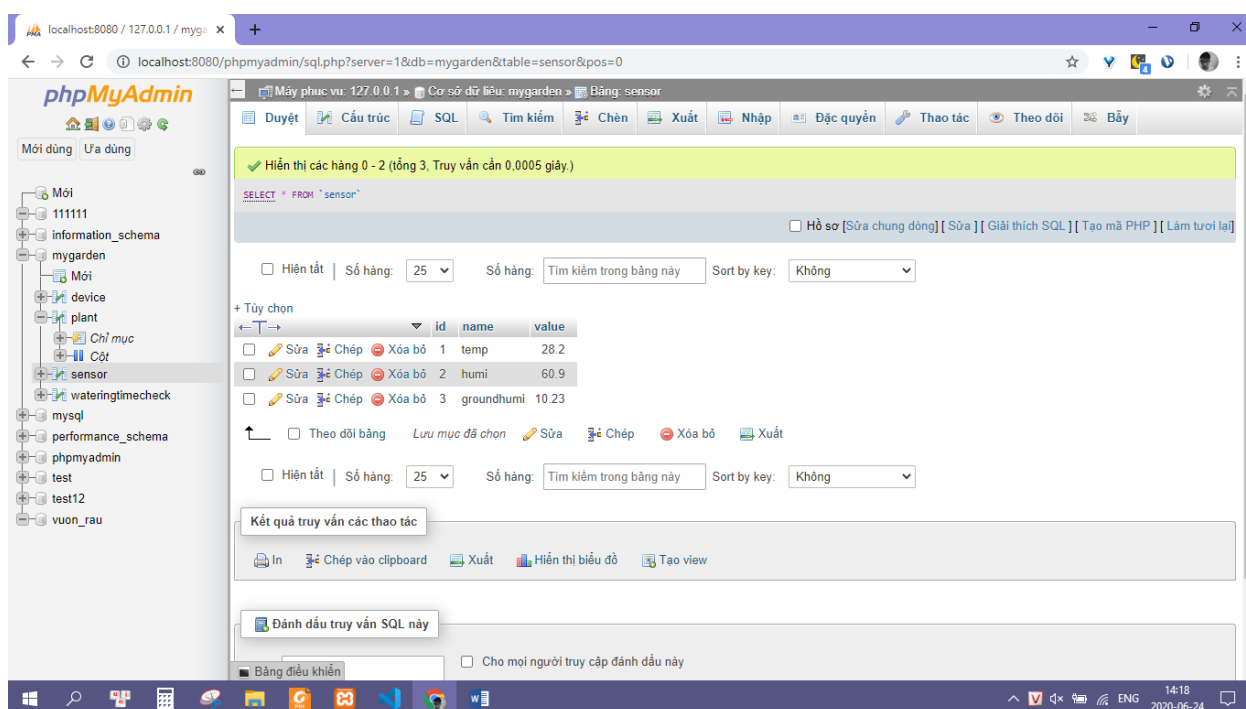
Trong mục này sẽ lưu trữ các thông tin của cây trồng sau khi người dùng cập nhật dữ liệu qua web cũng như ứng dụng Android:

- **id**: để xác định đối tượng.
- **name**: là nơi lưu trữ thông tin tên của cây trồng.
- **datebegin**: ngày trồng cây.
- **dateend**: ngày thu hoạch cây dự kiến.
- **wateringtime1**: thời điểm bật máy bơm lần 1.
- **wateringtime2**: thời điểm bật máy bơm lần 2.
- **wateringtime3**: thời điểm bật máy bơm lần 3.
- **wateringoff**: sau bao nhiêu phút thì máy bơm tự động tắt.

Tiếp theo tạo vùng cơ sở dữ liệu cho các cảm biến.

Vùng cơ sở dữ liệu này em đặt tên là “**sensor**”:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-27: Giao diện quản lý dữ liệu cho “sensor”.

Đề tài sử dụng cảm biến DHT22 để đo nhiệt độ môi trường và độ ẩm không khí. Sử dụng cảm biến độ ẩm đất để đo độ ẩm đất.

Như vậy có 3 giá trị cần quản lý đó là nhiệt độ, độ ẩm không khí và độ ẩm đất. Ba giá trị này được đặt tên tương ứng là “**temp**”, “**humi**”, “**groundhumi**” với các id lần lượt là 1, 2, 3.

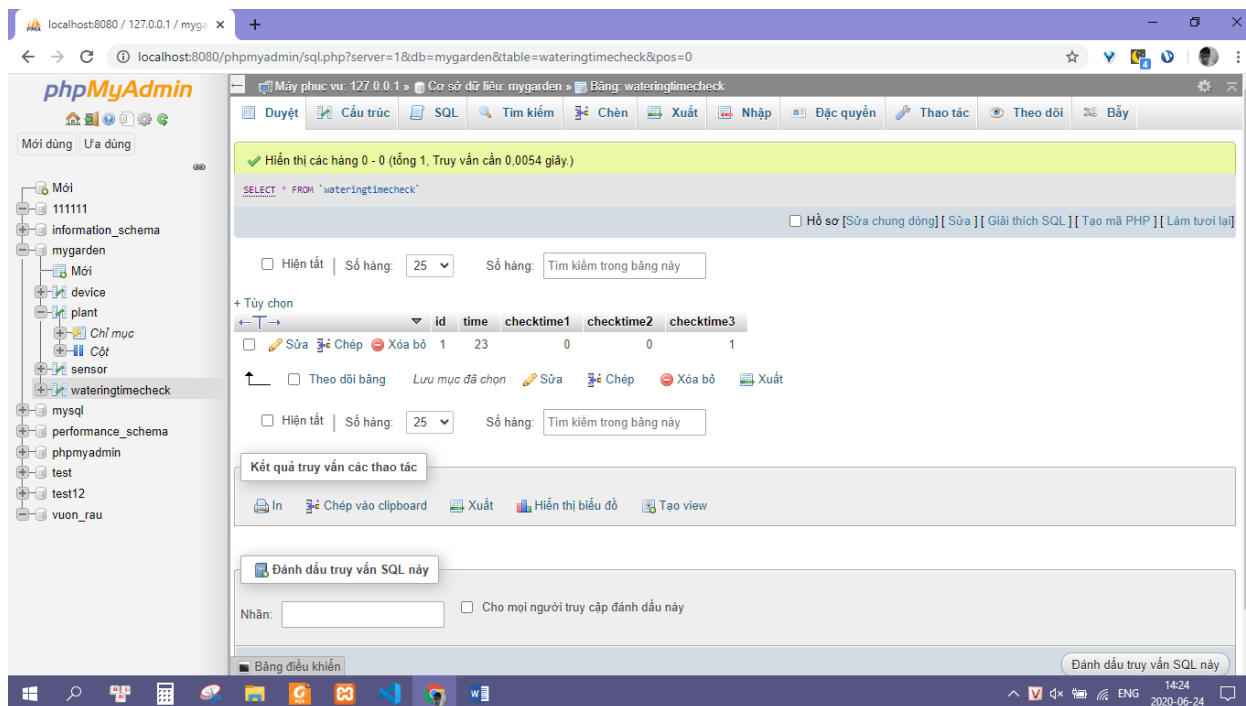
Giá trị khi đo được từ mô hình của 3 đối tượng này sẽ được gửi lên database này và được hiển thị ở cột “**value**”.

Như vậy trong cơ sở dữ liệu “**mygarden**” có 3 mục cơ sở dữ liệu con đó là “**device**”, “**plant**” và “**sensor**”.

Đến đây coi như hoàn thiện việc tạo cơ sở dữ liệu cho web và ứng dụng Android.

Tuy nhiên nếu vậy, khi demo thao tác hẹn giờ cho máy bơm sẽ tạo ra khó khăn bởi giờ tưới được thiết lập ở kiểu dữ liệu số nguyên (int). Cho nên chỉ có thể tưới vào những thời điểm giờ như 7h, 8h, 22h,... chứ không hẹn giờ tưới được ở những thời điểm như 7h 10 phút hay 9h 21 phút. Chính vì thế em sẽ tạo thêm 1 vùng cơ sở dữ liệu con có tên là “**wateringtimecheck**” để demo giờ tưới.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-28: Giao diện quản lý dữ liệu cho “wateringchecktime”.

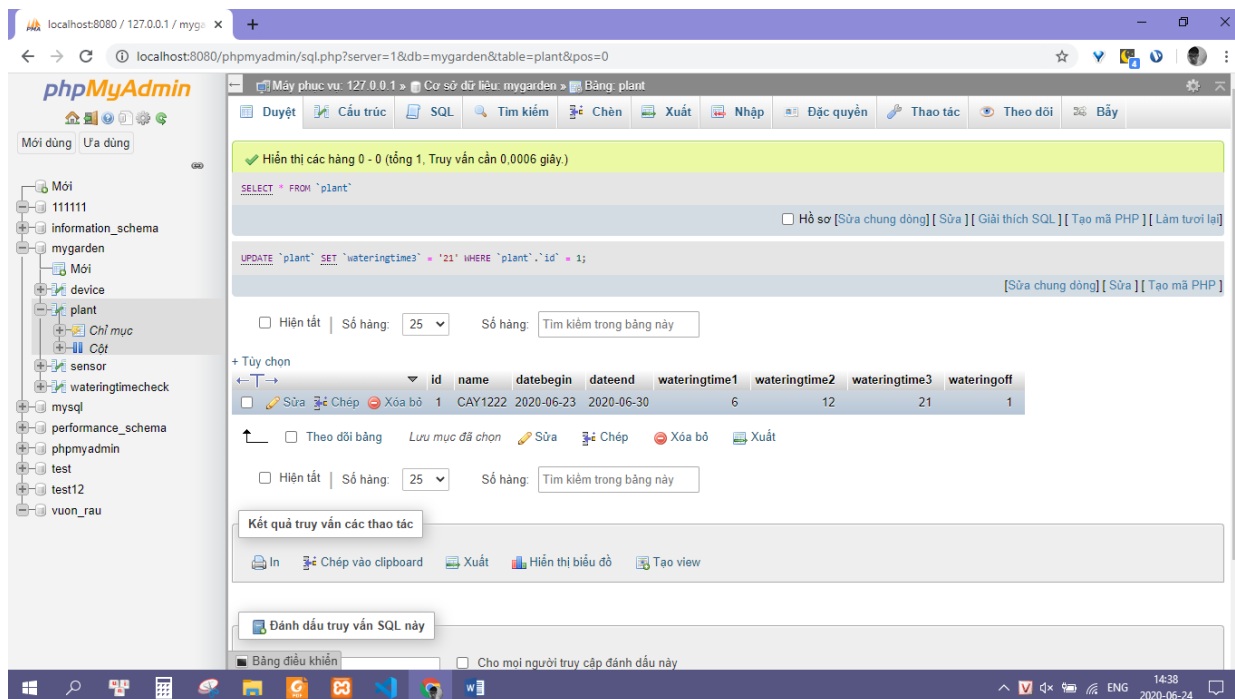
Ở mục cơ sở dữ liệu này, sẽ có 5 cột:

- **id**: xác định đối tượng (thời gian tưới).
- **checktime1**: thời điểm tưới lần thứ 1 (không nhập giá trị gì ở đây, mặc định sẽ là 0).
- **checktime2**: thời điểm tưới lần thứ 2 (không nhập giá trị gì ở đây, mặc định sẽ là 0).
- **checktime3**: thời điểm tưới lần thứ 3 (không nhập giá trị gì ở đây, mặc định sẽ là 0).
- **time**: thời điểm mà muốn demo.

Giải thích hoạt động:

Giả sử hện các thời điểm tưới trên ứng dụng cũng như web là: 6h, 12h, 21h:

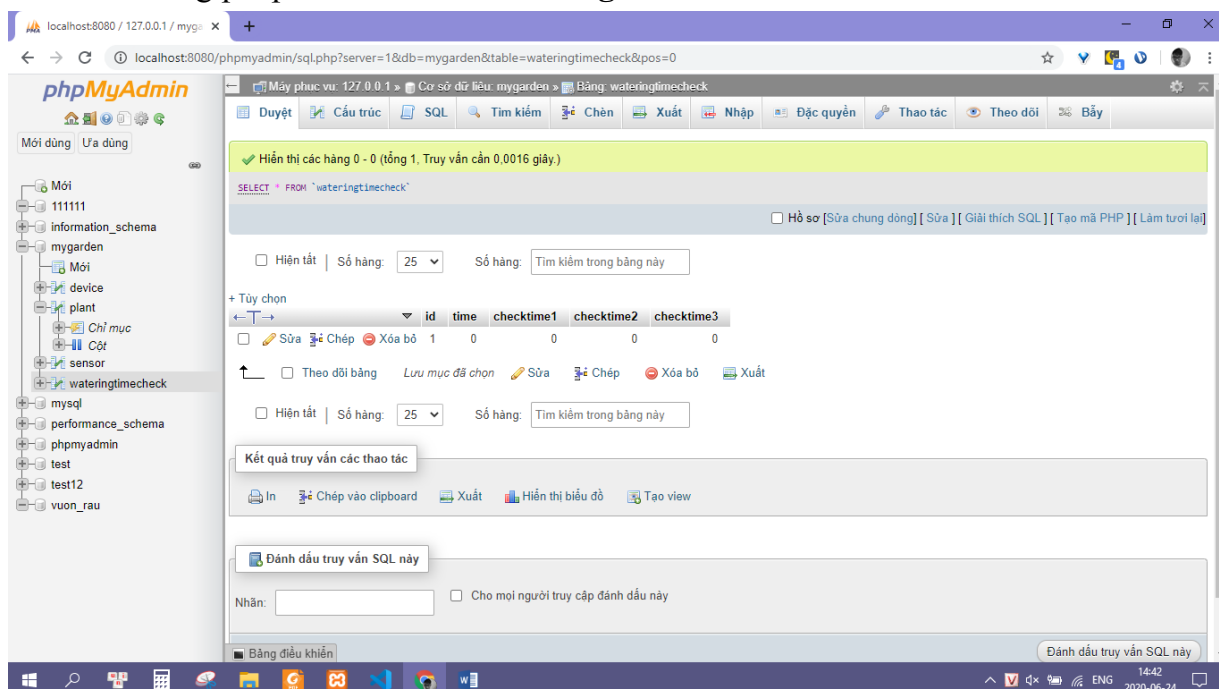
CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-29: Nhập thử thời gian tưới cho cây trên database.

