ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

--------------------------------

A map of the world

Description automatically generated with low confidence

Môn học: Hệ thống nhúng mạng không dây

NT131.M21

**BÁO CÁO ĐỒ ÁN**

**Đề Tài: Hệ thống tưới cây tự động**

SV thực hiện: Nhóm 04

Nguyễn Quang Huy 19520600

Cao Hoàng Tú 19521067

Trương Hồng Phương Anh 19521224

GV hướng dẫn: Th.S Nguyễn Khánh Thuật

*Thành phố Hồ Chí Minh – Năm 2022*

**NHẬN XÉT CỦA GIÁO VIÊN**

**LỜI CÁM ƠN**

Để hoàn thành đồ án này, nhóm em xin gởi đến thầy Nguyễn Khánh Thuật – người đã tận tình hướng dẫn, cung cấp đầy đủ các kiến thức cần thiết để nhóm có thể hiện thực hóa đồ án này lời cảm ơn sâu sắc nhất.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực tập, hoàn thiện chuyên đề này, nhóm chúng em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ thầy.

Chúng em xin chân thành cảm ơn.

*TP.HCM, ngày 22 tháng 05 năm 2022*

*Nguyễn Quang Huy*

*Cao Hoàng Tú*

*Trương Hồng Phương Anh*

**MỤC LỤC**

[**DANH MỤC HÌNH ẢNH** 6](#_Toc104130068)

[**DANH MỤC BẢNG** 6](#_Toc104130069)

[**I.** **Tổng quan đề tài** 7](#_Toc104130070)

[**I.1.** **Lí do chọn đề tài:** 7](#_Toc104130071)

[**I.2.** **Mục tiêu đề tài:** 7](#_Toc104130072)

[**I.3.** **Đối tượng:** 7](#_Toc104130073)

[**I.4.** **Phạm vi đề tài:** 7](#_Toc104130074)

[**II.** **Cơ sở lý thuyết** 8](#_Toc104130075)

[**II.1.** **Phần cứng** 8](#_Toc104130076)

[**II.1.1.** **NodeMCU ESP8266** 8](#_Toc104130077)

[**II.1.2.** **DHT11 – cảm biến nhiệt độ và độ ẩm không khí** 10](#_Toc104130078)

[**II.1.3.** **Cảm biến độ ẩm đất (Soil Moisture sensor)** 12](#_Toc104130079)

[**II.2.** **Phần mềm** 14](#_Toc104130080)

[**II.2.1.** **MQTT Mosquitto Broker** 14](#_Toc104130081)

[**II.2.2.** **Node-RED** 15](#_Toc104130082)

[**II.2.3.** **Arduino IDE** 16](#_Toc104130083)

[**II.2.4.** **Thư viện Pubsubclient** 19](#_Toc104130084)

[**II.3.** **Dữ liệu cây trồng** 20](#_Toc104130085)

[**II.3.1.** **Cây dâu tây** 20](#_Toc104130086)

[**II.3.2.** **Cây ớt** 20](#_Toc104130087)

[**II.3.3.** **Cây đậu cô-ve** 20](#_Toc104130088)

[**III.** **Thiết kế hệ thống** 21](#_Toc104130089)

[**III.1.** **Nguyên lý hoạt động** 21](#_Toc104130090)

[**III.1.1.** **Thành phần của hệ thống:** 21](#_Toc104130091)

[**III.1.2.** **Nguyên lý hoạt động:** 21](#_Toc104130092)

[**III.2.** **Sơ đồ nối dây** 21](#_Toc104130093)

[**IV.** **Hiện thực đề tài** 22](#_Toc104130094)

[**IV.1.** **Giao diện web UI** 22](#_Toc104130095)

[**IV.2.** **Quá trình thực hiện** 22](#_Toc104130096)

[**IV.2.1.** **Lập trình hệ thống** 22](#_Toc104130097)

[**IV.2.2.** **Cài đặt và cấu hình phần mềm** 23](#_Toc104130098)

[**V.** **Thực nghiệm đề tài** 27](#_Toc104130099)

[**V.1.** **Mô hình thực tế** 27](#_Toc104130100)

[**V.2.** **Hiển thị dữ liệu đo được lên webUI của Node-RED** 28](#_Toc104130101)

[**VI.** **Kết luận** 29](#_Toc104130102)

[**VI.1.** **Kết quả đạt được** 29](#_Toc104130103)

[**VI.2.** **Những khó khăn khi thực hiện** 29](#_Toc104130104)

[**VI.3.** **Khuyết điểm** 29](#_Toc104130105)

[**VI.4.** **Hướng phát triển** 29](#_Toc104130106)

[**TÀI LIỆU THAM KHẢO** 30](#_Toc104130107)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Các phiên bản khác nhau của ESP8266 8](#_Toc104128406)

[Hình 2. NodeMCU ESP8266 8](#_Toc104128407)

[Hình 3. Sơ đồ chân NodeMCU ESP8266 9](#_Toc104128408)

[Hình 4. Mô-đun DHT11 10](#_Toc104128409)

[Hình 5. Cảm biến DHT11 10](#_Toc104128410)

[Hình 6. Sơ đồ chân cảm biến DHT11 11](#_Toc104128411)

[Hình 7. Sơ đồ chân mô-đun DHT11 11](#_Toc104128412)

[Hình 8. Sơ đồ chân mô-đun DHT11 và cảm biến DHT11 11](#_Toc104128413)

[Hình 9. Sơ đồ kết nối mô-đun DHT11 với Arduino UNO 12](#_Toc104128414)

[Hình 10. Cảm biến độ ẩm đất 13](#_Toc104128415)

[Hình 11. Sơ đồ chân cảm biến độ ẩm đất 13](#_Toc104128416)

[Hình 12. Sơ đồ kết nối cảm biến độ ẩm đất với Arduino 13](#_Toc104128417)

[Hình 13. Mô hình hoạt động của MQTT Broker 14](#_Toc104128418)

[Hình 14. Giao diện Dashboard của Node-RED 15](#_Toc104128419)

[Hình 15. Arduino 16](#_Toc104128420)

[Hình 16. Giới thiệu Adruino IDE 17](#_Toc104128421)

[Hình 17. Thư viện của Adruino IDE 18](#_Toc104128422)

[Hình 18. Giao diện sử dụng Adruino IDE 18](#_Toc104128423)

[Hình 19. Giao diện webUI của hệ thống 22](#_Toc104128424)

[Hình 20. Giao diện bắt đầu của Node-RED 23](#_Toc104128425)

[Hình 21. Flow tổng hợp của hệ thống 24](#_Toc104128426)

[Hình 22. Khối gửi dữ liệu từ webUI đến MQTT Broker 24](#_Toc104128427)

[Hình 23. Các nút gửi thông tin loại cây trồng đến ESP8266 24](#_Toc104128428)

[Hình 24. Các nút gửi hành động “Tưới” hoặc “Ngừng tưới” đến ESP8266 24](#_Toc104128429)

[Hình 25. Khối phục vụ debugging 25](#_Toc104128430)

[Hình 26. Khối nhận dữ liệu từ MQTT Broker 25](#_Toc104128431)

[Hình 27. Cấu trúc subflow trong hệ thống 25](#_Toc104128432)

[Hình 28. Khối function 25](#_Toc104128433)

[Hình 29. Khối chuyển định dạng payload 26](#_Toc104128434)

[Hình 30. Khối phục vụ debugging 26](#_Toc104128435)

[Hình 31. Khối hiển thị 26](#_Toc104128436)

[Hình 32. Mô hình thực tế 1 27](#_Toc104128437)

[Hình 33. Mô hình thực tế 2 28](#_Toc104128438)

[Hình 34. Thực nghiệm - Hiển thị dữ liệu đo được lên webUI 28](#_Toc104128439)

