

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ VIỄN THÔNG



ĐỒ ÁN

**XÂY DỰNG HỆ THỐNG IOT CHĂM SÓC CÂY HÚNG QUẾ
TRONG NHÀ KÍNH BẰNG VI ĐIỀU KHIỂN TRUNG TÂM
ESP32 KẾT HỢP BỘ ĐIỀU KHIỂN PID VÀ XUNG PWM ĐỂ
ĐIỀU KHIỂN CƠ CẤU CHẤP HÀNH, THU THẬP DỮ LIỆU
QUA CẢM BIẾN PT100, DHT11 VÀ CAPACITIVE SOIL.**

Học phần: **SCADA NÂNG CAO**

Chuyên ngành: **TỰ ĐỘNG HÓA CÔNG NGHIỆP**

Giảng viên hướng dẫn: ThS. Lê Thị Ngọc Quyên

Nhóm sinh viên thực hiện:

- | | | |
|---------------------------|------------|-------|
| 1. Trương Quang Bảo Khanh | 2051050130 | TD20B |
| 2. Hoàng Minh Khôi | 2051050134 | TD20B |
| 3. Nguyễn Duy Nhật | 2051050154 | TD20B |
| 4. Bùi Tá Khang | 2051050128 | TD20B |
| 5. Phan Huy Ngọ | 2051050149 | TD20B |
| 6. Nguyễn Thái Hòa | 2051050116 | TD20B |

TP Hồ Chí Minh, ngày 13, tháng 11, năm 2023

LỜI CAM ĐOAN

Nhóm em xin cam đoan tất cả nội dung và thông tin trong bài báo cáo bộ môn SCADA Nâng cao là trung thực và do nhóm tự thực hiện và lên ý tưởng trong suốt quá trình nghiên cứu tìm hiểu hoàn thiện đồ án.

Các thông tin trích dẫn trong báo cáo có nguồn gốc rõ ràng và được phép công bố. Nhóm em xin hoàn toàn chịu trách nhiệm về nội dung của bài báo cáo đồ án của mình.

Thành phố Hồ Chí Minh, ngày 13, tháng 11, năm 2023

Nhóm sinh viên thực hiện

Trương Quang Bảo Khanh

Hoàng Minh Khôi

Nguyễn Duy Nhật

Bùi Tá Khang

Phan Huy Ngọ

Nguyễn Thái Hòa

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành đến Cô Lê Thị Ngọc Quyên đã tận tình chỉ bảo, giúp đỡ nhóm hoàn thành môn học và báo cáo đồ án môn học SCADA nâng cao. Không thể quên được, nhóm nghiên cứu xin gửi lời biết ơn đến những đấng sinh thành dưỡng dục đã luôn hỗ trợ, động viên và cũng là niềm động lực lớn lao để nhóm có thể hoàn thành đề tài một cách tốt nhất. Tuy đã có sự chuẩn bị, song bài báo cáo đồ án môn học không thể tránh khỏi những sai sót, nhóm chúng em rất mong được sự cảm thông và góp ý từ cô. Một lần nữa, nhóm chúng em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	2
1.1. Mở đầu – Lý do chọn đề tài:	2
1.2. Mục Tiêu công trình.....	4
1.3. Phương pháp nghiên cứu	4
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	6
2.1. Tổng quan về MQTT – Mosquitto Broker.....	6
2.1.1. MQTT là gì ?	6
2.1.2. Lịch sử hình thành MQTT	6
2.1.3. Thành phần MQTT	7
2.1.4. Chức năng của MQTT	8
2.1.5. Cách thức hoạt động	9
2.1.6. Ưu nhược điểm của MQTT.	10
2.1.7. MQTT Broker Mosquitto	10
2.1.8. Cài đặt và cấu hình Mosquitto lên PC	11
2.2. Node – RED	19
2.2.1. Node – red là gì?.....	19
2.2.2. Ứng dụng của node red trong hệ thống internet	20
2.2.2.1. Node red giúp loại bỏ mọi quy trình rườm rà trong hệ thống dây điện .20	
2.2.2.2. Hạ thanh kỹ thuật nhờ node red	21
2.2.3. Một số thao tác cơ bản của Node - red	21
2.2.4. Ứng dụng node – red vào đề tài.....	25
2.3. Tổng quan về PID (Proportional Integral Derivative)	27
2.3.1. PID là gì?	27
2.3.2. Lịch sử ra đời của PID:.....	29
2.3.3. Các loại bộ điều khiển PID.....	31
2.3.4. Vì sao lại cần PID?	31
2.3.5. Một số ứng dụng của bộ điều khiển PID	31
2.3.6. Ứng dụng PID về điều khiển nhiệt độ:	32
2.3.7. Công thức PID:.....	33
2.4. Các mạch điện sử dụng.....	34
2.4.1. Tìm hiểu về mạch kích Triac:	34
2.4.1.1. Mạch kích Triac là gì?	34
2.4.1.2. Sơ đồ mạch kích Triac:.....	34

2.4.1.3. Nguyên lý hoạt động của mạch kích Triac:	36
2.4.2. Tổng quan về mạch phát hiện điểm 0:.....	36
2.4.2.1. Mạch phát hiện điểm 0 là gì?	36
2.4.2.2. Một số ứng dụng phổ biến của mạch phát hiện điểm 0:	37
2.4.2.3. Tại sao chúng ta cần sử dụng mạch phát hiện điểm 0?.....	37
2.4.2.4. Sơ đồ mạch phát hiện điểm 0:	38
2.4.3. Mạch level shifter.	40
2.5. Ví điều khiển ESP32	44
2.5.1. Giới thiệu ESP32	44
2.5.2. Cấu hình của ESP32	44
2.5.3. Sơ đồ chân của DOIT ESP32 DEVKIT V1 Board	46
2.5.4. Điểm mạnh của ESP32 so với các dòng khác	46
2.6. Cảm biến nhiệt độ và cảm biến độ ẩm đất.....	47
2.6.1. Cảm biến DHT11.....	47
2.6.1.1. Tổng quan DHT11	47
2.6.1.2. Thông số DHT11	47
2.6.1.3. Sơ đồ chân.....	47
2.6.1.4. Các bước đọc dữ liệu từ DHT11	49
2.6.2. Cảm biến nhiệt độ PT100-MAX31865	50
2.6.2.1. Max31865 và PT100 là gì?	50
2.6.2.2. Quá trình kết nối PT100 với MAX31865	52
2.6.2.3. Chuyển đổi công thức Callenda – Van Dusen thành ngôn ngữ lập trình.	53
2.6.3. Cảm biến độ ẩm đất Capacitive Soil V1.2.....	55
2.6.3.1. Tổng quan về cảm biến	55
2.6.3.2. Lịch sử ra đời	55
2.6.3.3. Cấu tạo cảm biến độ ẩm đất Moisture Sensor V1.2	56
2.6.3.4. Nguyên lý hoạt động	57
2.6.3.5. Ưu điểm và nhược điểm	58
2.6.3.6. Thực nghiệm	59
2.7. Các linh kiện cần thiết cho cơ cấu chấp hành.....	65
2.7.1. Module relay 5v 2 kênh	65
2.7.2. Bóng đèn dây tóc	67
2.7.3. Quạt thông gió	68
2.7.4. Động cơ bơm phun sương	69
2.8. Arduino –IDE	70
2.8.1. Phần mềm Arduino IDE là gì?.....	70

2.8.2.	Arduino IDE hoạt động như thế nào?.....	71
2.8.3.	Ưu điểm Arduino IDE	71
2.8.4.	Cài đặt bổ sung ESP32 board cho arduino IDE.....	71
2.9.	EASY EDA.....	75
2.9.1.	Tổng quan về phần mềm Easy EDA:	75
2.9.2.	Cách tạo một project mới trên phần mềm Easy EDA:	76
2.10.	Mạch giảm áp LM2596	77
2.10.1.	Khái niệm	77
2.10.2.	Thông số kỹ thuật	78
2.10.3.	Nguyên lý hoạt động	78
2.10.4.	Ưu điểm và nhược điểm	79
2.11.	Bảo vệ phần cứng hệ thống	80
2.11.1.	Vì sao cần bảo vệ phần cứng hệ thống?	80
2.11.2.	Bảo vệ hệ thống cho mô hình trồng cây trong nhà kính:.....	80
2.11.2.1.	Bảo vệ mạch điện:	80
CHƯƠNG 3. LẬP TRÌNH – THIẾT KẾT VÀ KẾT QUẢ ĐẠT ĐƯỢC		82
3.1.	Yêu cầu cấu trúc hệ thống – Sơ đồ khái – Chức năng của các khái:.....	82
3.1.1.	Yêu cầu của hệ thống “trồng cây húng quế trong nhà kính”	82
3.1.2.	Sơ đồ khái và chức năng mỗi khái	82
3.2.	Quá trình xây dựng – hoạt động của hệ thống:	83
3.3.	Cấu hình và bắt đầu lập trình ESP32 trên Arduino IDE	84
3.3.1.	Các thư viện sử dụng	84
3.3.2.	Cấu hình giao tiếp SPI cho ESP32 và MAX31865	85
3.3.3.	Chuyển đổi thuật toán PID thành ngôn ngữ C/C++	86
3.3.4.	Viết thuật toán bơm nước theo phần trăm độ ẩm đất.....	87
3.4.	Sơ đồ nguyên lý toàn bộ hệ thống	89
3.5.	Tiến hành kết nối các mạch điện với mô hình nhà kính.	90
3.6.	Giám sát – cài đặt - điều khiển trên Node-RED Dashboard.....	92
3.7.	Kết luận và hướng phát triển.....	94

MỤC LỤC HÌNH ẢNH

Hình 1: Vị điêu khiển ESP32 Dev Kit V1	2
Hình 2:MQTT Broker.....	3
Hình 3: Hình ảnh minh họa MQTT	6
Hình 4 Cách thức hoạt động Broker	9
Hình 5 : Mosquitto broker	11
Hình 6: Cài đặt Mosquitto lên PC	12
Hình 7: Cài đặt Mosquitto lên PC	12
Hình 8: Cài đặt Mosquitto lên PC	13
Hình 9: Cài đặt Mosquitto lên PC	13
Hình 10: Cài đặt Mosquitto lên PC	14
Hình 11: Cài đặt Mosquitto lên PC.....	14
Hình 12: Cấu hình Mosquitto	15
Hình 13: Cấu hình Mosquitto	15
Hình 14: Cấu hình Mosquitto	15
Hình 15: Cấu hình Mosquitto	16
Hình 16: Cấu hình Mosquitto	17
Hình 17: Cấu hình Mosquitto	17
Hình 18: Cấu hình Mosquitto	18
Hình 19: Cấu hình Mosquitto	18
Hình 20: NODE-RED.....	19
Hình 21: <i>Cửa sổ làm việc của Node RED</i>	22
Hình 22 Tạo Flow trong Node RED.....	22
Hình 23: Cách xóa Flow trong Node RED.....	23
Hình 24: Cửa sổ ghi chú thích của Node RED	24
Hình 25: Ví trí thiết lập thư viện trong Node RED	25
Hình 26: Giao diện Giám sát Node-Red.....	26
Hình 27: Giao diện Điều khiển Node-Red	26
Hình 28: Lịch sử hoạt động Node-Red.....	27
Hình 29: Minh họa thuật toán PID	28
Hình 30: Lịch sử ra đời PID	29
Hình 31: Điều khiển mức nước bằng PID	32
Hình 32: Các công thức PID sử dụng trong đồ án.....	33
Hình 33: Sơ đồ mạch kích Triac	34
Hình 34: MOC3023 DIP-6 Triac Driver Optocoupler.....	35
Hình 35: Triac BTA16 - 600B	35
Hình 36: Sơ đồ chân Triac BTA16 - 600B	36
Hình 37: Sơ đồ mạch phát hiện điện áp điểm 0	38
Hình 38: Cầu Diode 2A 700V DB207 DIP-4 (2A vuông).....	39
Hình 39: Diode Zener 1W 5.1V DIP 1N4733A	39
Hình 40: Opto PC817 DIP-4 (817).....	40
Hình 41: Mạch Level Shifter	41
Hình 42: Mosfet driver 2xAOD4184	42
Hình 43: Mối liên hệ giữa V _{GS} và I _D	43

Hình 44: Bo mạch phát triển dựa trên vi điều khiển ESP32.....	44
Hình 45: Sơ đồ chân của DOIT ESP32 DEVKIT V1 Board.....	46
Hình 46: Cảm biến DHT11.....	48
Hình 47: Mạch ứng dụng điển hình cho cảm biến độ ẩm và nhiệt độ DHT11.....	48
Hình 48: Đọc dữ liệu từ DHT11	49
Hình 49: :Đọc dữ liệu từ DHT11	50
Hình 50: Đọc dữ liệu từ DHT11	50
Hình 51: Mối tương quan giữa nhiệt độ và điện trở.....	51
Hình 52: MAX31865 Adafruit	52
Hình 53: PT100 và MAX31865	53
Hình 54: Cảm biến độ ẩm đất Moisture Sensor V1.2.....	55
Hình 55: Mạch PCB cảm biến điện dung Moisture Sensor v1.2.....	57
Hình 56: Khối lượng của ly nhựa	60
Hình 57: Giá trị analog ứng với độ ẩm đất 0% khoảng 2750.....	62
Hình 58: Giá trị analog ứng với độ ẩm đất 20% khoảng 2325.....	62
Hình 59: Giá trị analog ứng với độ ẩm đất 40% khoảng 2020.....	63
Hình 60: Giá trị analog ứng với độ ẩm đất 60% khoảng 1670.....	64
Hình 61: Giá trị analog ứng với độ ẩm đất 80% khoảng 1520.....	64
Hình 62: Đồ thị tín hiệu đầu ra của cảm biến.....	65
Hình 63 Module relay 5v 2 kênh.....	66
Hình 64: Sơ đồ chân Module relay 5v 2 kênh	66
Hình 65: Quạt thông gió	68
Hình 66: Động cơ bơm phoi sương	69
Hình 67: LOGO giao diện IDE	70
Hình 68: Cài đặt bổ sung ESP32 board cho arduino IDE	71
Hình 69: Cài đặt bổ sung ESP32 board cho arduino IDE	72
Hình 70: : Cài đặt bổ sung ESP32 board cho arduino IDE	72
Hình 71: Cài đặt bổ sung ESP32 board cho arduino IDE	73
Hình 72: Cài đặt bổ sung ESP32 board cho arduino IDE	73
Hình 73: Cài đặt bổ sung ESP32 board cho arduino IDE	74
Hình 74: Cài đặt bổ sung ESP32 board cho arduino IDE	74
Hình 75: Cài đặt bổ sung ESP32 board cho arduino IDE	75
Hình 76: Giao diện làm việc trên phần mềm Easy EDA.....	75
Hình 77: Hướng dẫn tạo project mới trên Easy EDA.....	76
Hình 78: Hướng dẫn tạo project mới trên Easy EDA.....	76
Hình 79: Mạch giảm áp LM2596	77
Hình 80: Sơ đồ khối hệ thống.....	82
Hình 81: Thư viện sử dụng	84
Hình 82: Cấu hình SPI.....	85
Hình 83: Mode SPI	85
Hình 84: Tốc độ truyền tải SPI của MAX31865	86
Hình 85: Đặt biến PID cần thiết	86
Hình 86: Thuật toán PID code C/C++	87
Hình 87: Khai báo biến xuất xung PWM	88
Hình 88: Setup PIN PWM	88
Hình 89: Thuật toán bơm sương PWM	88
Hình 90: Lưu ý khi bơm sương	88
Hình 91: Schematic toàn bộ hệ thống.....	89

Hình 92: Tổng quan phần cứng	91
Hình 93: Đầu ra tủ điện	92
Hình 94: Tab giám sát thông số	93
Hình 95: Tab chế độ cảnh báo	93
Hình 96: Tab bảng số liệu.....	94

TÓM TẮT CÔNG TRÌNH

Hiện nay, khoa học kỹ thuật ngày càng phát triển, con người đã có cung như đang có nhiều bước tiến xa trong nhiều lĩnh vực, thực hiện được những việc mà trước đây tưởng chừng là không thể. Nhìn chung, tất cả các nỗ lực đó đều phục vụ cho nhu cầu đòi hỏi sống con người ngày càng đầy đủ, tiện nghi hơn ; và đặc biệt, một trong số đó là lĩnh vực nghiên cứu về hệ điều khiển tự động, xu hướng phát triển toàn cầu trong những năm gần đây.

Ở Việt Nam nói riêng, tình hình nghiên cứu về hệ thống trồng cây trong nhà kính đã có những tiến bộ đáng kể. Các nghiên cứu tập trung vào kiểm soát nhiệt độ, độ ẩm không khí, độ ẩm đất, độ PH, nồng độ CO₂, v.v.... giúp cho cây trồng được phát triển trong điều kiện thời tiết tốt nhất và cho ra sản lượng liên tục bất kể mùa màng hoặc thời tiết bên ngoài. Để tiếp nối, góp phần cải tiến vào công cuộc nghiên cứu ấy, nhóm nghiên cứu đã lên kế hoạch cho một đề tài chăm sóc cây trồng trong nhà kính với mục tiêu là kiểm soát nhiệt độ và độ ẩm đất cho cây húng quế ra lá xanh tốt.

Với những kiến thức và kinh nghiệm từ những đồ án trước, nhóm đã quyết định xây dựng một hệ thống nhúng IoT trồng cây húng quế trong nhà kính sử dụng vi điều khiển ESP32 – Module WiFi 2.4GHz, kết hợp thuật toán PID và thuật toán xuất xung PWM để điều khiển các cơ cấu chấp hành kiểm soát nhiệt độ và độ ẩm đất.

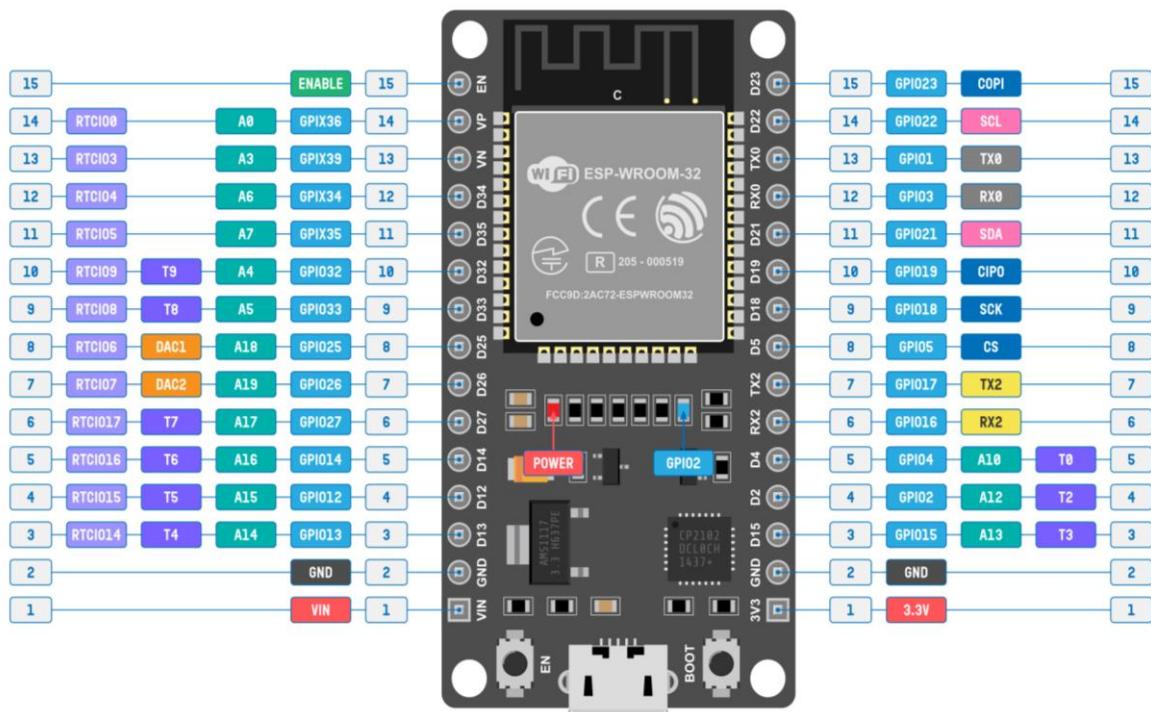
CHƯƠNG 1. TỔNG QUAN

Ở chương này, nhóm em sẽ trình bày lý do chọn đề tài, cũng như tính cấp thiết, tình hình nghiên cứu, mục tiêu, phương pháp nghiên cứu. Từ đó, người đọc sẽ dần hiểu được nguyên nhân, thực trạng cũng như phương án giải quyết của nhóm.

1.1. Mở đầu – Lý do chọn đề tài:

Ngày nay, việc ứng dụng khoa học công nghệ vào lĩnh vực phát triển nông nghiệp - đặc biệt là nhóm ngành tự động hóa – đã trở thành một hướng đi lớn mạnh, phát triển chính giúp thúc đẩy nền kinh tế.

Qua quá trình tìm hiểu, nhóm nhận thấy ESP32 là vi điều khiển giá thành rẻ, tốc độ xử lý nhanh (240MHz), có nhiều giao thức truyền thông như SPI, USART, I2C v.v... được sử dụng nhiều trong các project IoT thực tế. Ngoài ra,

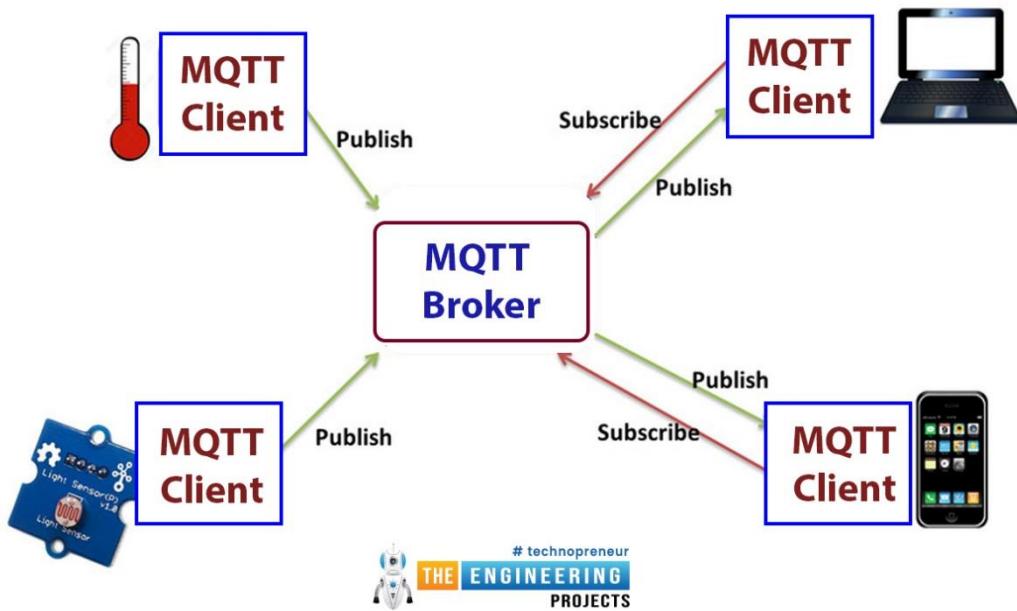


Hình 1: Vi điều khiển ESP32 Dev Kit V1

nghiên cứu ESP32 giúp cho người nghiên cứu có cái nhìn toàn diện về một hệ thống IoT – các giao thức trong IoT như MQTT, HTTP, v.v.... Qua đó cung cấp vững chắc cho sinh viên những kiến thức rõ ràng về cách xây dựng và hoàn thiện một project IoT.

Cũng với góc nhìn và hướng đi đó, nhóm chúng em đã quyết định lựa chọn nghiên cứu phát triển hệ tròng cây húng qué trong nhà kính, giám sát, cài đặt và điều khiển thông qua Node-Red Dashboard. Ở đồ án này, nhóm em sẽ sử dụng giao thức MQTT dựa trên mạng LAN để truyền và nhận tín hiệu giữa các Client và MQTT Broker.

Fig 2: MQTT Publish and Subscribe Architecture



Hình 2:MQTT Broker

Việc nghiên cứu 1 hệ thống ứng dụng thuật toán PID để gia nhiệt hoặc băm xung PWM cho động cơ không còn là điều mới mẻ. Tuy nhiên việc thiết kế và ứng dụng những kiến thức đó vào việc trồng cây trong nhà kính sử dụng vi điều khiển giá rẻ ESP32 có tốc độ xử lý nhanh, nhiều chuẩn giao tiếp, ngoại vi là việc vẫn chưa được nghiên cứu sâu tại Trường Đại học Giao Thông Vận Tải Thành phố Hồ Chí Minh nói riêng và khu vực Thành phố Hồ Chí Minh nói chung. Chính vì những lý do trên, cộng hưởng với sự đồng thuận của giảng viên hướng dẫn **Th.S Lê Thị Ngọc Quyên**, nhóm nghiên cứu đã chọn đề tài này.

1.2. Mục Tiêu công trình

- Mục tiêu đầu tiên nhóm đề ra là có thể hiểu được vi điều khiển ESP32 để có thể truyền và nhận dữ liệu từ PC/Smartphone thông qua giao thức MQTT.
- Nghiên cứu Node-RED và tự design Dashboard thông qua lập trình kéo thả các Node và lập trình Javascript cho giao diện hiển thị và cài đặt.
- Xây dựng thuật toán PID gia nhiệt tự động cho nhà kính khi gặp trời lạnh có thời gian xác lập dưới 5 phút, sai số xác lập dưới 0.09 độ C và độ vọt lố dưới 3%.
- Kiểm soát được công suất của máy bơm phun sương tùy theo nhiệt độ không khí và độ ẩm của đất.
- Nghiên cứu đo lường và tự viết thư viện cho cảm biến PT100 – MAX31865 và Capacitive Soil, đọc hiểu thư viện DHT11 và kiểm chứng giá trị của nó.
- Thiết kế mô hình nhà kính thực tế kèm theo đó là thiết kế tủ điện có kích thước vừa phải để tiện cho việc vận chuyển. Ngoài ra nhóm nghiên cứu cũng cần nghiên cứu cách để bảo quản các linh kiện điện tử - cảm biến khỏi các yếu tố tự nhiên gây hư hại cho hệ thống.
- Cuối cùng cần phải không ngừng nâng cao kiến thức, mở rộng trải nghiệm để tiếp xúc với các dự án lớn trong thực tế.

1.3. Phương pháp nghiên cứu

Do đây là một đề tài khá mới mẻ với nhóm nên trong thời gian thực hiện công trình, chúng em đã gặp không ít khó khăn trong việc tiếp cận với các nguồn tài liệu.

Nhóm thực hiện công trình sử dụng hai phương pháp chủ yếu:

Phương pháp tham khảo tài liệu: Nguồn tài liệu chủ yếu bằng tiếng Anh/tiếng Việt được tìm kiếm trên mạng Internet. Về thông tin nguồn tài liệu, nhóm em sẽ liệt kê chi tiết ở phần “TÀI LIỆU LAM THẢO”

Phương pháp lập trình, thiết kế mạch trên phần mềm tin học: Song song với việc tham khảo tài liệu thì nhóm em đã sử dụng một số phần mềm tin học để có thể minh họa, phân tích và đào sâu vào cách thức xây dựng hệ thống trồng cây trong nhà kính một cách chi tiết và rõ ràng nhất.

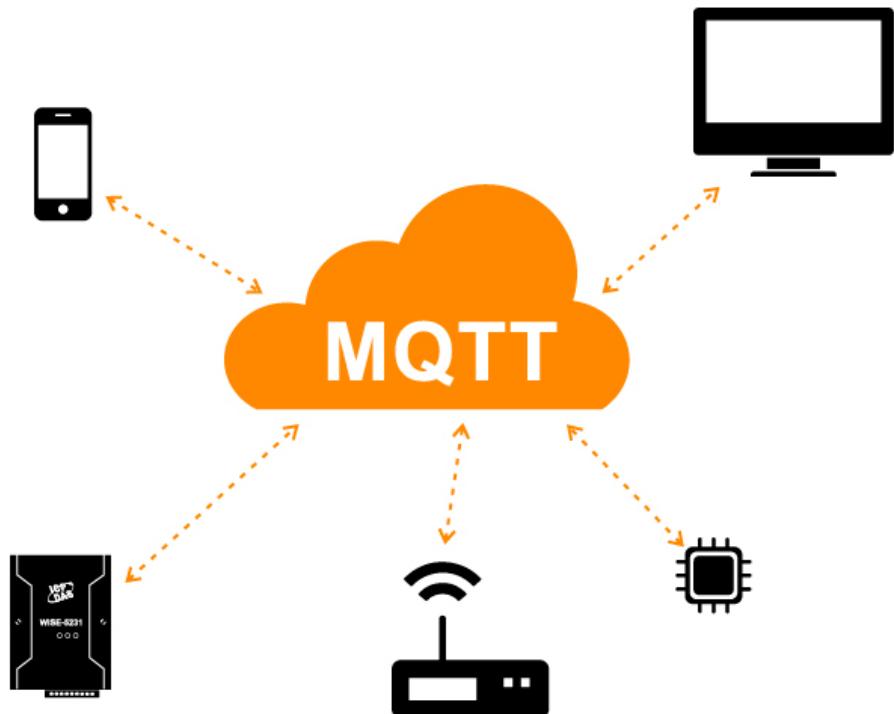
Một số phần mềm tin học nhóm nghiên cứu sử dụng có thể liệt kê như:

- + Arduino IDE – dùng để lập trình, chạy code, debug code, nạp code.
- + Easy EDA – dùng để thiết kế mạch điện tử, mạch PCB hệ thống.
- + Node-Red – dùng để lập trình giao tiếp MQTT với ESP32 thông qua MQTT Broker Mosquitto được cài đặt local trên laptop.

CHƯƠNG 2. CƠ SỞ LÍ THUYẾT

2.1. Tổng quan về MQTT – Mosquitto Broker

2.1.1. MQTT là gì ?



Hình 3: Hình ảnh minh họa MQTT

MQTT là từ viết tắt của cụm Message Queuing Telemetry Transport (tạm dịch: giao thức truyền thông điệp). Đây là một trong những giải pháp tiêu chuẩn của IoT (Internet of Things) vì quá trình truyền tải của MQTT rất nhẹ, độ chính xác cao và khả năng kết nối băng thông hiệu quả. MQTT còn được hiểu là một giao thức nhắn tin thông minh và đơn giản, được tạo ra nhằm phục vụ cho các thiết bị hạn chế về băng thông.

Nhiệm vụ của giao thức MQTT là hỗ trợ người dùng đọc, xuất bản dữ liệu. Bên cạnh đó, giao thức này còn giúp bạn gửi lệnh để điều khiển từ xa. Những hoạt động này đều thông qua nút cảm biến và một số tính năng khác. Nhiều người cho rằng MQTT là phương tiện để các thiết bị giao tiếp với nhau dễ dàng hơn.

2.1.2. Lịch sử hình thành MQTT

MQTT chính thức ra đời vào cuối những năm 1990. Giao thức này được phát minh bởi hai kỹ sư tài năng – Andy Stanford-Clark và Arlen Nipper. Nhiệm vụ của MQTT là “phương tiện” giao tiếp giữa hai thiết bị. Diễn hình là đường ống dầu khí và SCADA (hệ thống giám sát và thu thập dữ liệu).

- Trước khi MQTT xuất hiện, các thiết bị này không thể giao tiếp với nhau. Bởi vì chúng chỉ sử dụng các giao thức độc quyền, riêng biệt. Nhờ MQTT, vấn đề này đã được khắc phục nhanh chóng. Ngoài ra, điểm mạnh của giao thức này là: nhẹ, băng thông tối thiểu, dễ triển khai, cung cấp dữ liệu chất lượng,... Do đó, MQTT đã được ứng dụng mạnh mẽ hơn.
- MQTT chính thức ra đời vào cuối những năm 1990. Giao thức này được phát minh bởi hai kỹ sư tài năng – Andy Stanford-Clark và Arlen Nipper. Nhiệm vụ của MQTT là “phương tiện” giao tiếp giữa hai thiết bị. Diễn hình là đường ống dầu khí và SCADA (hệ thống giám sát và thu thập dữ liệu).
- Trước khi MQTT xuất hiện, các thiết bị này không thể giao tiếp với nhau. Bởi vì chúng chỉ sử dụng các giao thức độc quyền, riêng biệt. Nhờ MQTT, vấn đề này đã được khắc phục nhanh chóng. Ngoài ra, điểm mạnh của giao thức này là: nhẹ,
- Năm 1999: MQTT ban đầu được phát triển bởi Dr. Andy Stanford-Clark của IBM và Arlen Nipper của Arcom (nay là Eurotech). Họ đang làm việc trên một dự án liên quan đến kiểm soát môi trường và kiểm soát công nghiệp.
- Năm 2000: Giao thức MQTT đã được phát hành công khai dưới dạng một giao thức mã nguồn mở và đã thu hút sự quan tâm của cộng đồng phát triển phần mềm.
- Năm 2010: IBM chấp nhận MQTT như một dự án mã nguồn mở trong cộng đồng Eclipse. Giao thức được đặt tên là "Eclipse Paho," và nhóm phát triển Paho bắt đầu làm việc để cung cấp thư viện MQTT cho nhiều ngôn ngữ lập trình khác nhau.
- Năm 2013: MQTT trở thành một chuẩn OASIS (Organization for the Advancement of Structured Information Standards) và được đổi tên thành "MQTT Version 3.1.1." Phiên bản này đã cung cấp một số cải tiến và khắc phục lỗi so với phiên bản trước đó.
- Năm 2019: MQTT 5.0 được ra mắt như một phiên bản cải tiến hơn với nhiều tính năng mới và cải tiến bảo mật. Phiên bản này đã tạo ra nhiều cơ hội cho các ứng dụng IoT và các hệ thống máy chủ-client đòi hỏi tính linh hoạt và hiệu suất cao hơn.
- Năm 2020 và sau đó: MQTT tiếp tục phát triển và trở thành một phần quan trọng của cơ sở hạ tầng IoT. Nó được sử dụng trong các ứng dụng IoT, hệ thống giám sát, và nhiều ứng dụng khác.

2.1.3. Thành phần MQTT

- Client

- Publisher - Nơi gửi thông điệp
- Subscriber - Nơi nhận thông điệp
- Broker - Máy chủ môi giới

Trong đó Broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ Client (Publisher/Subscriber). Nhiệm vụ chính của Broker là nhận thông điệp (message) từ Publisher, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể. Nhiệm vụ phụ của Broker là nó có thể đảm nhận thêm một vài tính năng liên quan tới quá trình truyền thông như: bảo mật message, lưu trữ message, logs,

Client thì được chia thành hai nhóm là Publisher và Subscriber. Client chỉ làm ít nhất một trong 2 việc là publish các thông điệp (message) lên một/nhiều topic cụ thể hoặc subscribe một/nhiều topic nào đó để nhận message từ topic này.

2.1.4. Chức năng của MQTT

Giám sát từ xa

MQTT hoạt động tốt cho các ứng dụng liên quan đến giám sát từ xa như:

Đồng bộ hóa các cảm biến. Chẳng hạn đầu báo cháy hoặc cảm biến chuyển động để phát hiện trộm cắp. Để xác định mối nguy hiểm có hợp lệ hay không.

Giám sát các thông số sức khỏe bằng cảm biến cho bệnh nhân xuất viện và cảm biến cảnh báo người già nguy hiểm.

Ứng dụng nhắn tin

MQTT được Facebook sử dụng cho ứng dụng Messenger. Bên cạnh việc tiết kiệm pin trong quá trình nhắn tin giữa điện thoại di động. Giao thức này cho phép gửi tin nhắn hiệu quả lên đến mili giây. Kể cả khi kết nối internet không nhất quán trên toàn cầu.

Dịch vụ đám mây

Hầu hết các nhà cung cấp dịch vụ đám mây lớn, bao gồm Amazon Web Services (AWS), Google Cloud, IBM Cloud và Microsoft Azure đều có hỗ trợ giao thức MQTT.