Thời điểm hiện tại là 14h 38 phút, muốn biết chức năng tự động bật máy theo thời gian đã thiết lập có hoạt động hay không, có 2 phương pháp:

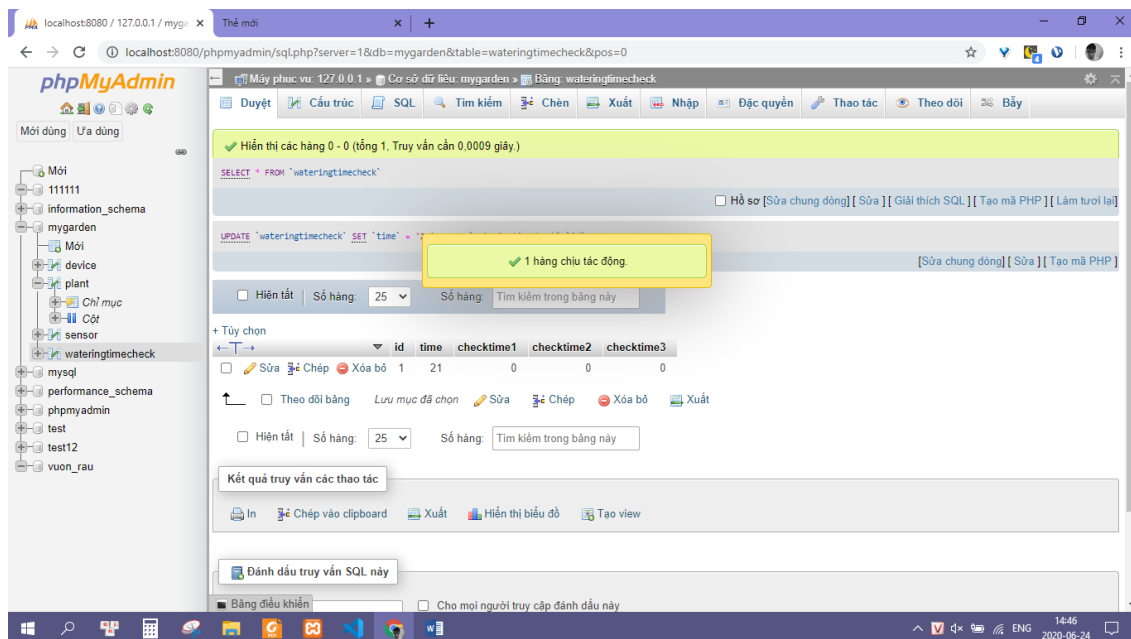
- Phương pháp 1: Chờ đến 21h.
- Phương pháp 2: Vào mục “wateringtimecheck”:



Hình 4-30: Điều chỉnh giờ để kiểm tra hệ thống tưới.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Chọn vào mục “time” và nhập “21” và ấn phím Enter trên bàn phím máy tính.

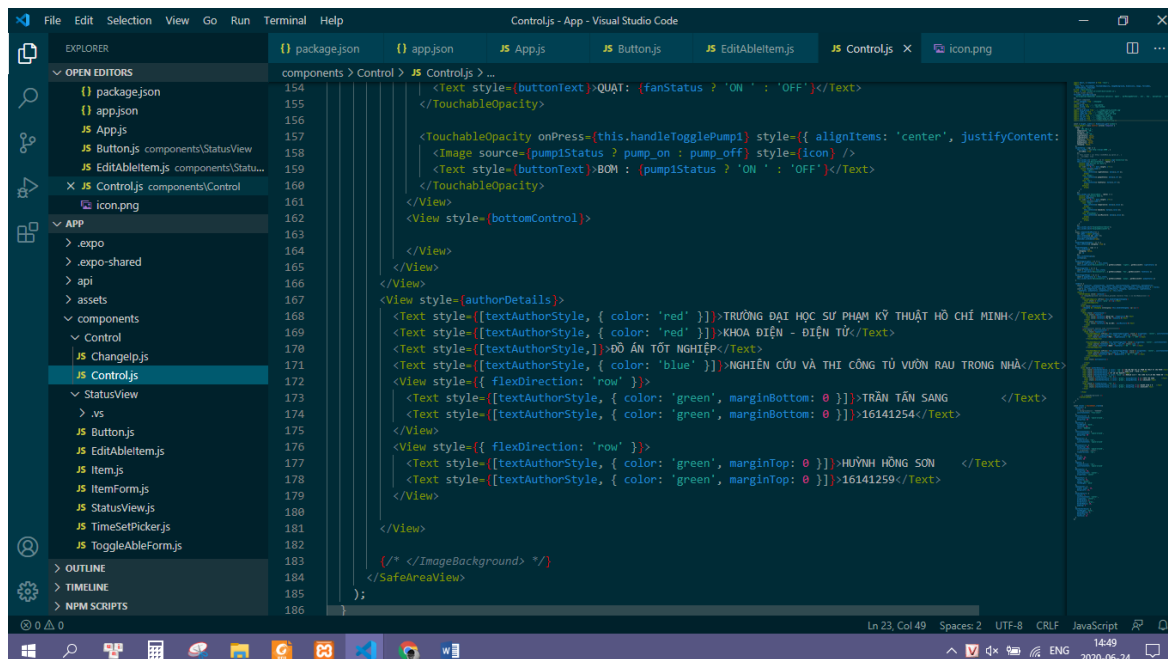


Hình 4-31: Giao diện sau khi đã nhập giờ để kiểm tra tưới.

Sau đó 1 phút thì máy bơm sẽ hoạt động, trạng thái của máy bơm sẽ được hiển thị trên mô hình, trên ứng dụng Android cũng như web.

Xây dựng ứng dụng Android:

Chương trình được viết trên Visual Studio Code:



Hình 4-32: Viết chương trình xây dựng ứng dụng Android trên Visual Studio Code.

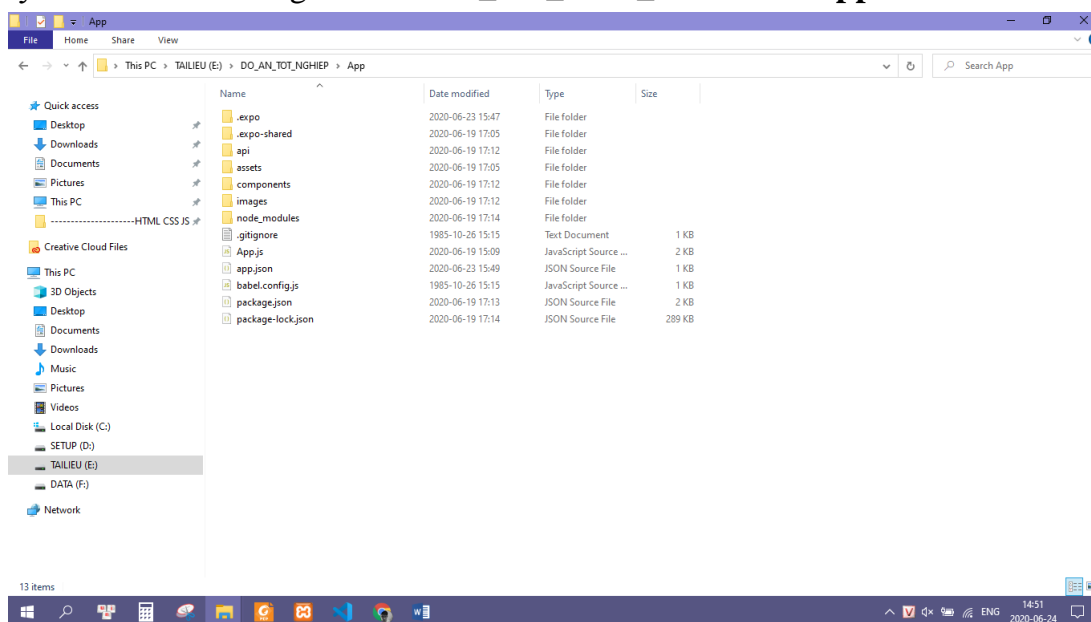
CHƯƠNG 4: THI CÔNG HỆ THỐNG

Sau khi viết xong chương trình, em sẽ chạy mô phỏng ứng dụng Android. Nếu không có lỗi thì sẽ build ra file .apk và tải về sau đó cài đặt.

Cách mô phỏng sau khi viết xong chương trình:

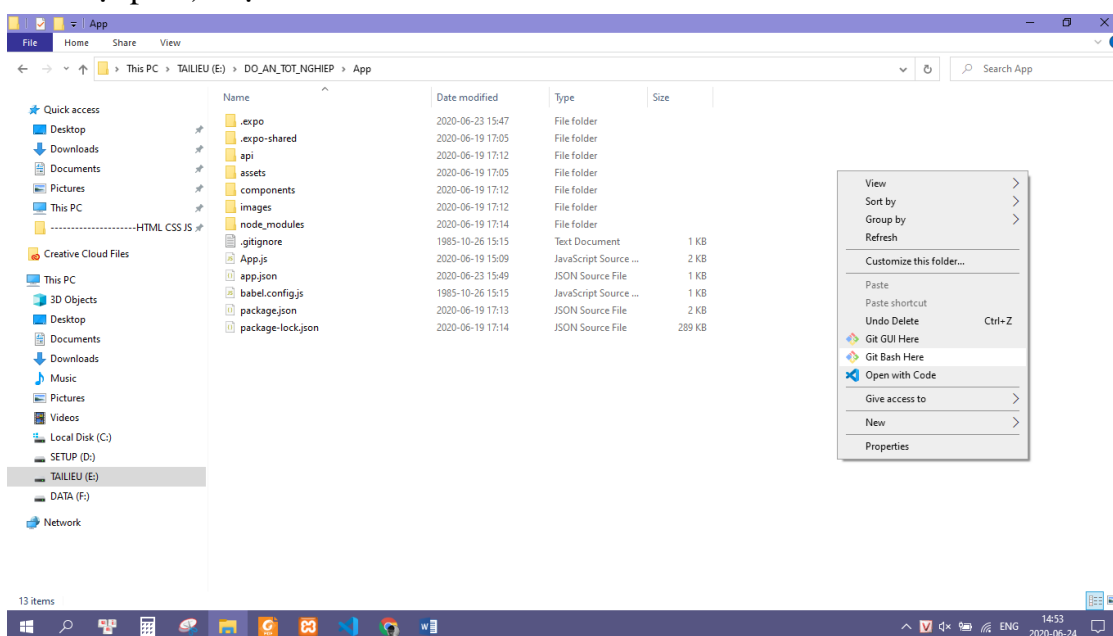
Vào thư mục chứa toàn bộ chương trình (code) trên máy tính.

Ở đây em lưu theo đường dẫn: **E>DO_AN_TOT_NGHIEP>App**



Hình 4-33: Truy cập vào thư mục chứa dữ liệu chương trình.