**DANH MỤC BẢNG**

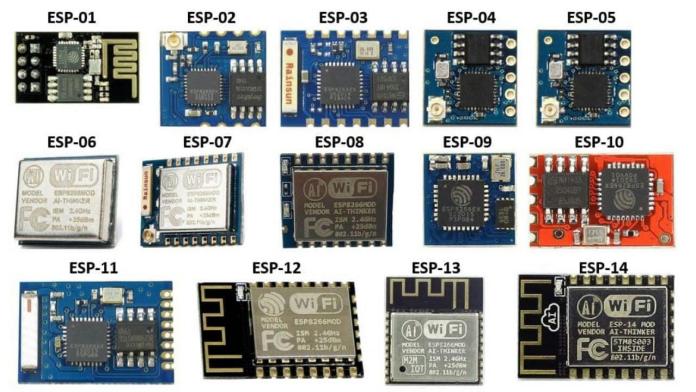
[Bảng 1. Sơ đồ nối dây của hệ thống 21](#_Toc104108735)

1. **Tổng quan đề tài**
   1. **Lí do chọn đề tài:**

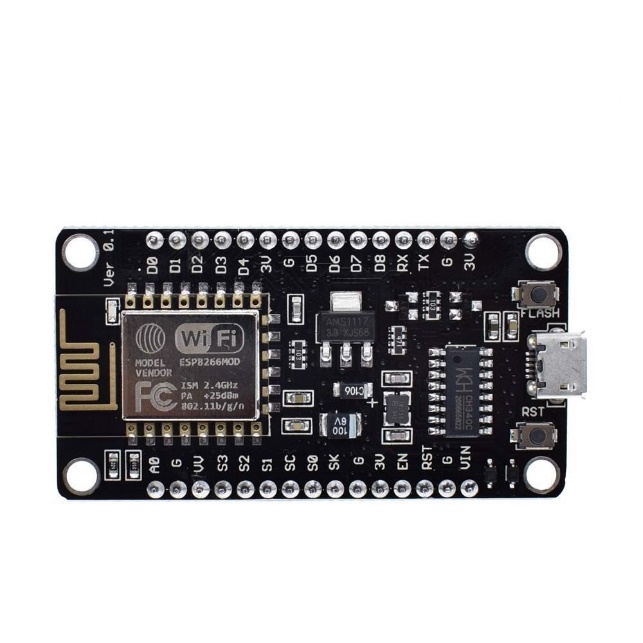
* Nước là một trong những yếu tố quan trọng nhất đối với cây trồng, việc cung cấp đủ nước cho cây là việc vô cùng cần thiết. Tuy nhiên, giữa cuộc sống bận rộn hiện nay, có nhiều người tuy rất thích trồng cây, chăm vườn nhưng thời gian có hạn khiến họ không thể có một khu vườn như ý.
* Chính vì vậy mà hệ thống tưới cây tự động ra đời nhằm đáp ứng nhu cầu về chăm sóc cây cối một cách hoàn toàn tự động, giữ cho khu vườn luôn đủ lượng nước cần thiết.
* Nhóm em chọn đề tài này với tôn chỉ giúp người dùng tiết kiệm thời gian nhất có thể nhưng vẫn có thể chăm sóc khu vườn của mình một cách hiệu quả.
  1. **Mục tiêu đề tài:**
* Xây dựng một hệ thống vườn thông minh có khả năng tự động tưới cây dựa trên các chỉ số môi trường.
* Có web UI để người dùng dễ dàng quản lý, giám sát khu vườn của mình.
  1. **Đối tượng:**
* Cá nhân, hộ gia đình, công ty muốn chăm sóc khu vườn, trồng cây hiệu quả mà tiết kiệm thời gian, không phải bỏ quá nhiều công sức.
  1. **Phạm vi đề tài:**
* Hệ thống vườn thông minh được xây dựng trên ESP8266, có khả năng tưới cây tự động và web UI hiển thị thông số môi trường.

1. **Cơ sở lý thuyết**
   1. **Phần cứng**
      1. **NodeMCU ESP8266**
         1. **Khái niệm và cấu tạo cơ bản** [1][2]

* ESP8266 là một chip được sản xuất bởi Espressif Systems có tích hợp công nghệ Wi-Fi đáp ứng nhu cầu của các ứng dụng IoT (Internet of Things) như chi phí thấp, sử dụng năng lượng hiệu quả, hiệu suất đáng tin cậy và thiết kế nhỏ gọn, tương thích với nhiều nền tảng. Nó thường được sử dụng trong các board mạch smart home giá rẻ như Sonoff Basic và các công tắc Wi-Fi. Module sơ khai nhất của ESP8266 là ESP-01.
* Có khả năng kết nối mạng WiFi hoàn chỉnh, ESP8266 có thể hoạt động như một thiết bị slave cho một bộ vi điều khiển chủ hoặc như một ứng dụng độc lập. Khi nói thiết bị tớ cho một vi điều khiển chủ, điều đó có nghĩa là nó có thể sử dụng làm bộ điều hợp WiFi cho bất kỳ vi điều khiển nào sử dụng giao tiếp SPI hoặc UART. Trong khi sử dụng độc lập, nó có thể thực hiện được các chức năng của một bộ vi điều khiển và mạng WiFi.
* ESP8266 dựa trên dòng L106 Diamond của Tensilica, là bộ xử lý 32-bit và có SRAM trên chip. Đồng thời tích hợp module nguồn, balun RF, bộ thu và phát RF, bộ thu và phát tương tự, băng tần số, bộ khuếch đại, bộ lọc và một số thành phần tối thiểu khác.
* ESP8266 có nhiều phiên bản khác nhau như hình dưới. Hiện nay, NodeMCU ESP-12E là phiên bản phổ biến nhất vì nó dễ kết nối, lập trình và có nhiều chân GPIO

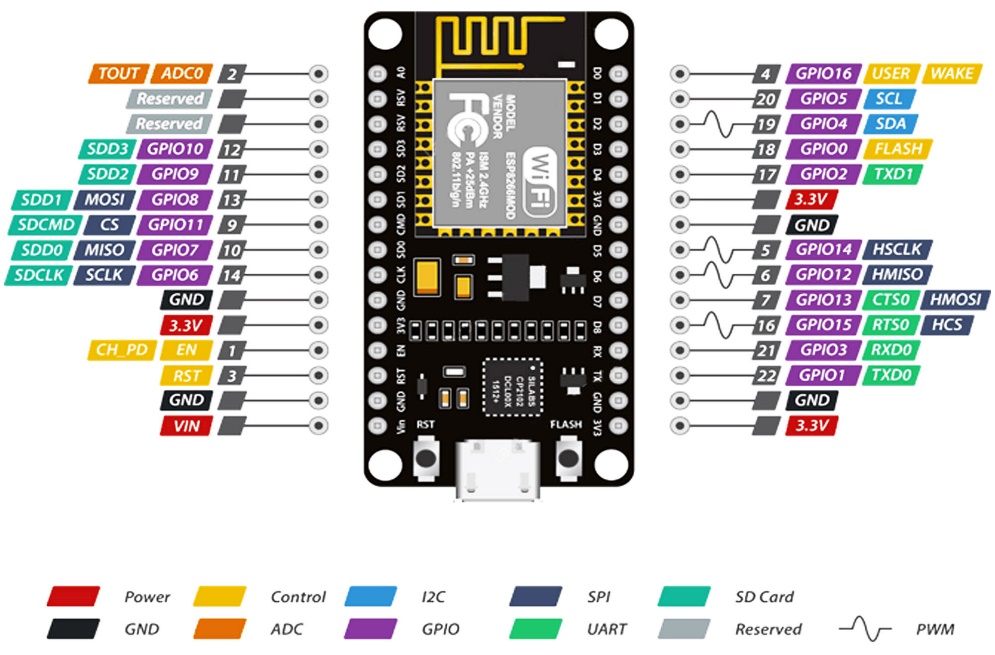


Hình 1. Các phiên bản khác nhau của ESP8266



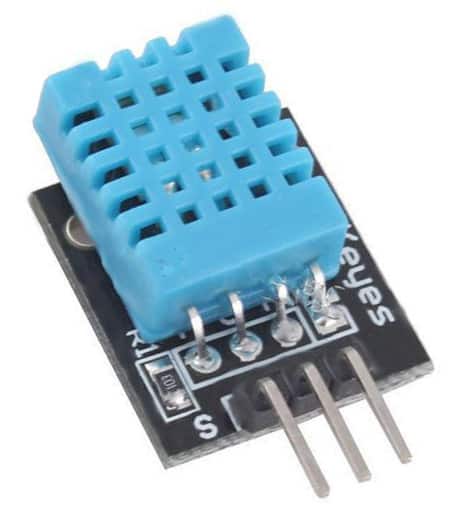
Hình 2. NodeMCU ESP8266

* + - 1. **Thông số kỹ thuật và sơ đồ chân** [1][2]
* Các thông số kỹ thuật chung của ESP8266:
  + Tần số hoạt động: 80 MHz
  + Bộ nhớ flash: 1MB
  + 17 GPIO pins
  + Giao tiếp: I2C, SPI, I²S, UART
  + Đặc biệt: 10bit ADC
* Các thông số kỹ thuật của NodeMCU ESP8266
  + WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
  + Điện áp hoạt động: 3.3V
  + Điện áp vào: 5V thông qua cổng USB
  + Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
  + Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
  + Bộ nhớ Flash: 4MB
  + Giao tiếp: Cable Micro USB ( tương đương cáp sạc điện thoại )
  + Hỗ trợ bảo mật: WPA/WPA2
  + Tích hợp giao thức TCP/IP
  + Lập trình trên các ngôn ngữ: C/C++, Micropython, Lua,…
* Sơ đồ chân của NodeMCU ESP8266