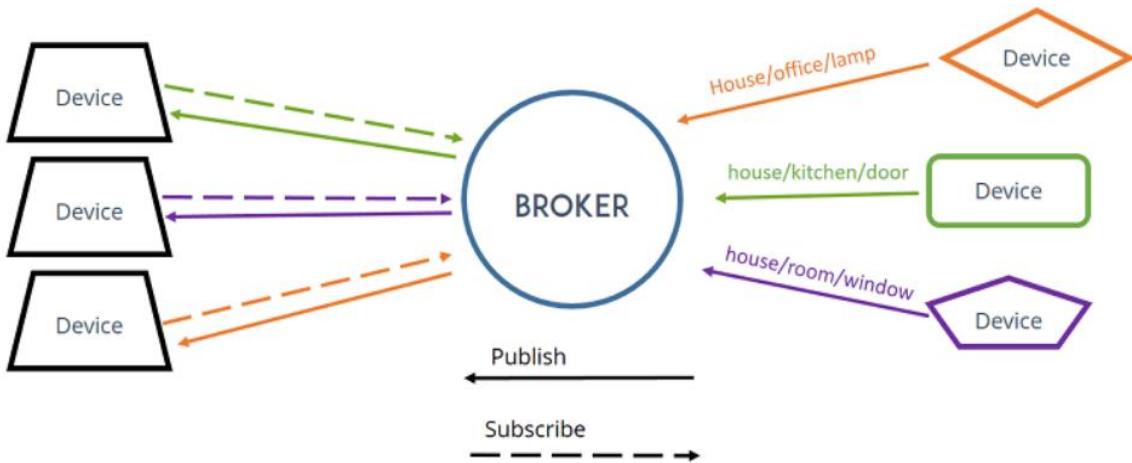
Thiết bị đo lường

Smart meter là một thiết bị điện tử ghi lại các thông tin như tiêu thụ năng lượng điện, mức điện áp, dòng điện và hệ số công suất. Thiết bị này sử dụng giao thức MQTT để truyền dữ liệu nhằm đảm bảo cho các chỉ số đồng hồ luôn chính xác trong thời gian thực.

Hệ thống thanh toán

MQTT giúp loại bỏ các tập tin trùng lặp hoặc bị mất trong việc lập hóa đơn, thanh toán.

2.1.5. Cách thức hoạt động



Hình 4 Cách thức hoạt động Broker

Giao thức này hoạt động theo mô hình đăng ký và công bố (publish-subscribe), giúp các thiết bị và ứng dụng giao tiếp thông qua một trung tâm thông điệp (message broker). Dưới đây là cách MQTT hoạt động:

Thiết lập kết nối: Thiết bị MQTT (thường gọi là MQTT client) thiết lập kết nối với một MQTT broker, là trung tâm quản lý thông điệp. Kết nối này có thể sử dụng giao thức TCP hoặc SSL/TLS để bảo mật kết nối.

Đăng ký (Subscribe): MQTT client có thể đăng ký (subscribe) để nhận thông điệp từ các chủ đề (topics) cụ thể. Một chủ đề là một chuỗi ký tự định danh thông điệp. Ví dụ, một thiết bị có thể đăng ký để nhận thông điệp từ chủ đề "sensor/temperature" hoặc "device/status". MQTT broker sẽ theo dõi các đăng ký này.

Công bố (Publish): MQTT client có thể gửi thông điệp đến một hoặc nhiều chủ đề bằng cách công bố (publish) thông điệp với nội dung và chủ đề tương ứng. Các thông điệp này được gửi đến MQTT broker.

Phân phối thông điệp: MQTT broker nhận thông điệp từ các MQTT client và chuyển nó đến các MQTT client đã đăng ký với chủ đề tương ứng. Điều này cho phép các thiết bị và ứng dụng nhận thông điệp mà họ quan tâm, giúp tối ưu hóa băng thông và tăng hiệu suất truyền thông.

Xác nhận (Acknowledgment): Sau khi MQTT broker đã chuyển thông điệp đến các subscriber, nó sẽ gửi xác nhận cho MQTT publisher để báo hiệu rằng thông điệp đã được gửi thành công.

Ngắt kết nối: MQTT client có thể ngắt kết nối khi không cần sử dụng nữa. MQTT broker cũng có thể ngắt kết nối nếu cần.

2.1.6. Ưu nhược điểm của MQTT.

Ưu điểm

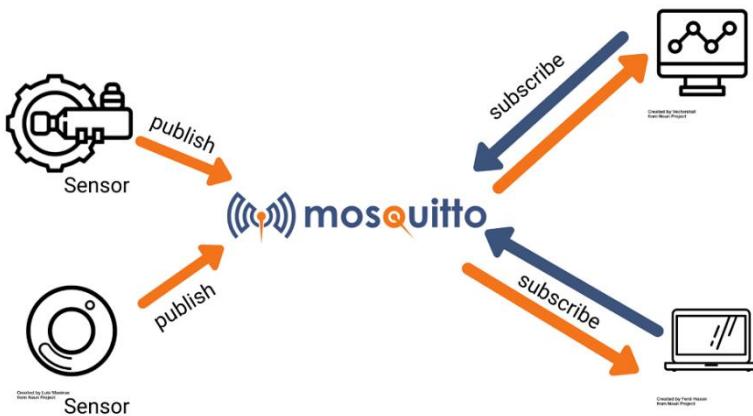
- Kết nối riêng rẽ, độc lập.
- Khả năng mở rộng.
- Thời gian tách biệt (Time decoupling).
- Đồng bộ riêng rẽ (Synchronization decoupling).

Nhược điểm

- Máy chủ môi giới (Broker) không cần thông báo về trạng thái gửi thông điệp. Do đó không có cách nào để phát hiện xem thông điệp đã gửi đúng hay chưa.
- Publisher không hề biết gì về trạng thái của subscribe và ngược lại. Vậy làm sao chúng ta có thể đảm bảo mọi thứ đều ổn.
- Những kẻ xấu (Malicious Publisher) có thể gửi những thông điệp xấu, và các Subscriber sẽ truy cập vào những thứ mà họ không nên nhận.

2.1.7. MQTT Broker Mosquitto

Mosquitto là một MQTT Broker mã nguồn mở cho phép thiết bị truyền nhận dữ liệu theo giao thức MQTT version 5.0, 3.1.1 và 3.1 – Một giao thức nhanh, nhẹ theo mô hình publish/subscribe được sử dụng rất nhiều trong lĩnh vực Internet of Things. Mosquitto cung cấp một thư viện viết bằng ngôn ngữ C để triển khai các MQTT Client và có thể dễ dàng sử dụng bằng dòng lệnh: “mosquitto_pub” và “mosquitto_sub”. Ngoài ra, Mosquitto cũng là một phần của Eclipse Foundation, là dự án iot.eclipse.org và được tài trợ bởi cedalo.com



Hình 5 : Mosquitto broker

Ưu điểm:

- Ưu điểm nổi bật của Mosquitto là tốc độ truyền nhận và xử lý dữ liệu nhanh, độ ổn định cao, được sử dụng rộng rãi và phù hợp với những ứng dụng embedded.
- Mosquitto rất nhẹ và phù hợp để sử dụng trên tất cả các thiết bị.
- Ngoài ra, Mosquitto cũng được hỗ trợ các giao thức TLS/SSL (các giao thức nhằm xác thực server và client, mã hóa các message để bảo mật dữ liệu).

Nhược điểm:

- Một số nhược điểm của mosquitto là khó thiết kế khi làm những ứng dụng lớn và ít phương thức xác thực thiết bị nên khả năng bảo mật vẫn chưa tối ưu.

2.1.8. Cài đặt và cấu hình Mosquitto lên PC

Cài đặt Mosquitto:

Bước 1: Lựa chọn phiên bản cài phù hợp cho máy tính (Em dùng Window 11 64bit nên chọn như hình dưới)

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- [mosquitto-2.0.14-install-windows-x64.exe](#) (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- [mosquitto-2.0.14-install-windows-x32.exe](#) (32-bit build 1, Windows Vista and up, built with Visual Studio Community 2019)

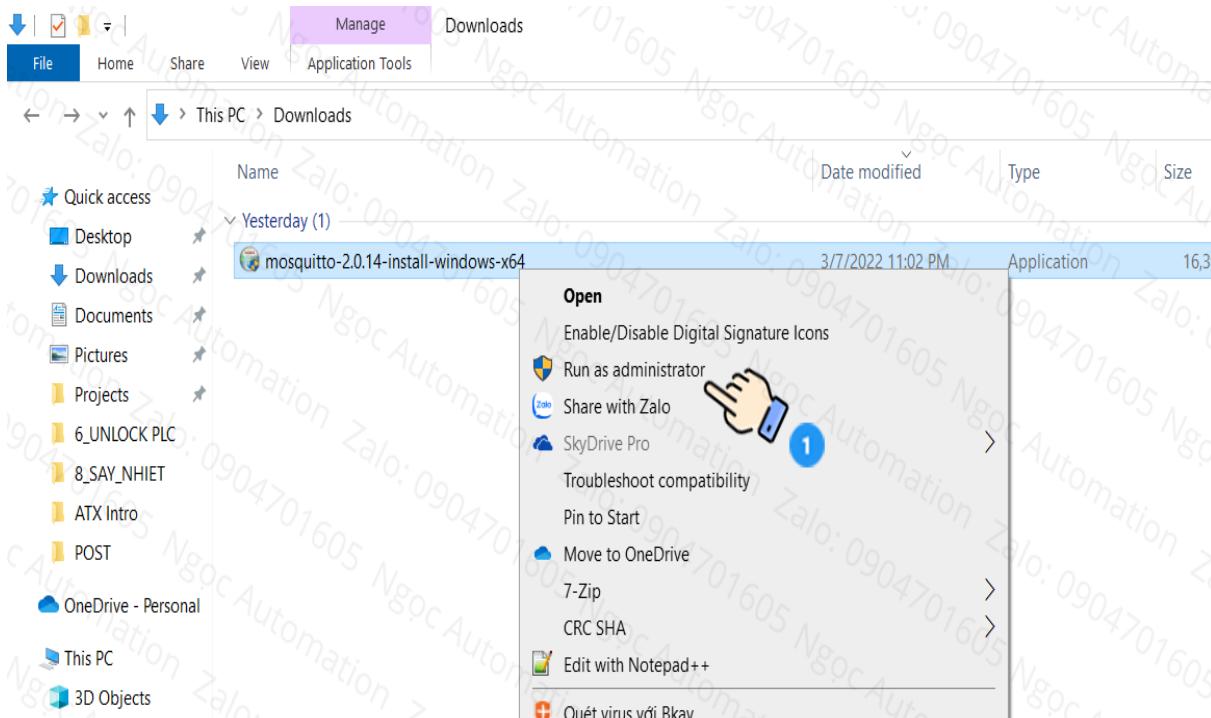
Older installers can be found at <https://mosquitto.org/files/binary/>.

See also README-windows.md after installing.

Mac

Hình 6: Cài đặt Mosquitto lên PC

Bước 2: Sau khi tải về chạy trình cài đặt Broker bằng quyền admin.



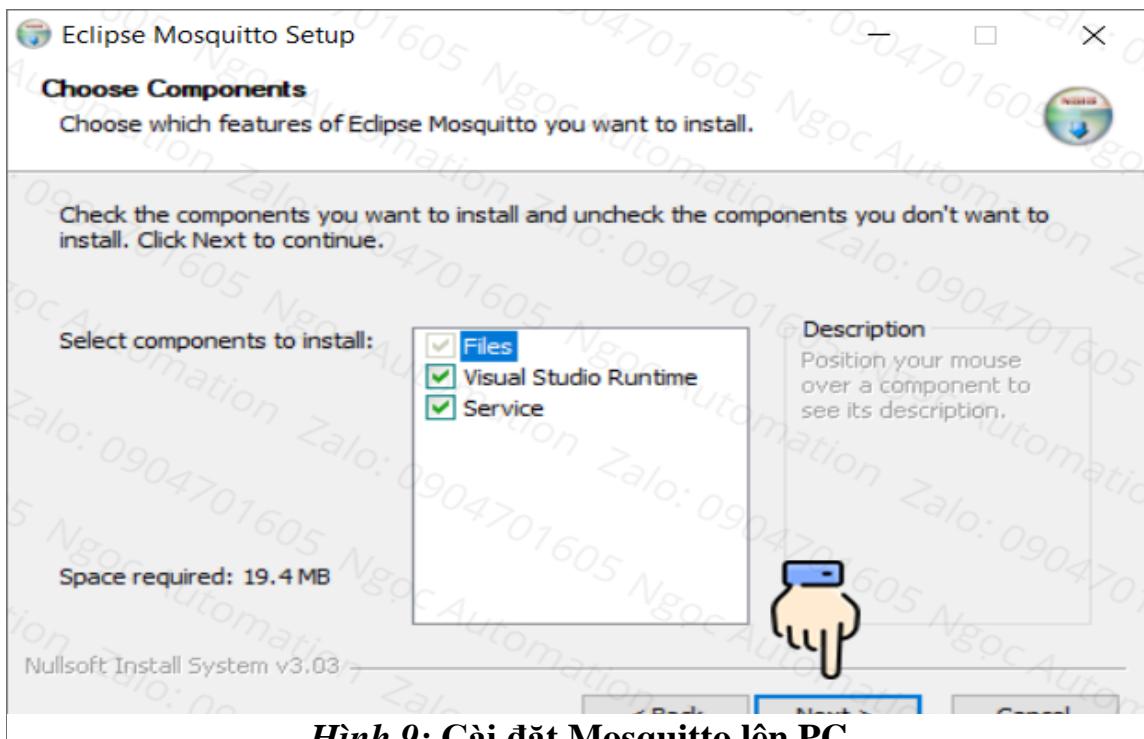
Hình 7: Cài đặt Mosquitto lên PC

Bước 3: Nhấn chọn Next



Hình 8: Cài đặt Mosquitto lên PC

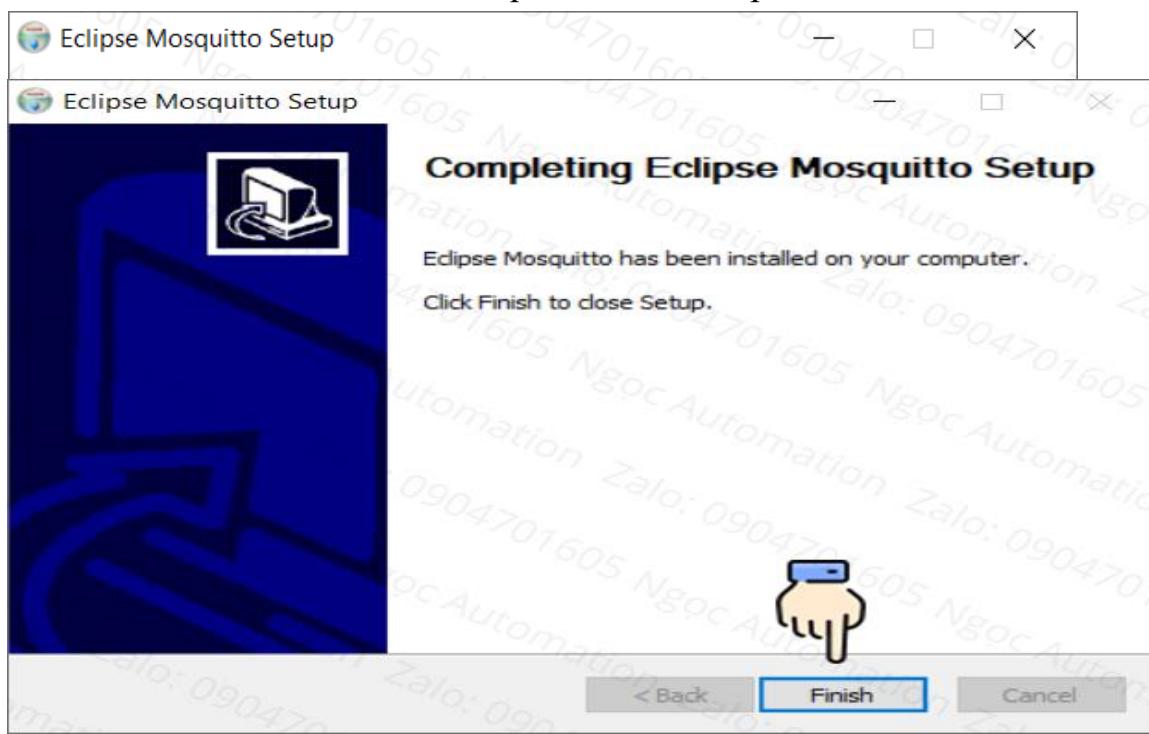
Bước 4: Tiếp tục nhấn Next



Hình 9: Cài đặt Mosquitto lên PC

Bước 5: Lựa chọn folder cài đặt là ổ C (chú ý), sau đó nhấn Install

Bước 6: Nhấn finish để kết thúc quá trình cài đặt phần mềm

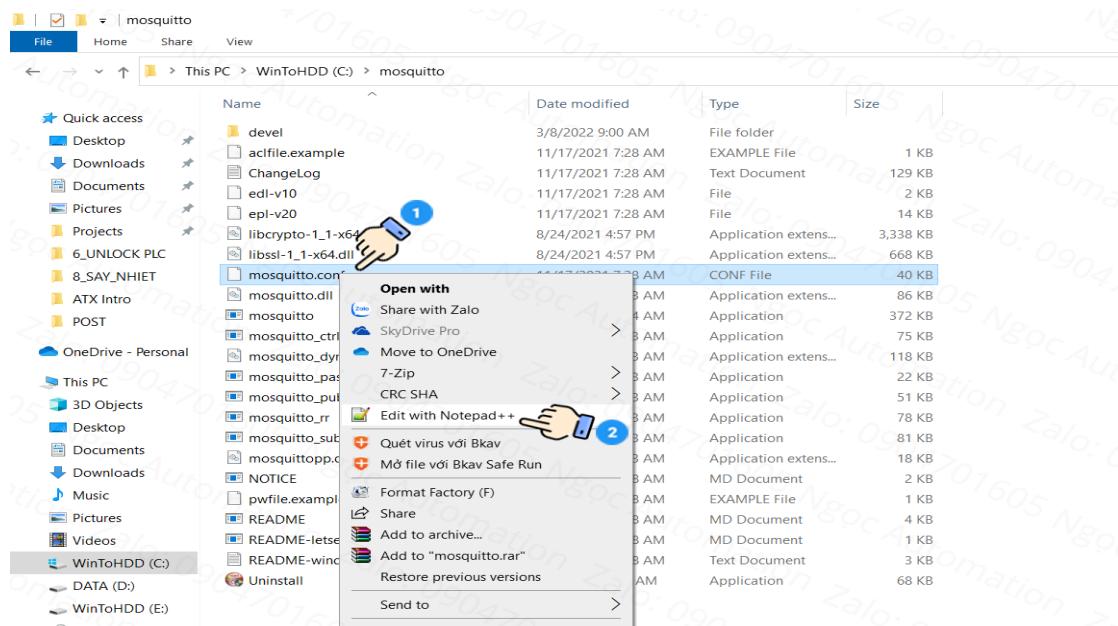


Hình 11: Cài đặt Mosquitto lên PC

Hình 10: Cài đặt Mosquitto lên PC

Cấu hình Mosquitto

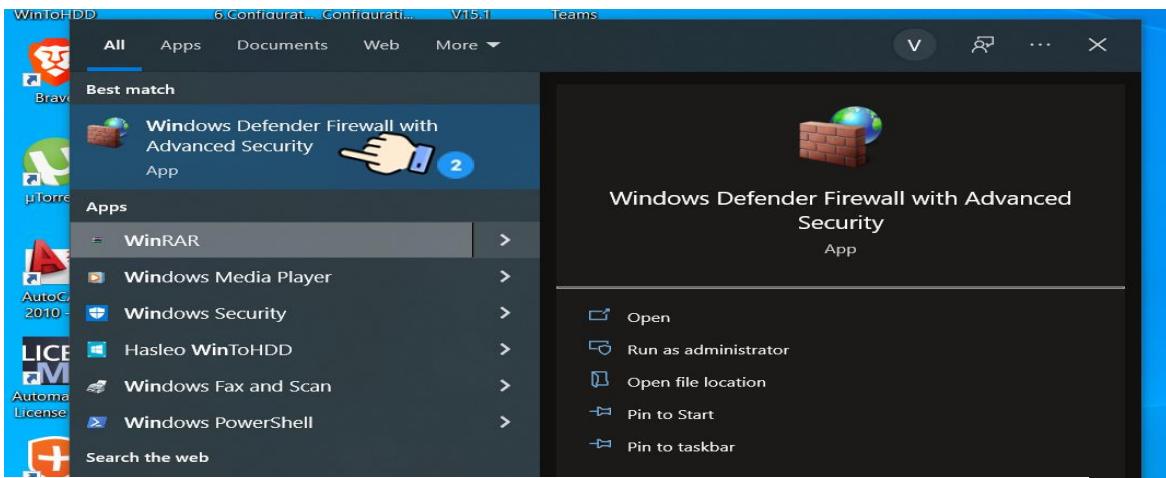
Bước 1: Vào folder vừa cài đặt phần mềm (C:\mosquitto) mở file config và dùng phần mềm notepad ++ để sửa;



Bước 2: Thêm 2 dòng lệnh sau vào file config của Mosquitto

listener 1883 allow_anonymous true

Bước 3: Mở window defender để thêm port cho Broker



Hình 12: Cấu hình Mosquitto



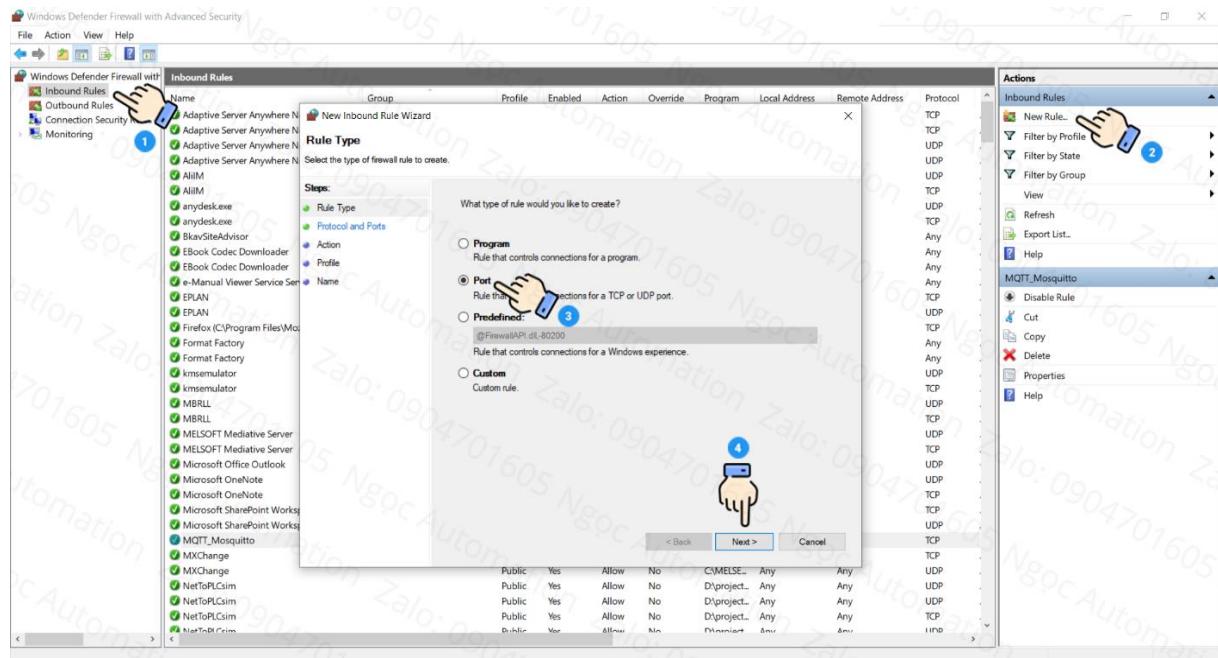
Hình 13: Cấu hình Mosquitto



Hình 14: Cấu hình Mosquitto

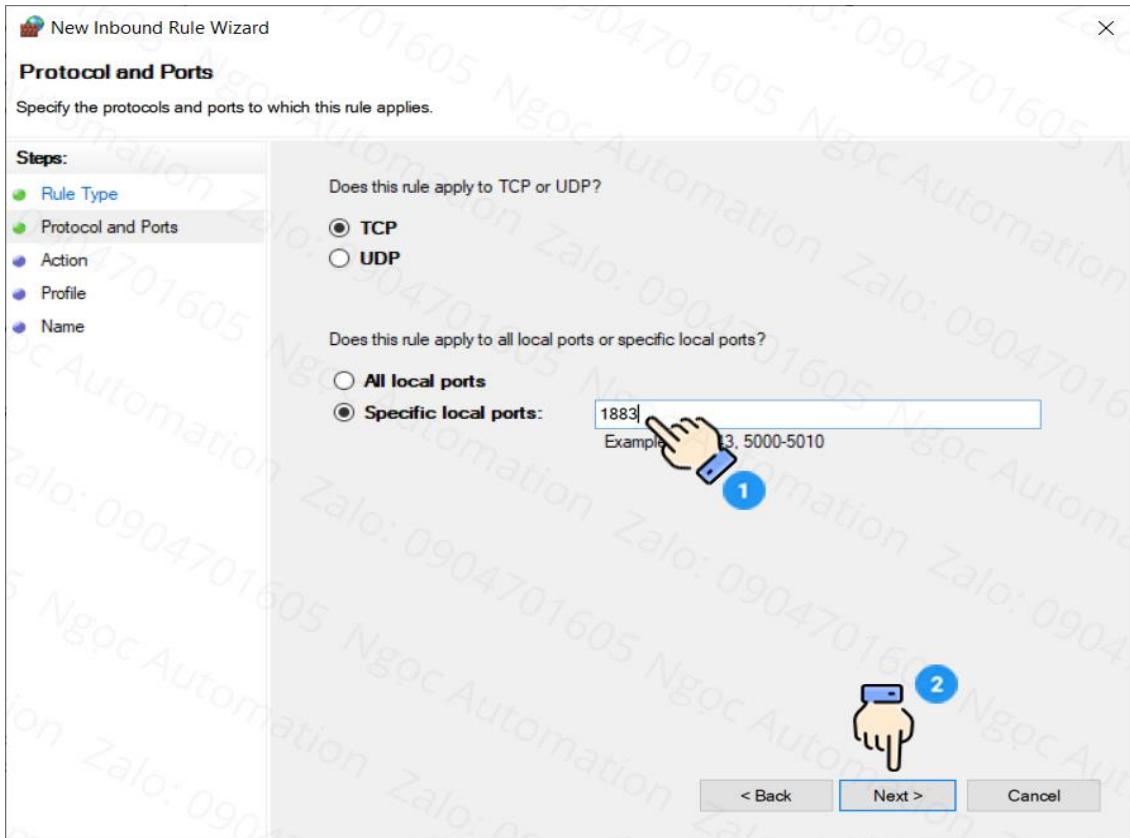
```
1 # Config file for mosquitto
2 #
3 # See mosquitto.conf(5) for more information.
4 #
5 # Default values are shown, uncomment to change.
6 #
7 # Use the # character to indicate a comment, but only if it is the
8 # very first character on the line.
9
10 # =====
11 # General configuration
12 # =====
13 listener 1883 
14 allow_anonymous true 
15
16 # Use per listener security settings.
17 #
18 # It is recommended this option be set before any other options.
19 #
20 # If this option is set to true, then all authentication and access control
21 # options are controlled on a per listener basis. The following options are
22 # affected:
23 #
24 # acl_file
25 # allow_anonymous
```

Bước 4: Thực hiện các bước sau để thêm port



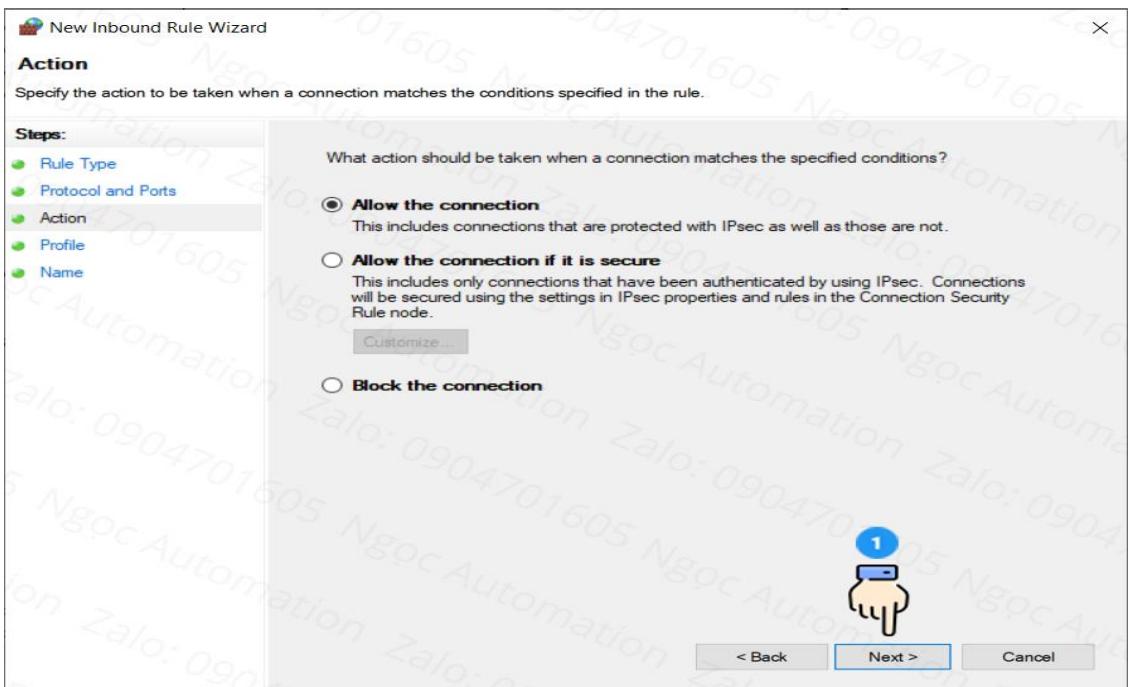
Hình 15: Cấu hình Mosquitto

Bước 5: Nhập port cho MQTT Broker là 1883 sau đó nhấn next



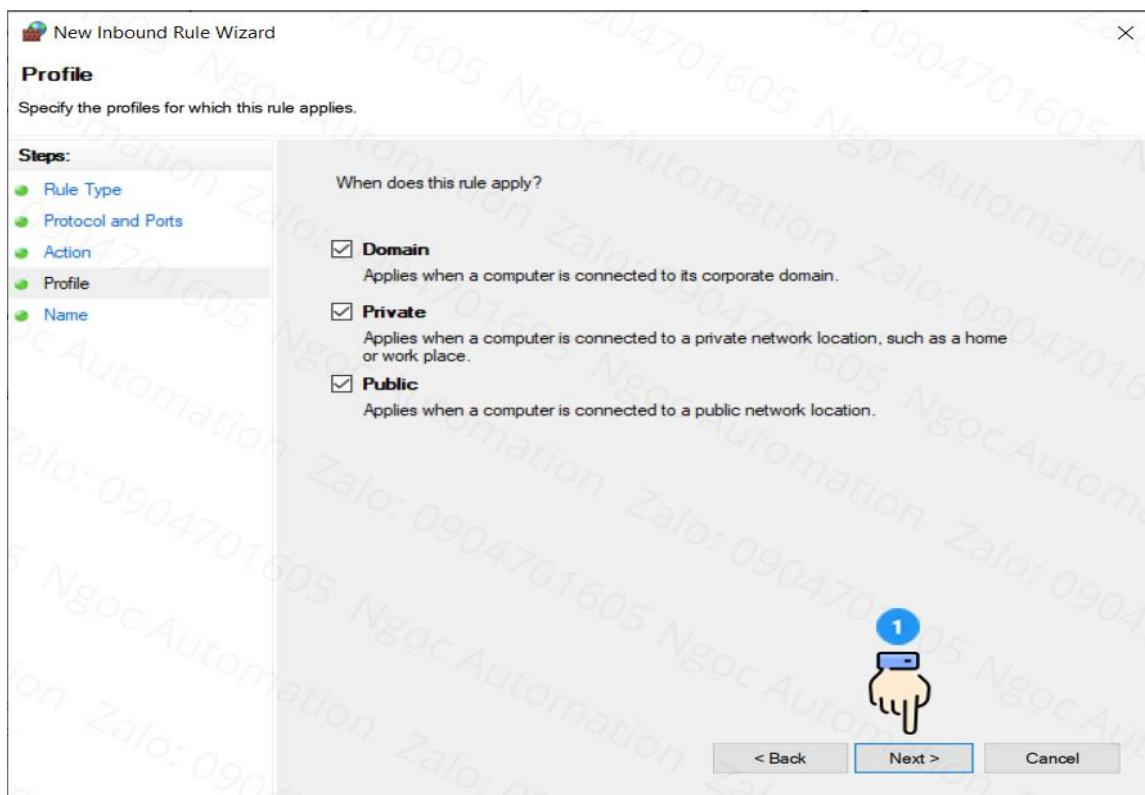
Hình 16: Cấu hình Mosquitto

Bước 6: Nhấn chọn next



Hình 17: Cấu hình Mosquitto

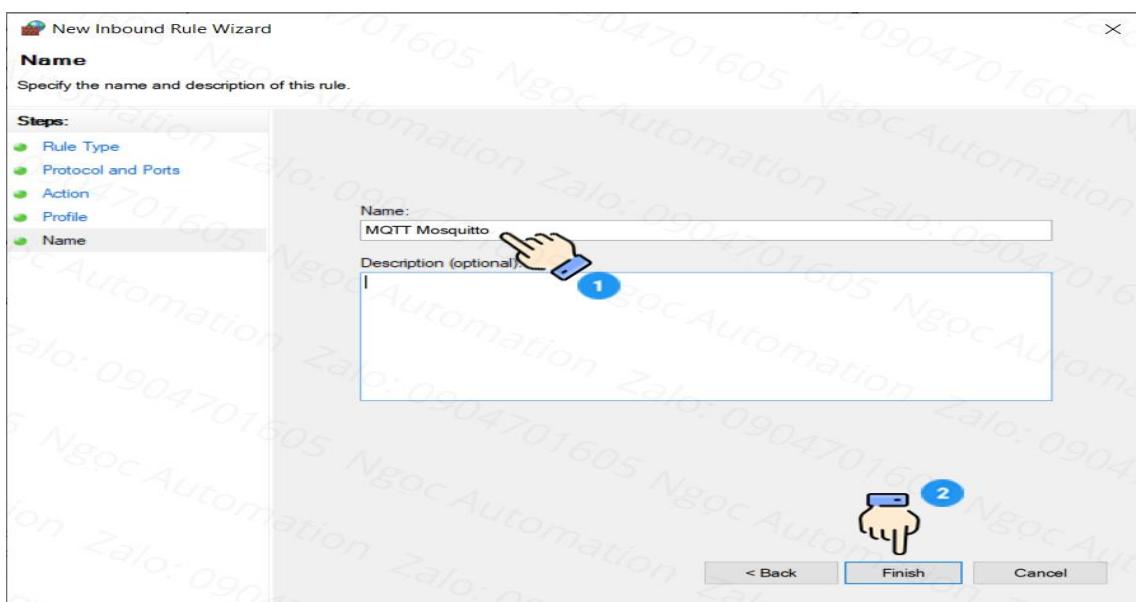
Bước 7: Tiếp tục nhấn next



Hình 18: Cấu hình Mosquitto

Bước 8: Đặt tên cho Rule (tên tùy chọn tiếng anh không dấu), sau đó nhấn finish để kết thúc

Với các bước trên, nhóm đã cấu hình thành công Mosquitto Broker lên máy tính của mình. Từ lúc này, máy tính sẽ đóng vai trò như máy chủ, nhận tin từ client và truyền tin đến client.