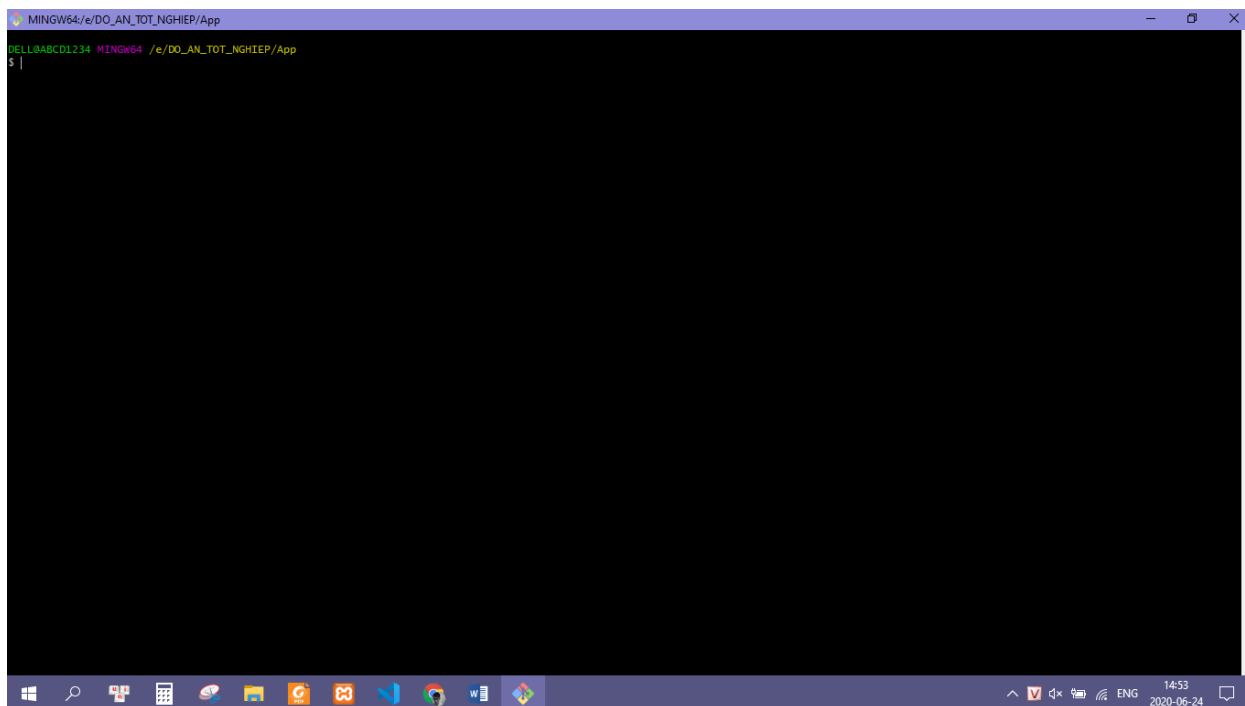
Click chuột phải, chọn: “**Git Bash Here**”.



Hình 4-34: Mở Git Bash Here.

Ô cửa sổ dòng lệnh của GitBash hiện ra:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-35: Cửa sổ Git Bash Here hiện ra.

Gõ “**expo start—android**” và nhấn ENTER.

Đợi khoảng 20 giây thì sẽ có một mã QR như hình:



Hình 4-36: Mã QR hiện ra để chạy thử ứng dụng.

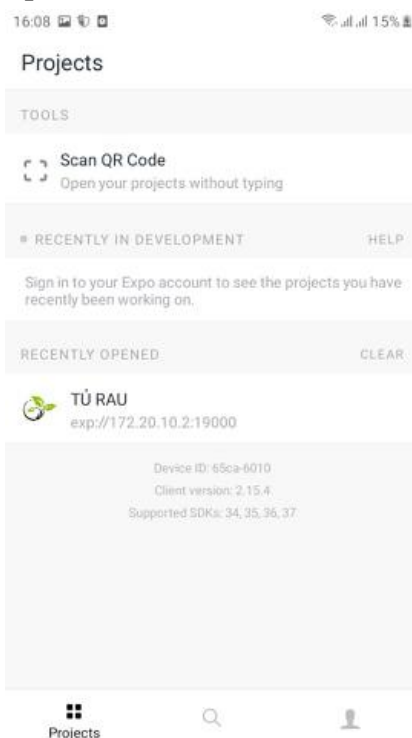
Sau đó mở điện thoại Android, tìm và cài đặt phần mềm có tên “**EXPO**” như hình.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



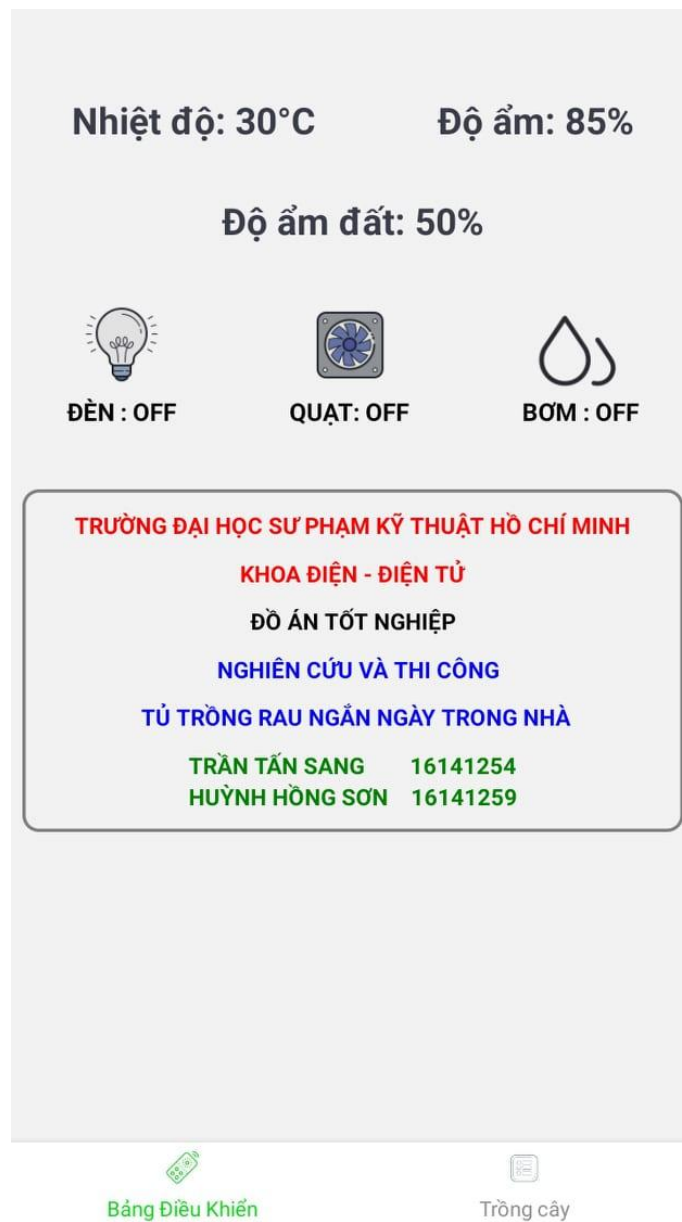
Hình 4-37: Tải và cài đặt ứng dụng EXPO trên Google Play.

Mở ứng dụng EXPO vừa tải, quét mã QR:



Hình 4-38: Khởi chạy EXPO.

Đợi 1 lúc và chúng ta đã có thể mô phỏng, hình dung ra ứng dụng Android được viết có giao diện, chức năng như thế nào.



Hình 4-39: Giao diện tab “Bảng điều khiển”.

Thêm cây trồng

Tên cây trồng:
Cà chua...

Ngày trồng: Dự kiến thu hoạch:
[] []

Hẹn giờ: -- ▾ -- ▾ -- ▾

Tắt sau: 1 ▾

Thêm **Hủy**

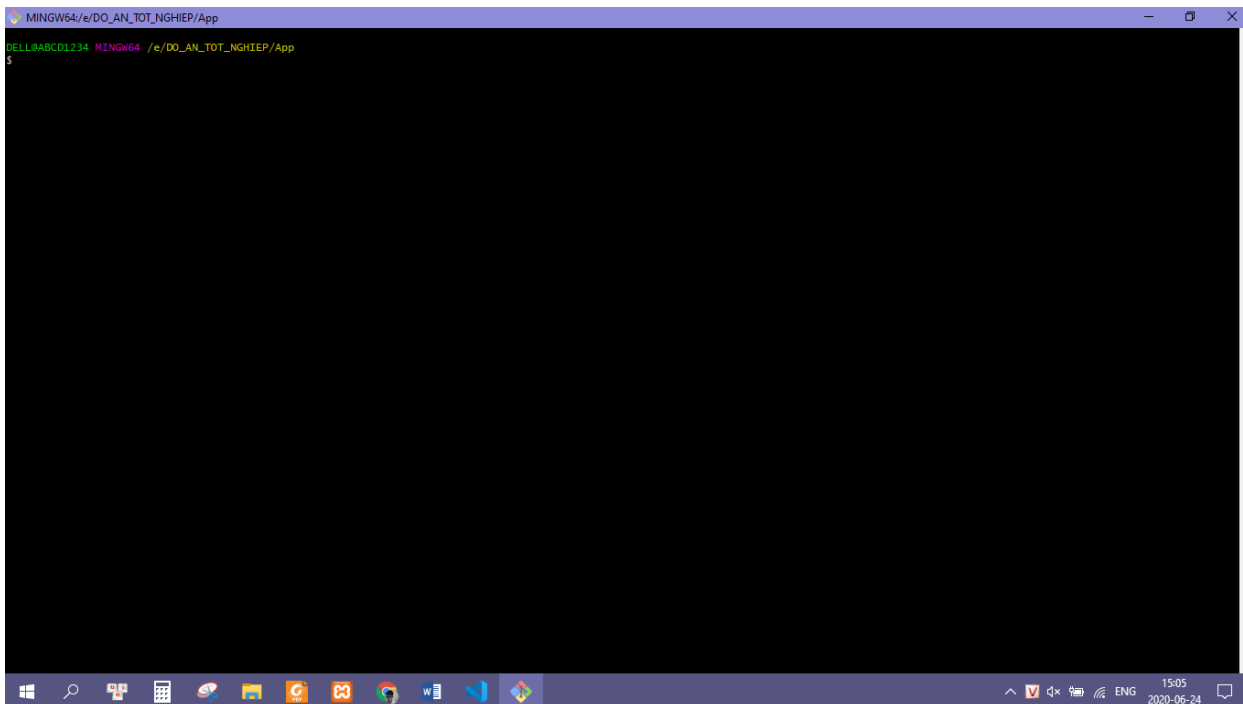
 Bảng Điều Khiển  Trồng cây

Hình 4-40: Giao diện tab “Trồng cây”.

Tạo ra file apk để tải về và cài đặt:

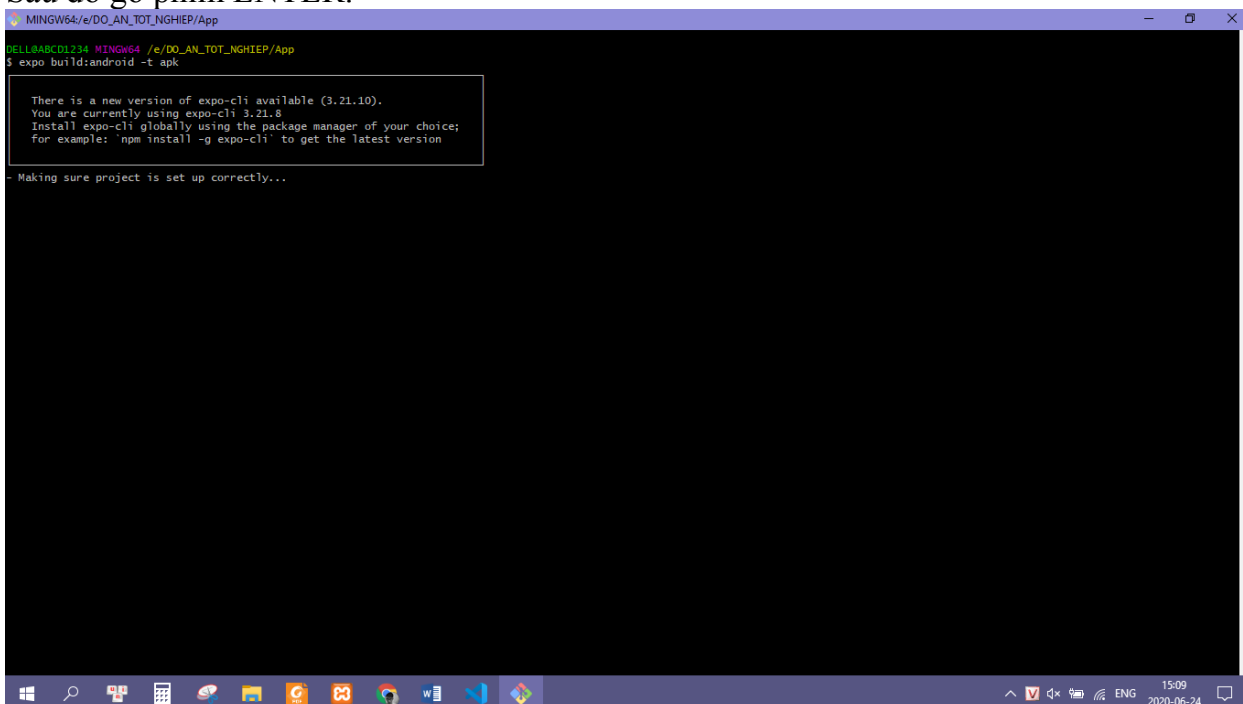
Sau khi đã mô phỏng và chỉnh sửa lại chương trình (nếu có), khi thấy ứng dụng đã hoạt động theo yêu cầu; chúng ta sẽ xuất ra file .apk để tải về và cài đặt cho điện thoại. Mở lại Git Bash Here:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-41: Khởi chạy Git Bash Here.