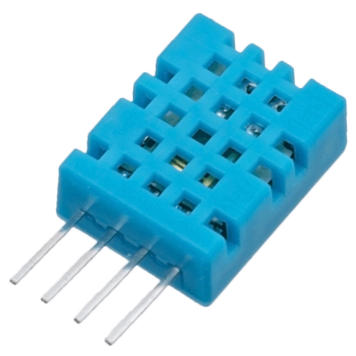


Hình 3. Sơ đồ chân NodeMCU ESP8266

* Một số ứng dụng cơ bản khi sử dụng ESP8266 trên Ubuntu
  + Điều khiển công tắc bật/tắt Led bằng openHAB
  + Đọc nhiệt độ trên cảm biến DHT11 bằng openHAB
  + Điều khiển bật/tắt Led bằng giọng nói sử dụng ứng dụng openHAB
    1. **DHT11 – cảm biến nhiệt độ và độ ẩm không khí**
       1. **Khái niệm** [3][4]
* DHT11 là một cảm biến kỹ thuật số giá rẻ để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kỳ bộ vi điều khiển vi nào như Arduino, Raspberry Pi, ... để đo độ ẩm và nhiệt độ ngay lập tức.
* DHT11 là một cảm biến độ ẩm tương đối. Để đo không khí xung quanh, cảm biến này sử dụng một điện trở nhiệt và một cảm biến độ ẩm điện dung.

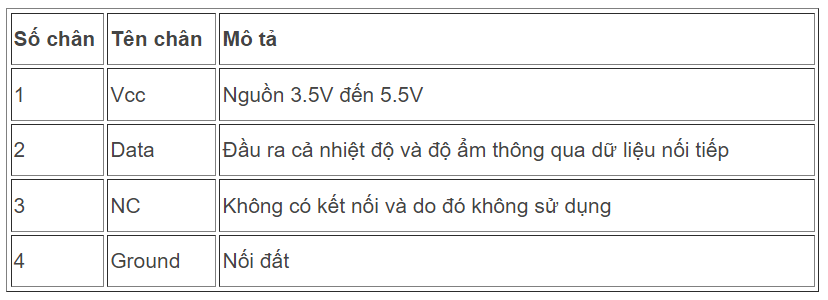


Hình 4. Mô-đun DHT11



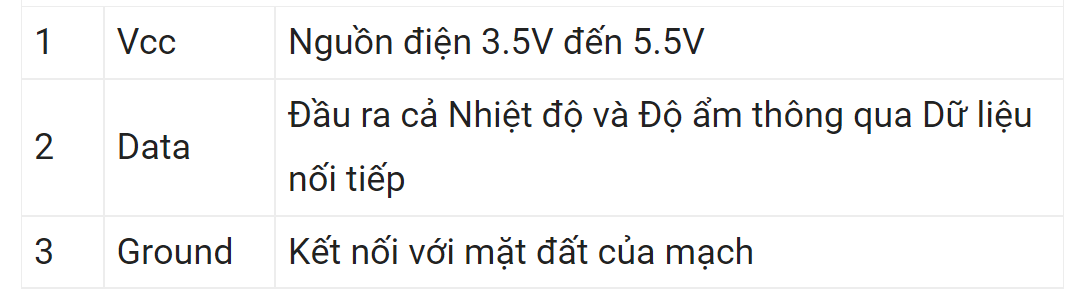
Hình 5. Cảm biến DHT11

* + - 1. **Cấu tạo cơ bản của bộ DHT11** [4]
* Cảm biến DHT11 bao gồm một phần tử cảm biến độ ẩm điện dung và một điện trở nhiệt để cảm nhận nhiệt độ. Tụ điện cảm biến độ ẩm có hai điện cực với chất nền giữ ẩm làm chất điện môi giữa chúng. Thay đổi giá trị điện dung xảy ra với sự thay đổi của các mức độ ẩm. IC đo, xử lý các giá trị điện trở đã thay đổi này và chuyển chúng thành dạng kỹ thuật số.
* Để đo nhiệt độ, cảm biến này sử dụng một nhiệt điện trở có hệ số nhiệt độ âm, làm giảm giá trị điện trở của nó khi nhiệt độ tăng. Để có được giá trị điện trở lớn hơn ngay cả đối với sự thay đổi nhỏ nhất của nhiệt độ, cảm biến này thường được làm bằng gốm bán dẫn hoặc polymer.
  + - 1. **Sơ đồ chân - Thông số kỹ thuật – Sơ đồ kết nối:** [3][4]
* Đối với cảm biến DHT11:

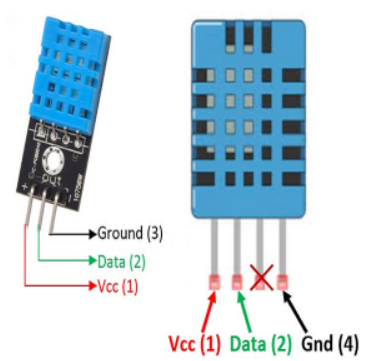


Hình 6. Sơ đồ chân cảm biến DHT11

* Đối với mô-đun DHT11:

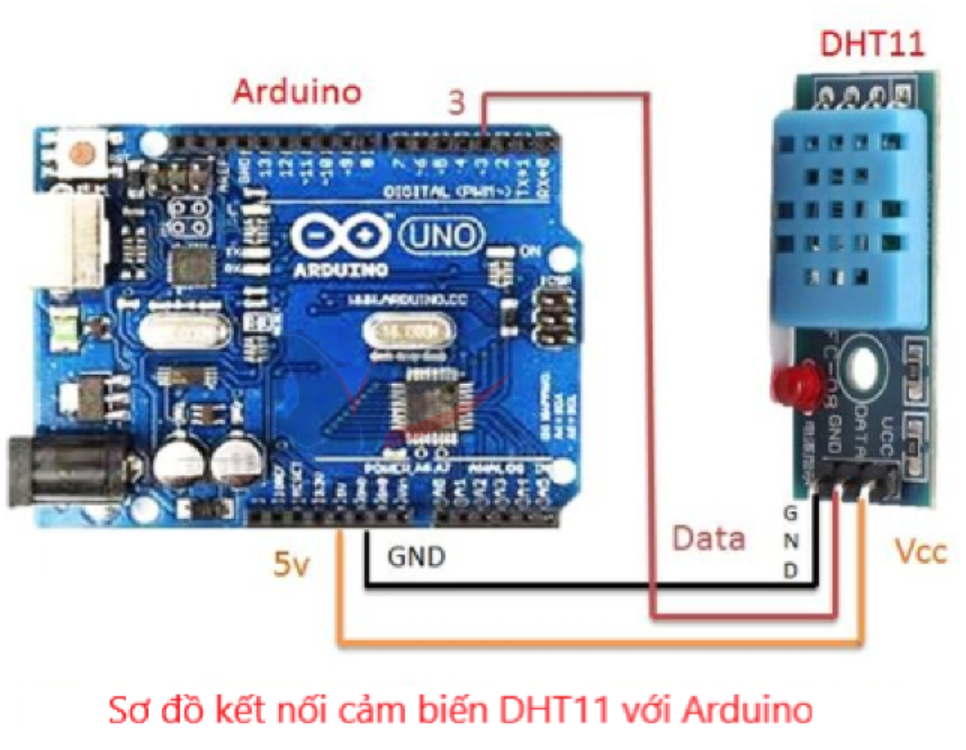


Hình 7. Sơ đồ chân mô-đun DHT11



Hình 8. Sơ đồ chân mô-đun DHT11 và cảm biến DHT11

* Thông số kỹ thuật:
  + Điện áp hoạt động: 3V - 5V DC
  + Dòng điện tiêu thụ: 2.5mA
  + Phạm vi cảm biến độ ẩm: 20% - 90% RH, sai số ±5%RH
  + Phạm vi cảm biến nhiệt độ: 0°C ~ 50°C, sai số ±2°C
  + Tần số lấy mẫu tối đa: 1Hz (1 giây 1 lần)
  + Kích thước: 23 \* 12 \* 5 mm
  + Độ phân giải: Nhiệt độ và Độ ẩm đều
* Sơ đồ kết nối