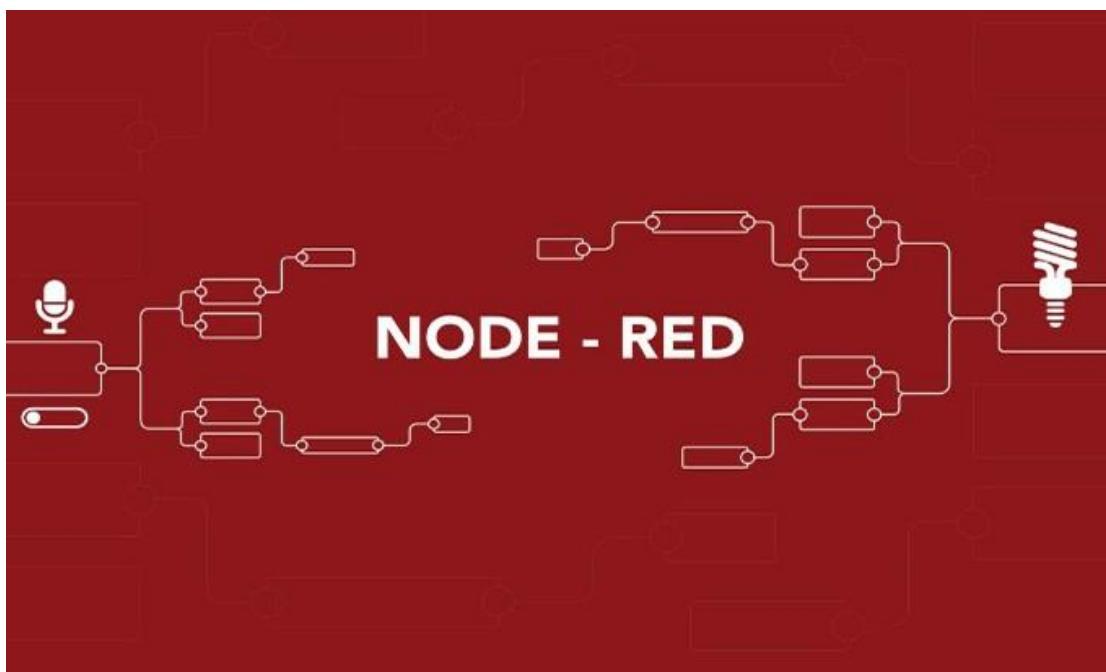


Hình 19: Cấu hình Mosquitto

2.2. Node – RED

2.2.1. Node – red là gì?

Trong quá trình tìm hiểu để xây dựng đề cương cho luận văn, chúng em nhận thấy việc cần xây một ứng dụng để hiển thị các dữ liệu thu thập được từ các cảm biến môi trường và điều khiển một số cơ cấu đơn giản của hệ thống giám sát cảnh báo cháy cho hộ gia đình. Ứng dụng sẽ có định dạng đơn giản, chạy trên web và hỗ trợ giao thức vận chuyển dữ liệu phổ biến như MQTT, và Node RED là một nền tảng khá phù hợp cho yêu cầu này. Đây là một công cụ lập trình dùng để kết nối các thiết bị phần cứng, API và các dịch vụ trực tuyến với nhau, có thể thấy rằng Node RED là một công cụ trực quan được thiết kế cho hệ thống IOT nói chung và nhà thông minh nói riêng.



Hình 20: NODE-RED

Node red chính là một công cụ lập trình, thực hiện vai trò kết 3 yếu tố bao gồm API, Phần cứng và online services. Không giống với các kết nối thông thường khác, Note red kết nối các yếu tố này lại theo cách hoàn toàn mới mẻ.

Node RED cung cấp cho lập trình viên một trình soạn thảo dựa trên trình duyệt giúp dễ dàng kết nối các luồng với nhau bằng cách sử dụng một loạt các node trong palette. Bản thân học viên không chuyên sâu về các ngôn ngữ lập trình gần với ngôn ngữ máy, cách lập trình trên nền kéo thả của Node RED sẽ giúp giảm đáng kể thời gian xây dựng ứng dụng. Chương trình phần mềm của Node RED cũng rất dễ dàng nhúng các API, các thư viện mở của bên thứ ba như API phục vụ đẩy dữ liệu lên MongoDB hay các thư viện kết hợp với Watson IoT để nâng cao trải nghiệm điều khiển của người sử dụng. Nhờ các yếu tố kể trên mà ứng

dụng của luận văn có thể được mở rộng sau này mà không gặp quá nhiều khó khăn.

Để thiết lập hoạt động của Node RED cần chuẩn bị môi trường có hệ điều hành Ubuntu để khởi chạy hệ thống. Đây là một hệ điều hành mã nguồn mở với rất nhiều tiện ích và là hệ điều hành thích hợp để phát triển các tính năng của hệ thống IoT nói chung và nhà thông minh trên nền tảng Node RED nói riêng. Ubuntu hoàn toàn miễn phí, và được chia sẻ rộng rãi trên trang chủ của Ubuntu với bản cập nhật mỗi sáu tháng một lần và luôn tăng cường bảo mật. Cách cài đặt Ubuntu vào máy tính cũng tương đối dễ dàng. Ngoài ra, người dùng cũng có thể chỉnh sửa tùy ý, sao chép hoặc cài tiến nó với giấy phép từ GNU GPL. Ubuntu hoạt động khá hiệu quả và tiêu tốn cực ít dung lượng phần cứng, đây là một điểm then chốt khi phát triển các thiết bị trong nhà thông minh bởi cần sự nhỏ gọn và tiết kiệm năng lượng. Điều này sẽ gia tăng tốc độ hoạt động của các máy tính đồng thời tăng hiệu suất hệ thống. Theo nhận định của nhiều chuyên gia về công nghệ thì cơ chế bảo mật của Ubuntu cao hơn so với Windows.

2.2.2. Ứng dụng của node red trong hệ thống internet

Note red là một sự cải cách vĩ đại trong dòng chảy của thời đại cách mạng công nghệ số 4.0 đó chính là khả năng kết nối hệ thống internet vạn vật. Trong thực tế, nếu đưa vấn đề đi theo hướng cơ bản thì mọi sự kết nối online đều cần phải vận dụng tới rất nhiều đối tượng với nhiều kiểu ngôn ngữ khác nhau.

Tuy nhiên các kỹ sư công nghệ thế hệ mới đã không cam tâm đi theo chiều hướng phức tạp và có phần thủ công đó. Họ hướng đến việc làm thế nào để có thể nhanh chóng lấy được dữ liệu vận hành. Và node red chính là một công cụ tuyệt vời giúp kết nối các IoT lại với nhau một cách dễ dàng, tạo ra những giá trị lợi ích vô cùng lớn.

Vậy thì ở cái nhìn cụ thể, node red sẽ được các chuyên gia công nghệ tận dụng những ứng dụng của nó phục vụ cho việc kết nối internet vạn vật như thế nào? Hãy đọc ngay nội dung được chia sẻ trong bài viết dưới đây.

2.2.2.1. Node red giúp loại bỏ mọi quy trình rườm rà trong hệ thống dây điện

Một số kỹ sư công nghệ đã tìm cách để có thể đơn giản hóa việc liên kết giữa hệ thống, thiết bị trong chiến lược phát triển hệ thống internet of things (**IoTs**). Trong nhận thức, ai cũng hiểu rõ, bất cứ sự kết nối nào cũng sẽ phải lặp đi lặp lại rất nhiều lần, những số lần không đếm xuể. Đó là một quá trình vô cùng phức tạp, khó khăn. Vì thế các kỹ sư trẻ đến từ nhóm Công nghệ của **IBM** muốn xây dựng hộp công cụ mã nhằm tối giản quá trình kết nối lặp đi lặp lại đó, nếu mục tiêu này thành công thì cơ hội hệ thống internet of things cũng sẽ được tăng tốc nhiều hơn.

Dù có cơ chế hoạt động khá đơn giản nhưng công cụ được tạo ra từ node red lại có thể đem đến khả năng kết nối dễ dàng nhiều sự kiện ở thế giới thực, có thể thêm dữ liệu thông tin hoặc truy cập được những nút đơn giản, từ đó tích hợp thành công chúng lại với nhau áp dụng vào các hệ thống nhắn tin cũng như ứng dụng cho những nền tảng mạng xã hội phổ biến bao gồm MongoDB, Twitter hay Watson IoT. Đây là cách tốt nhất để con người có thể tạo nên những ứng dụng có khả năng phản ứng lại với mọi thứ diễn ra trong thế giới của họ.

2.2.2.2. HẠ THANH KỸ THUẬT NHỜ NODE RED

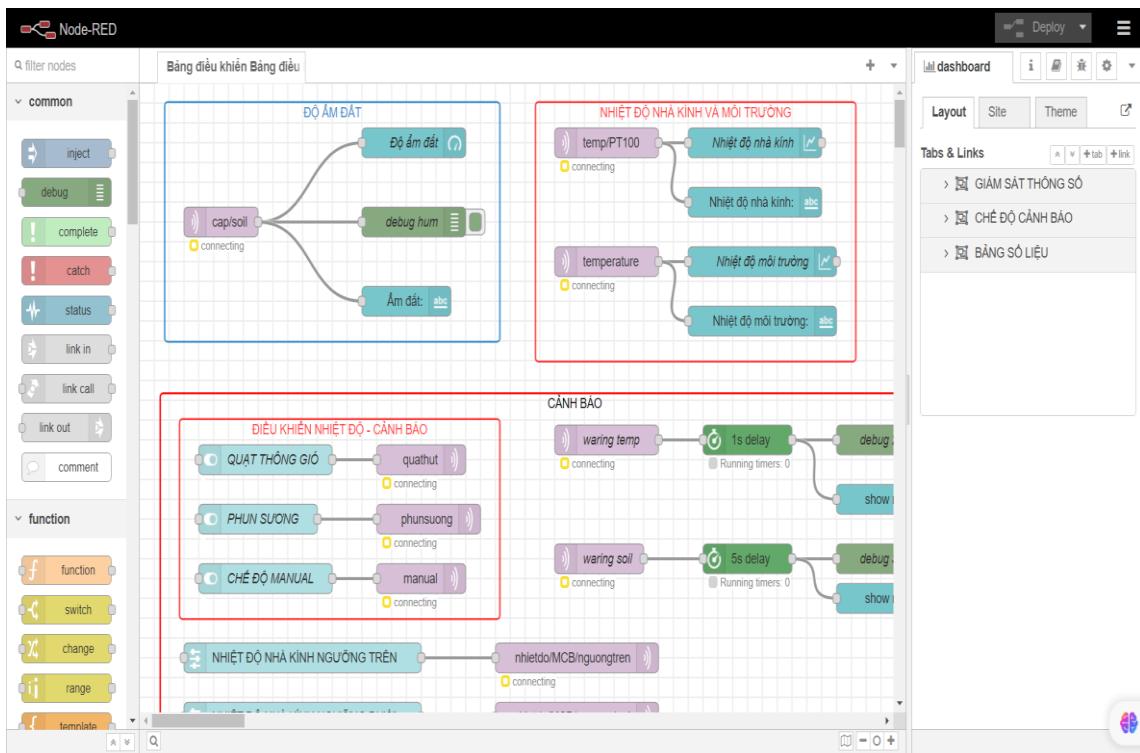
Node red có thể làm giảm đi tối đa nhu cầu viết mã, hạ thấp hơn những thanh kỹ thuật, giúp cho những ai đang quan tâm tới vấn đề cần phát triển IoT có thể thực hiện chức năng tạo. Chính vì thế mà node red trở nên phổ biến với rất nhiều đối tượng, nhiều hoàn cảnh do dễ dàng sử dụng.

Vai trò quan trọng của node red

Node red được đánh giá cao về độ ứng dụng của nó vì hầu hết mọi người đều có thể dễ dàng học được cách sử dụng nó, không phải chỉ lập trình viên mới có thể học và hiểu. So với IoT thì node red được đánh giá là công cụ hữu ích vô cùng trong việc kết nối vạn vật một cách hiệu quả trên internet.

2.2.3. MỘT SỐ THAO TÁC CƠ BẢN CỦA NODE - RED

Node RED vốn là một công cụ lập trình có nhiệm vụ nhanh chóng tạo ra kết nối các thiết bị phần cứng mà cụ thể là các cảm biến, các cơ cấu chấp hành có thể kết nối vào mạng, đồng thời cho phép trao đổi dữ liệu bằng cách tích hợp các API và các dịch vụ trực tuyến với nhau. Node-RED được xây dựng bằng Node.JS. Do đó, Node RED có thể chạy dễ dàng trên bất cứ trình duyệt web nào. Giao diện Node RED được chia làm ba phần đó là khu vực input nơi chứa các Block để xử lý luồng đầu vào, khu vực xây dựng các luồng xử lý đó là các Flow, nơi có thể chứa các Block và biểu diễn quan hệ giữa các Block với nhau. Khu vực output chứa thông tin và các Block, cách cấu hình chúng và debug.

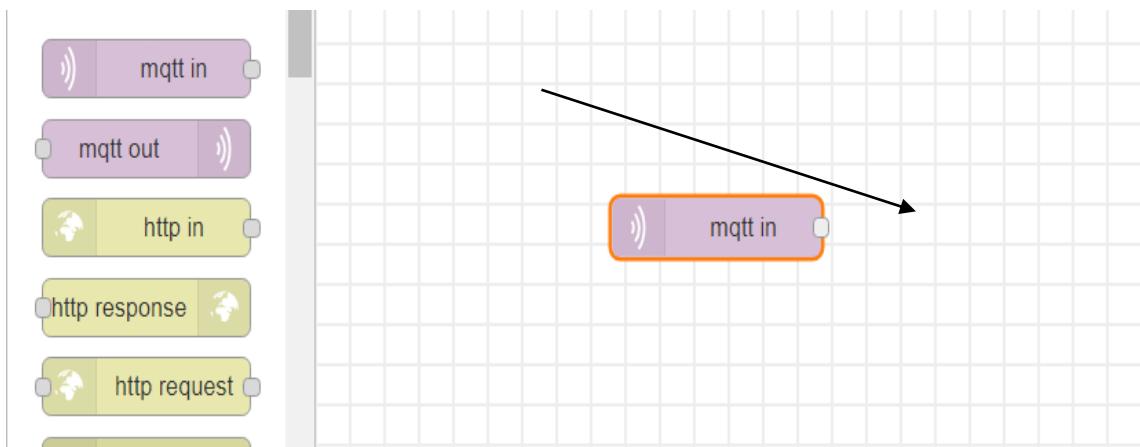


Hình 21: Cửa sổ làm việc của Node RED

Cửa sổ soạn thảo gồm 4 thành phần chính:

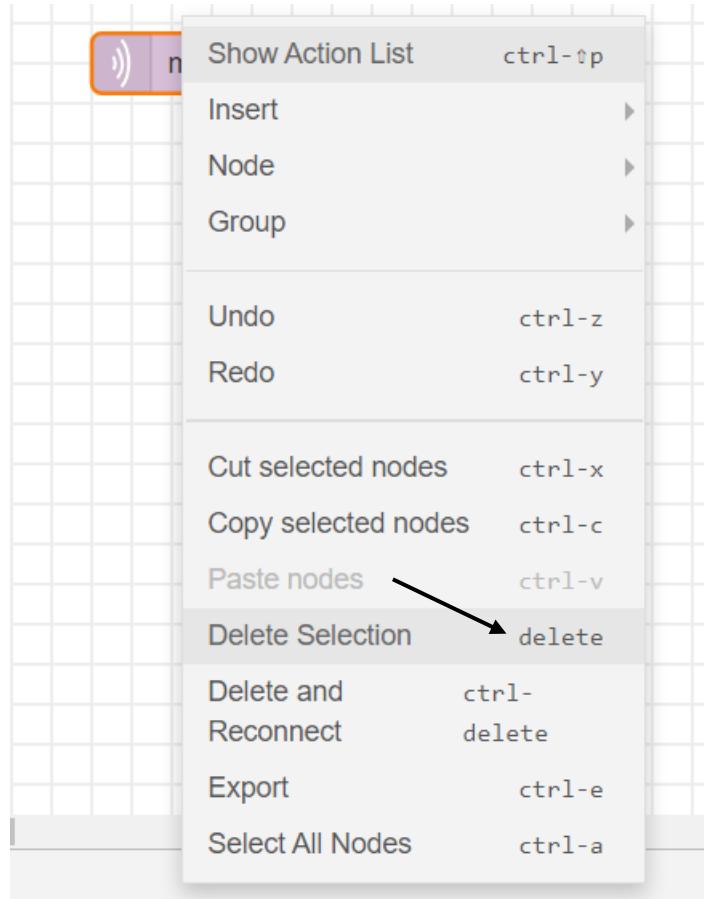
- Tiêu đề ở trên, chứa nút Deploy, menu chính.
- Bảng màu (palette) bên trái, chứa các nút có sẵn để sử dụng.
- Không gian làm việc chính (workspace) ở giữa, nơi các luồng được tạo.
- Thanh sidebar bên phải.

Trong môi trường Node RED các chương trình được tổ chức dưới dạng các Flow, các Flow có thể chạy song song và độc lập với nhau, người lập trình cũng có thể thêm, sửa xóa các Flow hết sức dễ dàng.



Hình 22 Tạo Flow trong Node RED

Để xóa một Flow chỉ cần kích đúp vào tab của Flow đó và chọn Delete ở phần panel hiện ra bên phải, cửa sổ này còn cho phép đổi tên và viết mô tả, giới thiệu các thông tin cơ bản của luồng.



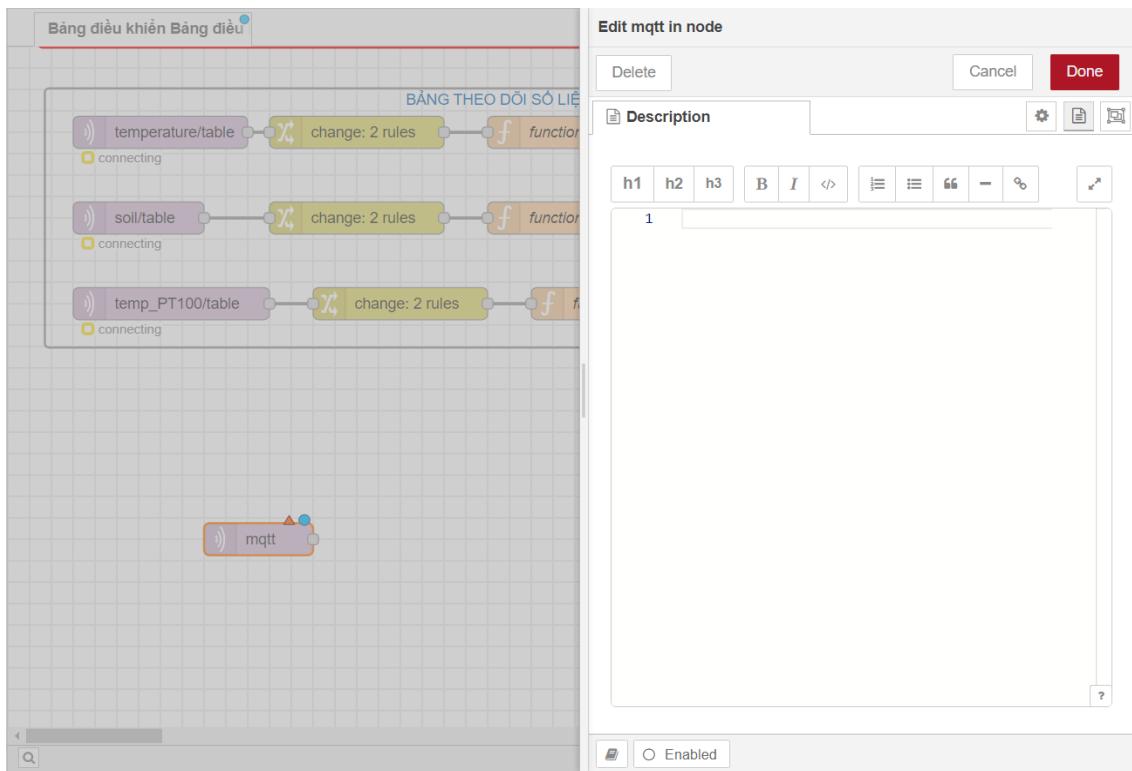
Hình 23: Cách xóa Flow trong Node RED

Việc thao tác giữa các luồng Flow cũng cần quan tâm đến việc Export (kết xuất dữ liệu) để tiến hành lưu trữ, đồng thời Import dữ liệu từ Flow trở lại màn hình làm việc. Node RED cho phép xuất nhập file dưới định dạng văn bản thông thường nên rất thuận tiện cho việc thay đổi và hiệu chỉnh chương trình. Các thao tác sao chép, di chuyển node giữa các Flow cũng được thực hiện hết sức dễ dàng qua các bước:

Bước 1: khôi chọn những nodes muốn sao chép hoặc di chuyển

Bước 2: dùng Ctrl + C để sao chép các nodes; dùng Ctrl + X để di chuyển các nodes
Bước 3: chuyển sang Flow khác và dùng Ctrl + V để dán các nodes lại

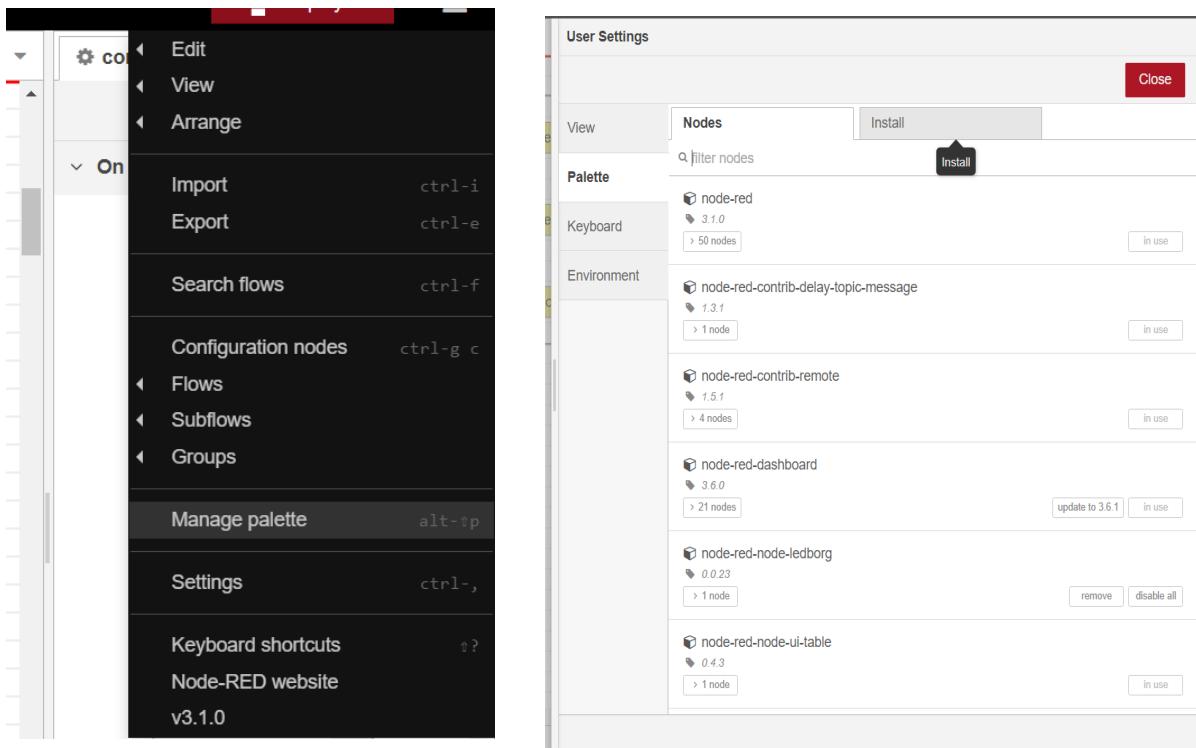
Mặc dù là Node-RED cung cấp cho chúng ta một cách lập trình trực quan, dễ hiểu. Tuy nhiên, việc ghi chú lại chức năng, cách sử dụng các nodes / flow là rất cần thiết. Vì nếu người dùng share Flow cho cộng đồng thì những ghi chú đó sẽ rất có giá trị. Đây là lý do Node RED cung cấp một node khá hữu ích đó là Comment. Để sử dụng, chỉ cần kéo thả node Comment vào workspace, sau đó double-click vào node Comment để thêm nội dung ghi chú.



Hình 24: Cửa sổ ghi chú thích của Node RED

Ngoài cách export ra Clipboard, Node-RED còn hỗ trợ người dùng export các Flow / Nodes ra thành Library. Dĩ nhiên cách này sẽ thuận tiện cho chúng ta hơn rất nhiều khi làm các dự án lớn vì không cần copy / paste phần export nữa. Để bắt đầu, người dùng khởi chọn các nodes muốn export và chọn **Menu -> Export -> Library**.

Sau đó điền tên Library theo ý muốn. Người dùng cũng có thể phân cấp Library bằng cách dùng dấu /. Như hình bên dưới có nghĩa là Library sẽ nằm trong thư mục mechasolution. Để sử dụng Library vừa tạo, người dùng chỉ việc vào **Menu -> Import -> Library -> mechasolution -> first-flow**



Hình 25: Ví trí thiết lập thư viện trong Node RED

Như vậy có thể thấy rằng Node RED cung cấp cho người dùng một môi trường làm việc với cách đối tượng được thiết kế sẵn và hỗ trợ tối đa việc tùy biến đối tượng cũng như phát triển các đối tượng mới. Việc nhúng các API vào trong Node RED cũng rất dễ dàng, từ đó liên kết hệ thống trở nên đơn giản hơn. Cơ bản các nhà phát triển sẽ sử dụng Node RED làm trạm trung chuyển để nhúng các giao thức mạng như MQTT, CoAP, http... đồng thời phát triển nhanh giao diện người dùng dashboard.

2.2.4. Úng dụng node – red vào đề tài

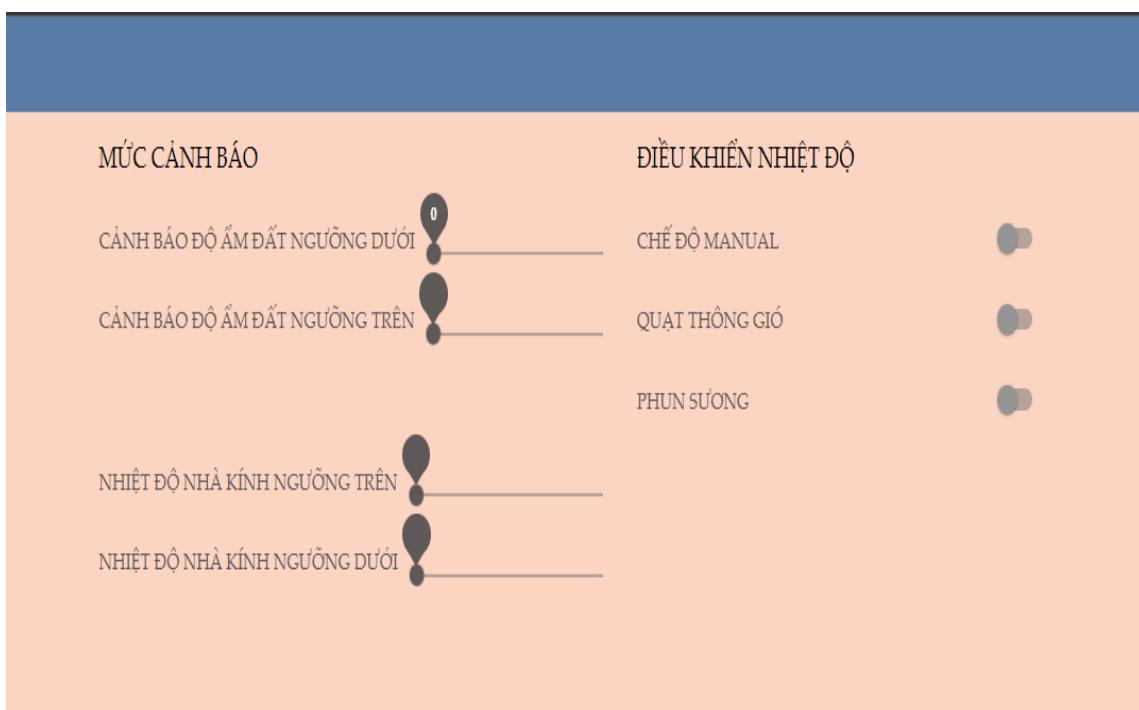
Như đã nói, Node – red là một phần mềm thiết kế giao diện, cụ thể đối với đề tài chính là giao diện để giám sát và điều khiển hệ thống chăm sóc cây húng quế trong nhà kính. Qua tìm hiểu, chúng em nhận thấy rằng, Node – red vừa là một phần mềm đơn giản để học hỏi và sử dụng nhưng cũng chứa đựng rất nhiều chức năng nhằm đảm bảo cho tính khả thi của đồ án nhóm chúng em.

Thông qua Node – red chúng em tiến hành kiểm soát 3 yếu tố nhiệt độ môi trường, nhiệt độ nhà kính và độ ẩm đất. Với Tab 1, chúng em thực hiện theo dõi thông số của 3 yếu tố trên đang ở mức độ nào, có phù hợp với cây húng quế hay không (Tab chính để theo dõi nhà kính từ xa).



Hình 26: Giao diện Giám sát Node-Red

Với tab 2, Node – red giúp nhóm em tiến hành điều chỉnh thông số, kiểm soát lại nhiệt độ cũng như độ ẩm để cây trồng phát triển một cách tốt nhất (Khi có những cảnh báo ở Tab 1 như hình trên, tụi em sẽ thực hiện điều chỉnh cho phù hợp lại với cây trồng như Phun sương để tăng độ ẩm và bật quạt thông hơi để giảm nhiệt độ).



Hình 27: Giao diện Điều khiển Node-Red

Với Tab 3, Node – red có chức năng giúp tự em liệt kê lại quá trình và diễn biến thay đổi ở trong nhà kính qua từng đơn vị giây nhằm để phát hiện nhanh chóng những vấn đề khi cây húng quế bị ảnh hưởng và nhận biết được cây húng quế phát triển trong môi trường ra sao.

Ngày/Giờ	Nhiệt độ Môi Trường	Độ Ẩm Độ	Ngày/Giờ	Nhiệt độ Nhà Kính
11/3/2023, 5:59:56 PM			11/3/2023, 5:59:56 PM	23.85
11/3/2023, 5:59:56 PM		0	11/3/2023, 5:59:56 PM	
11/3/2023, 5:59:56 PM	25.8		11/3/2023, 5:59:56 PM	
11/3/2023, 5:59:46 PM			11/3/2023, 5:59:46 PM	23.85
11/3/2023, 5:59:46 PM		0	11/3/2023, 5:59:46 PM	
11/3/2023, 5:59:46 PM	25.8		11/3/2023, 5:59:46 PM	
11/3/2023, 5:59:35 PM			11/3/2023, 5:59:35 PM	23.79
11/3/2023, 5:59:35 PM		0	11/3/2023, 5:59:35 PM	
11/3/2023, 5:59:35 PM	25.8		11/3/2023, 5:59:35 PM	
11/3/2023, 5:59:26 PM			11/3/2023, 5:59:26 PM	23.79
11/3/2023, 5:59:26 PM		0	11/3/2023, 5:59:26 PM	
11/3/2023, 5:59:26 PM	25.8		11/3/2023, 5:59:26 PM	
11/3/2023, 5:59:18 PM			11/3/2023, 5:59:18 PM	23.75
11/3/2023, 5:59:18 PM		0	11/3/2023, 5:59:18 PM	
11/3/2023, 5:59:18 PM	25.8		11/3/2023, 5:59:18 PM	
11/3/2023, 5:59:05 PM			11/3/2023, 5:59:05 PM	23.85
11/3/2023, 5:59:05 PM		0	11/3/2023, 5:59:05 PM	
11/3/2023, 5:59:05 PM	24.9		11/3/2023, 5:59:05 PM	
11/3/2023, 5:59:01 PM			11/3/2023, 5:59:01 PM	23.85
11/3/2023, 5:59:01 PM		0	11/3/2023, 5:59:01 PM	
11/3/2023, 5:59:01 PM	24.8		11/3/2023, 5:59:01 PM	
11/3/2023, 5:58:47 PM			11/3/2023, 5:58:47 PM	23.72
11/3/2023, 5:58:47 PM		0	11/3/2023, 5:58:47 PM	
11/3/2023, 5:58:47 PM	24.8		11/3/2023, 5:58:47 PM	
11/3/2023, 5:58:35 PM			11/3/2023, 5:58:35 PM	23.82
11/3/2023, 5:58:35 PM		0	11/3/2023, 5:58:35 PM	

Hình 28: Lịch sử hoạt động Node-Red

2.3. Tổng quan về PID (Proportional Integral Derivative)

2.3.1. PID là gì?

PID là tên viết tắt của Proportional Integral Derivative. Đây là một cơ chế phản hồi vòng điều khiển đã và đang được dùng phổ biến trong các hệ thống điều khiển công nghiệp. Bộ điều khiển này đã dùng nhiều nhất trong các hệ thống vòng điều khiển kín (có tín hiệu phản hồi).

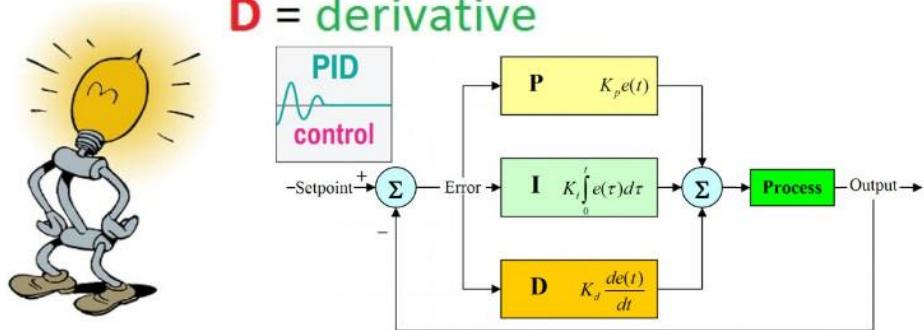
Bộ điều khiển này có thể tính toán được giá trị sai số là hiệu số giữa giá trị đặt mong muốn và giá trị của thông số biến đổi. PID có thể làm giảm tối đa tình trạng sai số bằng cách điều chỉnh các giá trị điều khiển ngay từ đầu vào.

Và để đạt được một kết quả tốt nhất, các thông số từ bộ điều khiển PID dùng trong tính toán cần phải điều chỉnh dựa vào tính chất của hệ thống. Đồng thời, nó cũng phụ thuộc vào những đặc thù của hệ thống.

P = proportional

I = integral

D = derivative



Hình 29: Minh họa thuật toán PID

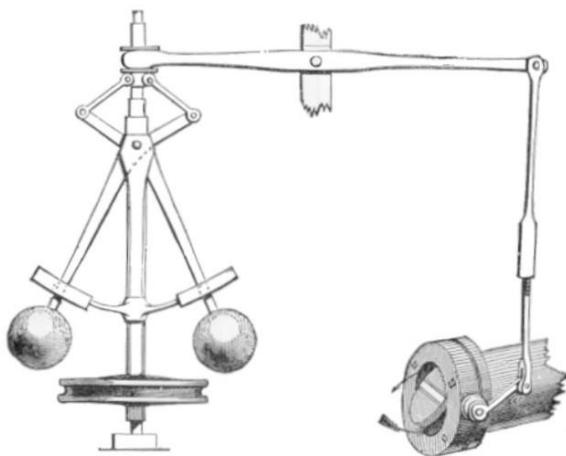
Trong đó:

P – Proportional: Đây là phương pháp dùng để điều chỉnh tỉ lệ và tạo ra các tín hiệu điều chỉnh tỷ lệ với phần sai lệch đầu vào dựa vào thời gian để lấy mẫu.

I – Integral: Đây là tích phân của sai số dựa vào thời gian lấy mẫu. Nó cũng là một phương pháp dùng để điều chỉnh nhằm tạo ra tín hiệu điều chỉnh sao cho độ sai lệch sẽ giảm tới 0. Từ đó, chúng ta cũng biết chính xác được tổng sai số tức thời theo thời gian hoặc các sai số tích lũy ở quá khứ. Khi mà thời gian càng ngắn thì việc điều chỉnh tích phân sẽ càng mạnh. Tương tương với đó chính là độ lệch sẽ càng thấp.

D – Derivative: Đây là vi phân sai lệch. Nó cũng để tạo ra các tín hiệu điều chỉnh sao cho tốc độ và tỷ lệ thay đổi sai lệch đầu vào. Thời gian càng dài thì phạm vi để điều chỉnh vi phân sẽ càng lớn. Điều này cũng tương ứng với bộ điều chỉnh có thể đáp ứng tốt những thay đổi từ đầu vào càng nhanh chóng.

2.3.2. Lịch sử ra đời của PID:



Hình 30: Lịch sử ra đời PID

Điều khiển liên tục, trước khi bộ điều khiển PID được hiểu và thực hiện đầy đủ, có một trong những nguồn gốc của nó trong bộ điều khiển ly tâm, sử dụng trọng lượng quay để điều khiển một quá trình. Điều này được phát minh bởi Christiaan Huygens vào thế kỷ 17 để điều chỉnh khoảng cách giữa các cối xay trong cối xay gió tùy thuộc vào tốc độ quay, và do đó bù cho tốc độ thay đổi của thức ăn ngũ cốc.

Với việc phát minh ra động cơ hơi nước tinh áp suất thấp, cần phải điều khiển tốc độ tự động, và bộ điều khiển "con lắc hình nón" tự thiết kế của James Watt, một bộ bi thép quay được gắn vào trục chính thẳng đứng bằng các cánh tay liên kết, đã trở thành một tiêu chuẩn công nghiệp. Điều này dựa trên khái niệm kiểm soát khe hở cối xay.