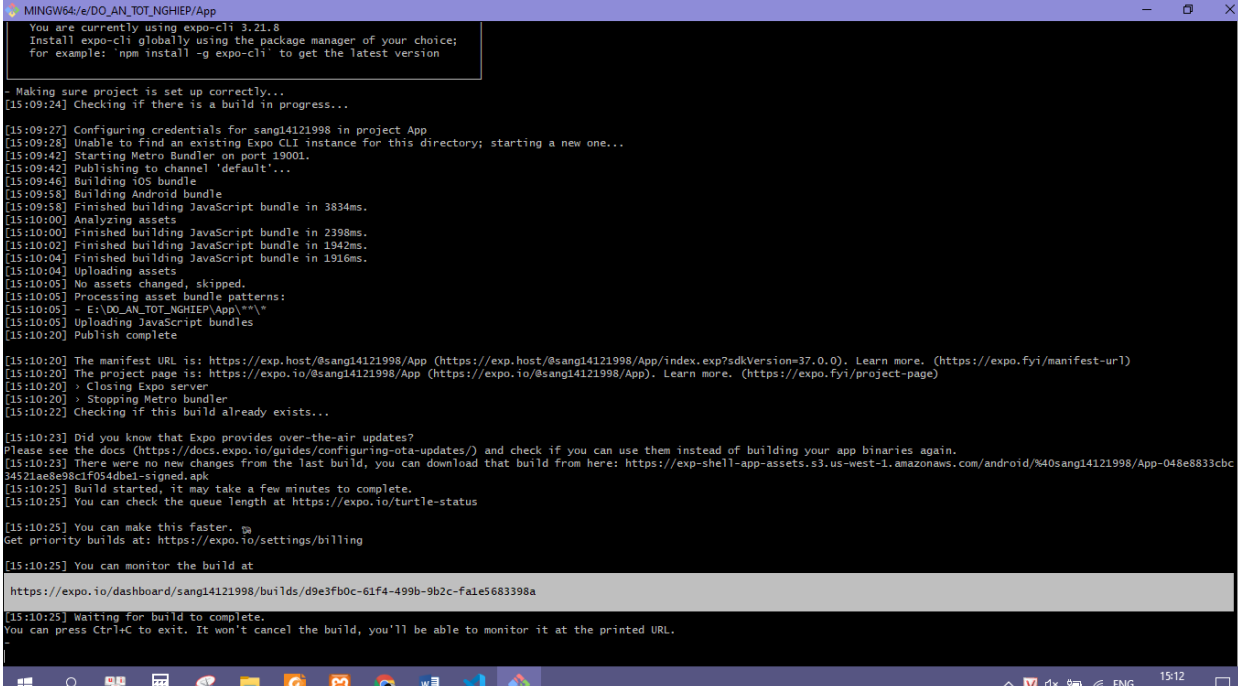
Nhập dòng lệnh: **expo build:android -t apk**
Sau đó gõ phím ENTER.



Hình 4-42: Chạy dòng lệnh để tạo file apk.

Sau đó đợi khi hiện ra thông báo có 1 đường dẫn liên kết để tải về như hình:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



```
MINGW64/e/DO_AN_TOT_NGHIEP/App
You are currently using expo-cli 3.21.8
Install expo-cli globally using the package manager of your choice;
For example: 'npm install -g expo-cli' to get the latest version

-Making sure project is set up correctly...
[15:09:24] Checking if there is a build in progress...
[15:09:27] Configuring credentials for sang14121998 in project App
[15:09:28] Unable to Find an existing Expo CLI instance for this directory; starting a new one...
[15:09:42] Starting Metro Bundler on port 19001.
[15:09:42] Publishing to channel 'default'...
[15:09:46] Building iOS bundle
[15:09:53] Building Android bundle
[15:09:58] Finished building JavaScript bundle in 3834ms.
[15:10:00] Analyzing assets
[15:10:00] Finished building JavaScript bundle in 2398ms.
[15:10:02] Finished building JavaScript bundle in 1942ms.
[15:10:04] Finished building JavaScript bundle in 1916ms.
[15:10:04] Uploading assets
[15:10:05] No assets changed, skipped.
[15:10:05] Processing asset bundle patterns:
[15:10:05] - E:\DO_AN_TOT_NGHIEP\App\**\*
[15:10:05] Uploading JavaScript bundles
[15:10:20] Publish complete

[15:10:20] The manifest URL is: https://exp.host/@sang14121998/App (https://exp.host/@sang14121998/App/index.exp?sdkVersion=37.0.0). Learn more. (https://expo.fyi/manifest-url)
[15:10:20] The project page is: https://expo.io/@sang14121998/App (https://expo.io/@sang14121998/App). Learn more. (https://expo.fyi/project-page)
[15:10:20] > Closing Expo server
[15:10:20] > Stopping Metro bundler
[15:10:22] Checking if this build already exists...

[15:10:23] Did you know that Expo provides over-the-air updates?
Please see the docs (https://docs.expo.io/guides/configuring-ota-updates/) and check if you can use them instead of building your app binaries again.
[15:10:23] There were no new changes from the last build, you can download that build from here: https://exp-shell-app-assets.s3.us-west-1.amazonaws.com/android/%40sang14121998/App-048e8833cbc34521a8e98c1f054dbe1-signed.apk
[15:10:25] Build started, it may take a few minutes to complete.
[15:10:25] You can check the queue length at https://expo.io/turtle-status

[15:10:25] You can make this faster.
Get priority builds at: https://expo.io/settings/billing

[15:10:25] You can monitor the build at
https://expo.io/dashboard/sang14121998/builds/d9e3fb0c-61f4-499b-9b2c-fa1e5683398a

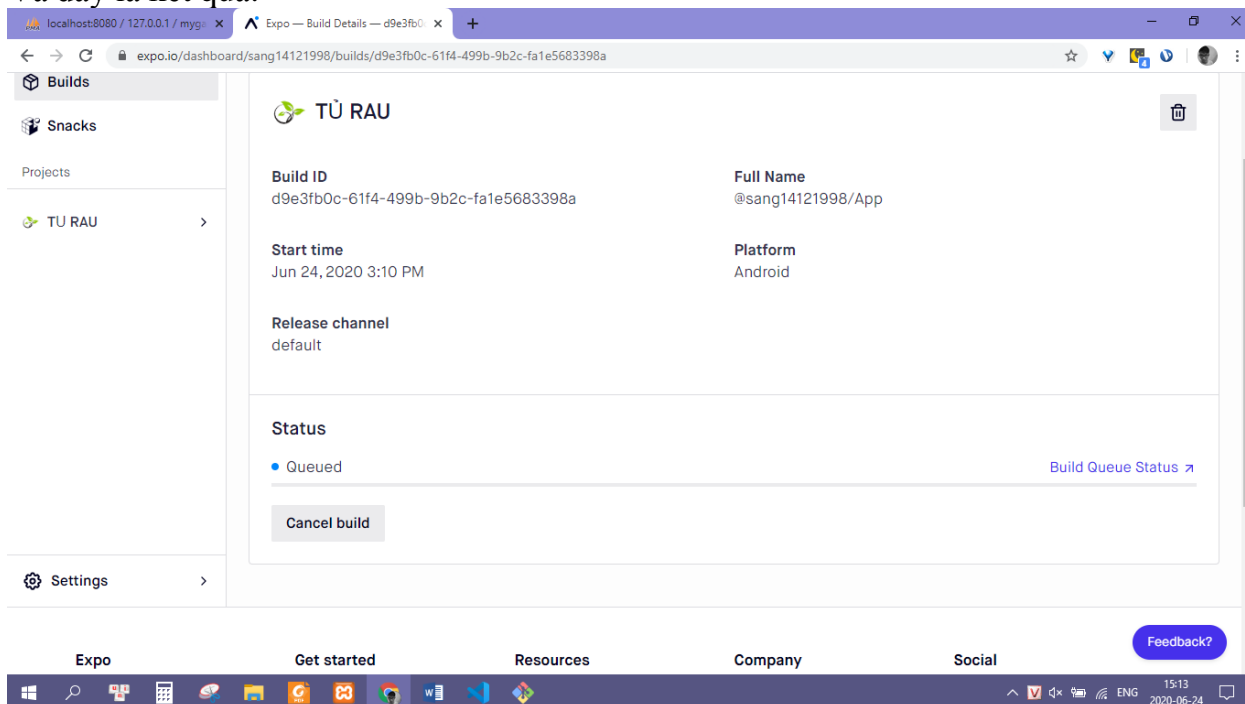
[15:10:25] Waiting for build to complete.
You can press Ctrl+C to exit. It won't cancel the build, you'll be able to monitor it at the printed URL.
```

Hình 4-43: Đường dẫn tải về xuất hiện.

Copy đường dẫn vào trình duyệt, sau đó tải về.

Lưu ý trước đó, khi cài đặt EXPO thì chúng ta cũng đồng thời tạo 1 tài khoản EXPO rồi.

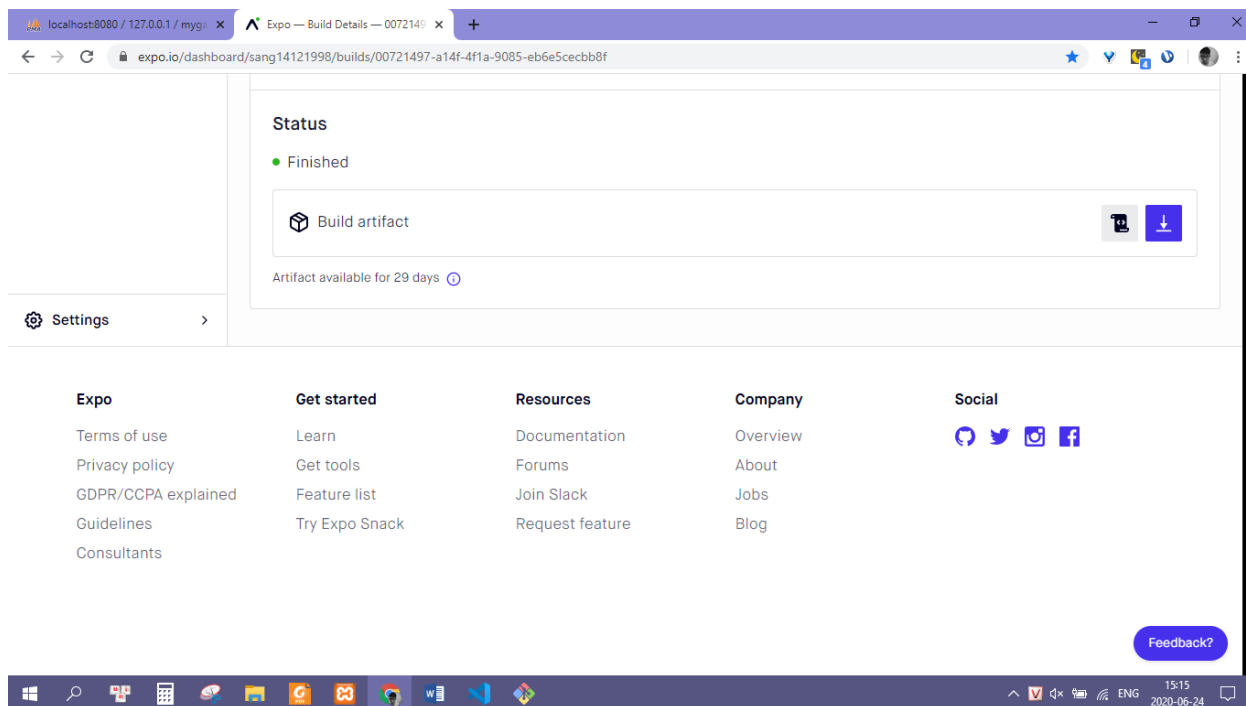
Và đây là kết quả:



Hình 4-44: Ứng dụng đang được tạo trên server của EXPO.

Đợi khoảng 30 phút thì sẽ có file .apk, sau đó ấn vào biểu tượng mũi tên tải về và cài đặt:

CHƯƠNG 4: THI CÔNG HỆ THỐNG

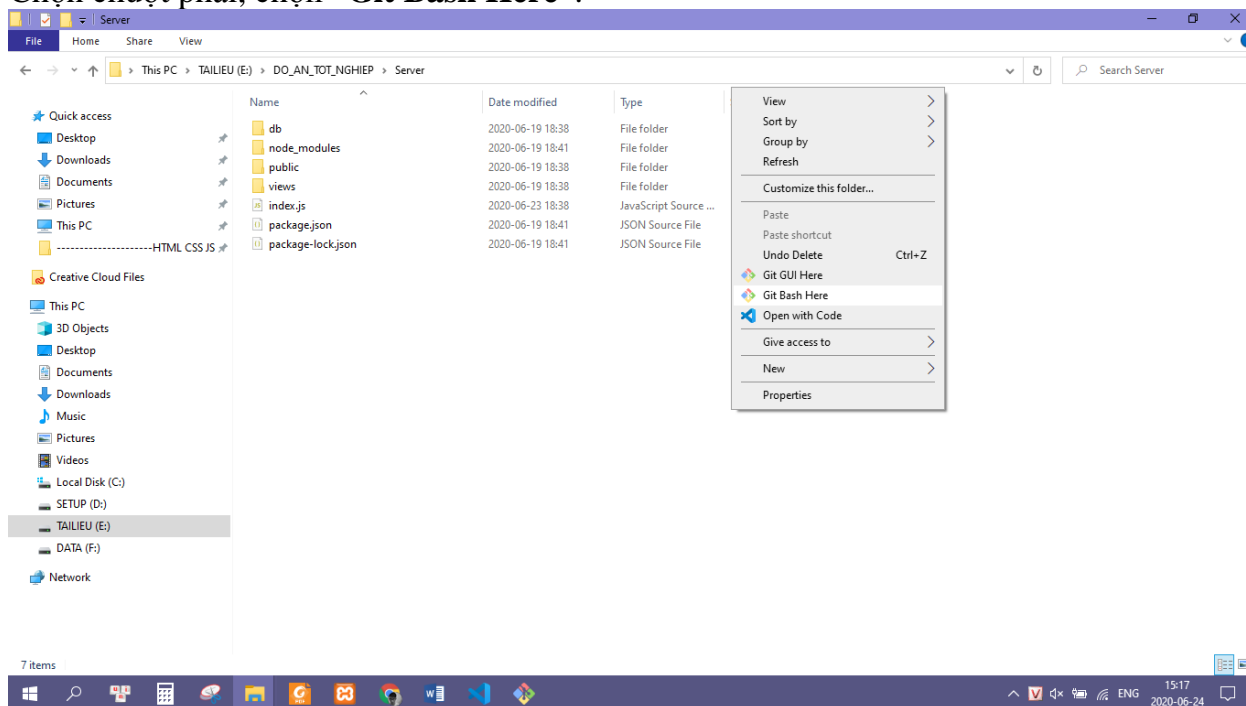


Hình 4-45: Tập tin cài đặt đã được tạo thành công.