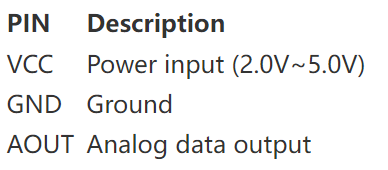
Hình 9. Sơ đồ kết nối mô-đun DHT11 với Arduino UNO

* + - 1. **Nên sử dụng DHT11 ở đâu và một số loại cảm biến nhiệt độ:** [3][4]
* Sử dụng DHT11 ở đâu:
  + DHT11 là một cảm biến nhiệt độ và độ ẩm thường được sử dụng. Cảm biến đi kèm với một NTC chuyên dụng để đo nhiệt độ và một bộ vi điều khiển 8 bit để xuất ra các giá trị nhiệt độ và độ ẩm dưới dạng dữ liệu nối tiếp. Cảm biến cũng được hiệu chuẩn tại nhà máy và do đó dễ dàng giao tiếp với các bộ vi điều khiển khác.
  + Cảm biến có thể đo nhiệt độ từ 0 ° C đến 50 ° C và độ ẩm từ 20% đến 90% với độ chính xác ± 1 ° C và ± 1%. Vì vậy, nếu bạn đang muốn đo trong phạm vi này thì cảm biến này có thể là lựa chọn phù hợp cho bạn.
* Một số loại cảm biến nhiệt độ:
  + Cảm biến nhiệt độ tương đương DHT11: DHT22, AM2302, SHT71
  + Cảm biến nhiệt độ khác: Cặp nhiệt điện, TMP100, LM75, DS18820, SHT15, LM35DZ, TPA81, D6T
    1. **Cảm biến độ ẩm đất (Soil Moisture sensor)** [5][6][7]
       1. **Tổng quát**
* Khái niệm:
  + Cảm biến độ ẩm đất Soil Moisture Sensor thường được sử dụng trong các mô hình tưới nước tự động, vườn thông minh,...cảm biến giúp xác định độ ẩm của đất qua đầu dò và trả về giá trị Analog để giao tiếp với vi điều khiển để thực hiện vô số các ứng dụng khác nhau.
* Thiết kế:
  + Có thiết kế giống như cái dĩa, giúp dễ dàng cắm vào đất
  + Điện áp đầu ra tăng cùng với độ ẩm của đất tăng lên
* Thông số kỹ thuật:
  + Độ sâu phát hiện: 38mm
  + Nguồn: 2.0V ~ 5.0V
  + Kích thước: 20,0mm \* 51,0mm
  + Kích thước lỗ lắp: 2.0mm



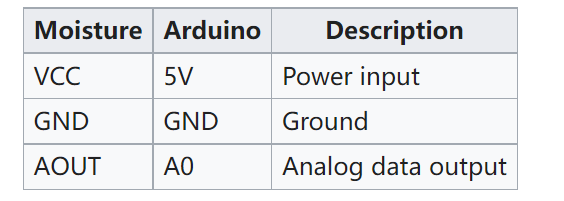
Hình 10. Cảm biến độ ẩm đất

* Sơ đồ chân:



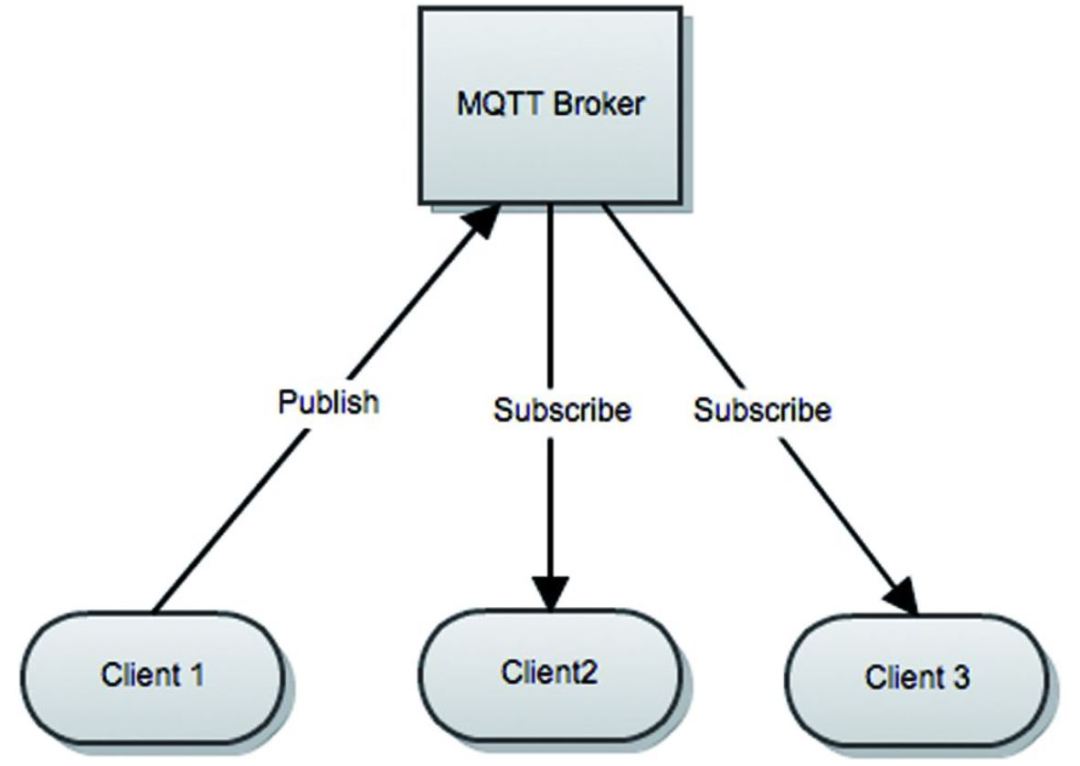
Hình 11. Sơ đồ chân cảm biến độ ẩm đất

* + - 1. **Nguyên lý hoạt động và các ứng dụng**
* Nguyên lý làm việc:
  + Đầu dò hình nĩa với hai dây dẫn để hở, hoạt động như một biến trở (giống như một chiết áp) có điện trở thay đổi tùy theo hàm lượng nước trong đất.
  + Càng nhiều nước trong đất có nghĩa là độ dẫn điện tốt hơn và sẽ dẫn đến điện trở thấp hơn.
  + Càng ít nước trong đất có nghĩa là độ dẫn điện kém và sẽ dẫn đến điện trở cao hơn.
  + Cảm biến tạo ra điện áp đầu ra theo điện trở, bằng cách đo chúng ta có thể xác định mức độ ẩm.
* Kết nối phần cứng với Adruino:



Hình 12. Sơ đồ kết nối cảm biến độ ẩm đất với Arduino

* Các ứng dụng:
  + Hệ thống tưới tự động
  + Phát hiện độ ẩm đất lọ hoa
  + Hệ thống rót tự động
* Cách sử dụng Trong trường hợp làm việc với MCU::
  + VCC ↔ 2.0V ~ 5.0V
  + GND ↔ mặt đất cung cấp điện
  + AOUT ↔ MCU.IO (đầu ra tương tự)
  1. **Phần mềm**
     1. **MQTT Mosquitto Broker** 
        1. **Khái niệm** [8]
* 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. 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.
  + - 1. **Giao thức MQTT** [9]
* MQTT viết tắt của Message Queuing Telemetry Transport.
* Đây là giao thức truyền tin đơn giản dựa vào truyền thông điệp (message) theo mô hình publish/subcribe.
* Được thiết kế cho những thiết bị IoT đơn giản với băng thông thấp, độ tin cậy cao và khả năng sử dụng trong mạng lưới không ổn định.
* MQTT cho phép các thiết bị IoT có thể phát và nhận bản tin, nhờ vậy người sử dụng gửi các lệnh điều khiển cũng như nhận các bản tin từ các thiết bị IoT.
  + - 1. **Cách hoạt động** [10]



Hình 13. Mô hình hoạt động của MQTT Broker

* Các client sẽ kết nối tới MQTT Broker để gửi hoặc nhận thông điệp (message) do các Client sẽ không kết nối trực tiếp với nhau.
* Khi một Client muốn gửi một dữ liệu đến các Client khác, Client sẽ thực hiện publish dữ liệu tới MQTT Broker sẽ đó từ MQTT Broker này sẽ gửi đến các Client khác.
* Ngược lại khi Client muốn nhận dữ liệu từ MQTT Broker, Client sẽ sử dụng subcribe để nhận dữ liệu gửi về từ MQTT Broker.

=> MQTT Broker đóng vai là một trạm trung gian để giao tiếp giữa các Client.

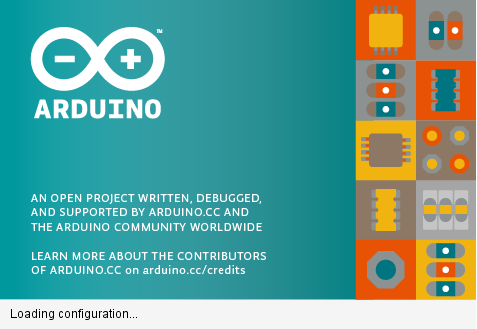
* + - 1. **Ưu và nhược điểm** [11]
* Ưu điểm:
  + Tốc độ truyền nhận và xử lí dữ liệu nhanh, độ ổn định cao, được sử dụng rộng rãi và phù hợp với những ứng dụng embedded.
  + Mosquitto rất nhẹ và phù hợp để sử dụng trên tất cả các thiết bị.
  + Ngoài ra, Mosquitto cũng được hỗ trợ các giao thức TLS/SSL (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.
    1. **Node-RED** [12]
       1. **Khái niệm**
* Node-RED là một công cụ lập trình dùng để kết nối các thiết bị phần cứng, API và các dịch vụ trực tuyến với nhau. Về cơ bản, đây là một công cụ trực quan được thiết kế cho IoT (Internet of Things), nhưng cũng có thể được sử dụng cho các ứng dụng khác nhằm liên kết nhanh các luồng (flow) dịch vụ khác nhau.
* Node-RED là mã nguồn mở và được phát triển bởi Emerging Technology Services của IBM và được tích hợp trong gói ứng dụng khởi động IoT Bluemix của IBM (Platform-as-a-Service hoặc PaaS). Node-RED cũng có thể được triển khai riêng bằng ứng dụng Node.js. Hiện tại, Node-RED là một dự án của JS Foundation.
* Node-RED cho phép người dùng kết hợp các dịch vụ Web và phần cứng bằng cách thay thế các tác vụ mã hóa cấp thấp phổ biến (như một dịch vụ đơn giản giao tiếp với một cổng nối tiếp) và điều này có thể được thực hiện với giao diện kéo thả trực quan. Các thành phần khác nhau trong Node-RED được kết nối với nhau để tạo ra một luồng (flow). Hầu hết mã lệnh (code) cần thiết được tạo tự động.



Hình 14. Giao diện Dashboard của Node-RED

* + - 1. **Các tính năng của Node-RED**
* Các tính năng chính của Node-RED:
  + Cho phép chỉnh sửa luồng (flow) ngay trên trình duyệt.
  + Vì được xây dựng dựa trên Node.js, nên Node-RED hỗ trợ môi trường thời gian thực nhẹ cùng với mô hình hướng sự kiện và không chặn.
  + Các luồng khác nhau được tạo trong Node-RED được lưu trữ bằng JSON, có thể dễ dàng nhập và xuất để chia sẻ với những người khác.
  + Bạn có thể chạy Node-RED cục bộ (hỗ trợ Docker, v.v.).
  + Node-RED có thể dễ dàng phù hợp với hầu hết các thiết bị được sử dụng rộng rãi như Raspberry Pi, BeagleBone Black, Arduino, các thiết bị dựa trên Android, v.v.
  + Node-RED có thể chạy trong môi trường điện toán đám mây như IBM Cloud, AWS, Microsoft Azure, v.v.
    - 1. **Node RED có thể được sử dụng ở đâu**
* Node-RED có thể được sử dụng trong một loạt các ứng dụng. Những cái chính được liệt kê dưới đây:

1. Trong Bluemix, để kết nối với IoT (với ReST và MQTT).
2. Để liên kết và kết nối với các cơ sở dữ liệu (MongoDB).
3. Để lưu trữ dữ liệu IoT cho tính toán hiện tại và tương lai.
4. Đối với các mạng xã hội, khi thực hiện hành động và khi các ứng dụng hướng sự kiện là cần thiết (như Twitter).

* Node-RED có hơn 225.000 gói repo, dễ dàng mở rộng và thêm các gói mới. Nó cũng có một cộng đồng chuyên dụng và được xây dựng với kiến trúc mạnh mẽ bằng cách sử dụng Node.js.
* Node-RED có thể được sử dụng trong các ứng dụng và dịch vụ theo hướng sự kiện và nhanh chóng đưa ra thị trường, với các bước có thể thực hiện dễ dàng.
  + 1. **Arduino IDE** [13]
       1. **Tổng quan**
* Giới thiệu:
  + Arduino IDE là một phần mềm với 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.

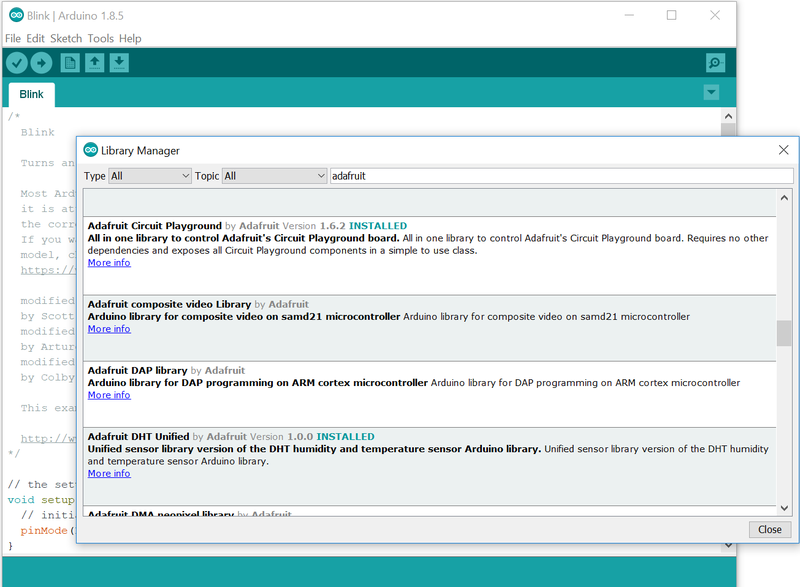
Hình 15. Arduino

* 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.
    - 1. **Các lý do nên sử dụng Arduino IDE:**
* Phần mềm lập trình mã nguồn mở miễn phí:
  + IDE trong Arduino IDE là phần có nghĩa là mã nguồn mở. Nghĩa là phần mềm này miễn phí cả về phần tải về lẫn phần bản quyền. Người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung được nhà phát hành cho phép mà không cần xin phép ai, điều mà họ không được phép làm đối với các phần mềm nguồn đóng.
  + Tuy là phần mềm mã nguồn mở nhưng khả năng bảo mật thông tin của Arduino IDE là vô cùng tuyệt vời, khi phát hiện lỗi nhà phát hành sẽ vá nó và cập nhật rất nhanh khiến thông tin của người dùng không bị mất hoặc rò rỉ ra bên ngoài.