Tuy nhiên, điều khiển tốc độ xoay vẫn thay đổi trong các điều kiện tải trọng khác nhau, trong đó thiếu sót của cái mà ngày nay được gọi là điều khiển tỷ lệ là rõ ràng. Sai số giữa tốc độ mong muốn và tốc độ thực tế sẽ tăng lên khi tải tăng. Vào thế kỷ 19, cơ sở lý thuyết cho hoạt động của các thống đốc lần đầu tiên được James Clerk Maxwell mô tả vào năm 1868 trong bài báo nổi tiếng hiện nay của ông về các thống đốc. Ông đã khám phá cơ sở toán học cho sự ổn định kiểm soát, và tiến bộ một cách tốt hướng tới một giải pháp, nhưng đã kêu gọi các nhà toán học kiểm tra vấn đề. Vấn đề đã được kiểm tra thêm vào năm 1874 bởi Edward Routh, Charles Sturm, và vào năm 1895, Adolf Hurwitz, tất cả đều góp phần thiết lập các tiêu chí ổn định kiểm soát. Trong các ứng dụng tiếp theo, các bộ điều khiển tốc độ đã được tinh chỉnh thêm, đáng chú ý là bởi nhà khoa học người Mỹ Willard Gibbs, người vào năm 1872 đã phân tích về mặt lý thuyết thống đốc con lắc hình nón của Watt.

Vào khoảng thời gian này, việc phát minh ra ngư lôi Whitehead đặt ra một vấn đề điều khiển đòi hỏi phải kiểm soát chính xác độ sâu chạy. Chỉ riêng việc sử dụng một cảm biến áp suất sâu tỏ ra không đủ, và một con lắc đo khoảng cách phía trước và phía sau của ngư lôi được kết hợp với đo độ sâu để trở thành bộ điều khiển con lắc và thủy lực. Kiểm soát áp suất chỉ cung cấp một điều khiển tỷ lệ, nếu độ lợi điều khiển quá cao, sẽ trở nên không ổn định và đi vào tình trạng quá mức với sự mất ổn định đáng kể của việc giữ độ sâu. Con lắc đã thêm cái mà ngày nay được gọi là điều khiển phái sinh, làm giảm dao động bằng cách phát hiện góc lăn / leo ngư lôi và do đó tốc độ thay đổi độ sâu. Sự phát triển này (được Whitehead đặt tên là "Bí mật" để không đưa ra manh mối nào về hành động của nó) là vào khoảng năm 1868.

Một ví dụ ban đầu khác về bộ điều khiển kiểu PID được Elmer Sperry phát triển vào năm 1911 để điều khiển tàu, mặc dù công việc của ông là trực quan hơn là dựa trên toán học.

Tuy nhiên, mãi đến năm 1922, một luật kiểm soát chính thức cho cái mà ngày nay chúng ta gọi là PID hoặc kiểm soát ba kỳ lần đầu tiên được phát triển bằng cách sử dụng phân tích lý thuyết, bởi kỹ sư người Mỹ gốc Nga Nicolas Minorsky. Minorsky đã nghiên cứu và thiết kế hệ thống lái tàu tự động cho Hải quân Hoa Kỳ và dựa trên phân tích của ông dựa trên những quan sát của một người lái tàu. Ông lưu ý rằng người lái tàu đã điều khiển con tàu không chỉ dựa trên lỗi hướng đi hiện tại mà còn dựa trên lỗi trong quá khứ, cũng như tốc độ thay đổi hiện tại; điều này sau đó đã được Minorsky đưa ra một phương pháp điều trị toán học. Mục tiêu của ông là sự ổn định, không phải kiểm soát chung, điều này đã đơn giản hóa vấn đề một cách đáng kể. Mặc dù kiểm soát tỷ lệ cung cấp sự ổn định chống lại các nhiễu nhỏ, nhưng nó không đủ để đối phó với sự xáo trộn ổn định, đáng chú ý là một cơn gió mạnh (do lỗi trạng thái ổn định), đòi hỏi phải thêm thuật ngữ tích phân. Cuối cùng, thuật ngữ phái sinh đã được thêm vào để cải thiện tính ổn định và kiểm soát.

Các thử nghiệm đã được thực hiện trên USS New Mexico, với các bộ điều khiển điều khiển vận tốc góc (không phải góc) của bánh lái. Kiểm soát PI mang lại yaw (sai số góc) duy trì $\pm 2^\circ$. Thêm phần tử D mang lại sai số ngáp $\pm 1/6^\circ$, tốt hơn hầu hết các người lái xe có thể đạt được.

Hải quân cuối cùng đã không áp dụng hệ thống này do sự kháng cự của nhân viên. Công việc tương tự đã được thực hiện và xuất bản bởi một số người khác trong những năm 1930.

2.3.3. Các loại bộ điều khiển PID

Bộ điều khiển này có thể phân thành các loại:

- PI hay có tên gọi đầy đủ là *Proportional and Integral Controller* là bộ điều khiển tỉ lệ, tích phân.
- P là *Proportional Controller*, bộ điều khiển tỉ lệ.
- PD là *Proportional and Derivative Controller* là bộ điều khiển đạo hàm.

2.3.4. Vì sao lại cần PID?

Bộ điều khiển PID (Proportional-Integral-Derivative) là một phần quan trọng trong hệ thống điều khiển tự động. Nó được sử dụng để điều chỉnh đầu ra của một hệ thống dựa trên sự đánh giá liên tục của sai số giữa giá trị đầu ra hiện tại và giá trị đặt trước (đầu vào).

Dưới đây là một số lý do chúng ta cần bộ điều khiển PID:

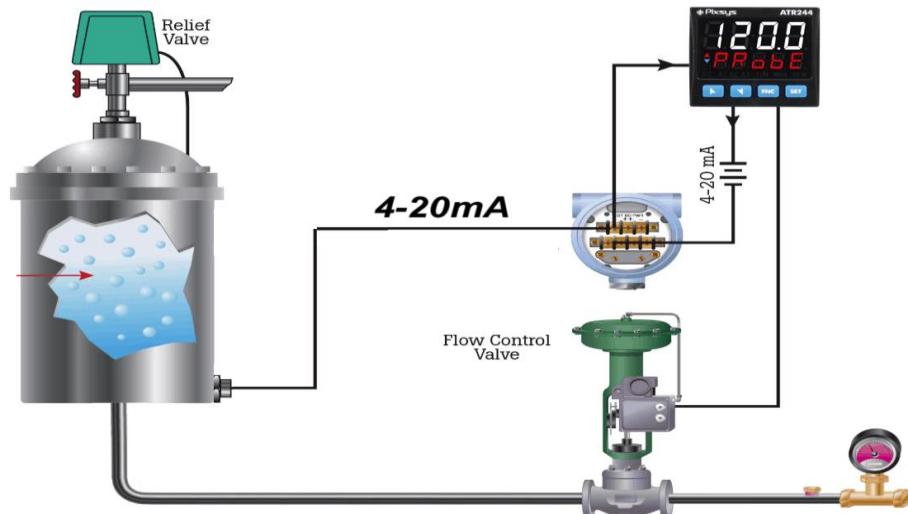
- Độ ổn định: Bộ điều khiển PID giúp duy trì sự ổn định của hệ thống. Nó phản hồi nhanh chóng đến các thay đổi trong đầu vào và điều chỉnh đầu ra để đảm bảo rằng hệ thống duy trì trạng thái cân bằng.
- Độ chính xác: Bộ điều khiển PID giúp giảm sai số giữa giá trị đầu ra thực tế và giá trị đặt trước. Bằng cách liên tục đánh giá sai số và điều chỉnh đầu ra, nó giúp hệ thống đạt được độ chính xác cao hơn.
- Đáp ứng nhanh: PID có khả năng điều chỉnh đầu ra nhanh chóng và linh hoạt đối với các thay đổi trong đầu vào. Điều này cho phép hệ thống phản ứng và thích ứng nhanh chóng với các điều kiện thay đổi, giúp đạt được hiệu suất tối ưu.
- Dễ dàng thực hiện: Bộ điều khiển PID có cấu trúc đơn giản và dễ dàng thực hiện trong các hệ thống điều khiển. Các thông số PID (proportional, integral, derivative) có thể được điều chỉnh để đáp ứng yêu cầu cụ thể của hệ thống.
- Ứng dụng rộng rãi: Bộ điều khiển PID được sử dụng phổ biến trong nhiều lĩnh vực như điều khiển công nghiệp, điều khiển quá trình, tự động hóa, robot học và nhiều ứng dụng khác. Sự linh hoạt và hiệu quả của PID làm cho nó trở thành một công cụ hữu ích trong việc điều khiển và điều chỉnh các hệ thống phức tạp.

2.3.5. Một số ứng dụng của bộ điều khiển PID

Hiện nay PID được ứng dụng trong nhiều ngành nghề khác nhau.

- Giảm các sai số, hạn chế sự dao động hay là giảm thời gian gian xác lập và độ vọt lô...

- Sử dụng để điều khiển mực nước: bộ điều khiển được tự động hóa nhờ vào các thiết bị điện tử như cảm biến, van điều khiển...
- Điều khiển biến tần: Các thiết bị điện tử kết hợp ở đây gồm có: van điều khiển lưu lượng, cảm biến nhiệt độ, biến tần điều khiển....
- Kiểm soát lưu lượng khí qua đường ống
- Điều khiển PID trong PLC: Ở trong PLC thường sẽ được thiết kế sẵn các hàm dùng để điều chỉnh nhiệt độ, áp suất, lưu lượng...



Hình 31: **Điều khiển mực nước bằng PID**

2.3.6. Ứng dụng PID về điều khiển nhiệt độ:

Một ví dụ cụ thể về việc sử dụng bộ điều khiển PID để điều khiển nhiệt độ là hệ thống điều khiển nhiệt độ của một lò nung.

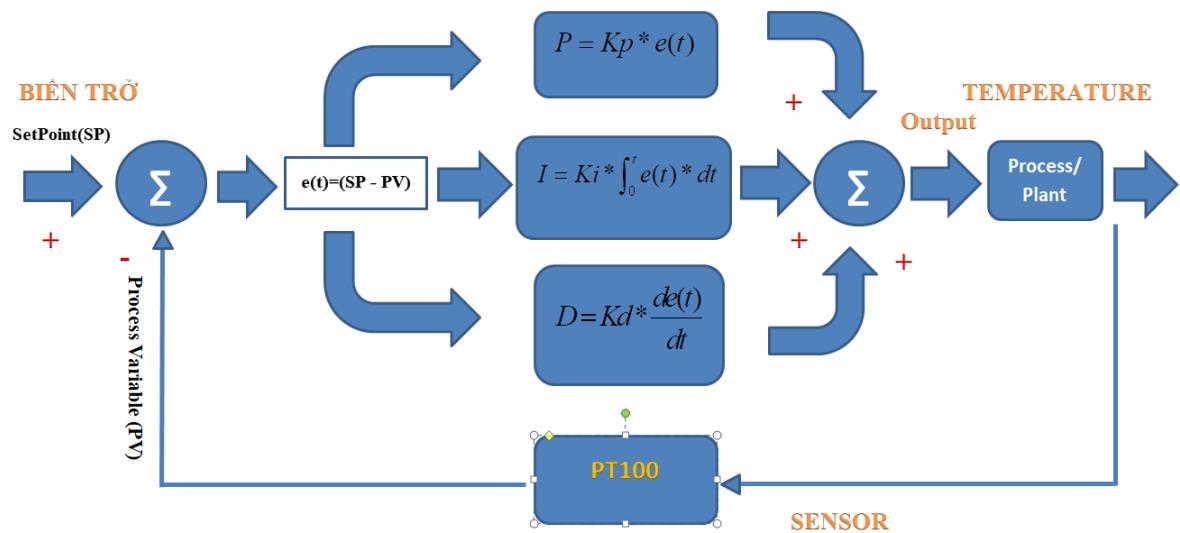
Trong một lò nung, bộ điều khiển PID có thể được sử dụng để duy trì nhiệt độ trong lò ở một mức ổn định. Các thành phần của bộ điều khiển PID được cấu hình như sau:

- Proportional (P): Đầu vào của thành phần P sẽ đo lường hiện tại của nhiệt độ trong lò nung và so sánh với giá trị đặt trước (setpoint). Sai số giữa hai giá trị này sẽ được nhân với hệ số tỷ lệ (gain) để tính toán giá trị điều chỉnh đầu ra của P.
- Integral (I): Thành phần I tích lũy các sai số trong thời gian, và điều chỉnh đầu ra dựa trên tổng sai số tích lũy. Điều này giúp giảm thiểu sai số ổn định dài hạn và đảm bảo rằng nhiệt độ trong lò nung đạt được giá trị đặt trước.
- Derivative (D): Thành phần D đo tốc độ thay đổi của nhiệt độ trong lò nung. Nếu nhiệt độ thay đổi quá nhanh, thành phần D sẽ tăng đầu ra để giảm thiểu độ dao động và giữ cho nhiệt độ ổn định hơn.

Bằng cách kết hợp ba thành phần trên, bộ điều khiển PID sẽ điều chỉnh đầu ra (ví dụ như công suất điện đưa vào lò) để duy trì nhiệt độ trong lò nung ở mức ổn định và gần giá trị đặt trước.

Trong quá trình hoạt động, bộ điều khiển PID liên tục đo nhiệt độ hiện tại, so sánh với giá trị đặt trước và tính toán giá trị điều chỉnh dựa trên các thông số được cấu hình. Sau đó, nó gửi tín hiệu điều chỉnh cho hệ thống điện tử của lò nung để điều chỉnh công suất điện đưa vào và duy trì nhiệt độ ổn định.

2.3.7. Công thức PID:



Hình 32: Các công thức PID sử dụng trong đồ án

Bộ điều khiển PID là một bộ điều khiển vòng kín được sử dụng rộng rãi trong hệ thống điện, hệ thống tự động, điện tử. Mục tiêu của bộ điều khiển PID là điều chỉnh giá trị điều khiển ở ngõ ra Output sao cho sai lệch Error $e(t) = (SP - PV)$ giữa giá trị đo được của hệ thống PV (Process Variable) với giá trị cài đặt SP (SetPoint) nhỏ nhất có thể (~ 0), đạt được sự ổn định và có đáp ứng nhanh.

Mục tiêu của bộ điều khiển PID là điều chỉnh giá trị điều khiển ở ngõ ra Output sao cho sai lệch Error $e(t) = (SP - PV)$ giữa giá trị đo được của hệ thống PV (Process Variable) với giá trị cài đặt SP (SetPoint) nhỏ nhất có thể (~ 0), đạt được sự ổn định và có đáp ứng nhanh.

2.4. Các mạch điện sử dụng.

2.4.1. Tìm hiểu về mạch kích Triac:

2.4.1.1. Mạch kích Triac là gì?

Mạch kích Triac là một mạch điện được sử dụng để điều khiển độ sáng hoặc công suất của các thiết bị điện, chẳng hạn như đèn, quạt hay máy giặt. Triac là một loại transistor hai hướng, cho phép dòng điện chạy qua nó theo cả hai chiều.

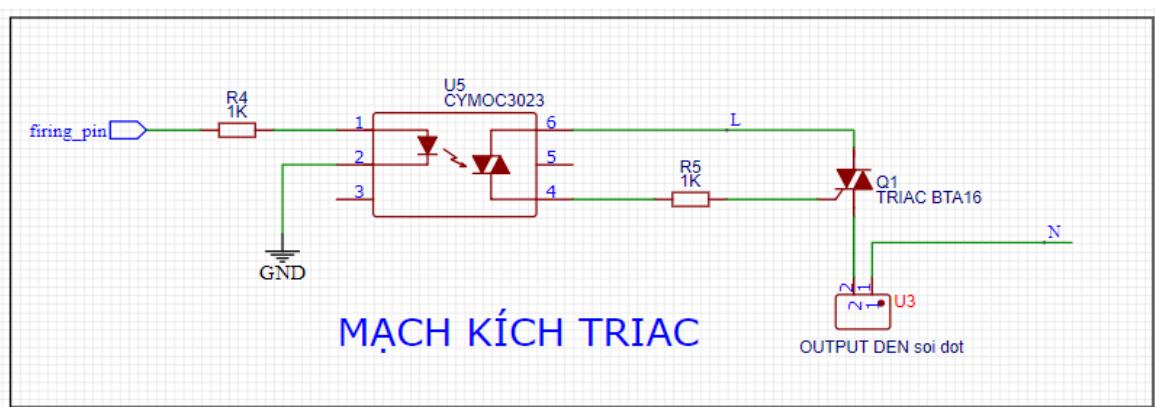
Triac được chế tạo từ hai transistor SCR (Silicon-Controlled Rectifier) kết hợp lại và chia sẻ một công điều khiển. Điều này cho phép Triac có khả năng điều khiển dòng điện chạy qua nó, bằng cách điều chỉnh thời gian mà nó dẫn điện trong mỗi nửa chu kỳ. Điều này làm cho Triac trở thành một công cụ hữu ích trong việc điều khiển tải điện biến thiên như đèn sáng.

Mạch kích Triac thường bao gồm các thành phần chính sau:

- *Triac*: Là linh kiện chính trong mạch kích. Nó có ba chân: cổng điều khiển (Gate), chân MT1 (Main Terminal 1) và chân MT2 (Main Terminal 2).
- *Ngõ vào điều khiển*: Được sử dụng để cung cấp tín hiệu điều khiển cho mạch kích Triac. Tín hiệu này thường được cung cấp từ mạch điều khiển hoặc mạch điện tử khác.
- *Optocoupler (quang cách ly)*: Đôi khi được sử dụng trong mạch kích Triac để cách ly tín hiệu điều khiển và mạch công suất, bảo vệ các linh kiện nhạy cảm và nâng cao an toàn.

Khi tín hiệu điều khiển được cung cấp vào cổng điều khiển của Triac, nó sẽ bắt đầu dẫn điện và cho phép dòng điện chạy qua nó. Thời gian mà Triac dẫn điện trong mỗi nửa chu kỳ được điều chỉnh bởi tín hiệu điều khiển. Bằng cách điều chỉnh thời gian này, ta có thể điều chỉnh độ sáng hoặc công suất của thiết bị điện được kết nối với mạch kích Triac.

2.4.1.2. Sơ đồ mạch kích Triac:



Hình 33: Sơ đồ mạch kích Triac

- CYMOC3023(U5): có tác dụng cung cấp cách ly điện và chuyển đổi tín hiệu điều khiển từ dạng ánh sáng thành tín hiệu điện để kích hoạt triac chính.



Hình 34: MOC3023 DIP-6 Triac Driver Optocoupler

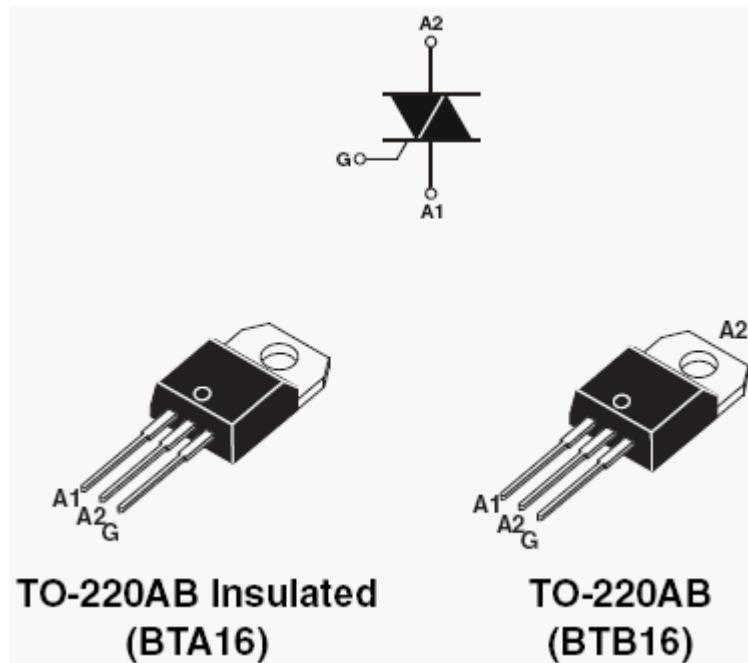
- TRIAC BTA16(Q1): điều khiển dòng điện và công suất đầu ra của các thiết bị điện thông qua việc điều chỉnh xung điện điều khiển áp dụng lên chân Gate của nó.



Hình 35: Triac BTA16 - 600B

Thông số kỹ thuật:

- Điện áp cực đại: 600V
- Dòng điện thuận cực đại: 16A
- Điện áp điều khiển mở van: 1.5V
- Dòng điều khiển mở van: 100mA
- Nhiệt độ làm việc: $-40^{\circ}C \sim 125^{\circ}C$



Hình 36: Sơ đồ chân Triac BTA16 - 600B

2.4.1.3. Nguyên lý hoạt động của mạch kích Triac:

Hoạt động của mạch kích triac dựa trên việc điều khiển kích triac thông qua tín hiệu điện áp áp dụng lên cổng Gate. Khi kích triac mở, nó cho phép dòng điện chạy qua nó và điều khiển dòng điện và công suất đầu ra của thiết bị điện liên quan.

2.4.2. Tổng quan về mạch phát hiện điểm 0:

2.4.2.1. Mạch phát hiện điểm 0 là gì?

Mạch phát hiện điểm 0 (zero crossing circuit) là một loại mạch điện được sử dụng để phát hiện sự thay đổi từ dương sang âm hoặc từ âm sang dương của một tín hiệu điện. Điểm 0 đề cập đến thời điểm khi tín hiệu điện chuyển từ một chiều sang chiều khác hoặc từ mức điện thế cao xuống mức điện thế thấp (hoặc ngược lại) trong chu kỳ tín hiệu.

Mạch phát hiện điểm 0 thường được sử dụng trong các ứng dụng như điều khiển động cơ, âm thanh, xử lý tín hiệu và các hệ thống đo lường. Mạch này có thể sử dụng để đếm số lượng điểm 0 trong một chu kỳ tín hiệu hoặc để đo tần số của tín hiệu dựa trên khoảng thời gian giữa các điểm 0 liên tiếp.

Có nhiều cách để thiết kế mạch phát hiện điểm 0, bao gồm sử dụng linh kiện như transistor, op-amp, hoặc vi điều khiển. Các mạch này thường sử dụng các nguyên lý như so sánh mức điện thế và sử dụng bộ lọc để loại bỏ nhiễu và chỉ tập trung vào sự thay đổi từ dương sang âm hoặc từ âm sang dương.

2.4.2.2.Một số ứng dụng phổ biến của mạch phát hiện điểm 0:

- Đo tần số: Mạch phát hiện điểm 0 có thể được sử dụng để đo tần số của một tín hiệu. Bằng cách đếm số lượng điểm 0 trong một khoảng thời gian, ta có thể tính được tần số của tín hiệu đó.

- Điều khiển động cơ: Mạch phát hiện điểm 0 có thể được sử dụng trong hệ thống điều khiển động cơ để xác định vị trí hoặc tốc độ của động cơ. Bằng cách theo dõi sự thay đổi từ dương sang âm hoặc từ âm sang dương, ta có thể xác định được khi nào động cơ đạt đến một điểm quan trọng trong quá trình quay.

- Xử lý tín hiệu: Mạch phát hiện điểm 0 có thể được sử dụng để xử lý tín hiệu âm thanh hoặc tín hiệu khác. Ví dụ, trong xử lý âm thanh, mạch này có thể được sử dụng để phân tách tín hiệu thành các phần dương và âm, hoặc để xác định các điểm tín hiệu đặc biệt như đỉnh sóng (peaks) hoặc vị trí trung tâm của các nhịp điệu.

- Điều khiển hệ thống: Mạch phát hiện điểm 0 có thể được sử dụng để xác định sự thay đổi trong một tín hiệu và điều khiển các hệ thống dựa trên sự thay đổi đó. Ví dụ, trong hệ thống đèn giao thông, mạch này có thể được sử dụng để phát hiện sự chuyển đổi từ đèn đỏ sang đèn xanh để kích hoạt các hành động tiếp theo.

Trên thực tế, ứng dụng của mạch phát hiện điểm 0 là rất đa dạng và phụ thuộc vào ngữ cảnh sử dụng của nó.

2.4.2.3.Tại sao chúng ta cần sử dụng mạch phát hiện điểm 0?

- Xác định trạng thái hoặc mức điện áp: Mạch phát hiện điểm 0 có thể xác định xem mức điện áp hoặc dòng điện có vượt quá ngưỡng cố định hay không. Điều này rất hữu ích trong việc xác định trạng thái của một hệ thống hoặc các thiết bị, như xác định trạng thái hoạt động hay ngừng hoạt động của một máy móc.

- Bảo vệ quá tải: Mạch phát hiện điểm 0 có thể được sử dụng để bảo vệ các thành phần điện tử khỏi quá tải. Khi một mức điện áp hoặc dòng điện vượt quá ngưỡng quy định, mạch phát hiện điểm 0 có thể kích hoạt các cơ chế bảo vệ để ngắt hoặc giảm dòng điện, từ đó ngăn chặn các thiệt hại tiềm năng cho hệ thống.

- Điều khiển chuyển đổi: Mạch phát hiện điểm 0 cũng có thể được sử dụng để điều khiển các chuyển đổi hoặc thiết bị khác. Ví dụ, khi một nguồn điện áp đạt đến mức quy định, mạch phát hiện điểm 0 có thể kích hoạt chuyển đổi hoặc kích thích một hành động nhất định, chẳng hạn như bật đèn hoặc bật máy móc.

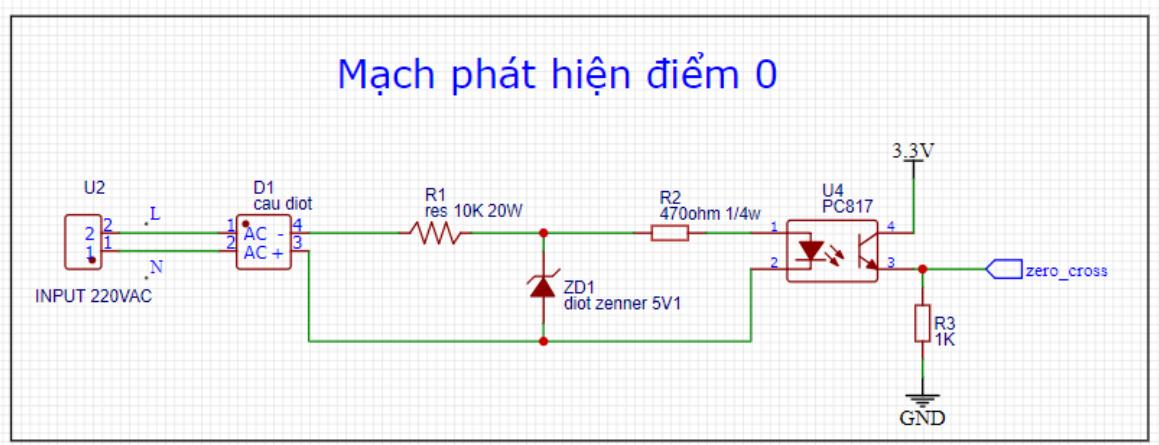
- Kiểm soát năng lượng: Mạch phát hiện điểm 0 có thể được sử dụng để kiểm soát năng lượng tiêu thụ của các thiết bị. Bằng cách xác định mức điện áp hoặc dòng

điện, mạch phát hiện điểm 0 có thể điều chỉnh hoạt động của các linh kiện hoặc hệ thống để tiết kiệm.

* Nếu không sử dụng mạch phát hiện điểm 0 và kích bừa bãi mà không có hệ thống kiểm soát, có thể xảy ra những hậu quả không mong muốn như sau:

1. Quá tải và hỏng thiết bị: Khi không có mạch phát hiện điểm 0, không có cơ chế để ngăn chặn mức điện áp hoặc dòng điện vượt quá giới hạn an toàn. Điều này có thể dẫn đến quá tải và hỏng hóc các thành phần hoặc thiết bị điện tử. Ví dụ, một máy móc có thể bị hỏng hoặc bị cháy khi quá tải điện.
2. Thiệt hại về tài sản: Nếu không có mạch phát hiện điểm 0 để ngắt hoặc giảm dòng điện khi cần thiết, năng lượng điện sẽ tiếp tục truyền qua các thiết bị và có thể gây ra thiệt hại cho tài sản. Điều này có thể bao gồm hỏng hóc linh kiện, đèn bị cháy, máy móc hỏng, hoặc thậm chí gây cháy nổ.
3. An toàn và nguy hiểm: Việc kích bừa bãi điện mà không có kiểm soát có thể tạo ra một môi trường nguy hiểm. Nếu dòng điện không được điều khiển và ngừng khi cần thiết, có thể xảy ra tai nạn, gây chấn thương hoặc gây nguy hiểm đến tính mạng của con người.
4. Tiêu thụ năng lượng không hiệu quả: Mạch phát hiện điểm 0 có thể giúp kiểm soát và điều chỉnh năng lượng tiêu thụ của hệ thống. Nếu không có mạch này, hệ thống có thể tiêu thụ năng lượng không cần thiết hoặc không hiệu quả, dẫn đến lãng phí năng lượng và tăng chi phí hoạt động.

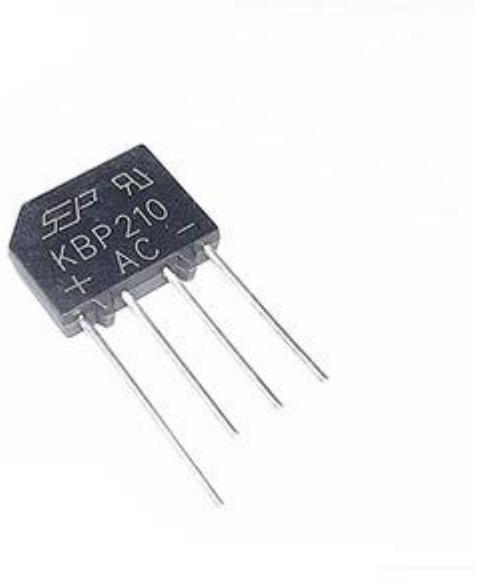
2.4.2.4. Sơ đồ mạch phát hiện điểm 0:



Hình 37: Sơ đồ mạch phát hiện điện áp điểm 0

Trong đó:

Cầu diot(D1): Dùng để chỉnh lưu dòng điện xoay chiều thành một chiều.



Hình 38: Cầu Diode 2A 700V DB207 DIP-4 (2A vuông)

Thông số kỹ thuật:

- Dòng điện định mức: 2A
- Điện áp tối đa: 700V
- Nhiệt độ hoạt động: $-55^{\circ}C$ đến $165^{\circ}C$

Điott Zener 5V(ZD1): Diode Zener thể hiện tính năng ổn áp khi được phân cực nghịch. Ban đầu, khi phân cực nghịch, dòng qua diode rất bé . Nhưng nếu tăng điện áp ngược lên 1 giá trị nào đó (V_z) thì dòng tăng mạnh trong khi điện áp giữa 2 đầu diode gần như không đổi.

Zener Diode



Hình 39: Diode Zener 1W 5.1V DIP 1N4733A

PC817(U4): là một loại optocoupler hoặc optoisolator, nó được sử dụng trong mạch phát hiện điểm 0 để cách ly và truyền tín hiệu giữa hai phần của mạch. Nhiệm vụ chính của PC817 trong mạch phát hiện điểm 0 là truyền tín hiệu từ phần nhận tín hiệu (đầu vào) sang phần xử lý (đầu ra) mà không có sự tiếp xúc vật lý trực tiếp giữa hai phần này.



Hình 40: Opto PC817 DIP-4 (817)

Các giá trị cần tính chọn trong mạch:

Giá trị hiệu dụng khi qua cầu Diot: $U_{hd} = 220 - (0,7 \times 2) = 218,6$ (V)

$$\text{Giá trị dòng điện: } I_R = \frac{U_{hd}-5V}{10k} = \frac{218,6V-5V}{10^4} \approx 0,02 \text{ (A)}$$

Điện áp rơi trên điện trở 10K: $U_R = I_R \times 10K = 0,02 \times 10^4 = 200$ (V)

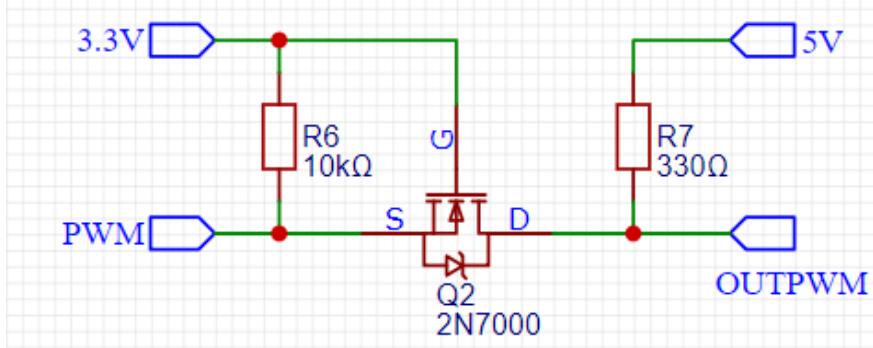
Công suất của trở: $P_R = I_R^2 \times 10k = 0,02^2 \times 10^4 = 4 \cdot 10^{-4} \times 10^4 = 4$ (W)

2.4.3. Mạch level shifter.

Mạch level shifter là một thành phần trong điện tử được sử dụng để chuyển đổi mức độ logic (voltage level) giữa hai hệ thống hoặc thiết bị khác nhau. Trong một số trường hợp, các thành phần hoặc hệ thống có thể sử dụng mức độ logic khác nhau, và để chúng có thể tương tác với nhau, một mạch level shifter được sử dụng để đảm bảo rằng tín hiệu được hiểu đúng.

Các mức độ logic thường được đo bằng điện áp, và mạch level shifter sẽ thực hiện chuyển đổi giữa các mức độ này.

Ở đồ án này, nhóm sử dụng mạch level shifter với mục đích dịch mức 0 - 3.3V PWM thành 0 - 5V PWM để cấp xung PWM 0 - 5V vào mạch AOD4184 Mosfet driver.



MẠCH LEVEL SHIFTER

Hình 41: Mạch Level Shifter

Mạch level shifter của nhóm em tự thiết kế sử dụng mosfet kênh 2N7000:

MOSFET kênh N tín hiệu nhỏ

Điện áp drain-source (VDS) là 60V

Dòng drain liên tục (ID) là 200mA

Dòng drain xung (đỉnh ID) là 500mA

Điện áp ngưỡng gate (VGS-th) là 3V

Điện áp source-gate là (VGS) là $\pm 20V$

Thời gian bật và tắt mỗi lần là 10ns. Tương đương tần số tối đa mà mosfet chịu được là 50MHz.

Nguyên lý hoạt động:

Khi chân PWM của ESP32 xuất ra điện áp 3.3V thì sẽ không có sự chênh lệch điện áp trên chân G và chân S ($V_{GS} = 0V$). Khi này điện áp 5V sẽ được cấp thẳng vào chân OUTPWM (mosfet công suất).

Khi chân PWM của ESP32 xuất ra điện áp 0V thì điện áp chênh lệch giữa chân G và chân S là 3.3V ($V_{GS} = 3.3V$). Khi này điện áp 5V sẽ qua điện trở 330 ôm và chạy xuống chân Source. Vì vậy điện áp đặt lên chân OUTPWM là 0V.

Kết luận: Vì mạch mosfet công suất (AOD4184) cần một điện áp $V_{GS} > 3.3V$ để có thể chịu được dòng cho tải (bơm) lớn hơn 5A. Chính vì vậy mạch level shifter sẽ giúp nhóm chúng em đảm bảo được áp cấp vào chân G của AOD4184 đủ lớn.

2.4.4. Mạch mosfet công suất 2xAOD4184

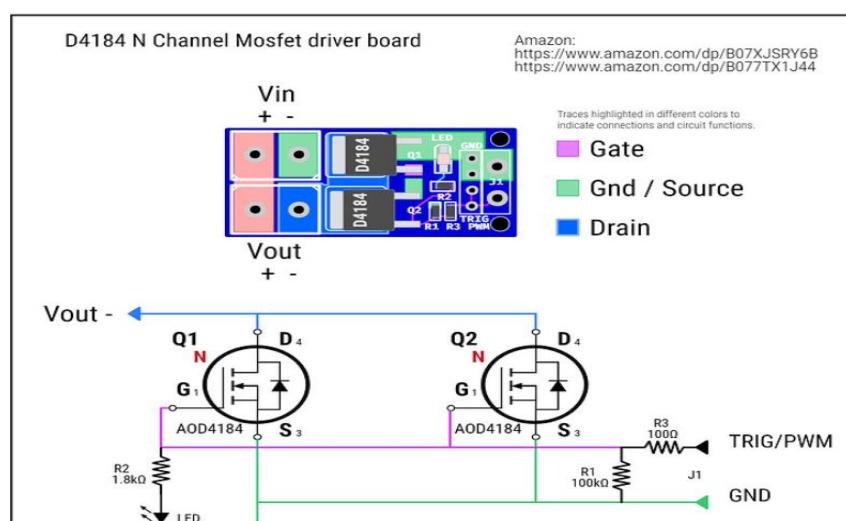
D4184 là một mạch công suất Mosfet có 2 kênh, thích hợp để điều khiển các bộ điều chế toàn xung như bộ nguồn LED PWM, cũng như để điều khiển tốc độ của động cơ DC. Hai Mosfet D4184 được kết nối song song cho phép tải liên tục lên đến 15A và cho phép tải đỉnh lên đến 30A.