Khởi chạy server:

Vào thư mục chứa file chương trình của web cũng như server.

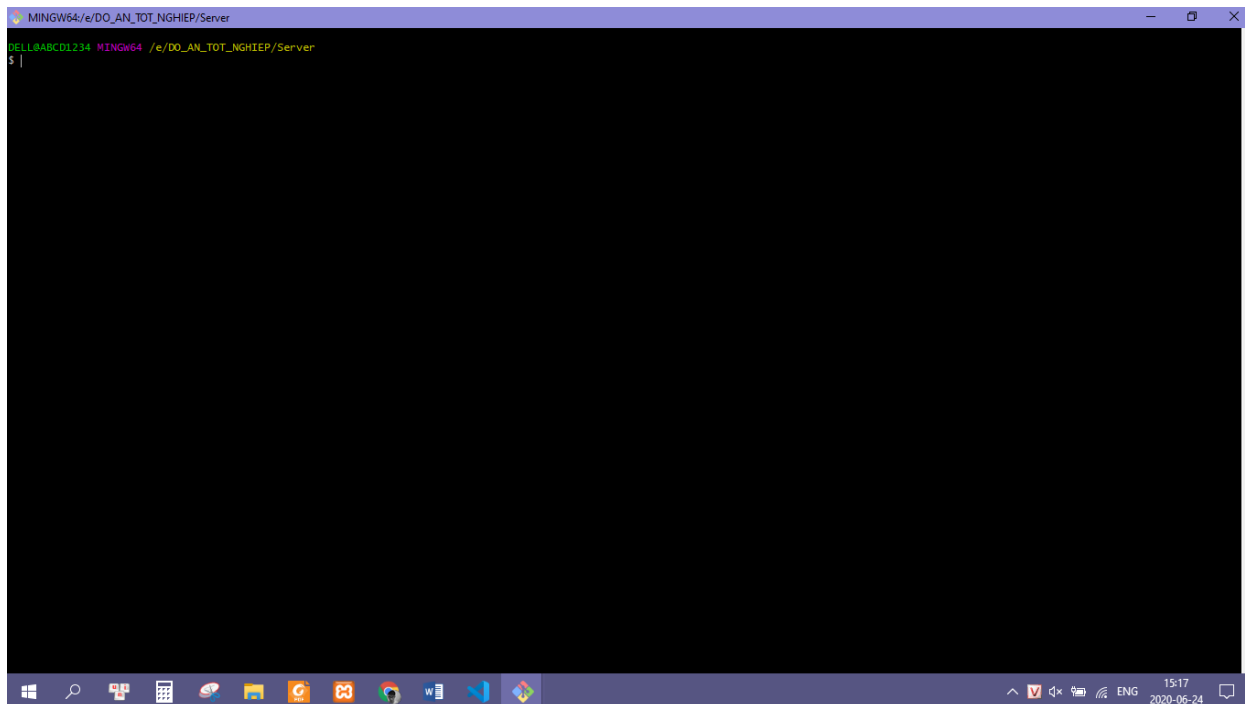
Chọn chuột phải, chọn “Git Bash Here”:



Hình 4-46: Khởi chạy Git Bash.

Giao diện cửa sổ của GitBash hiện ra:

CHƯƠNG 4: THI CÔNG HỆ THỐNG

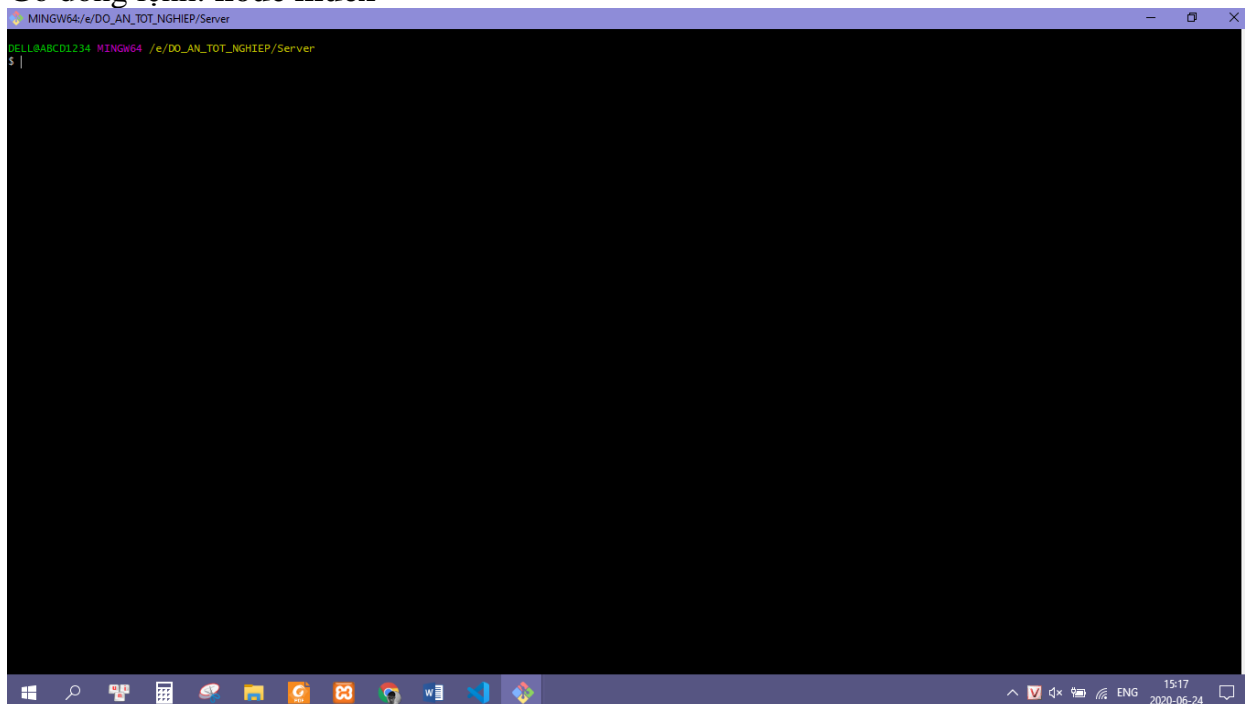


Hình 4-47: Giao diện của Git Bash.

Bây giờ em sẽ khởi chạy chương trình chính.

File chương trình này trong Visual Studio được em đặt tên (tùy ý) là index.js

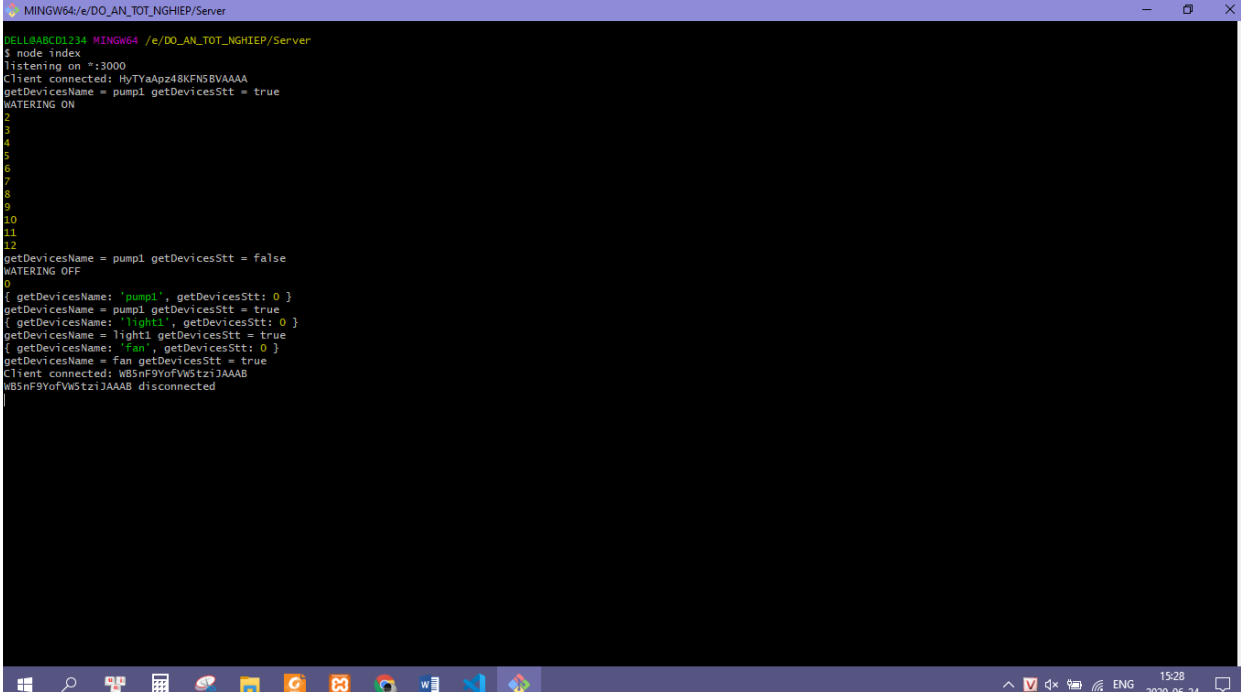
Gõ dòng lệnh: **node index**



Hình 4-48: Khởi chạy server.

Và ấn ENTER:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



```
MINGW64/e/DO_AN_TOT_NGHIEP/Server
DELLBABC01234 MINGW64 /e/DO_AN_TOT_NGHIEP/Server
$ node index
Listening on *:3000
Client connected: HyTYApz48KFNSBVAAAA
getDevicesName = pump1 getDevicesStt = true
WATERING ON
1
2
3
4
5
6
7
8
9
10
11
12
getDevicesName = pump1 getDevicesStt = false
WATERING OFF
0
{ getDevicesName: 'pump1', getDevicesStt: 0 }
getDevicesName = pump1 getDevicesStt = true
{ getDevicesName: 'light1', getDevicesStt: 0 }
getDevicesName = light1 getDevicesStt = true
{ getDevicesName: 'fan', getDevicesStt: 0 }
getDevicesName = fan getDevicesStt = true
Client connected: WBSnF9YofVw5tzjJAAAB
WBSnF9YofVw5tzjJAAAB disconnected
```

Hình 4-49: Thông báo đã khởi chạy thành công server.

Chú ý dòng lệnh: **listening on*:3000**

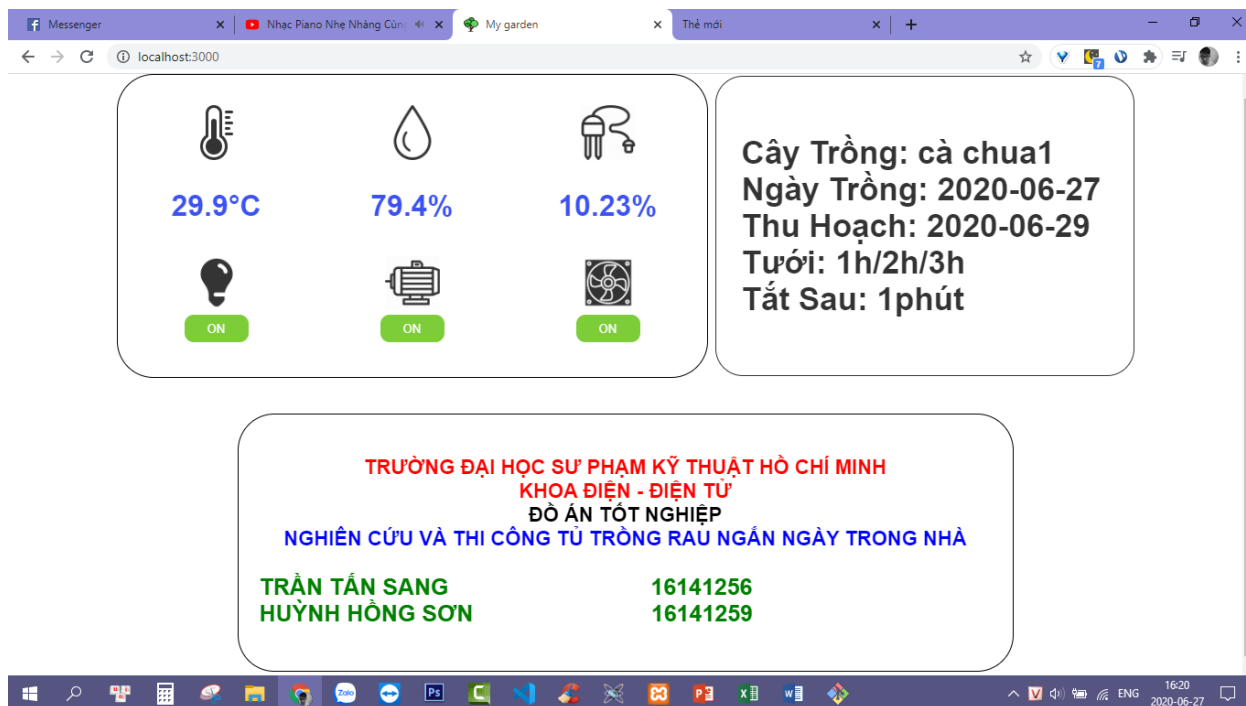
Số **3000** là địa chỉ port (được thiết lập trong chương trình viết trên Visual Studio Code) để giao tiếp giữa server dữ liệu trên phpMyAdmin với ứng dụng Android và web.