Hình 16. Giới thiệu Adruino IDE

* Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên: Arduino IDE sử dụng ngôn ngữ lập trình C/C++ rất phổ biến trong giới lập trình. Bất kỳ đoạn code nào của C/C++ thì Arduino IDE đều có thể nhận dạng, giúp các lập trình viên thuận tiện trong việc thiết kế chương trình lập cho các bo mạch Arduino.
* Hỗ trợ lập trình tốt cho bo mạch Arduino:
  + Arduino có một module quản lý bo mạch, nơi người dùng có thể chọn bo mạch mà họ muốn làm việc cùng và có thể thay đổi bo mạch thông qua Menu. Quá trình sửa đổi lựa chọn cũng liên tục tự động cập nhật để các dữ liệu có sẵn trong bo mạch và dữ liệu sửa đổi đồng nhất với nhau.
  + Bên cạnh đó, Arduino IDE cũng giúp bạn tìm ra lỗi từ code mà bạn viết, qua đó giúp bạn sửa lỗi kịp thời tránh tình trạng bo mạch Arduino làm việc với code lỗi quá lâu dẫn đến hư hỏng hoặc tốc độ xử lý bị giảm sút.
* Thư viện hỗ trợ phong phú: Arduino IDE tích hợp với hơn 700 thư viện, được viết và chia sẻ bởi nhà phát hành Arduino Software và thành viên trong cộng đồng Arduino. Mọi người có thể tận dụng chúng cho dự án của riêng mình mà không cần phải bỏ ra bất kỳ chi phí nào.



Hình 17. Thư viện của Adruino IDE

* Giao diện đơn giản, dễ sử dụng: Arduino IDE có một giao diện đơn giản, dễ sử dụng giúp người dùng thuận tiện hơn trong thao tác.

phần mềm Arduino IDE



Hình 18. Giao diện sử dụng Adruino IDE

* Hỗ trợ đa nền tảng như Windows, MacOS, Linux: Arduino IDE hoạt động trên 3 hệ điều hành phổ biến nhất. Đó là Windows, Mac OS và Linux. Nhờ vậy, người dùng có thể truy cập vào phần mềm ở bất cứ đâu, bất cứ khi nào. Ngoài ra, người dùng có thể truy cập vào công cụ từ đám mây. Điều này cho phép các nhà lập trình lựa chọn tạo và lưu dự án của mình trên đám mây. Hoặc họ có thể xây dựng chương trình trên máy tính và upload nó lên bo mạch Arduino.
  + 1. **Thư viện Pubsubclient**
       1. **Khái niệm** [14]
* PubSubClient là thư viện cung cấp cho Client các cách thực hiện publish/subcribe thông điệp (message) với server hỗ trợ MQTT.
  + - 1. **Một số hàm API hỗ trợ MQTT Client của PubSubClient** [15]
* Hàm **PubSubClient(Client &client)** để khởi động bộ thư viện. Chúng ta sẽ tạo 1 biến kiểu Client và pass địa chỉ biến đó vào hàm này để khởi động bộ thư viện.
* Hàm setServer(<domain name của Broker>, <TCP port>) để thiết lập địa chỉ của Broker server và port TCP sẽ kết nối đến.
* Hàm **setCallback(<callback>)** để đăng ký hàm callback sẽ được gọi khi thư viện nhận được giá trị mới từ Broker cho dữ liệu đã được subscribe.
* Hàm **connect(<client\_name>)** để bắt đầu quá trình kết nối đến Broker server. Hàm này sẽ đợi đến khi kết nối thành công hoặc timeout; giá trị trả về của hàm là kiểu boolean với True là kết nối thành công, False là kết nối không thành công.
* Hàm **disconnect()** để chủ động ngắt kết nối đến Broker server.
* Hàm **publish(<topic>, <value>)** để gửi giá trị <value> cho dữ liệu <topic> lên Broker server.
* Hàm **subscribe(<topic>)** để đăng ký nhận giá trị mới của dữ liệu <topic> từ Broker server.
* Hàm **connnected()** để kiểm tra trạng thái kết nối đến Broker server (True = kết nối OK).
* Hàm **loop()** là hàm xử lý các tác vụ theo giao thức MQTT. Chúng ta sẽ gọi hàm này liên tục trong hàm loop() của Arduino.
  + - 1. **Hạn chế của PubSubClient** [14]
* Chỉ có thể publish thông qua các message “QoS 0” và chỉ có thể subcribe ở “QoS 0” hoặc “QoS 1”.
* Kích thước tối đa của message bao gồm cả header là 256 bytes theo mặc định.
* Thời gian keep-alive theo mặc định là 15 giây.
* Các Client sử dụng phiên bản MQTT 3.1.1 theo mặc định.
  + - 1. **Các phần cứng tương thích** [14]
* Do thư viện sử dụng Arduino Ethernet Client api để tương tác với phần cứng mạng bên dưới nên thư viện chỉ có để hoạt động được một số board và shield sau:
  + Arduino Ethernet
  + Arduino Ethernet Shield
  + Arduino YUN
  + Arduino WiFi Shield
  + Sparkfun WiFly Shield
  + TI CC3000 WiFi
  + Intel Galileo/Edison
  + ESP8266
  + ESP32
  1. **Dữ liệu cây trồng**
     1. **Cây dâu tây** [16]
* Nhiệt độ tối thiểu: 18oC
* Nhiệt độ tối đa: 22oC
* Độ ẩm đất thích hợp: 70%
* Độ ẩm không khí thích hợp: 84%
  + 1. **Cây ớt** [17, 18]
* Nhiệt độ tối thiểu: 25oC
* Nhiệt độ tối đa: 28oC
* Độ ẩm đất thích hợp: 70%
* Độ ẩm không khí thích hợp: 70%
  + 1. **Cây đậu cô-ve** [19]
* Nhiệt độ tối thiểu: 20oC
* Nhiệt độ tối đa: 25oC
* Độ ẩm đất thích hợp: 70%
* Độ ẩm không khí thích hợp: 65%

1. **Thiết kế hệ thống**
   1. **Nguyên lý hoạt động**
      1. **Thành phần của hệ thống:**

* ESP8266: vi xử lý.
* DHT11: thu thập nhiệt độ và độ ẩm không khí.
* Cảm biến độ ẩm đất: thu thập độ ẩm đất.
* Relay: đóng cắt máy bơm.
* Máy bơm: bơm nước đến cây trồng.
* WebUI (Node-RED): để người dùng tương tác.
  + 1. **Nguyên lý hoạt động:**

1. DHT11 và cảm biến độ ẩm đất được kết nối với ESP8266 để cung cấp dữ liệu về thông số của môi trường.
2. ESP8266 sẽ tiếp nhận những tín hiệu từ cảm biến, sau đó xử lý những tín hiệu này và gửi yêu cầu tưới nước đến chân GPIO nối với relay khi đạt ngưỡng cần tưới.
3. Song song đó, ESP8266 cũng gửi thông tin từ cảm biến lên broker và lắng nghe chỉ thị từ người dùng khi người dùng tương tác với web.
4. Relay khi được kích hoạt sẽ kích hoạt máy bơm hoạt động, bơm nước đến cây.
   1. **Sơ đồ nối dây**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ESP8266** | **DHT11** | **Cảm biến độ ẩm đất** | **Relay** | **Máy bơm** |
| 3V3 | VCC (+) | VCC (+) | VCC, NO |  |
| GND | GND (-) | GND (-) | GND | - |
| A0 | - | S | - |  |
| D4 | - | - | S |  |
| D6 | OUT | - | - |  |
|  |  |  | COM | + |