Mạch công suất này xử lý tín hiệu PWM từ 3.3V đến 20V, có thể kết nối với cổng I/O vi điều khiển, do đó phù hợp với các ứng dụng vi điều khiển.

Thông số kỹ thuật:

- Dòng điện max: 30A
- Công suất: 400W
- Nguồn tín hiệu kích: 3.3-20V, có thể kết nối với cổng I/O vi điều khiển, I/O PLC...
- Tần số: 0-20KHz
- Nhiệt độ hoạt động: -40 ~ 85°C
- Kích thước: 34x17x12mm

Nguyên lý hoạt động: Tương tự như mạch level shifter, nhưng mạch AOD4184 cho phép hoạt động với tải có dòng cao như động cơ, thanh gia nhiệt, v.v... . Khi điện áp chân PWM > 3.3V => $V_{GS} > 3.3V$. Lúc này Mosfet sẽ đóng lại và thông chân D và chân S => Xuất điện áp PWM cho động cơ. Điện áp trung bình trên động cơ phụ thuộc vào duty cycle PWM được xuất ra trên chân PWM của ESP32.



Hình 42: Mosfet driver 2xAOD4184

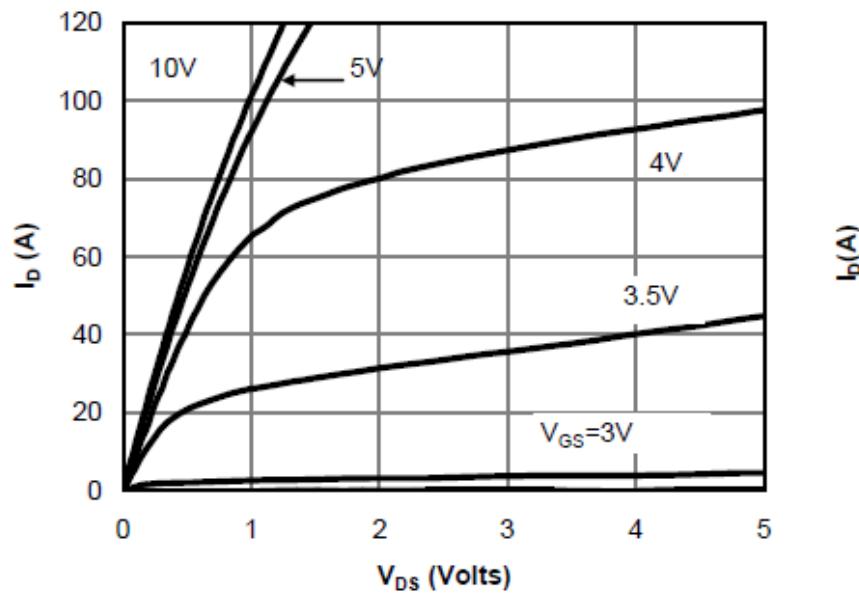


Fig 1: On-Region Characteristics (Note E)

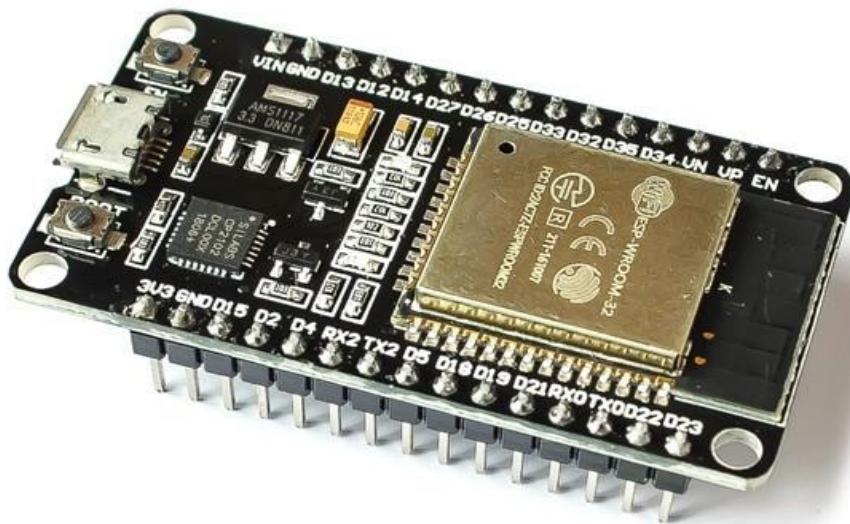
Hình 43: Mối liên hệ giữa V_{GS} và I_D

Trong datasheet của AOD4184 cho ta thấy, điện áp V_{GS} càng lớn thì dòng điện mà mosfet chịu được (I_D) càng lớn. Chính vì vậy, việc thiết kế mạch level shifter ở trên là cực kì quan trọng.

Kết luận: Với dòng max trên động cơ bơm 12VDC là 5A. Cho nên việc sử dụng mạch mosfet công suất AOD4184(100A) kết hợp mạch level shifter để kiểm soát mức bơm – phun sương cho động cơ là hoàn toàn hợp lý.

2.5. Vi điều khiển ESP32

2.5.1. Giới thiệu ESP32



Hình 44: Bo mạch phát triển dựa trên vi điều khiển ESP32

ESP32 là một vi điều khiển giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth (tạm dịch: Bluetooth ché độ kép). Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.

ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266.

2.5.2. Cấu hình của ESP32

CPU

- CPU: Xtensa Dual-Core LX6 microprocessor.
- Chạy hệ 32 bit
- Tốc độ xử lý 160MHZ up to 240 MHz
- Tốc độ xung nhịp đọc flash chip 40mhz --> 80mhz (tùy chỉnh khi lập trình)
- RAM: 520 KByte SRAM
- 520 KB SRAM liền chip –(trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

Hỗ trợ 2 giao tiếp không dây

- Wi-Fi: 802.11 b/g/n/e/i
- Bluetooth: v4.2 BR/EDR and BLE

Hỗ trợ tất cả các loại giao tiếp

- 8-bit DACs(digital to analog) 2 cổng
- Analog(ADC) 12-bit 16 cổng.
- I²C – 2 cổng
- UART – 3 cổng
- SPI – 3 cổng (1 cổng cho chip FLASH)
- I²S – 2 cổng
- SD card /SDIO/MMC host
- Slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN bus 2.0
- IR (TX/RX)
- Băm xung PWM (tất cả các chân)
- Ultra low power analog pre-amplifier'

Cảm biến tích hợp trên chip esp32

- 1 cảm biến Hall (cảm biến từ trường)
- 1 cảm biến đo nhiệt độ
- Cảm biến chạm (điện dung) với 10 đầu vào khác nhau.

Bảo mật

- IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

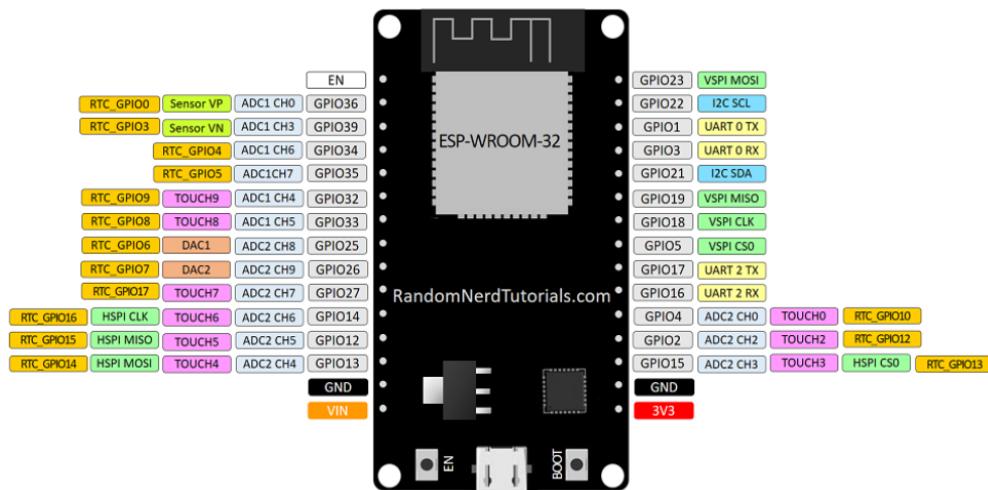
Nguồn điện hoạt động

- Nhiệt độ hoạt động -40 + 85C
- Điện áp hoạt động: 2.2-3.6V
- Số cổng GPIOs : 34

2.5.3. Sơ đồ chân của DOIT ESP32 DEVKIT V1 Board

Phiên bản có 30 GPIO

ESP32 DEVKIT V1 – DOIT
version with 30 GPIOs



Hình 45: Sơ đồ chân của DOIT ESP32 DEVKIT V1 Board

2.5.4. Điểm mạnh của ESP32 so với các dòng khác

Có thể nói ESP32 là sự nâng cấp hoàn hảo của ESP8266, với ESP8266 phù hợp với các dự án nhỏ và tiết kiệm chi phí. ESP32 lại phù hợp với các dự án phức tạp hơn, tốc độ xử lý cao hơn và tích hợp nhiều ngoại vi mạnh mẽ hơn.

ESP8266 là 17 chân GPIO, ADC độ phân giải 10 bit, 8 kênh PWM mềm trong khi đó ESP 32 có tới 30/36 chân GPIO, 18 kênh ADC độ phân giải 12-bit, 16 kênh PWM mềm, Touch Sensor, Hall Effect Sensor, Ethernet MAC Interface, Cảm biến nhiệt độ được tích hợp sẵn.

Về bộ nhớ ESP32 có thêm 4MB External Flash và 520KB SRAM (static random access memory) trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

ESP32 hỗ trợ Bluetooth 4.2 và BLE (Bluetooth Low Energy). Việc hỗ trợ cả bluetooth khiến ESP32 có thể tương tác với các thiết bị như là bàn phím, chuột, điện thoại khi mà không có wifi.

Ultra Low Power giải quyết vấn đề năng lượng cho ESP bởi vì sử dụng Wi-Fi sẽ rất ngốn điện đặc biệt khi chúng ta sử dụng pin phải tính toán rất kĩ.

2.6. Cảm biến nhiệt độ và cảm biến độ ẩm đất.

2.6.1. Cảm biến DHT11

2.6.1.1.Tổng quan DHT11

DHT11 là cảm biến nhiệt độ và độ ẩm kỹ thuật số cơ bản, chi phí cực thấp. Nó sử dụng cảm biến độ ẩm điện dung và nhiệt điện trở để đo không khí xung quanh, lấy dữ liệu thông qua giao tiếp 1-wire (sử dụng 1 dây data để truyền dữ liệu)

Cảm biến nhiệt độ và độ ẩm DHT11 bao gồm 3 thành phần chính. Một cảm biến độ ẩm loại điện trở, một điện trở nhiệt NTC (Negative Temperature Coefficient – hệ số nhiệt âm) để đo nhiệt độ và một vi điều khiển 8 bit có nhiệm vụ chuyển đổi tín hiệu tương tự từ cảm biến và gửi ra tín hiệu số duy nhất.

2.6.1.2.Thông số DHT11

Độ ẩm tương đối

- Độ phân giải: 16Bit
- Độ lặp lại: $\pm 1\%$ RH
- Độ chính xác: $25^{\circ}\text{C} \pm 5\%$ RH

Nhiệt độ

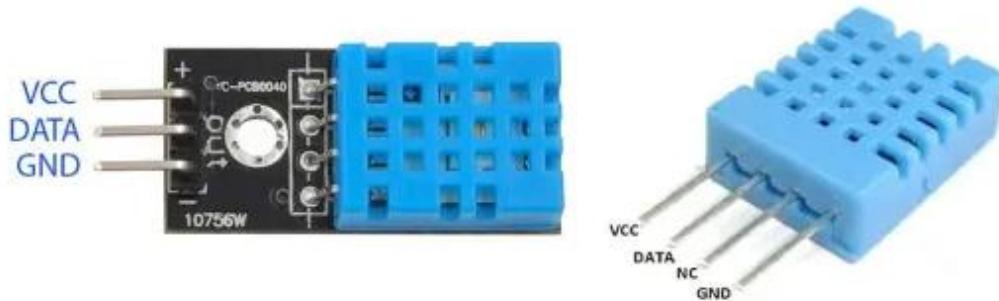
- Độ phân giải: 16Bit
- Độ lặp lại: $\pm 1^{\circ}\text{C}$
- Độ chính xác: $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$

Thông số hoạt động

- Điện áp hoạt động: 3-5VDC.
- Dải nhiệt độ đo: $0-50^{\circ}\text{C}$. Sai số $\pm 2^{\circ}\text{C}$.
- Dải độ ẩm đo: $20-80\%$. Sai số $\pm 5\%$.
- Tần số lấy mẫu: 1Hz (mỗi giây một lần).
- Để dữ liệu chính xác thì khoảng cách dây từ vi điều khiển đến cảm biến không quá 20m.

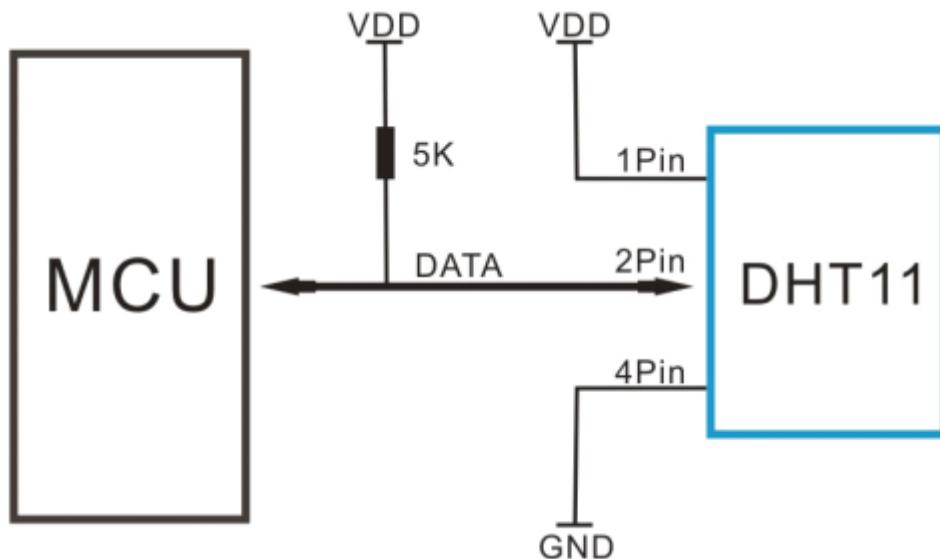
2.6.1.3.Sơ đồ chân

Cảm biến DHT11 gồm 2 chân cấp nguồn (VCC và GND), và 1 chân tín hiệu (DATA). Hiện nay, thông dụng ngoài thị trường có hai loại đóng gói cho DHT11: 3 chân và 4



Hình 46: Cảm biến DHT11

2.6.1.4. Mạch ứng dụng điển hình cho cảm biến độ ẩm và nhiệt độ DHT11



Hình 47: Mạch ứng dụng điển hình cho cảm biến độ ẩm và nhiệt độ DHT11

DHT11 gửi và nhận dữ liệu với một dây tín hiệu **DATA**, với chuẩn dữ liệu truyền 1 dây này, chúng ta phải đảm bảo sao cho ở chế độ chờ (idle) dây DATA có giá trị ở mức cao, nên trong mạch sử dụng DHT11, dây DATA phải được mắc với một điện trở kéo bên ngoài (thông thường giá trị là $5k\Omega$).

Dữ liệu truyền về từ cảm biến DHT11 bao gồm 40 bit và định dạng như sau:

8 bit biểu thị phần nguyên của độ ẩm + 8 bit biểu thị phần thập phân của độ ẩm + 8 bit biểu thị phần nguyên của nhiệt độ + 8 bit biểu thị phần thập phân của nhiệt độ + 8 bit kiểm tra

Ví dụ:

Xem xét dữ liệu nhận được từ cảm biến DHT11 là

00100101 00000000 00011001 00000000 00111110

High Humidity Low Humidity High Temperature Low Temperature Checksum (Parity)

Để kiểm tra xem liệu dữ liệu nhận được có chính xác hay không, chúng ta cần thực hiện một phép tính nhỏ. Cộng tất cả các giá trị nguyên và thập phân của độ ẩm và nhiệt độ và kiểm tra xem tổng có bằng giá trị checksum hay không, tức là dữ liệu 8 bit cuối cùng.

$$00100101 + 00000000 + 00011001 + 00000000 = 00111110$$

Giá trị này giống như 8 bit checksum và do đó dữ liệu nhận được là hợp lệ. Bây giờ để có được giá trị độ ẩm và nhiệt độ, chỉ cần chuyển đổi dữ liệu nhị phân thành dữ liệu thập phân.

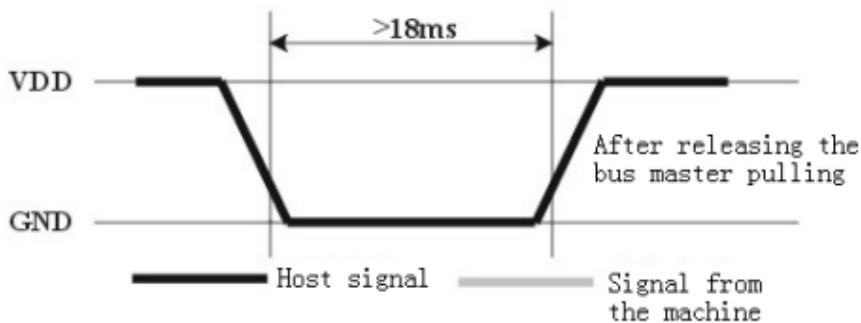
$$\text{Độ ẩm} = \text{Giá trị thập phân của } 00100101 = 37\%$$

$$\text{Nhiệt độ} = \text{Giá trị thập phân của } 00011001 = 25^{\circ}\text{C}$$

2.6.1.4. Các bước đọc dữ liệu từ DHT11

Bước một: Sau khi cấp nguồn cho DHT11 (sau khi cấp nguồn cho DHT11 1S để chờ qua trạng thái không ổn định trong khoảng thời gian này không thể gửi bất kỳ lệnh nào), kiểm tra nhiệt độ và độ ẩm môi trường, và ghi lại dữ liệu trong khi dây dẫn dữ liệu DATA của DHT11 được kéo lên bởi một điện trở kéo lên vẫn ở mức cao; DHT11 lúc này chân DATA là trạng thái đầu vào, luôn phát hiện tín hiệu bên ngoài.

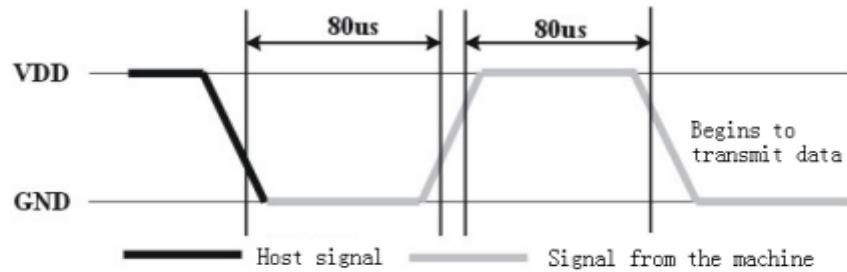
Bước hai: Vì điều khiển I / O xuất ra trong khi xuất ra được thiết lập ở mức thấp, và thời gian duy trì mức thấp không thể nhỏ hơn 18ms, sau đó vi điều khiển I / O được thiết lập để nhập trạng thái, do điện trở kéo lên, vi điều khiển I / O đó dây dẫn dữ liệu DHT11 cũng sẽ đi lên cao, chờ đợi để trả lời các tín hiệu DHT11 tín hiệu truyền như hình:



The host sends a start signal

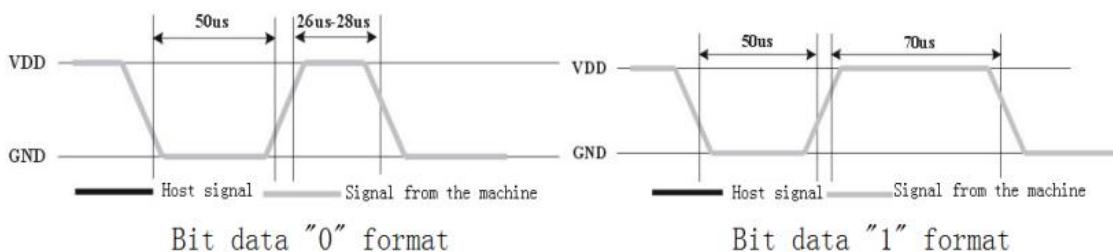
Hình 48: Đọc dữ liệu từ DHT11

Bước ba: Khi phát hiện tín hiệu bên ngoài ở mức thấp, chân DATA của DHT11 sẽ chờ đợi tín hiệu bên ngoài kết thúc mức thấp, sau một khoảng trễ DHT11 sẽ đưa chân DATA sang trạng thái xuất ra, xuất ra mức thấp trong 80 micro giây làm tín hiệu phản hồi, tiếp theo xuất ra mức cao trong 80 micro giây để thông báo cho thiết bị ngoại vi sẵn sàng nhận dữ liệu, vi điều khiển I / O lúc này ở trạng thái nhập, phát hiện I / O có mức thấp (tín hiệu phản hồi của DHT11) để chờ đợi 80 micro giây mức cao nhận và gửi tín hiệu dữ liệu như hình:



Hình 49: :Đọc dữ liệu từ DHT11

Bước bốn : 40 bit dữ liệu được xuất ra từ chân DATA của DHT11, vi điều khiển theo sự thay đổi của mức I/O nhận 40 bit dữ liệu, một định dạng dữ liệu của “0”: mức cao và mức thấp của 50 micro giây và 26-28 micro giây, định dạng dữ liệu “1”: mức thấp 50 micro giây cộng với 70 micro giây cao. Bit dữ liệu “0”, “1” định dạng tín hiệu được minh họa như hình:



Hình 50: Đọc dữ liệu từ DHT11

Kết thúc: Sau khi gửi xong 40 bit dữ liệu, cảm biến DHT11 sẽ tiếp tục xuất ra mức thấp trong 50 micro giây, sau đó chuyển sang trạng thái nhập, do điện trở kéo lên kèm theo sẽ kéo dây dẫn dữ liệu lên mức cao. Nhưng cảm biến DHT11 sẽ tiếp tục đo và kiểm tra lại nhiệt độ và độ ẩm bên trong, và lưu giữ dữ liệu, chờ đợi tín hiệu bên ngoài đến.

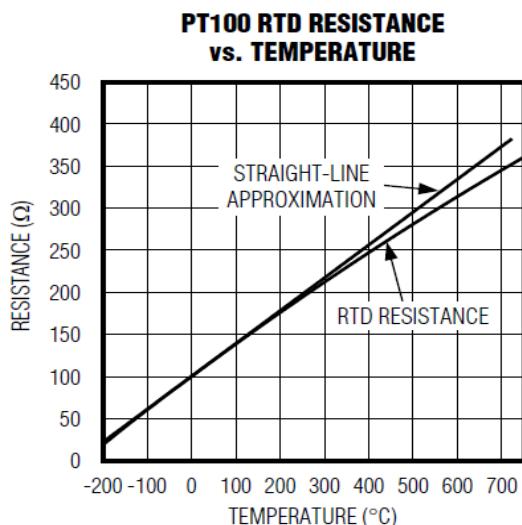
2.6.2. Cảm biến nhiệt độ PT100-MAX31865

2.6.2.1. Max31865 và PT100 là gì?

MAX31865 là một bộ chuyển đổi giá trị điện trở sang kỹ thuật số dễ sử dụng, được tối ưu hóa cho cảm biến nhiệt độ dựa trên điện trở platinum (RTD).

Cảm biến nhiệt độ PT100 là một loại cảm biến nhiệt độ dựa trên nguyên lý thay đổi điện trở của một RTD (Resistance Temperature Detector) được làm từ vật liệu platinum. PT100 có ý nghĩa là cảm biến có điện trở 100 ohm ở nhiệt độ 0°C.

PT100 sử dụng điện trở của RTD để đo nhiệt độ. Khi nhiệt độ tăng, điện trở của RTD tăng theo một quy luật cụ thể. Cảm biến PT100 được sử dụng rộng rãi trong các ứng dụng đo nhiệt độ chính xác, như trong các hệ thống kiểm soát nhiệt độ công nghiệp, phòng thí nghiệm, thiết bị y tế, và các ứng dụng khoa học và nghiên cứu khác. Hai giá trị được sử dụng rộng rãi nhất cho alpha (α) là 0.00385 và 0.00392, tương ứng với tiêu chuẩn IEC 751 (PT100) và tiêu chuẩn SAMA.



The resistance vs. temperature curve is reasonably linear, but has some curvature, as described by the Callendar-Van Dusen equation:

$$R(T) = R_0(1 + aT + bT^2 + c(T - 100)T^3)$$

where:

T = temperature (°C)

R(T) = resistance at T

R₀ = resistance at T = 0°C

IEC 751 specifies $\alpha = 0.00385055$ and the following Callendar-Van Dusen coefficient values:

$$a = 3.90830 \times 10^{-3}$$

$$b = -5.77500 \times 10^{-7}$$

$$c = -4.18301 \times 10^{-12} \text{ for } -200^\circ\text{C} \leq T \leq 0^\circ\text{C}, 0 \text{ for } 0^\circ\text{C} \leq T \leq +850^\circ\text{C}$$

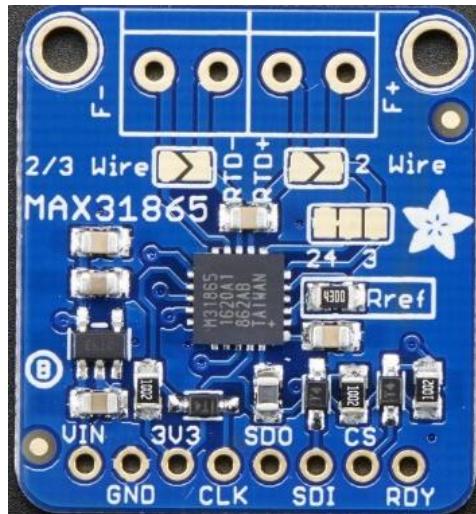
Hình 51: Mối tương quan giữa nhiệt độ và điện trở

Đường cong điện trở so với nhiệt độ có tính tuyến tính hợp lý, nhưng có một số độ cong, được mô tả bởi phương trình Callendar-Van Dusen ở hình trên.

Ở đề tài lần này, nhóm nghiên cứu sử dụng kết nối cảm biến PT100 loại 3 dây với MAX31865.

Dải đo trung bình của cảm biến PT100 dây dao động khoảng -40 đến 200 độ C, maximum có thể lên đến 400 độ C.

2.6.2.2. Quá trình kết nối PT100 với MAX31865



Hình 52: MAX31865 Adafruit

Vin - đây là chân cấp điện. Max31865 sử dụng điện áp 3V và các nhà thiết kế đã thêm 1 bộ ổn áp. Chính vì vậy với STM32F103C8T6 ta có thể cấp 3.3V hay 5V đều được.

3V3 - đây là chân đầu ra 3.3V từ bộ điều tiết áp, ta có thể lấy tối đa 100mA từ đây nếu cần.

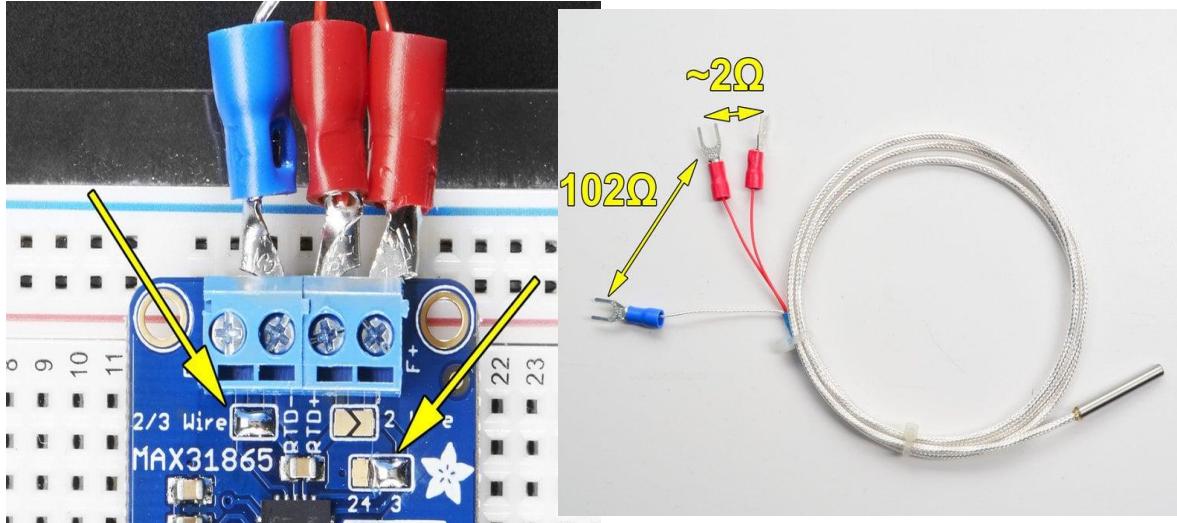
GND - chân mass chung cho nguồn và logic.

SCK - Đây là chân SPI Clock, là một chân INPUT của chip.

SDO - đây là chân Serial Data Out / Microcontroller In Sensor Out, dùng để gửi dữ liệu từ MAX31865 đến bộ xử lý STM32.

SDI - đây là chân Serial Data In / Microcontroller Out Sensor In, dùng để gửi dữ liệu từ bộ xử lý (STM32) của ta đến MAX31865.

CS - đây là chân Chip Select, khi có mức thấp, sẽ bắt đầu một truyền thông SPI. Đây là một chân đầu vào của chip.



Hình 53: PT100 và MAX31865

Sử dụng VOM để xác định các chân cần kết nối. Khi sử dụng loại cảm biến 3 dây, ta cần hàn kết nối tại các điểm mũi tên màu vàng – 2/3 Wire và “4 3”. Và không được quên phải loại bỏ kết nối giữa “2 4”.

Sau khi làm đủ các bước trên, ta đã sẵn sàng để kết nối SPI giữa MAX31865 và STM32 để thực hiện lập trình.

2.6.2.3. Chuyển đổi công thức Callenda – Van Dusen thành ngôn ngữ lập trình.

Để có thể biến đổi dữ liệu nhiệt độ nhận từ cảm biến PT100 và MAX31065, ta cần chuyển đổi công thức Callenda – Van Dusen thành ngôn ngữ lập trình để vi điều khiển hiểu được.

Sau khi chuyển đổi ra có giá trị Analog 15 bit được lưu vào biến “fullreg”

→ Cần là giá trị điện trở của PT100 để tính toán nhiệt độ

$$\text{Công thức (1)} : R_{RTD} = \frac{(ADC\ Code \times R_{Ref})}{2^{15}}$$

$$\Rightarrow RTD = ADC\ Code$$

$$RTD^* = R_{Ref}$$

$$RTD/ = 32768(2^{15})$$

→ Ta được: $RTD = R(T)$

The resistance vs. temperature curve is reasonably linear, but has some curvature, as described by the Callendar-Van Dusen equation:

$$R(T) = R_0(1 + aT + bT^2 + c(T - 100)T^3)$$

where:

T = temperature ($^{\circ}\text{C}$)

R(T) = resistance at T

R₀ = resistance at T = 0°C

IEC 751 specifies $\alpha = 0.00385055$ and the following Callendar-Van Dusen coefficient values:

$$a = 3.90830 \times 10^{-3}$$

$$b = -5.77500 \times 10^{-7}$$

$$c = -4.18301 \times 10^{-12} \text{ for } -200^{\circ}\text{C} \leq T \leq 0^{\circ}\text{C}, 0 \text{ for } 0^{\circ}\text{C} \leq T \leq +850^{\circ}\text{C}$$

Minh họa công thức Callenda – Van Dusen

Ta cần biến đổi giá trị điện trở R(T) sang nhiệt độ T($^{\circ}\text{C}$)

Ta có:

$$R(T) = R_0(1 + aT + bT^2)$$

Đặt biến r = R(T); RTDa = a; RTDb = b.

$$\Leftrightarrow r = 100(1 + RTDaT + RTDbT^2)$$

$$\Leftrightarrow \frac{r}{100} - 1 = RTDaT + RTDbT^2 \text{ (Phương trình bậc 2)}$$

$$\Leftrightarrow T = \frac{-RTDa + \sqrt{RTDa^2 - 4RTDb \times (1 - \frac{r}{100})}}{2RTDb}$$

$$\Leftrightarrow Z_1 = -RTDa$$

$$\Leftrightarrow Z_2 = RTDa^2 - 4RTDb$$

$$\Leftrightarrow Z_3 = \frac{4RTDb}{100}$$

$$\Leftrightarrow Z_4 = 2RTDb$$

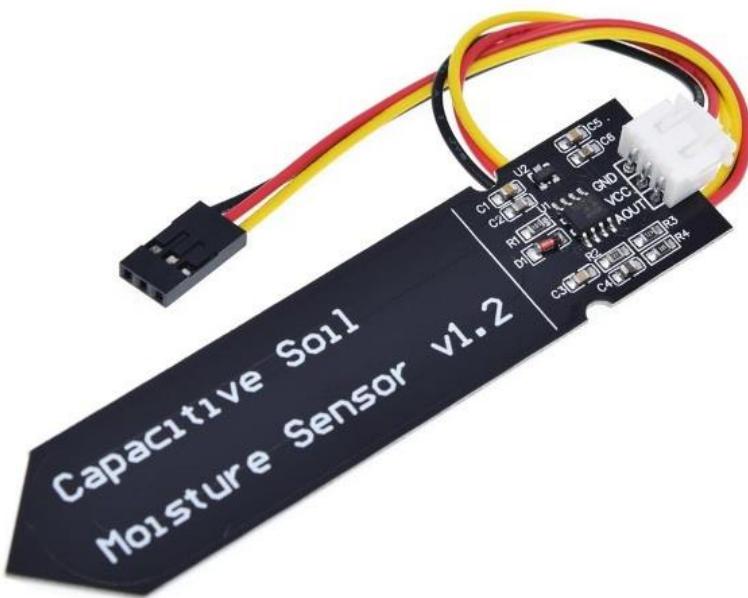
$$\rightarrow T = \frac{Z_1 + \sqrt{Z_2 + Z_3 \cdot r}}{Z_4}$$

Đây sẽ là tiền đề để nhóm lập trình cho cảm biến nhiệt độ PT100 và MAX31065.

2.6.3. Cảm biến độ ẩm đất Capacitive Soil V1.2

2.6.3.1. Tổng quan về cảm biến

- Cảm biến điện dung đo độ ẩm là một thiết bị điện tử được thiết kế để đo hoặc cảm nhận mức độ ẩm trong môi trường xung quanh nó bằng cách sử dụng nguyên lý hoạt động dựa trên sự thay đổi trong dung tích hoặc điện dung của một cấu trúc cảm biến khi môi trường xung quanh thay đổi độ ẩm.



Hình 54: Cảm biến độ ẩm đất Moisture Sensor V1.2

- Thông số kỹ thuật:

- + Điện áp hoạt động: 4.5~5.5VDC
- + Điện cực phủ sơn chống ăn mòn cho độ bền và độ ổn định cao
- + Điện áp xuất ra chân Analog: 0~VCC
- + Chuẩn giắc cắm: PH2.54-3P
- + Kích thước PCB: 98 x 23mm

2.6.3.2. Lịch sử ra đời

Cảm biến điện dung đo độ ẩm đất đã được phát triển vào thế kỷ 20 và có một lịch sử tương đối ngắn so với một số loại cảm biến khác. Dưới đây là một cái nhìn tổng quan về lịch sử ra đời của cảm biến điện dung đo độ ẩm đất:

Thập kỷ 1950:

Cảm biến điện dung đo độ ẩm đất đầu tiên xuất hiện vào cuối thập kỷ 1950. Chúng được phát triển để giúp nông dân và người làm vườn kiểm soát độ ẩm đất trong quá trình trồng cây và chăm sóc cây trồng.

Các cảm biến điện dung ban đầu có thiết kế đơn giản và chức năng cơ bản, và chúng thường được kết nối với các thiết bị hiển thị độ ẩm đất.