Khởi chạy web điều khiển:

Truy cập vào trang: **localhost:3000**

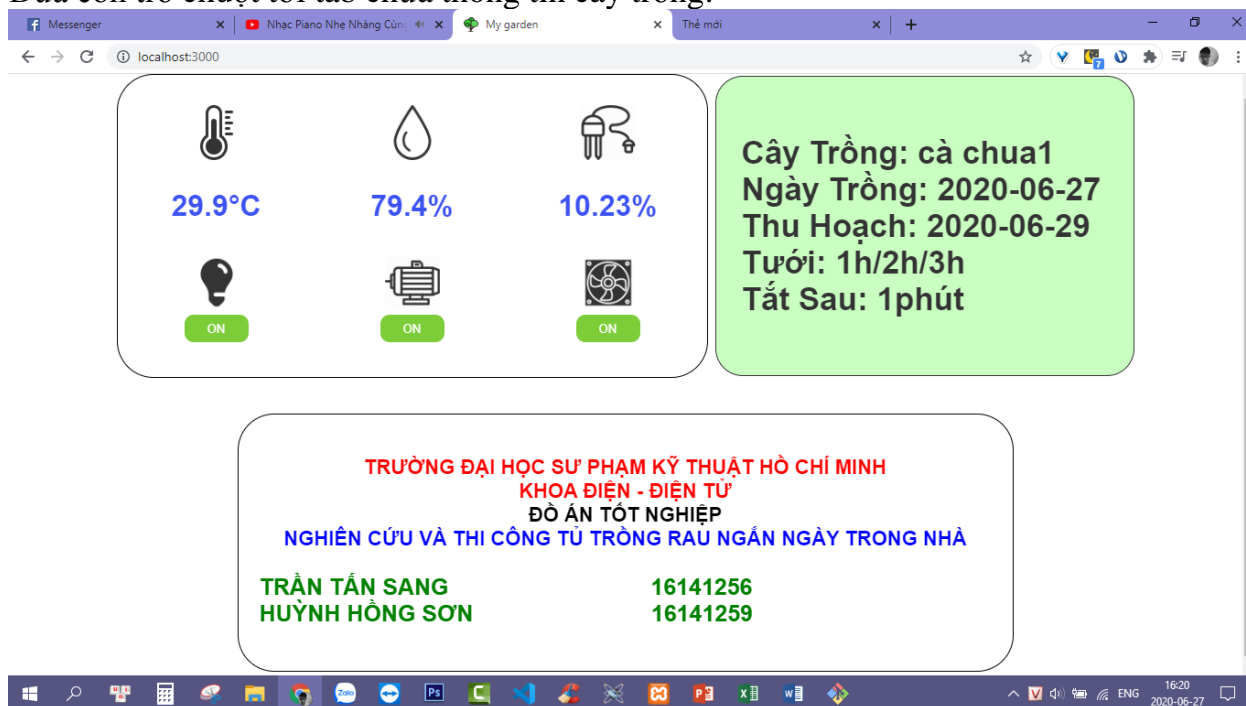
Giao diện điều khiển web lúc này hiện ra:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-50: Giao diện của web điều khiển.

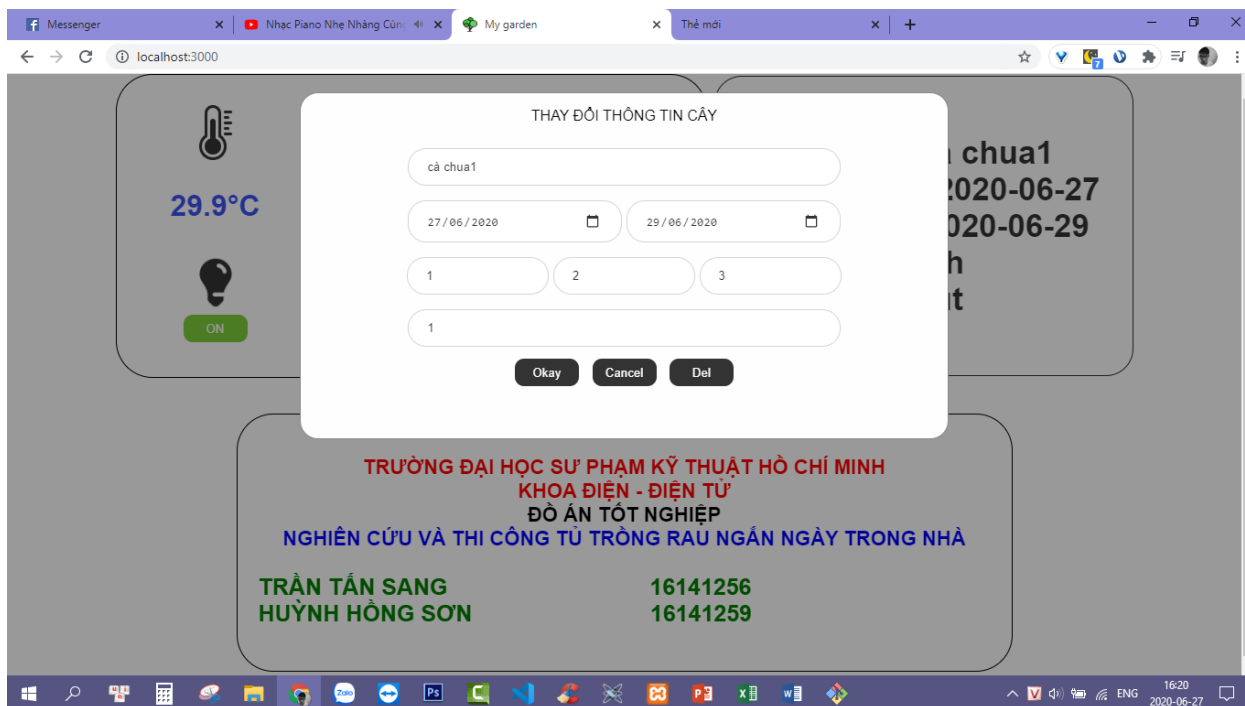
Đưa con trỏ chuột tới tab chứa thông tin cây trồng:



Hình 4-51: Bắt đầu nhập thông tin cây trồng.

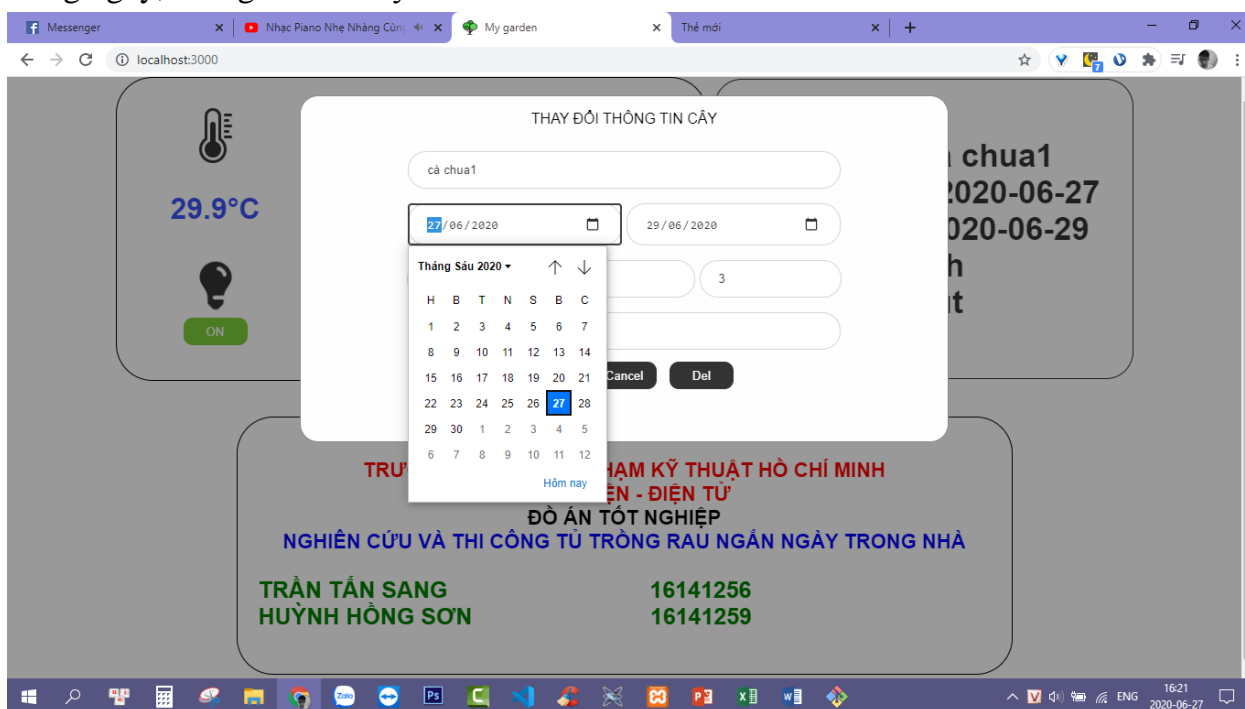
Giao diện nhập thông tin cho cây trồng hiện ra:

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4-52: Nhập các thông tin cơ bản cho cây trồng.

Nhập các thông tin như tên cây, ngày trồng, ngày thu hoạch dự kiến, thời điểm tưới nước trong ngày, thời gian tắt máy bơm:

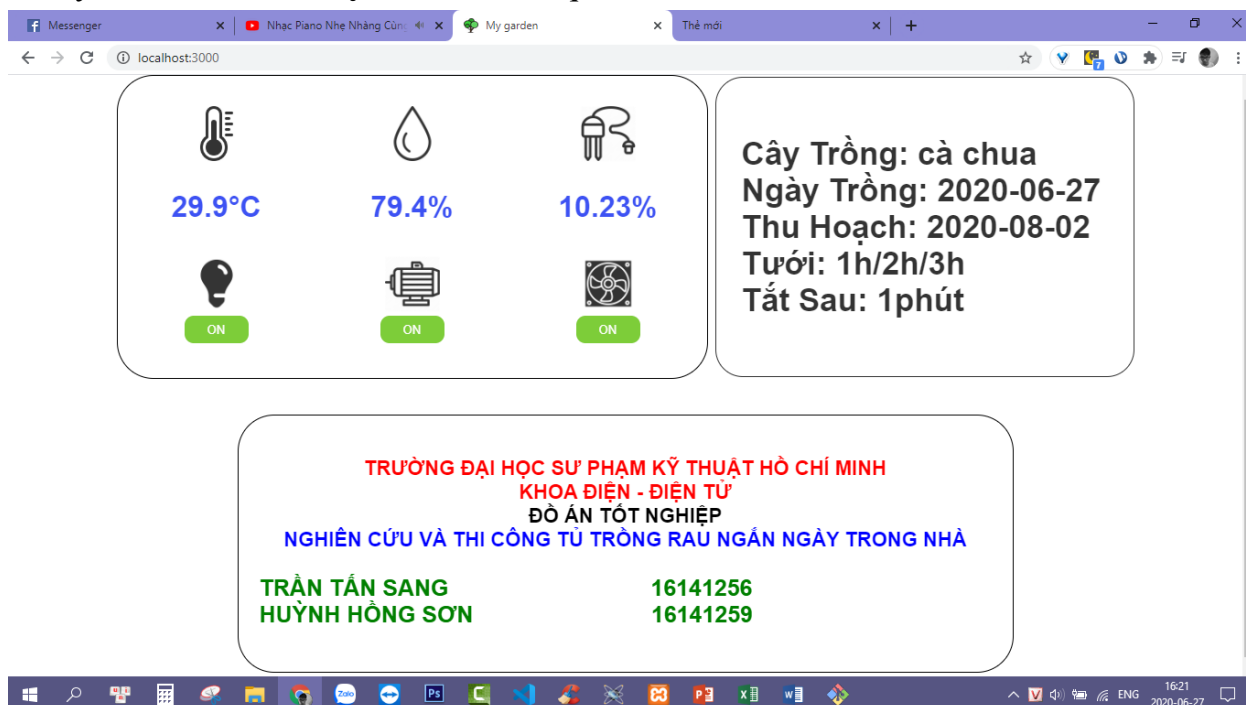


Hình 4-53: Giao diện chọn ngày trồng cây.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Sau đó ấn “**Okay**” để cập nhật, hoặc ấn “**Cancel**” để hủy bỏ dữ liệu vừa nhập và giữ lại giữ liệu trước đó. Nếu muốn xóa toàn bộ thông tin (các thông tin về cây trồng cũng như giờ tưới sẽ để ở trạng thái “**trống**”) thì ấn nút “**Del**”.