Bảng 1. Sơ đồ nối dây của hệ thống

1. **Hiện thực đề tài**
   1. **Giao diện web UI**

Graphical user interface

Description automatically generated

Hình 19. Giao diện webUI của hệ thống

* 1. **Quá trình thực hiện**
     1. **Lập trình hệ thống**
* Khai báo các giá trị:
  + Khai báo thư viện
    - Sử dụng thư viện ESP8266WiFi.h để kết nối Wifi.
    - Sử dụng thư viện PubSubClient.h để gửi dữ liệu lên MQTT Mosquitto Broker.
    - Sử dụng thư viện DHT.h để lấy dữ liệu từ cảm biến DHT11.
  + Khai báo chân
  + Khai báo thông tin Wifi
  + Khai báo thông tin MQTT broker
  + Khai báo biến
  + Khai báo Threshold
* Trong hàm setup, tiến hành khởi động DHT11 Sensor, khởi động cảm biến độ ẩm đất, khởi động kết nối wifi, khởi động chân tín hiệu điều khiển động cơ, đặt server mqtt broker, đặt callback cho MQTT broker.
* Trong hàm loop gọi thực hiện hàm con xử lý tín hiệu, kiểm tra kết nối wifi và thực hiện gửi dữ liệu lên broker mỗi 2 giây.
* Các hàm con:

1. Hàm gửi dữ liệu lên broker
2. Hàm xử lý tín hiệu
3. Hàm đọc dữ liệu cảm biến
4. Hàm callback
5. Hàm điều khiển động cơ
6. Hàm kết nối lại
7. Hàm setup wifi
   * 1. **Cài đặt và cấu hình phần mềm**
        1. **Cài đặt Mosquitto Broker**

Download Mosquitto Broker từ trang chủ <https://mosquitto.org/download/>

Tiến hành cài đặt như các phần mềm khác.

Sau khi cài đặt xong, các bạn mở cửa sổ Command Promt và gõ lệnh sau để bật mosquitto:

“C:\Program Files\mosquitto\mosquitto” -v và bấm Enter.

* + - 1. **Cài đặt và cấu hình Node-RED**
* Cài đặt Node-RED trên trang chủ <https://nodered.org>.
* Giao diện bắt đầu của Node-RED

Graphical user interface

Description automatically generated with low confidence

Hình 20. Giao diện bắt đầu của Node-RED

* Flow tổng hợp của hệ thống:

Diagram

Description automatically generated with low confidence

Hình 21. Flow tổng hợp của hệ thống

* Gồm 2 khối chính, khối gửi dữ liệu từ webUI và khối nhận dữ liệu.
  + Khối gửi dữ liệu từ webUI đến MQTT Broker:

Diagram

Description automatically generated

Hình 22. Khối gửi dữ liệu từ webUI đến MQTT Broker

* Bắt những hành động từ webUI và gửi đến MQTT Broker dưới topic “button”.
* Trong đó:
  + Các nút gửi thông tin loại cây trồng đến ESP8266.

A picture containing diagram

Description automatically generated

Hình 23. Các nút gửi thông tin loại cây trồng đến ESP8266

* + Các nút gửi hành động “Tưới” hoặc “Ngừng tưới” đến ESP8266.

Excel

Description automatically generated with low confidence

Hình 24. Các nút gửi hành động “Tưới” hoặc “Ngừng tưới” đến ESP8266

* + Khối “msg.payload” màu xanh lá là khối phục vụ debugging

A picture containing text, shoji, indoor, public

Description automatically generated

Hình 25. Khối phục vụ debugging

* + Khối nhận dữ liệu từ MQTT Broker:

Diagram

Description automatically generated

Hình 26. Khối nhận dữ liệu từ MQTT Broker

* + - Lắng nghe từ topic “event”, sau đó xác định các thông số Moisture (Độ ẩm đất), Humidity (Độ ẩm không khí), Temperature (Nhiệt độ không khí) bằng các subflow trùng tên tự định nghĩa.
    - Cấu trúc của 1 subflow trong hệ thống:

A picture containing text, shoji

Description automatically generated

Hình 27. Cấu trúc subflow trong hệ thống

Cụ thể:

* + Khối function “split moisture” là khối tách dữ liệu độ ẩm đất từ payload nhận được từ ESP8266 để hiển thị ra giao diện cho người dùng tương tác.

Graphical user interface, text, application

Description automatically generated

Hình 28. Khối function

* + Khối “set msg.payload” để chuyển định dạng của msg.payload (ở đây chuyển dạng chuỗi sang số )

Graphical user interface, text, application

Description automatically generated

Hình 29. Khối chuyển định dạng payload

* + Khối “msg.payload” màu xanh lá là khối phục vụ debugging.

Table

Description automatically generated with medium confidence

Hình 30. Khối phục vụ debugging

* + Khối “Độ ẩm đất” màu xanh dương để hiển thị dữ liệu ra giao diện người dùng.

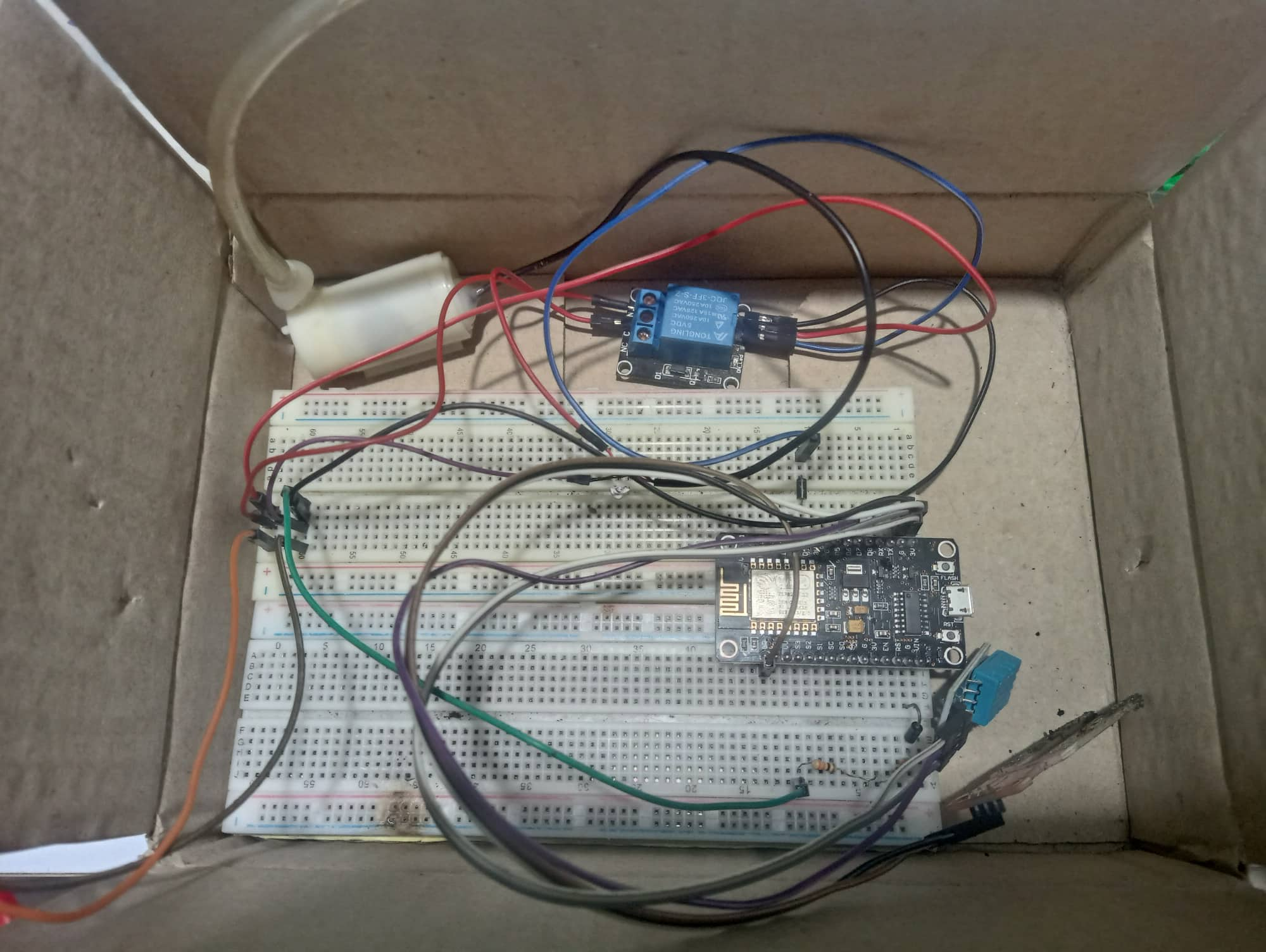
A picture containing text, shoji, window, building