Thập kỷ 1980:

Trong thập kỷ 1980, cảm biến điện dung đo độ ẩm đất đã trải qua sự phát triển đáng kể trong khả năng đo, độ chính xác và tích hợp công nghệ.

Cảm biến điện dung đã trở nên phổ biến hơn trong nông nghiệp và quản lý tài nguyên môi trường, giúp nâng cao hiệu suất và giảm thiểu sự lãng phí trong việc tưới tiêu và chăm sóc cây trồng.

Hiện tại:

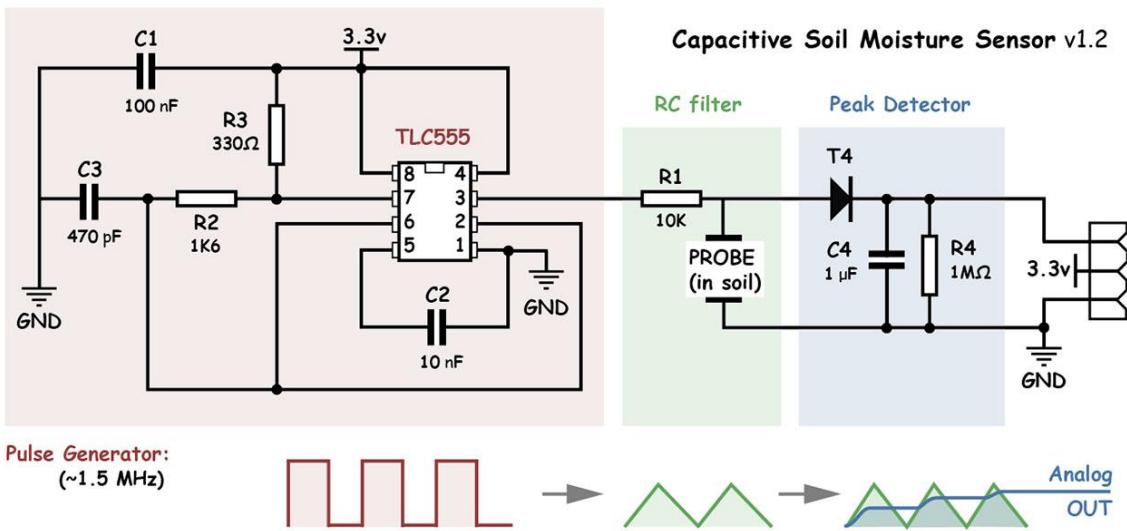
Cảm biến điện dung đo độ ẩm đất đã trở thành một công cụ quan trọng trong nhiều ứng dụng liên quan đến nông nghiệp thông minh, quản lý tài nguyên nước, quản lý môi trường, và các ứng dụng công nghiệp khác.

Các công nghệ tiên tiến, chẳng hạn như cảm biến không dây và tích hợp các hệ thống theo dõi tự động, đã làm cho việc sử dụng cảm biến điện dung trở nên dễ dàng và hiệu quả hơn.

Tổng cộng, cảm biến điện dung đo độ ẩm đất đã trải qua một sự phát triển liên tục và trở thành một công cụ quan trọng trong việc quản lý độ ẩm đất trong nhiều ngành khác nhau.

2.6.3.3. Cấu tạo cảm biến độ ẩm đất Moisture Sensor V1.2

Cảm biến điện dung đo độ ẩm đất thường có cấu trúc đơn giản với hai điện cực chính. Dưới đây là cấu trúc cơ bản của cảm biến điện dung đo độ ẩm đất:



Hình 55: Mạch PCB cảm biến điện dung Moisture Sensor v1.2

Đầu cảm biến:

Đầu cảm biến là phần tiếp xúc trực tiếp với đất hoặc môi trường cần đo độ ẩm. Nó thường được làm bằng vật liệu không dẫn điện, chẳng hạn như nhựa, sợi thủy tinh hoặc gốm, để tránh ảnh hưởng của dẫn điện và ẩm đất lên điện cực.

Hai điện cực:

Cảm biến điện dung sử dụng hai điện cực: một điện cực dẫn điện và một điện cực thứ cấp. Điện cực thứ cấp thường được chôn sâu vào đất để tiếp xúc trực tiếp với môi trường đất. Điện cực này có thể là một lá chất điện cực hoặc một thanh dẫn điện đặc biệt.

Không gian điện dung:

Giữa hai điện cực, có một khoảng không gian điện dung, thường là không khí hoặc môi trường không dẫn điện. Độ rộng của không gian điện dung phụ thuộc vào thiết kế cụ thể của cảm biến.

Dây cáp:

Dây cáp dẫn điện từ điện cực dẫn điện đến mạch điện tử hoặc thiết bị đo độ ẩm.

Mạch điện tử:

Cảm biến điện dung thường được kết nối với một mạch điện tử hoặc bo mạch đo để xử lý tín hiệu và chuyển đổi nó thành giá trị độ ẩm có thể đọc hoặc hiển thị.

2.6.3.4. Nguyên lý hoạt động

Nguyên lý hoạt động của cảm biến điện dung đo độ ẩm đất dựa trên sự thay đổi về điện dung của môi trường giữa hai điện cực khi độ ẩm của đất thay đổi. Dưới

đây là một cái nhìn chi tiết về nguyên lý hoạt động của cảm biến điện dung độ ẩm đất:

Kết cấu cảm biến:

Cảm biến bao gồm hai điện cực chính: một điện cực dẫn điện và một điện cực thứ cấp. Điện cực thứ cấp thường được đặt sâu vào đất hoặc môi trường cần đo độ ẩm, và điện cực này có thể được làm bằng chất liệu dẫn điện đặc biệt.

Điện dung ban đầu:

Ban đầu, không gian giữa hai điện cực là không khí hoặc môi trường không dẫn điện. Trong trường hợp này, điện dung của không gian giữa hai điện cực là điện dung ban đầu.

Tác động của độ ẩm:

Khi độ ẩm của môi trường (đất) xung quanh cảm biến thay đổi, đất sẽ thấm nước và làm cho không gian giữa hai điện cực thay đổi. Điều này dẫn đến sự thay đổi về điện dung của không gian đó.

Điện dung biến đổi:

Khi đất trở nên ẩm hơn, điện dung giữa hai điện cực tăng lên. Điện dung này tăng lên theo tỷ lệ với độ ẩm của đất. Khi đất khô đi, điện dung giảm xuống.

Đọc và xử lý dữ liệu:

Cảm biến được kết nối với một mạch điện tử hoặc thiết bị đo, và sự thay đổi về điện dung được ghi nhận và chuyển đổi thành giá trị độ ẩm. Giá trị độ ẩm có thể được hiển thị trực tiếp hoặc gửi đến hệ thống điều khiển để theo dõi và kiểm soát độ ẩm của đất hoặc môi trường cụ thể.

2.6.3.5. Ưu điểm và nhược điểm

***Ưu điểm:**

Giá trị tương đối thấp: Moisture Sensor V1.2 thường có giá thành thấp, làm cho nó trở thành một giải pháp kinh tế cho việc đo độ ẩm đất trong ứng dụng nông nghiệp và làm vườn.

Dễ sử dụng: Cảm biến này dễ dàng kết nối với các bo mạch điện tử hoặc thiết bị đo thông qua dây cáp tiêu chuẩn, và nó không đòi hỏi kiến thức kỹ thuật cao.

Khả năng theo dõi: Moisture Sensor V1.2 cung cấp dữ liệu độ ẩm đất liên tục, cho phép người dùng theo dõi và điều khiển tưới tiêu hoặc chăm sóc cây trồng dựa trên mức độ ẩm thực tế của đất.

Dễ bảo trì: Cảm biến này thường không yêu cầu bảo trì đặc biệt và có tuổi thọ tương đối dài.

*Nhược điểm:

Độ chính xác có thể bị ảnh hưởng: Moisture Sensor V1.2 không cung cấp độ chính xác cao như một số cảm biến đất tiền hơn. Điều này có thể dẫn đến sai lệch trong đo độ ẩm đất, đặc biệt trong các ứng dụng đòi hỏi độ chính xác cao.

Cần hiệu chuẩn định kỳ: Cảm biến này có thể cần phải được hiệu chuẩn định kỳ để đảm bảo tính đúng đắn của dữ liệu đo.

Dễ bị ảnh hưởng bởi yếu tố môi trường khác: Moisture Sensor V1.2 có thể bị ảnh hưởng bởi các yếu tố môi trường khác nhau, chẳng hạn như nhiệt độ, muối, hoặc chất lượng đất, dẫn đến độ chính xác bị giảm.

Khả năng dây cáp bị hỏng: Dây cáp của cảm biến này có thể bị hỏng hoặc bị ảnh hưởng bởi môi trường ngoài, đặc biệt là trong môi trường nông nghiệp.

2.6.3.6.Thực nghiệm

Công thức tính độ ẩm đất theo lý thuyết vật lý:

$$W = \frac{m_w}{m_d} \cdot 100\% \quad (*)$$

W: độ ẩm đất (%)

Mw: khối lượng nước có trong mẫu đất

Md: khối lượng đất khô trong mẫu đất (g)

Sau khoảng thời gian tìm hiểu, nhóm đã đưa ra công thức chuyển đổi tín hiệu đầu ra analog thành độ ẩm đất (%) dựa trên hàm Map như sau:

$$W = \frac{(g - z) \cdot (y - x)}{(t - z) + x} \quad (**)$$

W: độ ẩm đất (%)

x: giá trị ngưỡng dưới cần chuyển đổi

y: giá trị ngưỡng trên cần chuyển đổi

z: giá trị analog tương ứng với x

t: giá trị analog tương ứng với y

g: giá trị analog đo được ở hiện tại

Để kiểm chứng sai số của cảm biến, nhóm đã tiến hành đo độ ẩm đất ngoài thực tế theo các bước như sau:

* **Bước 1:** Chuẩn bị đất để đo (nhóm sử dụng đất trồng cây), phơi khô đất ngoài nắng để đảm bảo lượng nước trong đất không còn đáng kể. Lúc này độ ẩm đất gần như 0%. Ngoài ra, nhóm còn chuẩn bị các dụng cụ khác để đo như cân điện tử, ly nhựa để chứa đất, chai nước 10ml



Hình 56: Khối lượng của ly nhựa

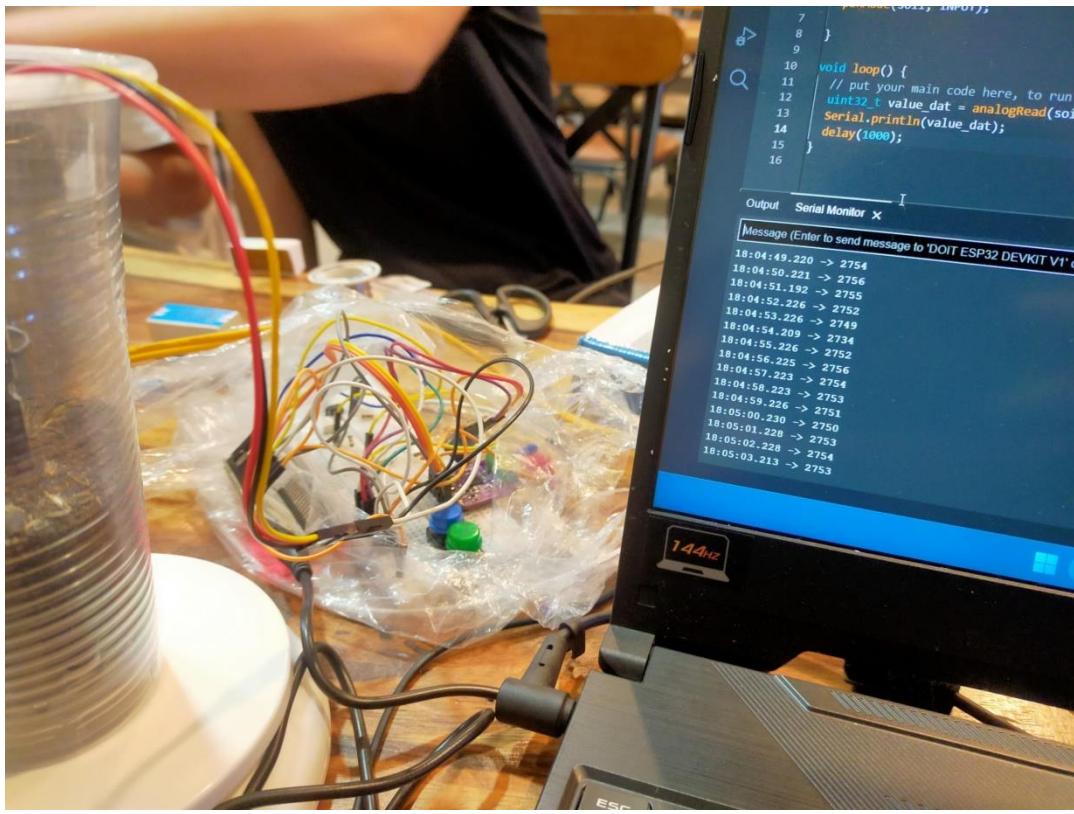


Hình 54: Khối lượng của ly nhựa

* **Bước 2:** Sử dụng 50g đất khô, 50ml nước. Ở đây, nhóm chia ra thành 5 lần đo:

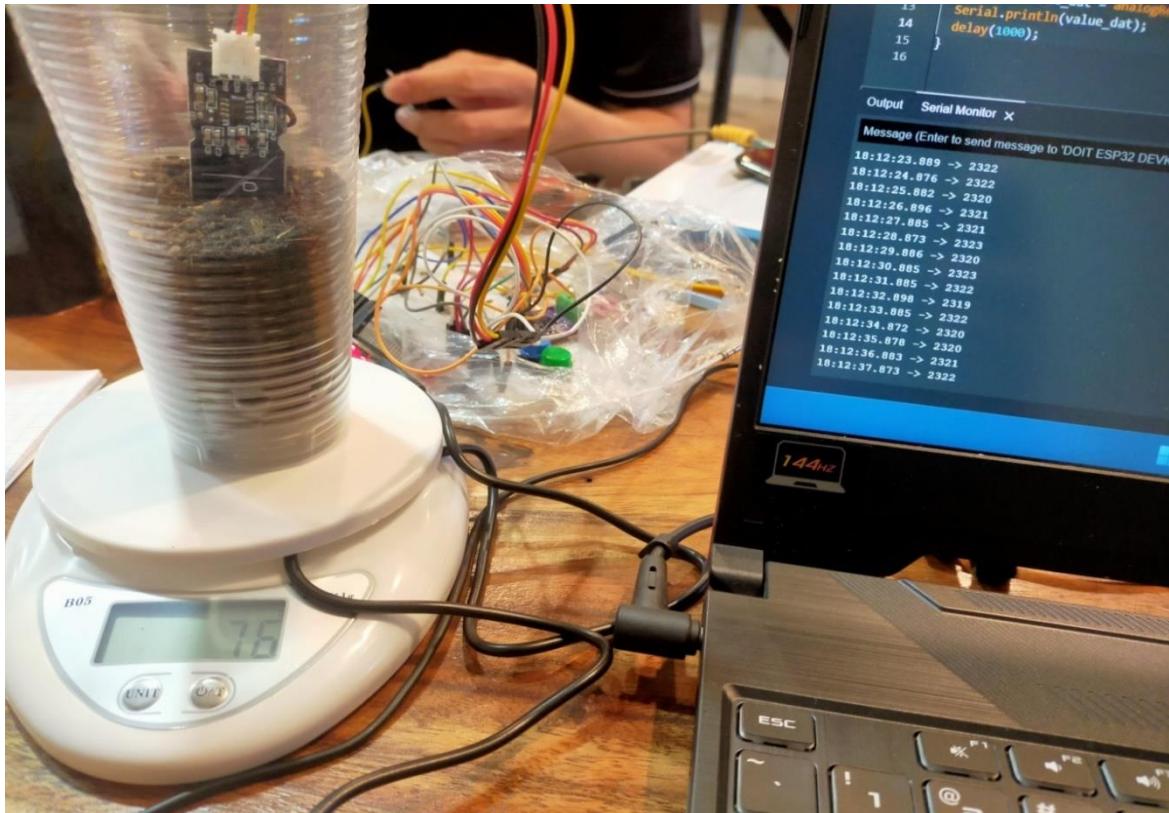
- Lần 1: Độ ẩm đất 0 - 20%

Tiến hành cho 50g đất vào ly nhựa, cắm cảm biến vào đất để đo lúc độ ẩm đất 0%. Lúc này, giá trị analog rơi vào khoảng 2750.



Hình 57: Giá trị analog ứng với độ ẩm đất 0% khoảng 2750

Sau đó, ta đổ 10ml nước vào 50g đất khô. Lúc này độ ẩm đất là 20%, giá trị analog trả về khoảng 2325.



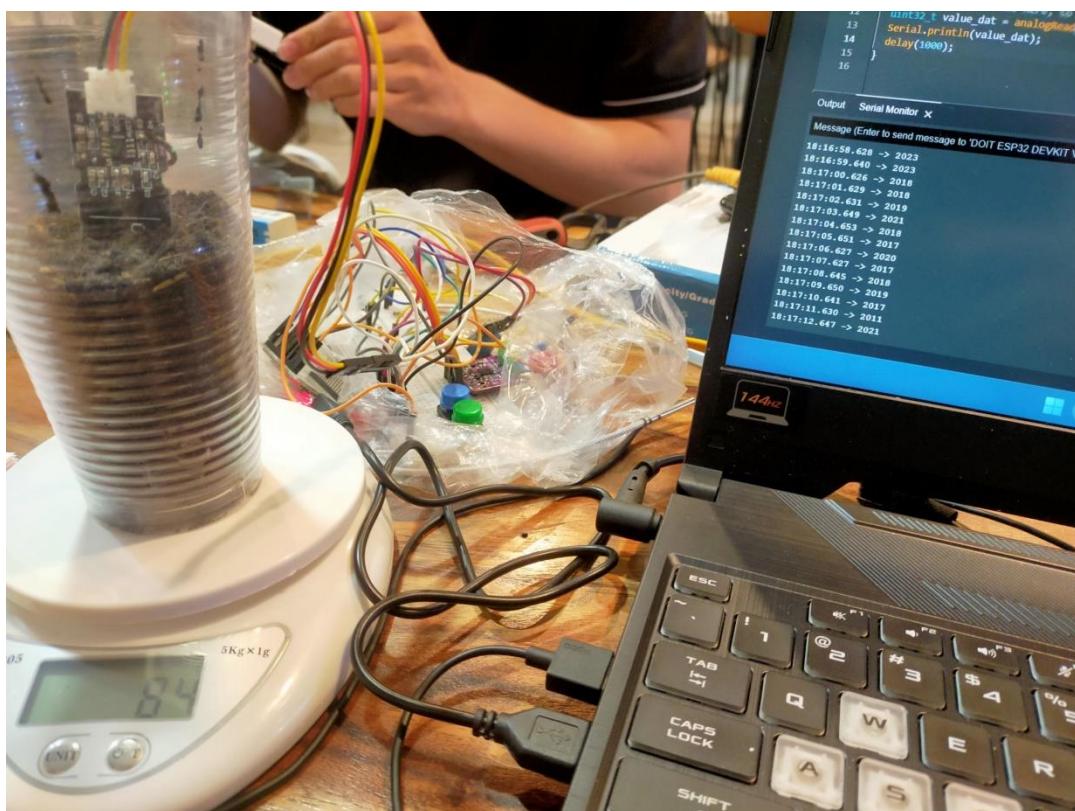
Hình 58: Giá trị analog ứng với độ ẩm đất 20% khoảng 2325

Sau khi đo 2 khoảng 0% và 20% ứng với giá trị analog lần lượt là 2750 và 2325. Ta thử áp dụng vào công thức (**) để kiểm tra tính chính xác của công thức:

$$W = \frac{(2750 - 2325).(20 - 0)}{(2325 - 2750) + 0} = 20\%$$

- Lần 2: Độ ẩm đất 20 - 40%

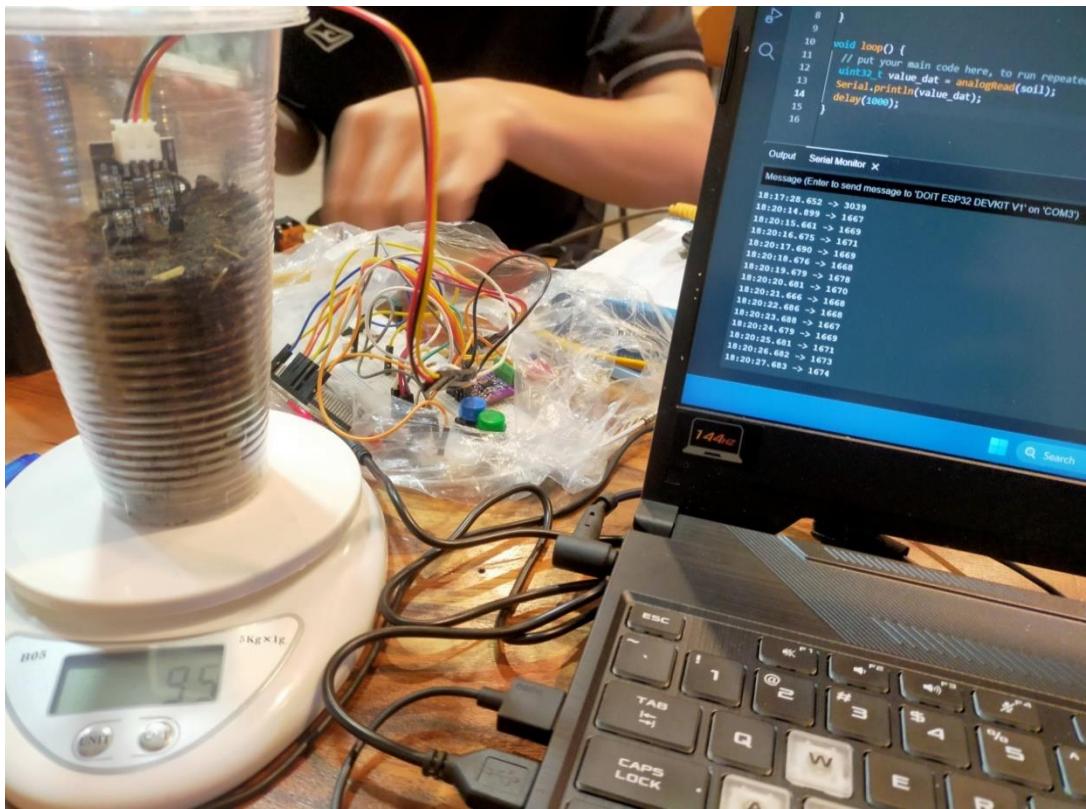
Sau khi đã thực hiện xong lần 1, ta tiếp tục cho thêm 10ml nước để đo lần 2. Lúc này, độ ẩm đất theo lý thuyết là 40%, giá trị analog trả về lúc này là 2020



Hình 59: Giá trị analog ứng với độ ẩm đất 40% khoảng 2020

- Lần 3: Độ ẩm đất 40 - 60%

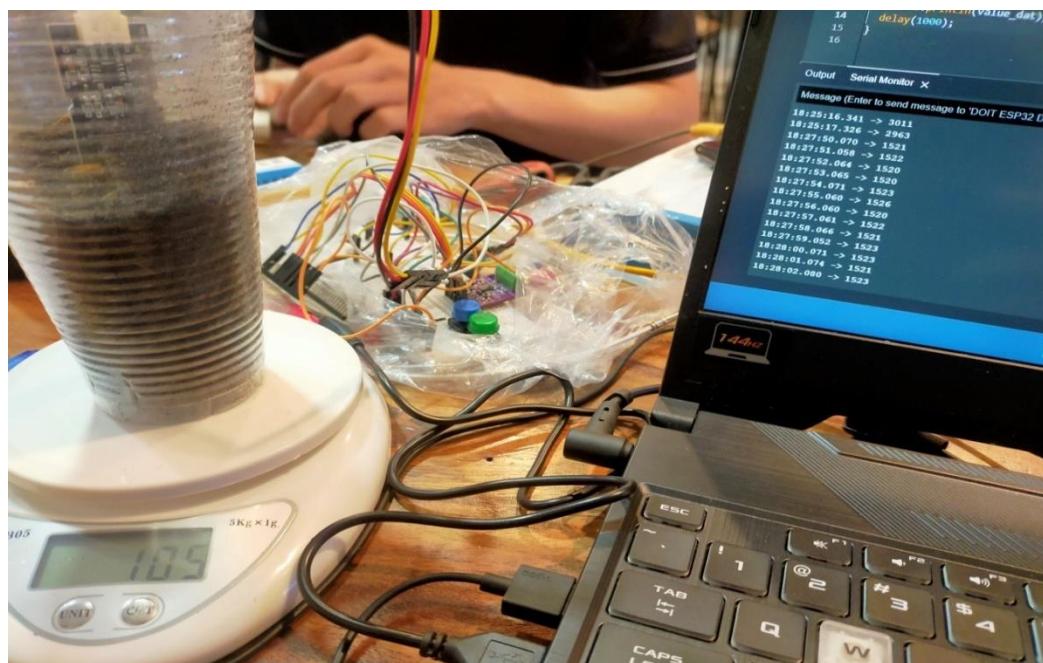
Sau khi đã thực hiện xong lần 2, ta tiếp tục cho thêm 10ml nước để đo lần 3. Lúc này, độ ẩm đất theo lý thuyết là 60%, giá trị analog trả về lúc này khoảng 1670.



Hình 60: Giá trị analog ứng với độ ẩm đất 60% khoảng 1670

- Lần 4: Độ ẩm đất 60 - 80%

Sau khi đã thực hiện xong lần 3, ta tiếp tục cho thêm 10ml nước để đo lần 4. Lúc này, độ ẩm đất theo lý thuyết là 60%, giá trị analog trả về lúc này khoảng 1520.

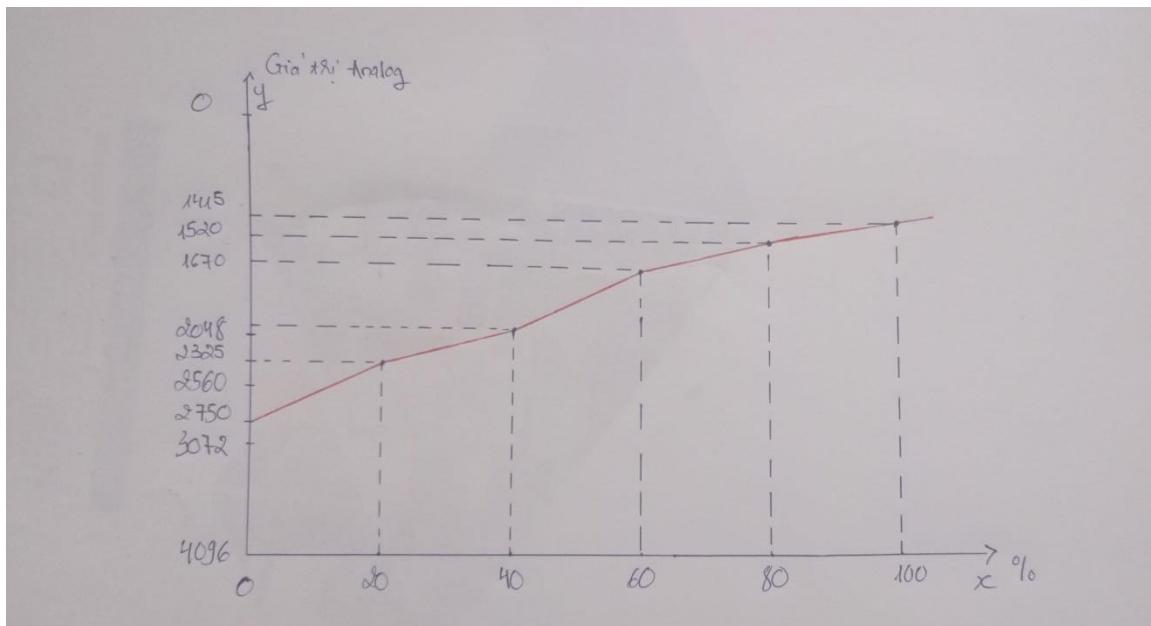


Hình 61: Giá trị analog ứng với độ ẩm đất 80% khoảng 1520

- Lần 5: Độ ẩm đất 80 - 100%:

Sau khi đã thực hiện xong lần 4, ta tiếp tục cho thêm 10ml nước để đo lần 5. Lúc này, độ ẩm đất theo lý thuyết là 100%, giá trị analog trả về lúc này khoảng 1415.

Tiến hành vẽ đồ thị tín hiệu đầu ra của cảm biến:



Hình 62: Đồ thị tín hiệu đầu ra của cảm biến

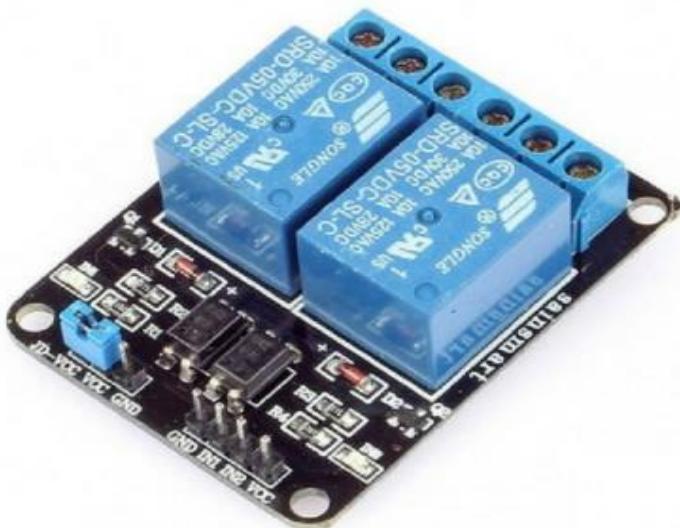
* **Bước 3: Kết luận**

Tóm lại, sau khi tìm hiểu nhiều tài liệu về cảm biến điện dung Moisture Sensor V1.2, nhóm tiến đến kết luận như sau:

Moisture Sensor V1.2 là cảm biến điện dung giá rẻ, chỉ phù hợp để làm đồ án sinh viên. Chính vì vậy, khả năng chính xác của nó không cao như những cảm biến độ ẩm đất trong lĩnh vực công nghiệp. Về mặt công thức chuyển đổi, nhóm đã kiểm chứng công thức trên là đúng với điều kiện trong từng khoảng chặn analog. Ở đây nhóm chặn 1 khoảng là 20% và tiến hành chặn 5 lần. Trong thực tế, giá trị analog mà cảm biến trả về sẽ không tuyến tính 1 đường thẳng (có thể do môi trường đất, nhiễu,...). Cứ trong 1 khoảng chặn, giá trị analog ngưỡng trên và dưới sẽ thay đổi, đó là lý do tại sao nhóm tiến hành đo ngoài thực tế để đưa ra các khoảng chặn analog gần đúng. Tất nhiên, độ chính xác sẽ tỉ lệ thuận với việc chia nhiều khoảng chặn, nhưng vì điều kiện không cho phép, nhóm chỉ chia ra 5 khoảng để lập trình.

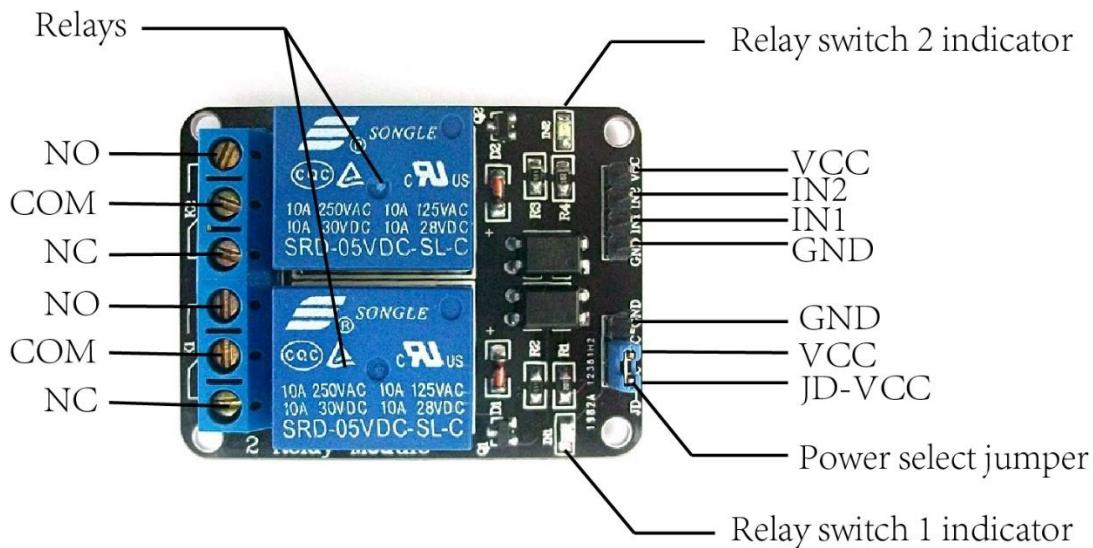
2.7. Các linh kiện cần thiết cho cơ cấu chấp hành.

2.7.1. Module relay 5v 2 kênh



Hình 63 Module relay 5v 2 kênh

Module Relay 2-Channel 5V được dùng nhiều trong các ứng dụng đóng ngắt các thiết bị tiêu thụ dòng điện lớn (<10A). Module có thể đóng ngắt cùng lúc hai kênh bằng tín hiệu điều khiển (với mức điện áp 3V3 hoặc 5V) từ các vi điều khiển khác nhau như: Arduino, 8051, AVR, PIC, DSP, ARM, MSP430, logic TTL, đồng thời module được cách ly bằng optocoupler giúp bảo vệ tốt hơn cho các vi điều khiển.



Hình 64: Sơ đồ chân Module relay 5v 2 kênh

Module được kết nối với các board điều khiển bằng 4 chân header như sau:

VCC cung cấp nguồn cho các opto.

GND kết nối với GND của board điều khiển.

IN1 và **IN2** dùng để điều khiển relay 1 và relay 2, tích cực mức thấp

Ngoài ra còn một 3 chân header được dùng để cấp nguồn cho relay, header này sẽ có một jumper dùng để kết nối chân VCC với chân RY_VCC mục đích dùng chung nguồn VCC (5V) từ header 4 chân cho relay, thông thường jumper được nối lại với nhau. Nếu như muốn cách ly tín hiệu điều khiển với nguồn cấp cho relay thì có thể bỏ jumper này ra và cấp nguồn riêng 5V cho chân RY_VCC.

Thông số kỹ thuật

Đóng ngắt được dòng điện cao: AC250V 10A, DC30V 10A

2 led báo trạng thái relay

Điện áp điều khiển: 5V

Mạch cách ly bằng opto

Kích thước: 50x45 mm

2.7.2. Bóng đèn dây tóc



Hình 63:Bóng đèn dây tóc

Thông số kỹ thuật:

- Thương hiệu: Điện Quang
- Công suất: 60W
- Đuôi đèn:E 27 phô biến
- Nguồn điện: 220V
- Loại bóng: Sợi đốt (dây tóc)
- Đường kính bóng: 50mm

Đặc tính: ánh sáng vàng ấm tỏa nhiệt, có thể sử dụng úm gà, vịt. Có thể thắp sáng cho thanh long.....

2.7.3. Quạt thông gió



Hình 65: Quạt thông gió

Thông Số kỹ thuật:

- Điện Áp 220v
- Kích Thước 12x12cmx3.8cm
- Dùng để làm mát máy móc thiết bị
- Thông gió phòng ăn phòng ngủ...
- Quạt Tản Nhiệt ORIX cao cấp
- Các thông số của quạt ORIX 220VAC
- Điện áp: 200V - 240V
- Dòng điện: 0.14A
- Kích thước: 120x120x38mm
- Tốc độ: 2800-3250rpm
- Cánh quạt làm bằng nhựa Plastic cao cấp
- Khung quạt làm bằng kim loại: chắc chắn

2.7.4. Động cơ bơm phun sương



Hình 66: Động cơ bơm phun sương

Thông số kỹ thuật

- Tên sản phẩm: Máy bơm áp lực nước mini 12V
- Thương hiệu: SAMAFA
- Phân loại: máy áp lực nước bơm màng có thể tự hút nước
- Mã SP: SM-MB1
- Áp sử dụng: 12VDC
- Dòng điện tối đa: 5A
- Áp lực nén: 7.0 bar
- Lưu lượng: 6-9 L/Phút
- Công suất: 60-65W
- Công tắc áp lực: ngắt điện máy bơm áp khi khóa van nước
- Chất liệu: đồng, nhựa, cao su
- Máy bơm tăng áp mini 12v + 2 đầu chuyển ống 8mm
- Tình trạng: mới

Ưu điểm

- Dòng máy tăng áp lực nước chính hãng mạnh mẽ và bền bỉ.
- Máy bơm áp lực 12v có công tắc áp lực tự ngắt khi đường ống quá áp.