Ở đây em ấn chọn “**Okay**” và được kết quả như hình:



Hình 4-54: Kết quả sau khi đã nhập.

Trên web hiển thị các thông số về nhiệt độ, độ ẩm không khí, độ ẩm đất. Có thể thao tác bật/tắt điều khiển quạt, đèn, máy bơm trực tiếp trên web.

4.6 QUY TRÌNH TRỒNG RAU MẦM CƠ BẢN

4.6.1 Dụng cụ trồng rau mầm

- khay trồng: có thể dùng khay kín hoặc khay hở đáy. Nếu đáy khay có lỗ to có thể dùng nilon lót dưới đáy.
- Giấy ăn hoặc giấy vệ sinh.
- Bình tưới cây: 1 bình phun.
- Kéo: 1 cái.

4.6.2 Nguyên liệu

- Giá thể: mùn cưa, mùn mía đã qua xử lý hoại mục hoặc hỗn hợp của xơ dừa, than hoa, phân giun quế, trấu hun, khăn giấy. Đơn giản và tiện lợi hơn thì người trồng có thể mua các loại đất được bán sẵn chuyên cho mục đích trồng cây.
- Hạt giống: có thể dùng hạt rau muống, các loại hạt cải, đậu, giá đỗ,... để gieo mầm. Nên chọn hạt giống có nguồn gốc xuất xứ và còn hạn sử dụng để đảm bảo độ nảy mầm của hạt.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.6.3 Cách trồng

- Hạt giống: ngâm hạt giống trong nước ấm khoảng 3-4 giờ. Để tăng khả năng hút nước của hạt có thể phơi hạt trước khi ngâm. Sau đó dùng vải ẩm, sạch, bọc kín để ủ. Tùy từng loại hạt, sau thời gian từ 6-12 giờ thì hạt nứt vỏ và đem gieo.

- Làm tơi giá thể, tránh vón cục và cho vào khay, dàn phẳng đều. Độ dày giá thể yêu cầu 1,5-2 cm. Dùng giấy ăn phủ kín trên mặt giá thể nhằm tránh cho hạt tiếp xúc với giá thể, đảm bảo độ sạch của rau mầm khi tưới nước và thu hoạch. Phun nước ướt đều khay đựng giá thể rồi tiến hành gieo hạt. Rắc hạt đều tay sao cho hạt phủ kín đều mặt khay, phun nước tưới và để khay trồng ở nơi thoáng mát hoặc che khay bằng bìa carton.

- Hàng ngày tiến hành tưới nước từ 2-3 lần. Không được tưới nước quá nhiều tránh nhằm úng cây. Sau khi tưới, không có nước đọng ở góc nghiêng khay là được. Khi mầm cao có thể tưới gốc rồi nghiêng khay cho nước thấm đều.

- Sau 2-3 ngày, có thể để khay ở nơi có ánh sáng nhưng không cho tiếp xúc trực tiếp với ánh sáng mặt trời.

4.6.4 Thu hoạch

- Sau khi gieo hạt từ 5-7 ngày, lúc này cây đã sinh trưởng khoảng 15 cm thì tiến hành thu hoạch bằng cách dùng kéo sạch cắt sát gốc.

4.6.5 Chế biến

- Rau mầm sau khi thu hoạch nên dùng ngay để tránh bị hỏng, héo; làm giảm giá trị dinh dưỡng.

- Rau mầm có thể dùng để ăn sống, trộn salad, nấu canh, làm các món xào hoặc trần qua nước sôi để chấm nước mắm,....

CHƯƠNG 5: KẾT QUẢ - ĐÁNH GIÁ – NHẬN XÉT

5.1 KẾT QUẢ ĐẠT ĐƯỢC

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm chúng em cũng bổ sung cho mình những kiến thức hay và bổ ích như:

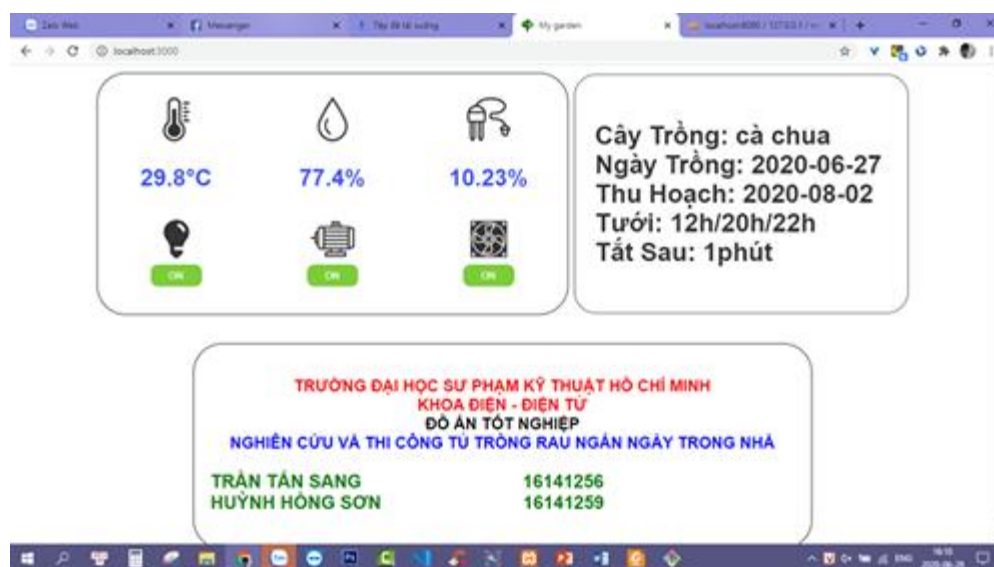
- Hiểu biết sâu hơn về cách sử dụng và các tính năng của Arduino như giao tiếp giữa Arduino với các module cảm biến như: cảm biến độ ẩm đất, cảm biến độ ẩm nhiệt độ DHT22, động cơ bơm nước, các module relay.
- Tìm hiểu được các chuẩn giao tiếp với vi xử lý như I2C, UART,....
- Sử dụng được thêm nhiều ngôn ngữ để phát triển web và ứng dụng Android như HTML, CSS, JavaScript, MySQL, NodeJS cũng như giao tiếp Socketio.
- Quản lý dữ liệu của thiết bị trên database hợp lý. Biết sử dụng, thêm dữ liệu, thay đổi dữ liệu, quản lý cơ sở dữ liệu theo bảng.

Sau thời gian nghiên cứu, thi công đồ án tốt nghiệp với đề tài “**NGHIÊN CỨU VÀ THI CÔNG TỦ TRỒNG RAU NGẮN NGÀY TRONG NHÀ**” nhóm đã được kết quả sau:

- Hệ thống tưới theo giờ đã thiết lập sẵn một cách chính xác.
- Cung cấp ánh sáng cho cây quang hợp.
- Có hệ thống quạt để đổi lưu.
- Có thể điều khiển bằng tay trực tiếp qua bảng điều khiển.
- Xây dựng được web và ứng dụng Android để điều khiển cũng như thiết lập, hiển thị trạng thái của các thiết bị và cài đặt thời gian tưới cho cây trồng.

5.2 KẾT QUẢ THỬ NGHIỆM HỆ THỐNG

5.2.1 Quá trình chạy ứng dụng Android và trên Web



Hình 5-1: Giao diện web điều khiển khi hoạt động.

CHƯƠNG 5: KẾT QUẢ - ĐÁNH GIÁ - NHẬN XÉT



Hình 5-2: Giao diện ứng dụng điều khiển khi hoạt động.

Nội dung	Số lần thành công/ Số lần thử nghiệm
Hiển thị thông số môi trường (nhiệt độ, độ ẩm không khí, độ ẩm đất).	30/30
Bật/tắt các nút điều khiển.	30/30
Thời gian đáp ứng < 5s	30/30
Khả năng đồng bộ dữ liệu giữa ứng dụng và web khi cập nhật, thay đổi trạng thái bật/tắt của các thiết bị chấp hành cũng như dữ liệu cập nhật cho cây trồng.	30/30

Bảng 5-1: Kết quả chạy trên ứng dụng điều khiển qua Wifi.

CHƯƠNG 5: KẾT QUẢ - ĐÁNH GIÁ - NHẬN XÉT

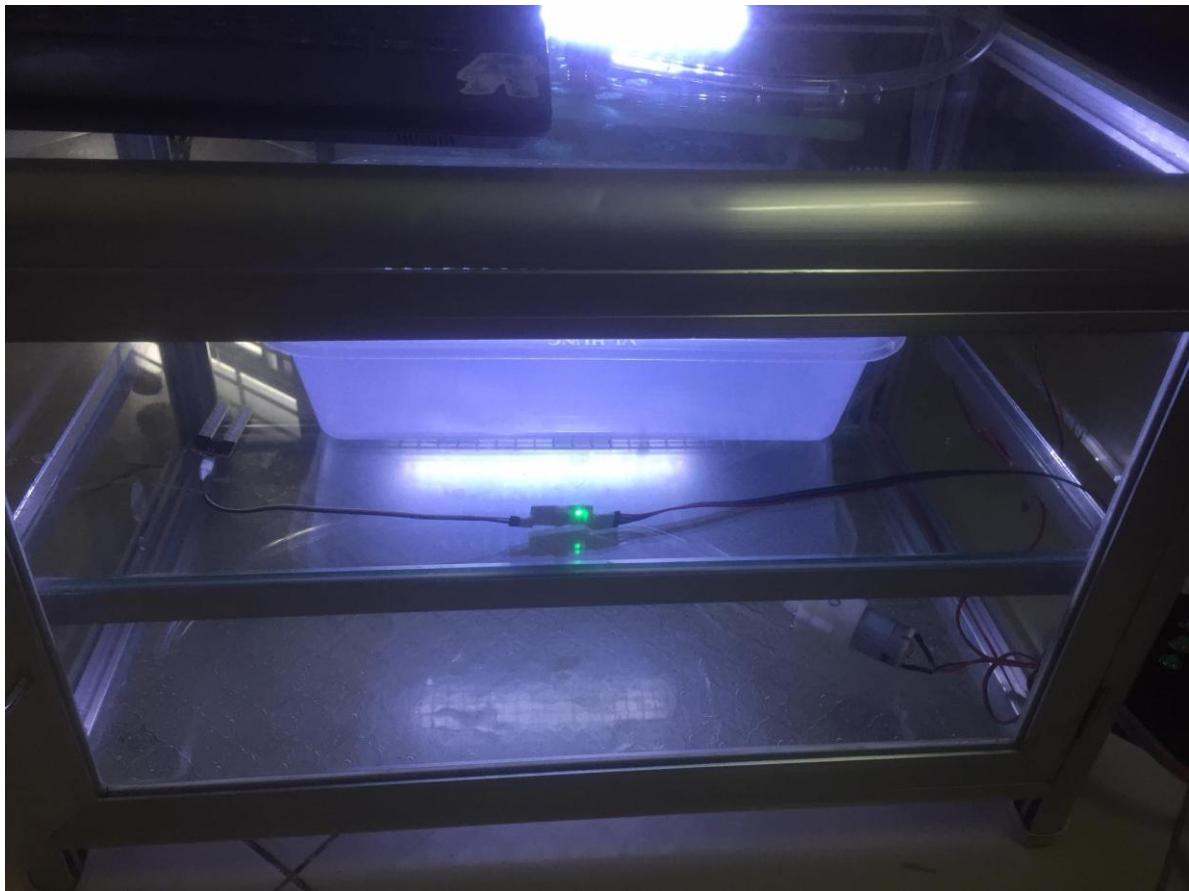
Nội dung	Số lần thành công / Số lần thử nghiệm
Hiển thị thông số môi trường (nhiệt độ, độ ẩm không khí, độ ẩm đất).	30/30
Bật/tắt các nút điều khiển.	30/30
Thời gian đáp ứng < 5s.	30/30
Khả năng đồng bộ dữ liệu giữa ứng dụng và web khi cập nhật, thay đổi trạng thái bật/ tắt của các thiết bị chấp hành cũng như dữ liệu cập nhật cho cây trồng.	30/30

Bảng 5-2: Kết quả chạy trên web điều khiển qua Wifi.

5.2.2 Quá trình vận hành trên phần cứng hệ thống

Sau khi cấp nguồn cho hệ thống:

- Giá trị đo được từ các cảm biến sẽ hiển thị trên màn hình LCD 16x2.
- Trạng thái quạt, bơm, đèn sẽ mặc định tắt.
- Thời gian tưới cây sẽ được thiết lập như trên web và ứng dụng, đến giờ sẽ tưới.



Hình 5-3: Mô hình đang hoạt động với đèn được bật.



Hình 5-4: Trạng thái các thiết bị và thông tin cảm biến được hiển thị trên.

CHƯƠNG 5: KẾT QUẢ - ĐÁNH GIÁ - NHẬN XÉT

Nội dung	Số lần thành công /Số lần thử nghiệm
Bật/tắt các nút điều khiển.	25/30
Thời gian đáp ứng < 5s	25/30
Độ chính xác khi hiển thị thông tin của các cảm biến.	30/30

Bảng 5-3: Kết quả thử nghiệm trên mô hình.

5.3 NHẬN XÉT- ĐÁNH GIÁ

- Nhìn chung hệ thống hoạt động ổn định, đạt yêu cầu đề tài.
- Hệ thống sử dụng nguồn cấp 12 VDC nên an toàn cho người sử dụng trước nguy cơ điện giật.
- Web cũng như ứng dụng Android điều khiển có giao diện dễ sử dụng, hiển thị thông tin trực quan.
- Thời gian đáp ứng điều khiển trong khoảng 2-3 giây. Thời gian đáp ứng dữ liệu lên web mỗi khi có người cập nhật thì lâu hơn một chút khoảng từ 3-5 giây.

Tuy nhiên hệ thống vẫn còn nhiều hạn chế tồn tại như:

- Không có chức năng hẹn giờ tưới trên bảng điều khiển phần cứng, bắt buộc phải thiết lập trên web hoặc ứng dụng điện thoại.
- Các nút nhấn đôi khi đáp ứng chưa tốt.
- Không có nguồn năng lượng dự trữ cung cấp cho mô hình khi mất điện.
- Chức năng hiển thị ngày trồng và ngày thu hoạch dự kiến chỉ có tác dụng hiển thị, chưa gửi được thông báo khi đến ngày thu hoạch.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Sau quá trình nghiên cứu, thi công đề tài “ **NGHIÊN CỨU VÀ THI CÔNG TỦ TRỒNG RAU NGẮN NGÀY TRONG NHÀ**” nhóm chúng em đã hoàn thành xong.

- Mô hình hoạt động ổn định, có thể làm việc liên tục.
- Relay đóng/cắt tải tốt.
- Thiết kế được ứng dụng Android và web điều khiển, quản lí dữ liệu qua wifi.

6.2 HƯỚNG PHÁT TRIỂN

- Phát triển theo hướng quy mô trang trại tự động, thay vì trồng trong phạm vi trong nhà.
- Thiết kế nguồn điện dự phòng khi mất điện.
- Có thể lắp đặt thêm hệ thống camera ghi lại quá trình sinh trưởng của cây để phục vụ cho mục đích nghiên cứu.
- Thiết kế hệ thống tự động bật tắt đèn dựa vào cảm biến ánh sáng.
- Tự động tưới nước dựa vào độ ẩm đất, nhiệt độ môi trường.
- Hệ thống sử dụng được 3G/4G, không phụ thuộc vào wifi.

TÀI LIỆU THAM KHẢO

Sách tham khảo:

- [1] Nguyễn Đình Phú, “*Giáo Trình Vi Xử Lý*”, Xuất bản ĐH Quốc Gia Tp.HCM, 2016.
- [2] “*Giáo Trình Mạch Điện*” của TS Nguyễn Minh Tâm, Ths Trần Tùng Giang – Ths Lê Thị Thanh Hoàng.
- [3] “*Giáo Trình Kỹ Thuật Số*” của Nguyễn Việt Hùng, Hà A Thôi. – Tp.HCM: Đại học Sư phạm kỹ thuật HCM, 2007.
- [4] “*Giáo Trình Điện Tử Cơ Bản*” của Trần Thu Hà, Trương Thị Bích Nga, Nguyễn Thị Lương,...--Tp.HCM: Đại học Quốc Gia Tp.HCM, 2013.
- [5] Đồ án tốt nghiệp: “**THI CÔNG MÔ HÌNH TRỒNG HOA LAN**” – Nguyễn Quang Thạnh (MSSV:14141294), Phan Thanh Triều (MSSV: 14141330) --Trường Đại học Sư phạm kỹ thuật Hồ Chí Minh, 2019.
- [6] Đồ án tốt nghiệp: “**THIẾT KẾ VÀ THI CÔNG MÔ HÌNH GIÁM SÁT VÀ ĐIỀU KHIỂN MÔ HÌNH TRỒNG NẤM RƠM**” – Vũ Ngọc Duy Tín (MSSV:14141321), Phan Thanh Triều (MSSV: 14141142) --Trường Đại học Sư phạm kỹ thuật Hồ Chí Minh, 2019.
- [7] Các ví dụ về ngôn ngữ HTML,CSS, JavaScript; <http://www.w3schools.com/>, <https://reactnative.dev/>
- [8] Giới thiệu về NodeJS, <https://www.codehub.vn/Co-ban-ve-Node-js> .
- [9] Cộng đồng Arduino Việt Nam, <http://arduino.vn>
- [10] Hướng dẫn tạo ứng dụng Android từ expo: <https://docs.expo.io/distribution/building-standalone-apps/> .
- [11] Cơ bản về Socketio: <https://viblo.asia/p/co-ban-ve-socketio-bJzKm0kY59N> , <https://topdev.vn/blog/socket-la-gi-websocket-la-gi/> , <https://topdev.vn/blog/co-ban-ve-giao-thuc-websocket-va-thu-vien-socket-io/> .

DATASHEET:

[12] Thông tin về các loại cảm biến, module sử dụng trong đề tài được tham khảo tại <http://icdayroi.com> , <http://hshop.vn>.

PHỤ LỤC

Chương trình cho Arduino Nano:

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include <ArduinoJson.h>
#include <Arduino_JSON.h>
#include <Adafruit_Sensor.h>

SoftwareSerial ss(10,11);

#define pin_dht A3
#define pin_sensorSoil A2

const int DHT_type = DHT22;
DHT dht(pin_dht, DHT_type);
LiquidCrystal_I2C lcd(0x27,16,2);

const int button[5]= {9,12,13};
const int relay[5] = {4,5,6};
int statusRelay[5] = {1,1,1};

float Temp = 0;
float Humi = 0;
int sensorSoil = 0;
bool checkLCD = true;

void initFunction()
{
  Serial.begin(9600);
  ss.begin(9600);

  dht.begin();

  lcd.init();
  lcd.backlight();

  for (int i =0; i<5; i++)
  {
    pinMode(button[i],INPUT);
    pinMode(relay[i],OUTPUT);
  }
  pinMode(pin_sensorSoil, INPUT);

  for (int i=0; i<5; i++)
  {
    digitalWrite(relay[i], statusRelay[i]);
  }
}

void readSensor_DHT()
{
  Humi = dht.readHumidity();
  Temp = dht.readTemperature();
  Serial.println("Gia tri nhiet do: " + (String)Humi + " Gia tri do am: " +
  (String)Temp);
}
```

```
}
void readSensor_Soil ()
{
    sensorSoil = analogRead(pin_sensorSoil);
    Serial.println("Gia tri do am dat: " + (String)sensorSoil);
}

void controlDevice_Button ()
{
    if (!digitalRead(button[0]))
    {
        while(!digitalRead(button[0]))
        {
            Serial.print("Vao button 0");
        }
        statusRelay[0]=!statusRelay[0];
        digitalWrite(relay[0], statusRelay[0]);
        ss.print ("light1."+(String) statusRelay[0]);
        checkLCD=true;
    }
    if (!digitalRead(button[1]))
    {
        while(!digitalRead(button[1]))
        {
            Serial.print("Vao button 1");
        }
        statusRelay[1]=!statusRelay[1];
        digitalWrite(relay[1], statusRelay[1]);
        ss.print ("pump1."+(String) statusRelay[1]);
        checkLCD=true;
    }
    if (!digitalRead(button[2]))
    {
        while(!digitalRead(button[2]))
        {
            Serial.print("Vao button 2");
        }
        statusRelay[2]=!statusRelay[2];
        digitalWrite(relay[2], statusRelay[2]);
        ss.print ("fan."+(String) statusRelay[2]);
        checkLCD=true;
    }
}

void sendDataSensor ()
{
    String T = String(int(Temp*100));
    String H = String(int(Humi*100));
    String S = String(int(sensorSoil));
    Serial.println("T: "+T+" S: "+S+" H: "+H);
    ss.print ("temp."+T+".humi."+H+".groundhumi."+S);
}

void pareJson_Data (String strJson)
{
    if (strJson != "")
    {
```

```
DynamicJsonBuffer jsonBuffer;
JsonObject &root = jsonBuffer.parseObject(strJson);
if (!root.success())
{
    Serial.println("parseArray() failed");
}
else
{
    int A=root.size();
    statusRelay[0] = root["light"];
    statusRelay[1] = root["pump"];
    statusRelay[2] = root["fan"];
    strJson = "";
    for (int i=0; i<5; i++)
    {
        digitalWrite(relay[i], statusRelay[i]);
    }
    checkLCD=true;
}
}
void displayLCD()
{
    if (checkLCD)
    {
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print("L:");
        if (statusRelay[0])
        {
            lcd.print("OFF");
        }
        else lcd.print("ON");
        lcd.setCursor(6,1);
        lcd.print("P:");
        if (statusRelay[1])
        {
            lcd.print("OFF");
        }
        else lcd.print("ON");

        lcd.setCursor(12,1);
        lcd.print("F:");
        if (statusRelay[2])
        {
            lcd.print("OFF");
        }
        else lcd.print("ON");
        checkLCD=false;
    }
    lcd.setCursor(0,0);
    lcd.print("T:");
    lcd.print((String)((int)Temp));

    lcd.setCursor(6,0);
    lcd.print("H:");
    lcd.print((String)((int)Humi));
}
```

```
lcd.setCursor(12,0);  
lcd.print("S:");  
lcd.print((String)((int)sensorSoil/100));  
}
```

Chương trình chính:

```
#include "function.h"

void setup()
{
  initFunction();
}

void loop()
{
  displayLCD();
  if (ss.available() != 0)
  {
    char temp= ss.read();
    if (temp=='a')
    {
      sendDataSensor();
    }
    Else
    {
      String payLoad = "";
      payLoad = temp;
      while(ss.available() !=0)
      {
        temp = ss.read();
        payLoad += temp;
        delay(10);
      }

      Serial.println("Payload: " + payLoad);
      pareJson_Data(payLoad);
    }
  }
  controlDevice_Button();

  readSensor_DHT();
  readSensor_Soil();
}
```

Chương trình cho ESP 8266:

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <Arduino_JSON.h>
#include <SoftwareSerial.h>
#include <SocketIoClient.h>

SoftwareSerial ss(D5, D1);
SocketIoClient webSocket;

const char* Host_Socket = "192.168.137.1";
unsigned int Port_Socket=3000;

const String ssid = "SANG1412";
const String pwdWifi = "11223344";
int devicesStatus[3] = {0,0,0};
int beginTime = 0;
int endTime = 0;

void setup()
{
  Serial.begin(9600);
  ss.begin(9600);
  WiFi.begin(ssid, pwdWifi);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  webSocket.begin(Host_Socket, Port_Socket,
"/socket.io/?transport=websocket");
  Serial.println("Connected");
  webSocket.emit("update-status", "");
  webSocket.on("DevicesSttData", sendNano);
  beginTime = millis();
}

void loop()
{
  if (endTime - beginTime > 3000)
  {
    String data = "";
    ss.print('a');
    delay(500);
    if(ss.available() !=0)
    {
      while(ss.available() !=0)
      {
        char temp = ss.read();
        data = data + temp;
      }
      int len=1;
      while (data[len] != NULL)
    {
```

```
        len++;
    }
    char str[len+1];
    data.toCharArray(str, len+1);
    websocket.emit("SensorsFromHard", str);
}
beginTime = millis();
}
if (ss.available()>0)
{
    String data = "";
    int beginTimeSS, endTimeSS = millis();
    while (ss.available()>0)
    {
        char temp = ss.read();
        data = data + temp;
        delay(10);
    }
    int len=1;
    while (data[len] != NULL)
    {
        len++;
    }
    char str[len+1];
    data.toCharArray(str, len+1);
    Serial.print(str);
    websocket.emit("DevicesControlNodeMCU", str);
}
endTime = millis();
websocket.loop();
}
void sendNano(const char * payload, size_t length)
{
    pareJson_Data(payload);

ss.print("{\"light\":\"+(String) devicesStatus[0]+\", \"pump\":\"+(String) device
sStatus[1]+\", \"fan\":\"+(String) devicesStatus[2]+\"}");
}
void pareJson_Data(String strJson)
{
    if (strJson != "")
    {
        DynamicJsonBuffer jsonBuffer;
        JsonArray &root = jsonBuffer.parseArray(strJson);
        if (!root.success())
        {
            Serial.println("parseArray() failed");
        }
    }
    Else
    {
        int A=root.size();
        devicesStatus[0] = !root[0]["stt"];
        devicesStatus[1] = !root[1]["stt"];
        devicesStatus[2] = !root[2]["stt"];
    }
}
}
```