Ví dụ: Khi bóp cần xịt nước để rửa xe, áp đường ống giảm, bơm sẽ chạy. Khi ngưng bóp cần xịt nước thì ngõ ra bị bịt lại dẫn tới áp đường ống tăng, bơm tự ngắt. Đây là dòng bơm áp lực mini sử dụng điện áp 12V, là điện áp chuẩn của bình acquy chuyên dùng nên tính linh động cao nên máy tăng áp nước tiện dụng sử dụng ở những nơi không có điện.

- Máy bơm áp lực mini có thể mang đi bất cứ đâu để phục vụ cho công việc sau:
 - + Máy tăng áp mini dùng cho vòi tưới cây ở những nơi áp lực nước yếu như sân thượng, vườn cây.
 - + Tăng áp lực cấp nước cho vòi hoa sen, máy giặt, nguồn nước dùng trong nhà.
 - + Phun thuốc trừ sâu, phân bón cho đồng ruộng. Thay thế cho máy xăng và cần xịt bơm tay lạc hậu.
 - + Vệ sinh máy lạnh, xe máy, xe tải, tàu thuyền ở những nơi không có điện hoặc cần sự cơ động.
 - + Chế máy rửa chén đĩa, vật dụng, dụng cụ.

2.8. Arduino –IDE

2.8.1. Phần mềm Arduino IDE là gì?



Hình 67: LOGO giao diện IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.

Đây là một phần mềm Arduino chính thống, giúp cho việc biên dịch mã trở nên dễ dàng, ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được.

2.8.2. Arduino IDE hoạt động như thế nào?

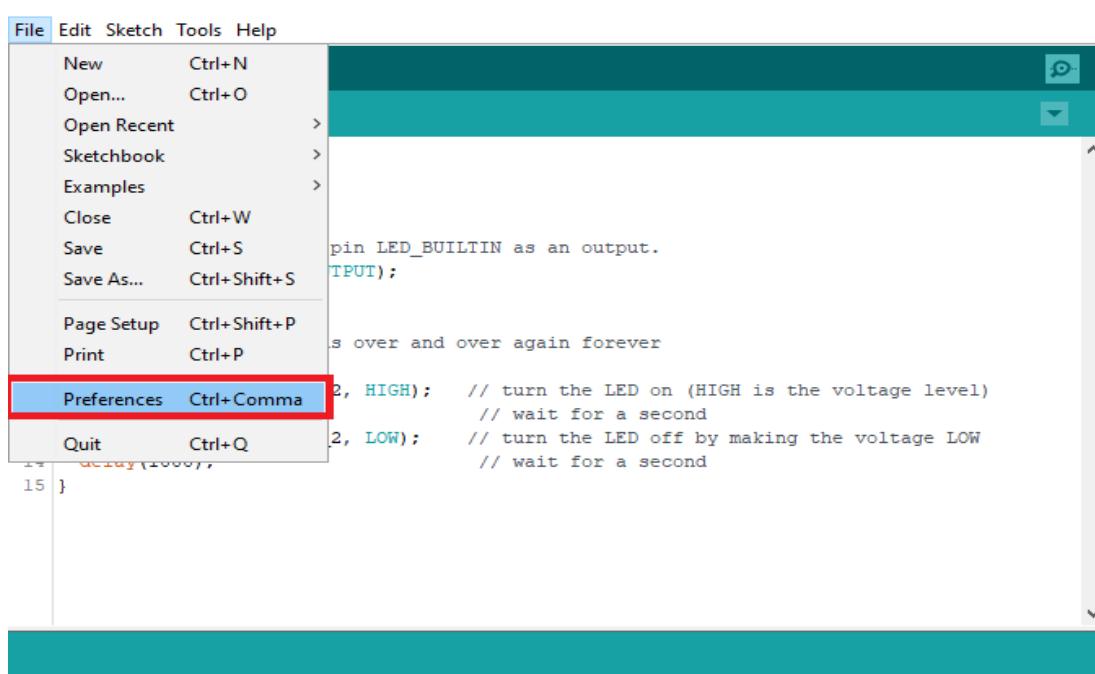
Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

2.8.3. Ưu điểm Arduino IDE

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

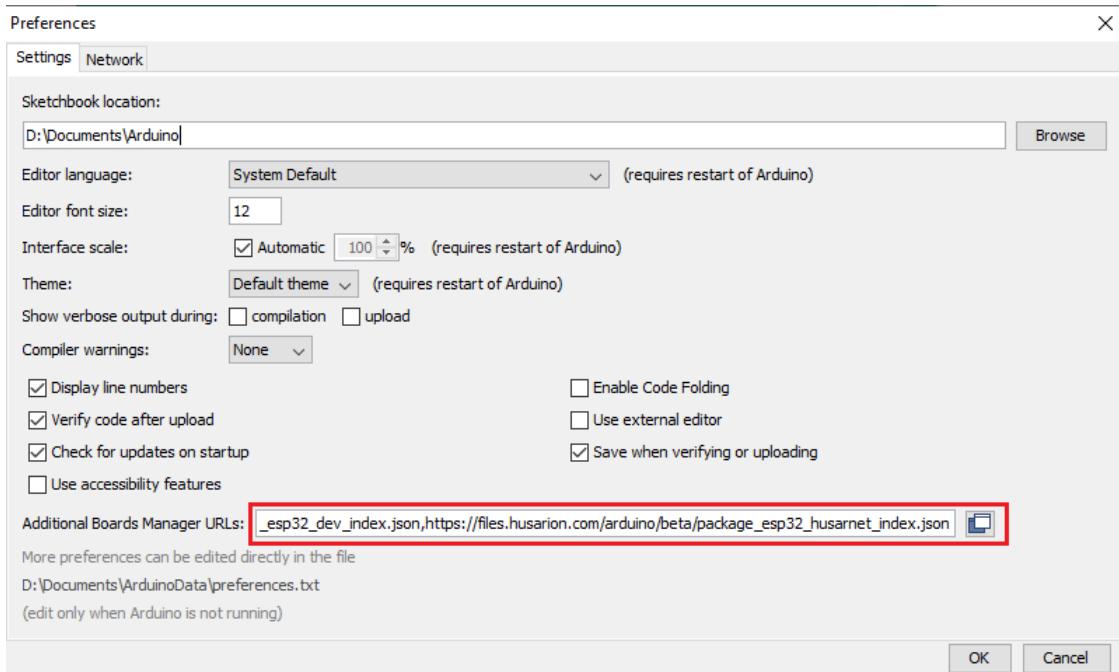
2.8.4. Cài đặt bổ sung ESP32 board cho arduino IDE

Mở arduino IDE chọn **File>Preferences**



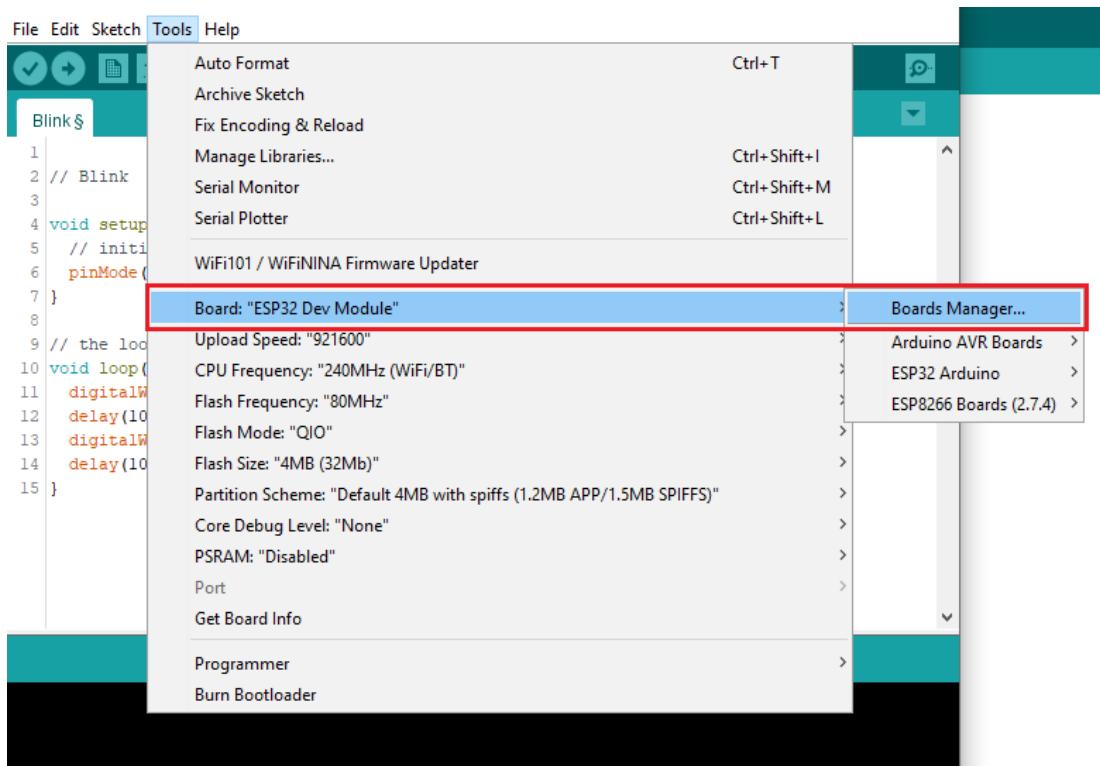
Hình 68: Cài đặt bổ sung ESP32 board cho arduino IDE

Nhập https://dl.espressif.com/dl/package_esp32_index.json vào trường “additional Board Manager URLs” trong hình sau đó nhấn “OK”.



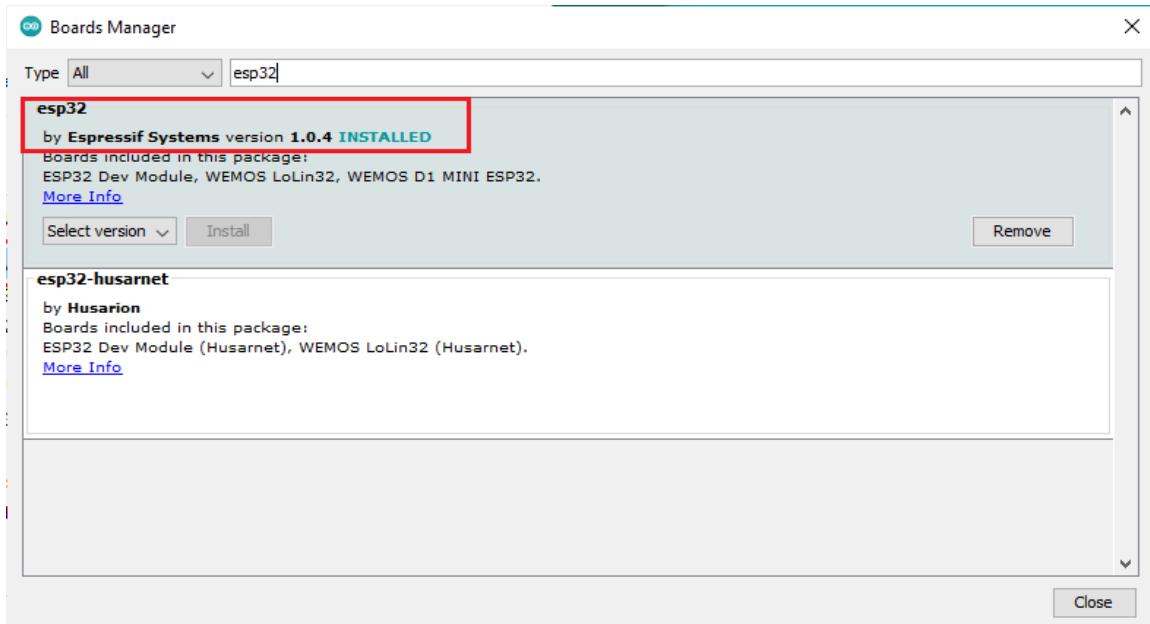
Hình 69: Cài đặt bổ sung ESP32 board cho arduino IDE

Mở Board Manager > Tools > Board > Boards Manager...



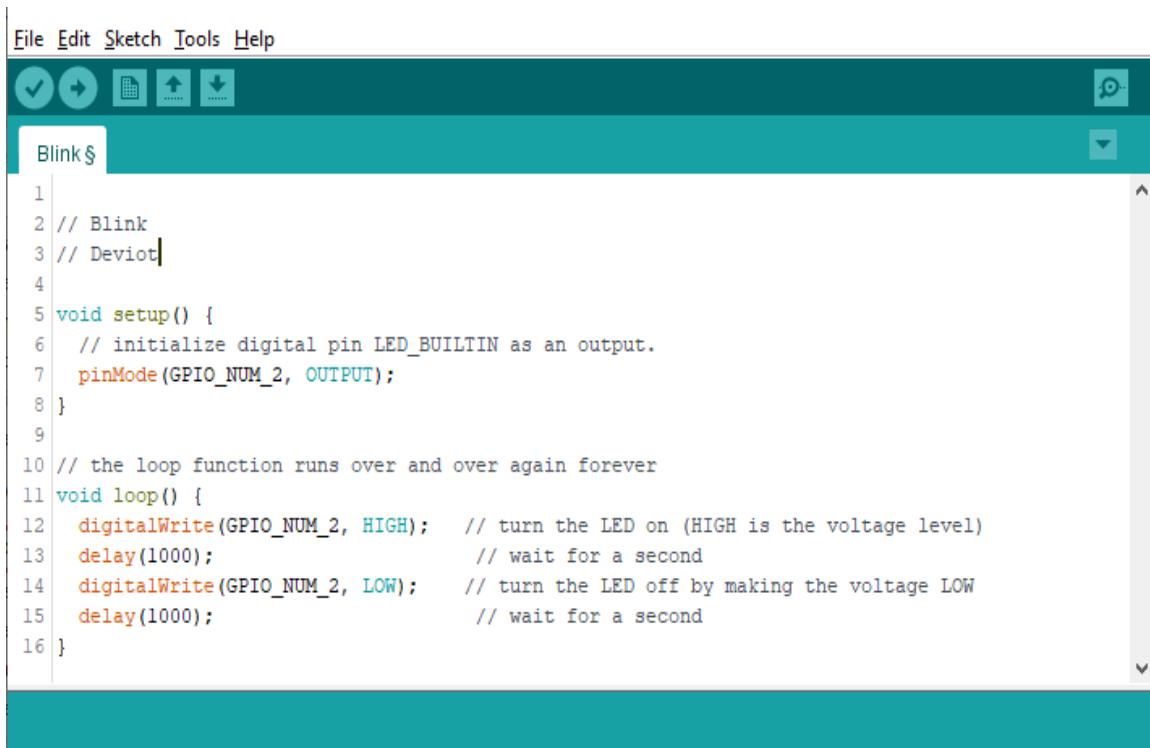
Hình 70: Cài đặt bổ sung ESP32 board cho arduino IDE

Tìm ESP32 và nhấn cài đặt cho “ESP32 by Espressif”



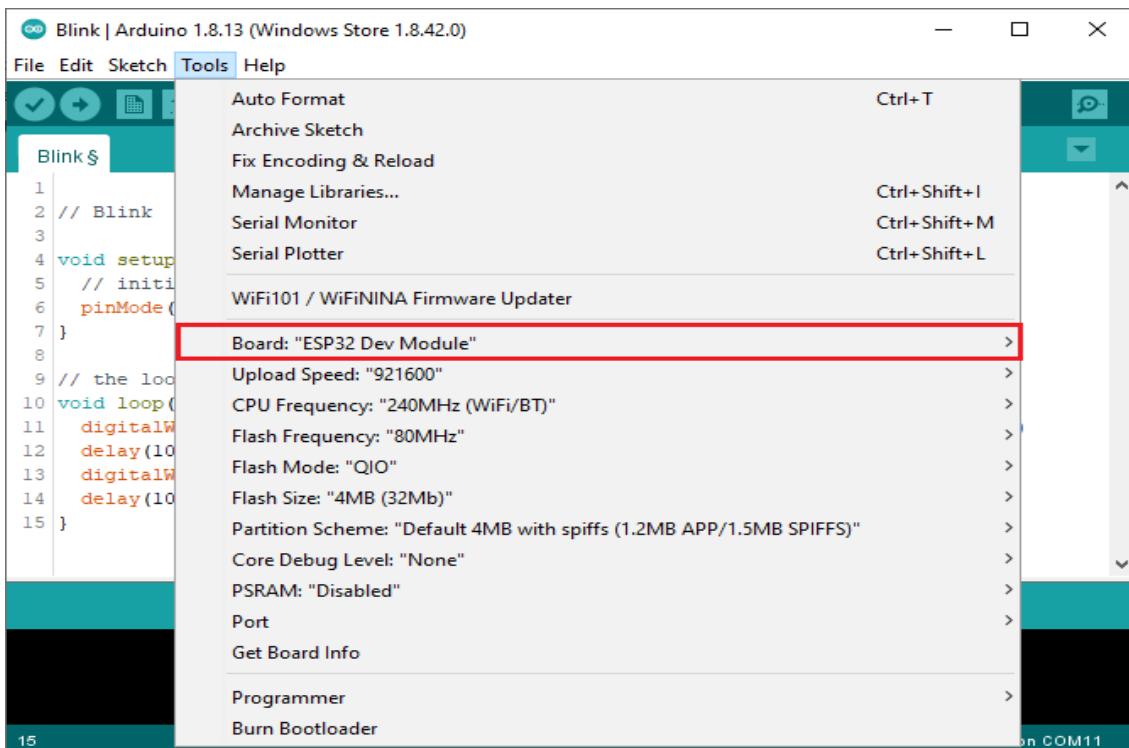
Hình 71: Cài đặt bổ sung ESP32 board cho arduino IDE

Thử một chương trình với ESP32.Tìm đến example Blink của Arduino



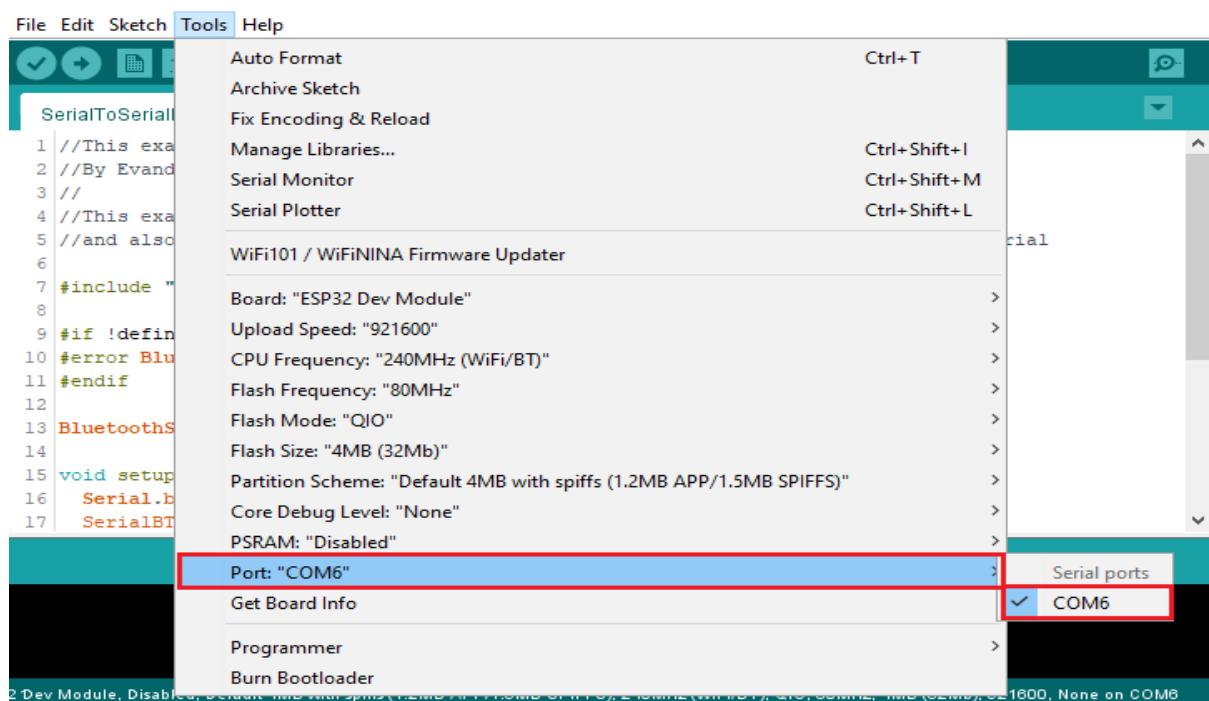
Hình 72: Cài đặt bổ sung ESP32 board cho arduino IDE

Chon board là **ESP32 Dev Module** hoặc chọn **DOIT ESP32 DEVKIT V1**



Hình 73: Cài đặt bổ sung ESP32 board cho arduino IDE

Chọn Port ,nếu như đã cắm ESP32 vào máy tính mà bạn không thấy cổng COM trong Arduino IDE của mình thì bạn hãy cài đặt **CP210x USB UART Bridge VCP Drivers .**



Hình 74: Cài đặt bổ sung ESP32 board cho arduino IDE

Upload chương trình chạy , quan sát LED xanh trên Board.

```

File Edit Sketch Tools Help
Blink
1
2 // Blink
3 // Deviot
4
5 void setup() {
6     // initialize digital pin LED_BUILTIN as an output.
7     pinMode(GPIO_NUM_2, OUTPUT);
8 }
9
10 // the loop function runs over and over again forever
11 void loop() {
12     digitalWrite(GPIO_NUM_2, HIGH);    // turn the LED on (HIGH is the voltage level)
13     delay(1000);                   // wait for a second
14     digitalWrite(GPIO_NUM_2, LOW);   // turn the LED off by making the voltage LOW
15     delay(1000);                   // wait for a second
16 }

```

Compiling sketch...

3 ESP32 Dev Module on COM11

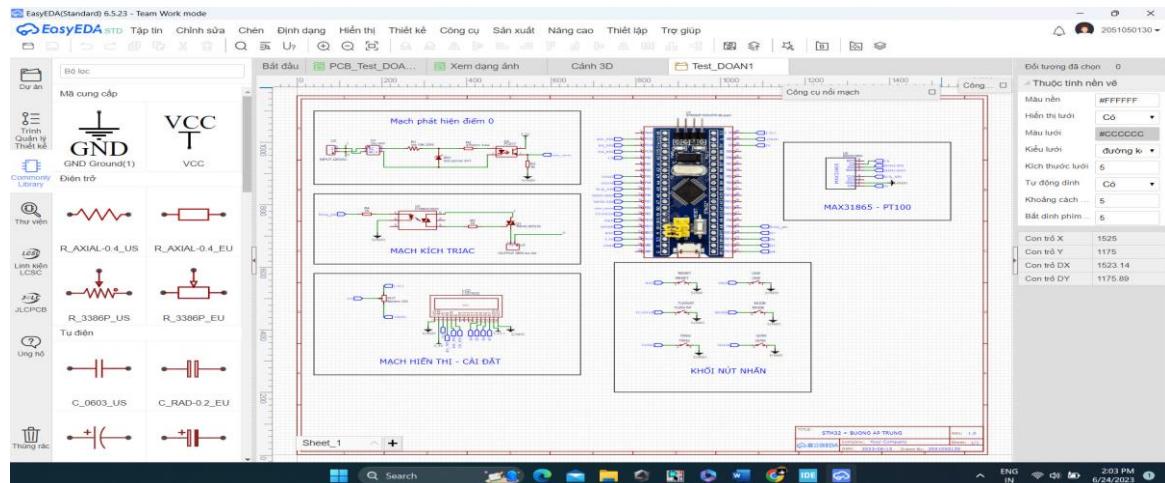
Hình 75: Cài đặt bổ sung ESP32 board cho arduino IDE

2.9. EASY EDA

2.9.1. Tổng quan về phần mềm Easy EDA:

EasyEDA là một phần mềm thiết kế mạch điện trực tuyến (EDA - Electronic Design Automation) mạnh mẽ và dễ sử dụng. Nó cung cấp một môi trường đồ họa để bạn có thể thiết kế mạch điện, mô phỏng, và làm việc với các linh kiện điện tử.

Phần mềm Easy EDA cho phép bạn thiết kế mạch điện một cách trực quan, mô phỏng và mô phỏng ngược các linh kiện trong mạch điện trước khi tiến hành sản xuất, thiết kế bo mạch in PCB một cách chính xác,... Nhìn chung, đây là phần mềm phù hợp và dễ sử dụng đối với những người làm điện tử.

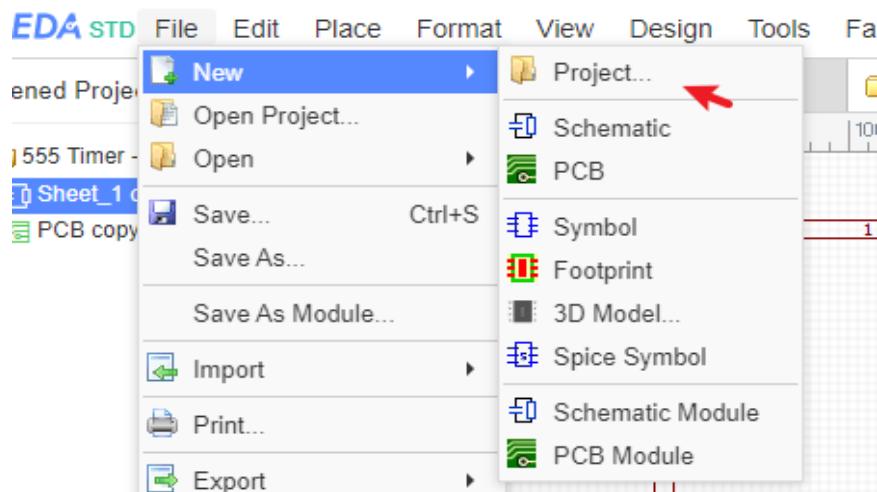


Hình 76: Giao diện làm việc trên phần mềm Easy EDA

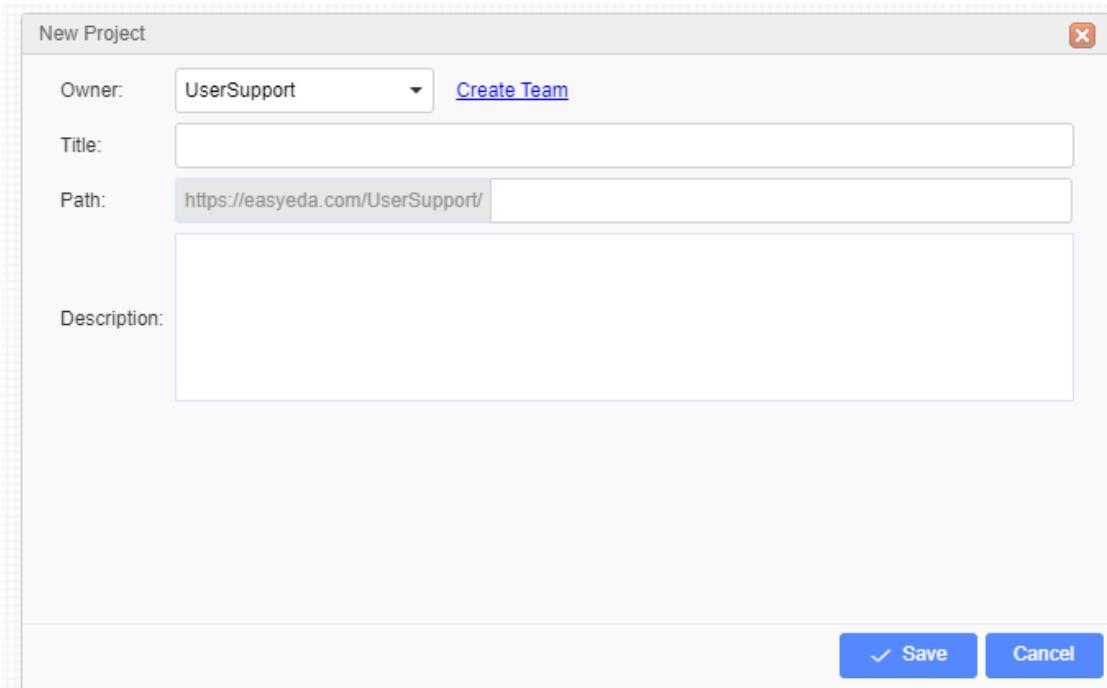
2.9.2. Cách tạo một project mới trên phần mềm Easy EDA:

Sau khi mở phần mềm, để tạo một project mới, ta thực hiện các thao tác:

File > New > Create a new project/Schematic..etc



Hình 77: Hướng dẫn tạo project mới trên Easy EDA



Hình 78: Hướng dẫn tạo project mới trên Easy EDA

Trong đó:

Owner: Đây là mục cho phép thay đổi chủ sở hữu của dự án; ta có thể chuyển chủ sở hữu cho nhóm mà mình đã tham gia.

Title: Tiêu đề hiển thị của dự án.

Path: EasyEDA cho phép ta đặt đường dẫn cho dự án, giúp dễ dàng chia sẻ với mọi người.

Description: Thêm một mô tả ngắn giúp bất kỳ ai bạn chia sẻ dự án này hiểu rõ về nội dung của dự án.

2.10. Mạch giảm áp LM2596

2.10.1. Khái niệm



Hình 79: Mạch giảm áp LM2596

Mạch giảm áp LM2596 là một loại mạch giảm áp điện áp (Voltage Regulator) sử dụng linh kiện điện tử, được sản xuất bởi hãng Linear Technology (nay thuộc về công ty Analog Devices) và cũng có các phiên bản từ các nhà sản xuất khác dưới tên gọi tương tự. LM2596 là một loạt mạch giảm áp dựa trên kỹ thuật điều khiển

độ rộng xung (PWM - Pulse Width Modulation), được sử dụng để điều chỉnh điện áp đầu ra từ một nguồn điện áp cao (thường từ 9V đến 40V) xuống một giá trị điện áp thấp hơn (thường từ 1.2V đến 35V), tùy theo phiên bản cụ thể.

LM2596 cung cấp khả năng ổn định và ổn định điện áp đầu ra, đồng thời có khả năng điều chỉnh điện áp đầu ra theo yêu cầu. Điều này làm cho nó trở thành một giải pháp phổ biến cho các ứng dụng yêu cầu cung cấp điện áp ổn định và độ chính xác cao như trong các dự án điện tử tự chế, nguồn cung cấp điện ổn định, sạc pin, và nhiều ứng dụng khác.

2.10.2. Thông số kỹ thuật

- Điện áp đầu vào: 4V-35V
- Điện áp đầu ra: 1.23V-30V
- Dòng đầu ra: 3A (max)
- Hiệu suất chuyển đổi: 92% (tối đa)
- Tần số hoạt động module hạ áp: 150kHz
- Nhiệt độ hoạt động: -40 °C đến +85 °C
- Kích thước: 23x14x8(mm)

2.10.3. Nguyên lý hoạt động

LM2596 hoạt động dựa trên nguyên tắc điều khiển độ rộng xung (PWM - Pulse Width Modulation) để giảm điện áp đầu ra từ nguồn điện áp đầu vào cao hơn xuống giá trị điện áp mong muốn. Dưới đây là nguyên lý hoạt động cơ bản của LM2596:

Ngõ vào (Input): LM2596 được cung cấp điện áp từ một nguồn điện áp cao hơn (V_{in}), thường từ 9V đến 40V. Nguồn điện này có thể là nguồn điện năng lượng xung (chẳng hạn như nguồn từ pin, nguồn điện tử), hoặc có thể là nguồn điện cố định từ mạng điện.

So sánh và Điều khiển: LM2596 sử dụng một bộ so sánh để so sánh điện áp đầu ra với một điện áp tham chiếu (được thiết lập bởi người dùng). Khi điện áp đầu ra cao hơn hoặc thấp hơn điện áp tham chiếu, bộ so sánh tạo ra một tín hiệu điều khiển

PWM: Tín hiệu điều khiển được đưa vào một bộ điều khiển PWM. Bộ điều khiển này sẽ điều chỉnh thời gian mở (duty cycle) của xung PWM dựa trên tín hiệu điều khiển. Khi điện áp đầu ra cao hơn so với điện áp tham chiếu, thời gian mở của xung PWM sẽ giảm, giúp giảm điện áp đầu ra. Khi điện áp đầu ra thấp hơn so với điện áp tham chiếu, thời gian mở của xung PWM sẽ tăng, giúp tăng điện áp đầu ra.

Điện áp đầu ra (Output): Điện áp đầu ra được điều chỉnh bằng cách điều chỉnh thời gian mở của xung PWM. Điện áp đầu ra ổn định và độ chính xác cao, và nó sẽ duy trì ở mức mong muốn dù có biến động trong nguồn điện đầu vào.

Lọc và ổn định: Sau khi xung PWM đi qua một cuộn cảm và một tụ lọc, điện áp đầu ra trở nên ổn định và không có các dao động cao tần.

Với nguyên lý hoạt động này, LM2596 cho phép bạn điều chỉnh điện áp đầu ra theo yêu cầu, cung cấp điện áp ổn định và độ chính xác cao cho các ứng dụng điện tử và nguồn cung cấp điện.

2.10.4. **Ưu điểm và nhược điểm**

***Ưu điểm:**

- **Hiệu suất cao:** LM2596 thường có hiệu suất chuyển đổi cao, giúp giảm sự tiêu thụ năng lượng và tạo ra ít nhiệt hơn so với các mạch giảm áp truyền thống dựa trên linear regulators.

- **Điều chỉnh điện áp đầu ra linh hoạt:** LM2596 cho phép điều chỉnh điện áp đầu ra theo yêu cầu bằng cách thay đổi các thông số linh kiện bên trong mạch. Điều này làm cho nó trở thành một lựa chọn tốt cho nhiều ứng dụng.

- **Ôn định và độ chính xác cao:** LM2596 cung cấp điện áp đầu ra ổn định và độ chính xác cao, giúp bảo vệ các linh kiện điện tử khỏi biến đổi điện áp không mong muốn.

- **Bảo vệ và tích hợp tính năng an toàn:** Một số phiên bản của LM2596 có tích hợp các tính năng bảo vệ như bảo vệ quá nhiệt độ, bảo vệ quá dòng, và bảo vệ quá điện áp, giúp ngăn ngừa hỏng hóc và sự cố.

- **Ứng dụng đa dạng:** LM2596 có thể được sử dụng trong nhiều ứng dụng khác nhau như nguồn cung cấp điện cho các dự án tự chế, sạc pin, nguồn cung cấp cho vi xử lý, và nhiều ứng dụng điện tử khác.

***Nhược điểm:**

- **Tiêu thụ dòng điện đứng yên (quiescent current):** Một số phiên bản của LM2596 có tiêu thụ dòng điện đứng yên tương đối cao, điều này có thể ảnh hưởng đến hiệu suất năng lượng trong ứng dụng yêu cầu tiêu thụ điện năng thấp.

- **Nhiệt độ hoạt động:** LM2596 có thể tạo ra một lượng nhiệt độ tương đối cao khi hoạt động ở dải điện áp và dòng lớn. Việc quản lý nhiệt độ trong thiết kế là quan trọng để đảm bảo hiệu suất và tuổi thọ của mạch.

- **Giá cả:** Một số phiên bản của LM2596 có giá thành tương đối cao so với một số mạch giảm áp đơn giản hơn. Tuy nhiên, điều này có thể được bù đắp bằng hiệu suất chuyển đổi cao và tính linh hoạt trong việc điều chỉnh điện áp đầu ra.

- Tùy thuộc vào ứng dụng cụ thể, các ưu điểm và nhược điểm của LM2596 có thể quyết định xem nó có phù hợp hay không. Việc lựa chọn mạch giảm áp nên dựa trên yêu cầu cụ thể của dự án và điều kiện hoạt động.

2.11. Bảo vệ phần cứng hệ thống

2.11.1. Vì sao cần bảo vệ phần cứng hệ thống?

Bảo vệ phần cứng hệ thống cũng chính là bảo vệ cơ cấu hoạt động sao cho tăng độ hiệu quả, khả năng hoạt động và tính chính xác của toàn bộ hệ thống.

Mục đích của việc bảo vệ hệ thống:

Bảo vệ dữ liệu: Phần cứng hệ thống là nơi lưu trữ dữ liệu, nếu bị hỏng hoặc bị đánh cắp sẽ dẫn đến mất dữ liệu.

Bảo vệ tính toàn vẹn và khả năng vận hành của hệ thống: Phần cứng hệ thống là nền tảng để hệ thống hoạt động, nếu bị hỏng hoặc bị tấn công sẽ khiến hệ thống ngừng hoạt động.

Tăng tuổi thọ của thiết bị: Việc bảo vệ các thành phần hệ thống sẽ giúp từng linh kiện được đảm bảo hoạt động trong các môi trường khác nhau, từ đó tuổi thọ linh kiện sẽ tăng cao.

Tiết kiệm chi phí: Bảo vệ hệ thống cũng chính là tiết kiệm chi phí bảo trì, sửa chữa và thay thế linh kiện mỗi khi có sự cố.

2.11.2. Bảo vệ hệ thống cho mô hình trồng cây trong nhà kính:

2.11.2.1. Bảo vệ mạch điện:

- Mạch điện sử dụng trong mô hình (bao gồm vi điều khiển ESP32, mạch kích triac, mạch phát hiện điểm 0, mạch level shifter, mạch mosfet công suất) là những phần cần được bảo vệ khỏi các tác nhân gây tổn hại như: nước, ánh sáng cường độ cao, nhiệt độ, chất ăn mòn, các tổn hại vật lý do động thực vật gây ra...

Trong đồ án lần này, nhóm quyết định thực hiện xây dựng mô hình hộp bảo vệ để đựng các mạch điện:

Kích thước: Dài x Rộng x Cao: 21 x 15 x 7 (Cm)

Chất liệu: Giấy PVC Foam, keo nén định hình và băng keo đen

Tính năng: Chống thấm nước, cách điện, chịu được nhiệt độ nhất định, chống oxy hóa.

CHƯƠNG 3. LẬP TRÌNH – THIẾT KẾT VÀ KẾT QUẢ ĐẠT ĐƯỢC

Ở chương này, nhóm nghiên cứu sẽ tiến hành phân tích sơ đồ khói, sơ đồ nguyên lý hệ thống, phân tích thuật toán, chạy code – nạp code, cũng như đấu nối thực tế và kiểm thử thực tế. Từ đó, người đọc chắc chắn có thể hiểu sâu về hướng đi và quá trình thực hiện của nhóm, đồng thời nhìn thấy rõ những kết quả mà nhóm nghiên cứu đạt được.

3.1. Yêu cầu cấu tạo hệ thống – Sơ đồ khói – Chức năng của các khối:

3.1.1. Yêu cầu của hệ thống “trồng cây húng quế trong nhà kính”

Hệ thống có các chức năng sau:

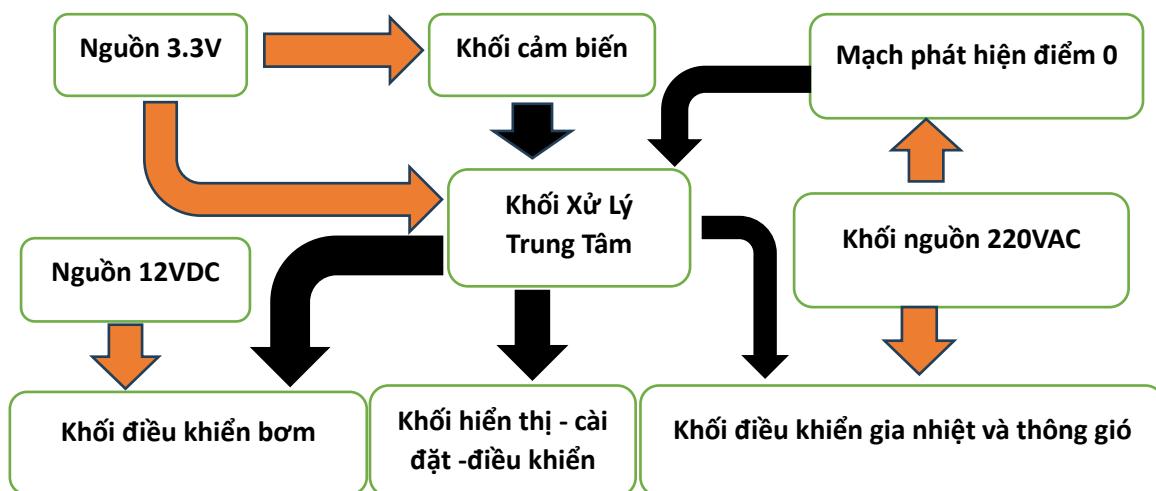
Hệ thống có thể giám sát, cài đặt, và điều khiển nhiệt độ không khí và độ ẩm đất bám sát theo thông số thích hợp cho cây húng quế phát triển và ra lá.

Hệ thống có thể giám sát – cài đặt – điều khiển ngay tại mạng local hoặc bất kì nơi nào trên thế giới.

Hệ thống sẽ gửi thông báo khi nhiệt độ và độ ẩm đất vượt ngưỡng.

3.1.2. Sơ đồ khói và chức năng mỗi khối

Sơ đồ khói:



Hình 80: Sơ đồ khói hệ thống

Trong đó, đường cam, đường đen lần lượt là dây nguồn và dây tín hiệu.

❖ Chức năng mỗi khối:

Khối nguồn 3.3VDC: Cấp nguồn cho khối xử lý trung tâm và khối cảm biến.

Khối cảm biến: Sử dụng MAX31865 đọc tín hiệu điện trở từ cảm biến nhiệt độ PT100, sau đó giao tiếp SPI với khối xử lý trung tâm. Sử dụng cảm biến nhiệt độ

DHT11 làm nhiệt độ tham chiếu môi trường và gửi về ESP32. Sử dụng cảm biến Capacitive Soil để đo độ ẩm đất và gửi tín hiệu về ESP32.

Khối hiển thị - cài đặt – điều khiển: Sử dụng Node-Red Dashboard UI để giám sát nhiệt độ và độ ẩm đất, cài đặt ngưỡng cảnh báo, chế độ hoạt động, xem lịch sử cảm biến và hiển thị thông báo.

Mạch phát hiện điểm 0: Mạch này sẽ lấy thời điểm tín hiệu điện AC gần bằng 0, ngay lúc này sẽ gửi tín hiệu đến khối xử lý trung tâm để thực hiện kiểm soát công suất cho nhiệt độ nhà kính.

Mạch kích Triac: nhận tín hiệu điểm 0, kết hợp với thuật toán PID từ đó sẽ xuất ra tín hiệu Digital phù hợp để kiểm soát nhiệt độ nhà kính.

Khối nguồn 220VAC-50Hz: Sử dụng nguồn điện gia đình cấp nguồn cho bóng đèn sợi đốt, quạt và mạch phát hiện điểm 0.

Khối xử lý trung tâm: Sử dụng VDK ESP32 DEV KIT V1 để lập trình chuyển đổi giá trị điện trở từ PT100 về giá trị nhiệt độ chuẩn xác, nhận tín hiệu điểm 0 kết hợp với thuật toán PID, ngắt ngoài. Xuất xung PWM tỉ lệ thuận với độ ẩm đất, chuyển đổi analog thành phần trăm độ ẩm đất. Đóng vai trò là 1 client để kết nối với MQTT Broker, điều khiển quạt, bơm, gửi giá trị cảm biến thông qua giao thức MQTT v.v...

Khối điều khiển nhiệt độ và thông gió: Khối này sử dụng relay 5V 2 kênh để điều khiển quạt và bật tắt chế độ PID nhiệt độ. Khối bao gồm mạch kích triac.

Khối điều khiển bơm: Khối này gồm mạch level shifter và mạch mosfet công suất, với mục đích là xuất điện áp phù hợp để bơm phun sương một cách đúng đắn.

3.2. Quá trình xây dựng – hoạt động của hệ thống:

Giai đoạn 1: Xây dựng khối cảm biến nhiệt độ và cảm biến độ ẩm đất.

B1: Tìm hiểu về VDK ESP32, lập trình chớp tắt led cơ bản, lý thuyết ngắt, timer, SPI, ADC...

B2: Chọn 1 phần mềm để lập trình VDK xuyên suốt đồ án, ở đây nhóm nghiên cứu sử dụng Arduino IDE để lập trình, nạp code và debug.

B3: Đọc Datasheet và viết thư viện cho Max31865 và Capacitive Soil, add thư viện vào source code. Điều chỉnh giá trị Rref trên MAX31865 đúng với giá trị đo được từ VOM. Nghiên cứu phương pháp đo độ ẩm đất theo tiêu chuẩn được công bố.

Giai đoạn 2: Xây dựng mạch phát hiện điểm 0, mạch kích Triac, mạch level shifter, mạch mosfet công suất.

B1:Cần tìm hiểu kỹ lý thuyết về 4 mạch này (đã được giải thích ở chương 2), sau khi nắm vững lý thuyết ta tiến hành đấu nối trên testboard.

B2: Ta kiểm tra mạch phát hiện điểm 0 có hoạt động tốt hay không bằng cách sử dụng ngắt ngoài ở ESP32, đặt 1 biến đếm và cho biến đếm tăng lên mỗi khi xảy ra ngắt. Ta nghiên cứu datasheet của mosfet AOD4184 liệu có xuất đủ dòng cho động cơ hay không, nếu muốn xuất đủ dòng, điện áp đặt vào cổng Gate là bao nhiêu.

B3: Thực hiện debug và quan sát biến đếm có tăng đều với tần số 100Hz hay không, nếu không đều, cần xử lý chống nhiễu trên chân ngắt ngoài.Thực hiện xây dựng mạch level shifter với mosfet 2N7000.

Giai đoạn 3: Bắt đầu viết thuật toán PID, thuật toán PWM.

B1: Tìm hiểu về PID, nắm rõ công thức và chuyển đổi thành code. Đặt các biến như PID_error, PID_value. Nghiên cứu về PWM như duty cucle, điện áp trung bình v.v...

B2: Kích hoạt 1 timer trên ESP32 với mục đích tạo delay Microsecond, từ đó kiểm soát công suất của bóng đèn. Kiểm thử giá trị analogWrite 8 bit, xem thử động cơ bơm tốt ở ngưỡng bao nhiêu.

B3: Sau khi hiệu chỉnh các tham số Kp, Ki, Kd, duty cucle ổn. Ta tiến hành lập trình giải thuật theo cài đặt người dùng.

3.3. . Cấu hình và bắt đầu lập trình ESP32 trên Arduino IDE

3.3.1. Các thư viện sử dụng

Ở đồ án này, nhóm em sử dụng các thư viện sau:

```
#include <stdlib.h>
#include <SPI.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#include "Capacitive_Soil.h"
#include "Max31865_Khanh.h"
```

Hình 81: Thư viện sử dụng

Để kết nối và sử dụng giao thức MQTT, nhóm em cần add thư viện “PubSubClient.h”, “WiFi.h”.

Để đọc nhiệt độ và độ ẩm đất, nhóm em add thư viện DHT11 và tự viết 2 thư viện “Max31865_Khanh.h” và “Capacitive_Soil.h”.

Để có thể giao tiếp SPI với Max31865, nhóm em add thư viện “SPI.h”.

3.3.2. Cấu hình giao tiếp SPI cho ESP32 và MAX31865

Với ESP32 lập trình bằng Arduino IDE, ta chỉ cần khai báo các pin tương ứng với Pin trên chân ESP32 là ta đã có thể cấu hình trực tiếp Pinout cho ESP32.

Để giao tiếp SPI với MAX31865, ta cần cấu hình các pin như: MISO, MOSI, CS, CLK.

Sau đó ta cần lập trình để chọn các chế độ cho ESP32:

```
SPI.beginTransaction(SPISettings(500000, MSBFIRST, SPI_MODE1));
```

Hình 82: Cấu hình SPI

- + Xung nhịp trên chân CLK là 50KHz.
- + Truyền gói dữ liệu bit cao trước – MSBFIRST.
- + SPI_MODE1:

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

Hình 83: Mode SPI

SPI_MODE1 tương ứng với CPOL là 0 và CPHA là 1. Điều này có nghĩa là dữ liệu sẽ được slave và master truyền và nhận khi xung clock chuyển từ mức cao xuống mức thấp. Khi chân CS xuống mức thấp chân CLK của master (ESP32) sẽ xuất ra mức thấp trước.

Trong datasheet của MAX31865, nhà sản xuất đã ghi rõ tốc độ truyền dữ liệu tối đa của MAX31865 là 5MBit/s. Với tần số thạch anh 240MHz của ESP32 ta có thể dễ dàng cấp xung 50KHz cho slave.

AC Electrical Characteristics: SPI Interface

($3.0V \leq V_{DD} \leq 3.6V$, $T_A = -40^{\circ}C$ to $+125^{\circ}C$, unless otherwise noted. Typical values are $T_A = +25^{\circ}C$, $V_{DD} = V_{DVDD} = 3.3V$.) (Notes 3 and 7) (Figure 1 and Figure 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Data to SCLK Setup	t_{DC}	(Notes 8, 9)	35			ns
SCLK to Data Hold	t_{CDH}	(Notes 8, 9)	35			ns
SCLK to Data Valid	t_{CDD}	(Notes 8, 9, 10)			80	ns
SCLK Low Time	t_{CL}	(Note 9)	100			ns
SCLK High Time	t_{CH}	(Note 9)	100			ns
SCLK Frequency	t_{CLK}	(Note 9)	DC		5.0	MHz
SCLK Rise and Fall	t_R, t_F	(Note 9)		200		ns
\bar{CS} to SCLK Setup	t_{CC}	(Note 9)	400			ns
SCLK to \bar{CS} Hold	t_{CCH}	(Note 9)	100			ns
\bar{CS} Inactive Time	t_{CWH}	(Note 9)	400			ns
\bar{CS} to Output High-Z	t_{CDZ}	(Notes 8, 9)		40		ns
Address 01h or 02h Decoded to DRDY High	t_{DRDYH}	After RTD register read access (Note 9)		50		ns

Hình 84: Tốc độ truyền tải SPI của MAX31865

3.3.3. Chuyển đổi thuật toán PID thành ngôn ngữ C/C++

Từ thuật toán PID ở chương 2, ta dễ dàng chuyển đổi lưu đồ thành code C như sau:

Đặt các biến toàn cục cần thiết:

```
float setpoint = 28.0f; // nhiệt độ tốt - đặt cho cây phát triển.  
//Đồ án sẽ kiểm soát nhiệt độ của nhà kính dao động từ 27,5 đến 28,5.  
float PID_error = 0;  
float previous_error = 0;  
float elapsedTime, Time, timePrev;  
int PID_value = 0;  
//PID constants  
int kp = 2000;  
float ki = 50;  
int kd = 1500;  
float PID_p = 0;  
float PID_i = 0;  
float PID_d = 0;
```

Hình 85: Đặt biến PID cần thiết

“setpoint” là điểm nhiệt độ đặt tham chiếu.

- PID_error là biến lưu trữ sai số giữa nhiệt độ đặt và nhiệt độ thực tế.
- Previous_error, elapsedTime, Time, timePrev là các biến dùng để lưu trữ và tính toán cho PID_d.
- PID_value là biến lưu trữ giá trị PID: $PID_value = PID_p + PID_i + PID_d$.

- K_p, K_i, K_d lần lượt là hệ số tỷ lệ, hệ số tích phân và hệ số đạo hàm của điều khiển PID.

```

unsigned long now = millis();
if (now - time_hoitiem > 1000) [
    PID_error = setpoint - PT100_Temperature;
    if (PID_error > 1.1)
    { PID_i = 0; }

    PID_p = kp * PID_error;           //Calculate the P value
    PID_i = PID_i + (ki * PID_error); //Calculate the I value
    timePrev = Time;                 // the previous time is stored before the actual time read
    Time = millis();                  // actual time read
    elapsedTime = (Time - timePrev) / 1000;
    PID_d = kd * ((PID_error - previous_error) / elapsedTime); //Calculate the D value
    PID_value = PID_p + PID_i + PID_d;           //Calculate total PID value

    //We define firing delay range between 0 and 7000. Read above why 7000!!!!!!
    if (PID_value < 0) { PID_value = 0; }
    if (PID_value > 7000) { PID_value = 7000; }
    previous_error = PID_error;
    Serial.print("PID_Value = ");
    Serial.println(PID_value);
    time_hoitiem = now;
]

```

Hình 86: Thuật toán PID code C/C++

- Biến PID_p là biến tỷ lệ, sai số càng lớn thì PID_p càng lớn và ngược lại.
- Biến PID_i là biến tích phân, biến này sẽ bắt đầu cộng dồn khi sai số bé hơn 1.1 độ C. Vì khi gần đạt ngưỡng nhiệt độ đặt, PID_p sẽ rất nhỏ dẫn đến không đủ nhiệt độ cấp cho buồng, khi này PID_i sẽ được tăng lên và đảm bảo nhiệt độ sẽ vẫn tăng đều, tránh vọt lố.
- PID_d là biến đạo hàm, đây là biến sẽ được so sánh với biến trước đó, nếu nhiệt độ có sự thay đổi lớn một cách đột ngột, PID_d sẽ đóng vai trò đảm bảo công suất đầu ra cho bóng đèn, giúp cho hệ thống không bị tăng hay giảm nhiệt độ đột ngột.
- Các hệ số K_p, K_i, K_d sẽ được chọn thông qua quá trình thử nghiệm thực tế.

3.3.4. Viết thuật toán bơm nước theo phần trăm độ ẩm đất.

Để có thể lập trình xuất xung PWM cho ESP32 bằng Arduino IDE, ta sử dụng hàm “ledcX”.

Đầu tiên ta khai báo biến, chọn PIN trên ESP32 sẽ được xuất xung và chọn channel cho PIN đó từ 0 đến 15:

```
// setting PWM properties
const int suongPin = 13;      // GPIO13 - PWM phun sương
const int freq = 15000;        // Tần số 15KHz
const int suongChannel = 1;    // chọn channel 0
const int resolution = 8;     // độ phân giải analog 8 bit, tối đa là 16 bit
```

Hình 87: Khai báo biến xuất xung PWM

Sau đó, trong void Setup(), ta lập trình như sau:

```
ledcSetup(suongChannel, freq, resolution);
ledcAttachPin(suongPin, suongChannel);
```

Hình 88: Setup PIN PWM

Và đây là thuật toán điều khiển xuất xung PWM:

```
void tuoi_cay()
{
if(millis() - laymau_dat > 3000)
{
    uint32_t phantramdat = convert_SOIL();
    if(phantramdat < 70){
        ledcWrite(suongChannel, 0); // tắt bom trước khi lấy mẫu mỗi 3 giây
        delay(200);
        uint32_t phantramdat = convert_SOIL();
        //uint32_t phantramdat_codinh = phantramdat;
    }
    client.publish("cap/soil", data_soil);
    if(phantramdat_codinh >= 0 && phantramdat_codinh < 30)
    {
        ledcWrite(suongChannel, 255);
        //batbom = 0;
    }
    else if(phantramdat_codinh >= 30 && phantramdat_codinh < 60)
    {
        ledcWrite(suongChannel, 210);
    }
    else if(phantramdat_codinh >= 60 && phantramdat_codinh < 70)
    {
        ledcWrite(suongChannel, 180);
    }
}
```

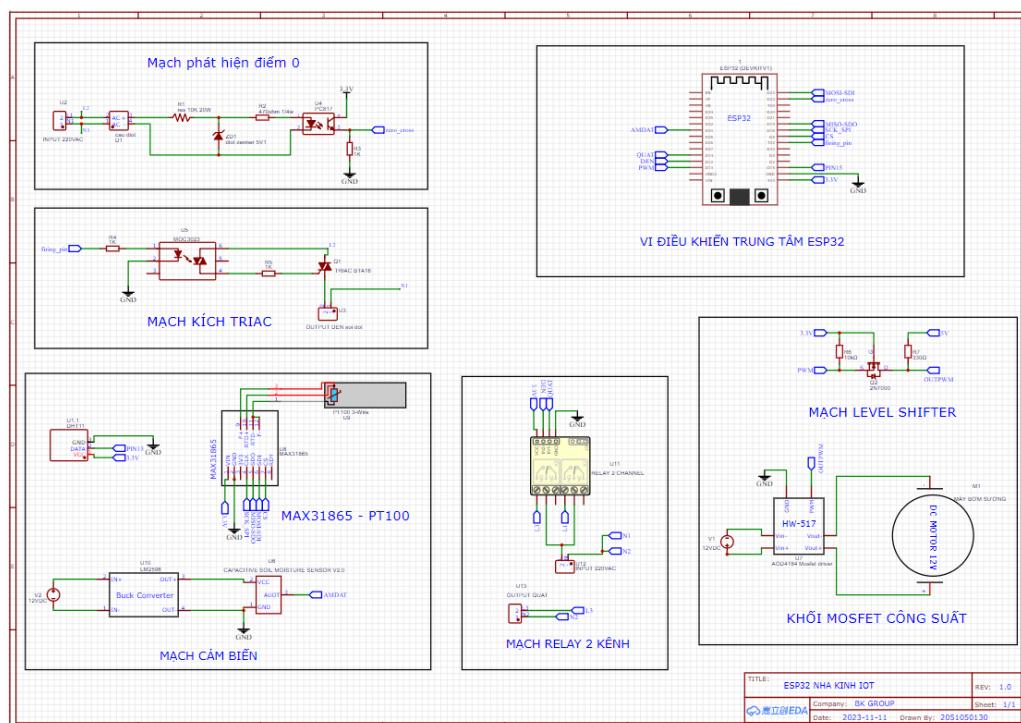
Hình 87: Thuật toán bơm sương PWM

Khi thực hiện tưới cây, ta tắt hết tất cả chế độ tự động điều khiển nhiệt độ khác. Khi độ ẩm đạt được mức mong muốn, ta sẽ bật các chế độ điều khiển nhiệt độ về bình thường.

Một lưu ý khi bơm sương đó là khi bơm bắt đầu chạy, nhiễu điện từ của động cơ sẽ phát ra làm giá trị đo đặc của cảm biến độ ẩm đất bị sai lệch. Chính vì vậy trước khi lấy mẫu độ ẩm đất, ta cần tắt động cơ trong 200ms, khi này chỉ số phần trăm đất sẽ là đúng nhất.

3.4. Sơ đồ nguyên lý toàn bộ hệ thống

Sơ đồ nguyên lý cho toàn bộ hệ thống được nhóm nghiên cứu và vẽ trên phần mềm EASY EDA



Hình 89: Schematic toàn bộ hệ thống

❖ Sơ đồ Schematic gồm 6 khối:

Khối mạch phát hiện điểm 0:

Có nhiệm vụ phát hiện điểm 0 của điện áp xoay chiều 220V-50Hz, đầu ra sẽ là chân ngắt ngoài trên ESP32.

Khi hệ thống phát hiện có điểm 0, ngay lập tức sẽ bật và tắt chân firing pin theo giá trị PID được tính toán.

Khối mạch kích Triac:

Sử dụng Triac BTA16, triac này có thể xuất dòng lên đến 16A và đóng vai trò như công tắc điện tử của điện áp xoay chiều.

Đầu vào của mạch này là chân firing (Pin Firing).

Đầu ra của mạch là bóng đèn già nhiệt.

Khối cảm biến:

Sử dụng cảm biến PT100 và MAX31865 giao tiếp với ESP32 qua giao thức SPI để đọc nhiệt độ.

Sử dụng cảm biến DHT11 để đọc nhiệt độ từ môi trường thông qua giao thức 1-Wire.

Sử dụng cảm biến Capacitive Soil để đo độ ẩm đất và trả về tín hiệu Analog, từ đó người dùng có thể nghiên cứu đo đạc theo các tiêu chuẩn để viết thư viện cho cảm biến.

Khối Mạch relay 2 kênh:

Mạch relay điều khiển bật/tắt công tắc cho mạch kiểm soát nhiệt độ - bật tắt nguồn cho mạch kích triac và mạch phát hiện điểm 0. Điều này là cần thiết vì khi nhiệt độ ngoài môi trường bằng hoặc lớn hơn nhiệt độ đặt (nhiệt độ cho cây húng quế ra lá) thì ta cần tắt nguồn cấp vào 2 mạch này để tránh lãng phí điện tiêu hao trên điện trở 20W và các linh kiện khác.

Mạch relay điều khiển bật/tắt trực tiếp cho động cơ quạt 220VAC.

Khối mosfet công suất:

Mạch level shifter có nhiệm vụ chuyển mức điện áp 0 – 3.3v thành 0 -5V và cấp vào mạch mosfet công suất.

Mạch mosfet công suất nhận đầu vào là đầu ra của mạch level shifter, từ đó xuất ra xung PWM 0-12V cho động cơ bơm phun sương.

3.5. Tiến hành kết nối các mạch điện với mô hình nhà kính.

Nhóm nghiên cứu xây dựng mô hình trồng cây trong nhà kính có kích thước và nguồn cấp như hình sau:



Hình 90: Tổng quan phần cứng

Kích thước mô hình nhà kính là 40x60x70cm (RxDxC).

Kích thước tủ điện là 16x21x9.

Nguồn cấp 220VAC - 5VDC cấp cho ESP32.

Nguồn cấp 220VAC – 12VDC cấp cho mạch mosfet

Nguồn cấp 220VAC cấp cho quạt và bóng đèn thông qua relay là mạch kích triac.

Nguồn cấp 12VDC – 3.3VDC cấp riêng cho cảm biến Capacitive Soil (tránh nhiễu và sụt áp).



Hình 91: Đầu ra tủ điện

Tủ điện có các đầu ra như sau:

- Đầu ra cảm biến độ ẩm đất Cap Soil có chiều dài được kết nối đến 1m.
- Đầu dò cảm biến PT100 có chiều dài là 50cm.
- Đầu ra đèn, quạt và bơm sẽ được kết nối tương ứng với đầu ra của mô hình nhà kính.

3.6. Giám sát – cài đặt - điều khiển trên Node-RED Dashboard

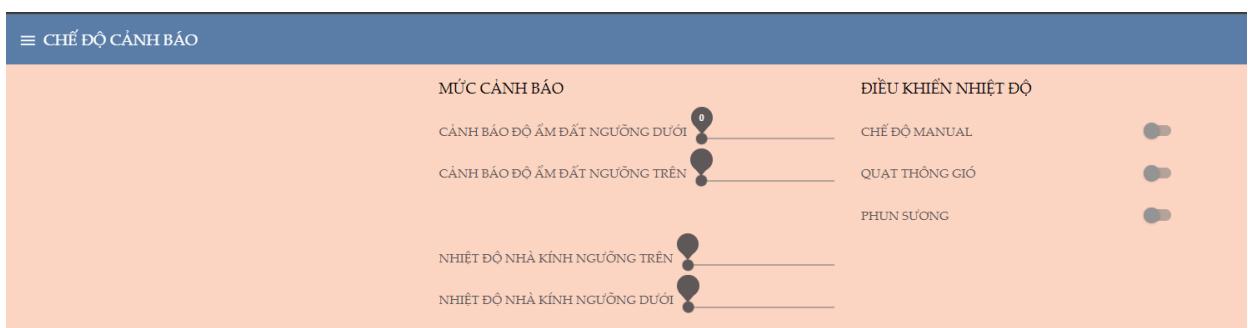
Hệ thống được giám sát, cài đặt và điều khiển thông qua 3 tab trên Dashboard node-red.

Ở tab “ GIÁM SÁT THÔNG SỐ” nhóm em hiển thị giá trị dạng text cho 3 cảm biến. Biểu đồ Gause cho độ ẩm đất, vào biểu đồ line cho nhiệt độ bên trong và bên ngoài nhà kính.



Hình 92: Tab giám sát thông số

Ở tab “CHẾ ĐỘ CẢNH BÁO” nhóm em có thể cài đặt ngưỡng cảnh báo cho 2 giá trị độ ẩm đất và nhiệt độ không khí.



Hình 93: Tab chế độ cảnh báo

Ngoài ra, có thể chọn chế độ manual hoặc auto cho hệ thống và bật tắt cơ cấu chấp hành bằng tay.

Ở Tab “BẢNG SỐ LIỆU”, nhóm em sẽ gửi dữ liệu thu thập từ 3 cảm biến mỗi 10 giây 1 lần lên tab này. Dữ liệu sẽ được thu thập real time từ lúc hệ thống khởi động cho đến khi hệ thống bị tắt.

BẢNG 1		BẢNG 2		BẢNG 3	
Ngày/Giờ	Nhiệt độ Môi Trường	Ngày/Giờ	Độ Ẩm Đất	Ngày/Giờ	Nhiệt độ Nhà Kính
11/3/2023, 5:59:56 PM		11/3/2023, 5:59:56 PM		11/3/2023, 5:59:56 PM	23.85
11/3/2023, 5:59:56 PM		11/3/2023, 5:59:56 PM	0	11/3/2023, 5:59:56 PM	
11/3/2023, 5:59:56 PM	25.8	11/3/2023, 5:59:56 PM		11/3/2023, 5:59:56 PM	
11/3/2023, 5:59:46 PM		11/3/2023, 5:59:46 PM		11/3/2023, 5:59:46 PM	23.85
11/3/2023, 5:59:46 PM		11/3/2023, 5:59:46 PM	0	11/3/2023, 5:59:46 PM	
11/3/2023, 5:59:46 PM	25.8	11/3/2023, 5:59:46 PM		11/3/2023, 5:59:46 PM	
11/3/2023, 5:59:35 PM		11/3/2023, 5:59:35 PM		11/3/2023, 5:59:35 PM	23.79
11/3/2023, 5:59:35 PM		11/3/2023, 5:59:35 PM	0	11/3/2023, 5:59:35 PM	
11/3/2023, 5:59:35 PM	25.8	11/3/2023, 5:59:35 PM		11/3/2023, 5:59:35 PM	
11/3/2023, 5:59:26 PM		11/3/2023, 5:59:26 PM		11/3/2023, 5:59:26 PM	23.79
11/3/2023, 5:59:26 PM		11/3/2023, 5:59:26 PM	0	11/3/2023, 5:59:26 PM	
11/3/2023, 5:59:26 PM	25.3	11/3/2023, 5:59:26 PM		11/3/2023, 5:59:26 PM	
11/3/2023, 5:59:18 PM		11/3/2023, 5:59:18 PM		11/3/2023, 5:59:18 PM	23.75
11/3/2023, 5:59:18 PM		11/3/2023, 5:59:18 PM	0	11/3/2023, 5:59:18 PM	
11/3/2023, 5:59:18 PM	25.3	11/3/2023, 5:59:18 PM		11/3/2023, 5:59:18 PM	
11/3/2023, 5:59:05 PM		11/3/2023, 5:59:05 PM		11/3/2023, 5:59:05 PM	23.85
11/3/2023, 5:59:05 PM		11/3/2023, 5:59:05 PM	0	11/3/2023, 5:59:05 PM	
11/3/2023, 5:59:01 PM		11/3/2023, 5:59:01 PM		11/3/2023, 5:59:01 PM	23.85
11/3/2023, 5:59:01 PM		11/3/2023, 5:59:01 PM	0	11/3/2023, 5:59:01 PM	
11/3/2023, 5:59:01 PM	24.8	11/3/2023, 5:59:01 PM		11/3/2023, 5:59:01 PM	
11/3/2023, 5:58:47 PM		11/3/2023, 5:58:47 PM		11/3/2023, 5:58:47 PM	23.72
11/3/2023, 5:58:47 PM		11/3/2023, 5:58:47 PM	0	11/3/2023, 5:58:47 PM	
11/3/2023, 5:58:47 PM	24.8	11/3/2023, 5:58:47 PM		11/3/2023, 5:58:47 PM	
11/3/2023, 5:58:35 PM		11/3/2023, 5:58:35 PM		11/3/2023, 5:58:35 PM	23.82
11/3/2023, 5:58:35 PM		11/3/2023, 5:58:35 PM	0	11/3/2023, 5:58:35 PM	
11/3/2023, 5:58:35 PM	24.8	11/3/2023, 5:58:35 PM		11/3/2023, 5:58:35 PM	

Hình 94: Tab bảng số liệu

Kết luận cho hệ thống giám sát- cài đặt – điều khiển từ xa: Như vậy, nhóm em đã thành công trong việc gửi/nhận dữ liệu bằng Internet thông qua giao thức MQTT. Song song với đó là cài đặt mức cảnh báo, cài đặt chế độ manual/auto và cập nhật dữ liệu dưới dạng bảng.

Về chi tiết quá trình kiểm thử toàn bộ hệ thống, nhóm em đã thực hiện quay video để báo cáo tất cả các trường hợp có thể xảy ra trong quá trình trồng cây trong nhà kính, từ đó đưa ra kết luận được liệu mô hình của nhóm có đáp ứng được mục tiêu ban đầu đề ra hay không.

3.7. Kết luận và hướng phát triển

Sau hai tháng lập trình, thiết kế và xây dựng hệ thống – nhóm em có thể đưa ra kết luận sau:

Những thành công trong đề tài:

+ Nhóm đã thành công trong việc kiểm soát nhiệt độ nhà kính bám theo xu hướng sống của cây húng quế trong điều khiển trời nóng - trời mát và trời lạnh, từ 27 đến 28 độ C.

+ Thuật toán PID nhiệt độ - nâng nhiệt độ nhà kính khi trời lạnh đã đạt được những thành công nhất định, cụ thể là sai số xác lập bé hơn 0.07 độ C, độ vọt lốp bé hơn 0.3%, thời gian xác lập tùy thuộc vào nhiệt độ môi trường (từ 1 – 2 phút khi nhiệt độ mong muốn và nhiệt độ thực tế chênh nhau ít hơn 2 độ C).

- + Hệ thống đã có thể đăng nhập để giám sát – cài đặt – điều khiển cho trong điều kiện dùng chung 1 mạng LAN hoặc khác mạng đều được.
- + Hệ thống có sự ưu tiên tưới cây khi độ ẩm đất dưới mức sống của cây.
- + Hệ thống bơm sương cho cây là được kiểm soát thành công với thuật toán xuất xung PWM.
- + Bảo vệ các linh kiện điện tử trong hệ thống khỏi sự ảnh hưởng của nước và ánh sáng mặt trời.

Những hạn chế cần khắc phục:

- + Việc phun sương khi hạ nhiệt độ có thể sẽ tăng độ ẩm đất. Mặc dù nhóm đã có phun sương ít kết hợp quạt có kiểm soát, nhưng việc tăng độ ẩm đất khi hạ nhiệt độ là điều khó tránh khỏi.
- + Hệ thống vẫn chưa có giải pháp thoát nước. Trong quá trình test hệ thống, một điểm yếu lớn nhất của hệ thống là chưa tìm được cách để thoát nước cho nhà kính.
- + Hệ thống vẫn chưa thể xuất file ở Tab 3 (Tab bảng số liệu) thành file để in ấn.
- + Hệ thống vẫn chưa thể lưu các cảnh báo.

Những kiến thức nhóm nghiên cứu đã gặt hái:

- + Trong hơn 2 tháng nghiên cứu và xây dựng hệ thống, nhóm em đã học được nhiều kiến thức mới cũng như hiểu sâu về những kiến thức cũ.
- + Nhờ sự hướng dẫn của giảng viên, nhóm đã biết cách để đo đặc – kiểm chứng thông số chính xác cho dữ liệu cảm biến.
- + Các thành viên trong nhóm đều được giao những nhiệm vụ và những trách nhiệm cụ thể, điều này giúp cho kỹ năng làm việc nhóm của các thành viên được nâng cao.
- + Nhóm đã hiểu hơn về các phương thức thiết kế một hệ thống IoT cũng như các giao thức không dây MQTT, giao thức có dây SPI.

Hướng phát triển cho hệ thống:

- + Sử dụng cảm biến đầu dò chống nước như SHTCX thay cho DHT11.
- + Thiết kế thoát nước cho mô hình nhà kính.
- + Thiết kế mạch và hàn mạch PCB cho hệ thống.
- + Bổ sung khả năng in ấn và lưu cảnh báo cho hệ thống.

TÀI LIỆU THAM KHẢO

- Sách tham khảo:

- [1] MAX31865, Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc.
- [2] ESP32 Datasheet (PDF) - ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD
- [3] TIÊU CHUẨN QUỐC GIA TCVN 4196:2012, ĐÁT XÂY DỰNG - PHƯƠNG PHÁP XÁC ĐỊNH - ĐỘ ÂM VÀ ĐỘ HÚT ÂM TRONG PHÒNG THÍ NGHIỆM, 2012.

- Trang thông tin điện tử tham khảo:

- [4] Electronoobs (2022), “220V AC HEATER PID TEMPERATURE CONTROL”, ELECTRONOOBS BLOGS,
https://electronoobs.com/eng_arduino_tut39.php, truy cập ngày 31/05/2023.
- [5] Nguyen Van Dong Hai (2021), “Dieu khien nhiet do”, Kênh Youtube Nguyen van dong hai,
https://www.youtube.com/playlist?list=PL0TKVdGkFUunBxIk9EgpGK_XEc_b0fNYM, truy cập ngày 28/05/2023.