Description automatically generated

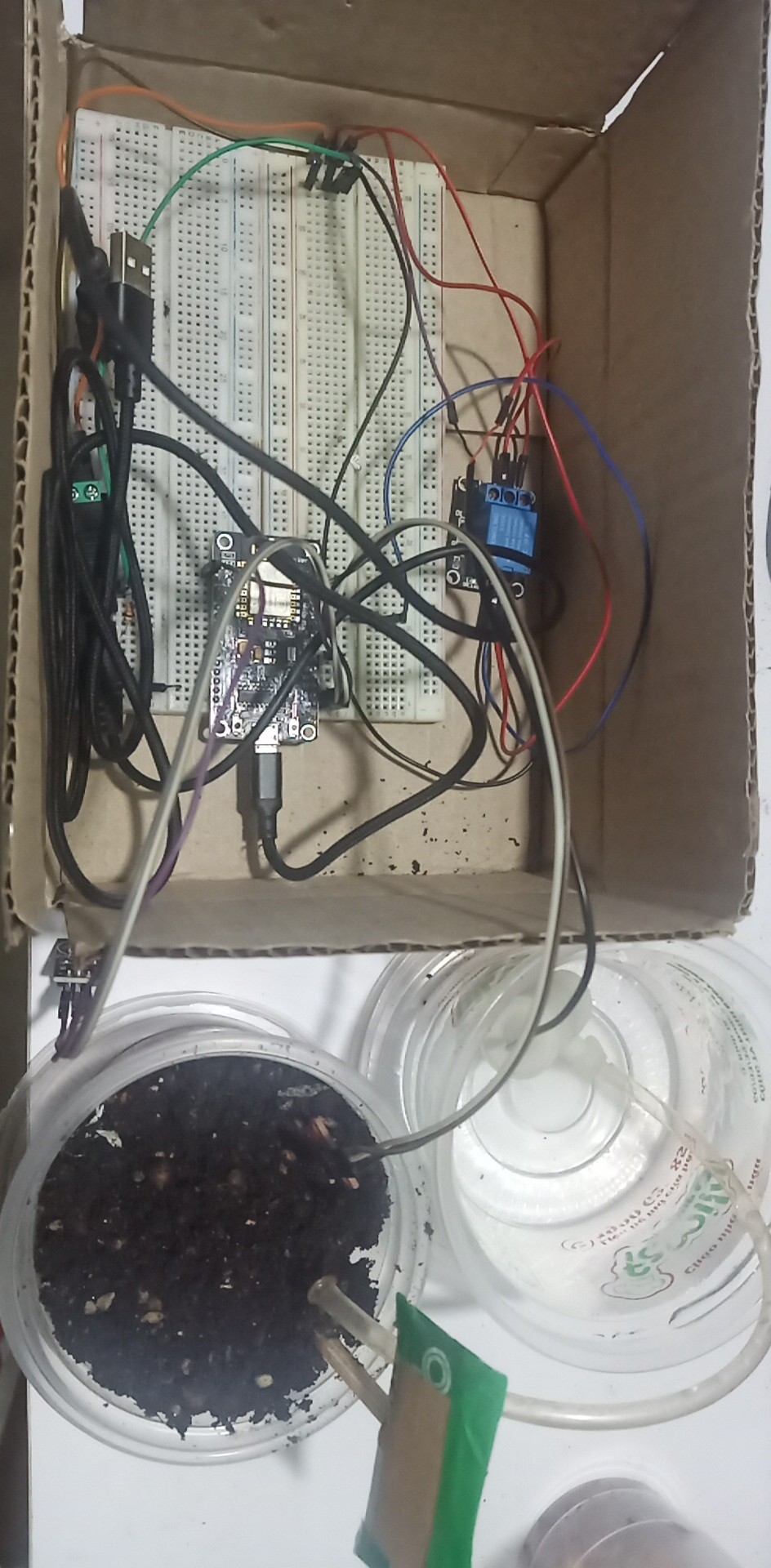
Hình 31. Khối hiển thị

Tương tự với 2 subflows còn lại về độ ẩm không khí và nhiệt độ.

1. **Thực nghiệm đề tài**
   1. **Mô hình thực tế**



Hình 32. Mô hình thực tế 1



Hình 33. Mô hình thực tế 2

* 1. **Hiển thị dữ liệu đo được lên webUI của Node-RED**

**Chart

Description automatically generated**

Hình 34. Thực nghiệm - Hiển thị dữ liệu đo được lên webUI

1. **Kết luận**
   1. **Kết quả đạt được**

* Hoàn thành hệ thống tưới cây tự động theo ba thông số: độ ẩm đất, độ ẩm không khí, nhiệt độ môi trường.
* Có webUI để người dùng tương tác: có thể nhấn nút yêu cầu tưới cây, chọn loại cây đang trồng để đặt ngưỡng hợp lý cho riêng loại cây đó.
* Hiển thị được dữ liệu cảm biến lên giao diện web.
  1. **Những khó khăn khi thực hiện**
* Nhóm thiếu thời gian cùng nhau lên lab thực hiện đồ án.
* Ban đầu, nhóm dự định sẽ làm giao diện tương tác của người dùng trên Blynk, nhưng thay đổi thành Node-red vào gần cuối nên gặp chút ít khó khăn.
* Dữ liệu cho các loại cây trồng khá khó tìm, nhóm mất khá nhiều thời gian tìm kiếm.
  1. **Khuyết điểm**
* Mô hình còn ít loại cây trồng, chưa có chức năng tự thêm loại cây trồng mới.
* Mô hình còn ít chức năng, chỉ mới có chức năng tự động tưới và tưới thủ công.
  1. **Hướng phát triển**
* Thêm các chức năng mới như chức năng hẹn giờ tưới cây.
* Mở rộng thêm nhiều loại cây trồng mới, cho phép người dùng tự nhập thông tin cây trồng.
* Chuyển Mosquitto broker và Node-red lên Raspberry Pi, tạo hệ thống hoạt động độc lập, không phụ thuộc vào máy tính cá nhân.
* Thêm nhiều cảm biến hơn để giám sát nhiều thông số của môi trường hơn.
* Thêm những thiết bị như phun sương, quạt để tối ưu hoá chức năng chăm sóc khu vườn.
* Mở rộng qui mô mô hình để phù hợp với khu vườn lớn hơn.

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | "dungat.com," [Online]. Available: https://dungat.com/blog/posts/gioi-thieu-ve-esp8266. [Accessed 21 05 2022]. |
| [2] | "lophocvui.com," [Online]. Available: https://lophocvui.com/iot-internet-of-things/smart-home/tong-quan-ve-esp8266/. [Accessed 21 05 2022]. |
| [3] | "blog.mecsu.vn," [Online]. Available: https://blog.mecsu.vn/dht11-cam-bien-nhiet-do-va-do-am/. [Accessed 21 05 2022]. |
| [4] | "dientutuonglai.com," [Online]. Available: https://dientutuonglai.com/cam-bien-nhiet-do-va-do-am-dht11.html. [Accessed 21 5 2022]. |
| [5] | "lastminuteengineers.com," [Online]. Available: https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/. [Accessed 21 5 2022]. |
| [6] | "www.waveshare.com," [Online]. Available: https://www.waveshare.com/moisture-sensor.htm. [Accessed 21 5 2022]. |
| [7] | "www.waveshare.com," [Online]. Available: https://www.waveshare.com/wiki/Moisture\_Sensor. [Accessed 21 5 2022]. |
| [8] | "viblo.asia," [Online]. Available: https://viblo.asia/p/tim-hieu-ve-mqtt-mosquitto-broker-va-cach-cai-dat-yMnKMjgrZ7P. [Accessed 21 5 2022]. |
| [9] | "viblo.asia," [Online]. Available: https://viblo.asia/p/mqtt-la-gi-vai-tro-cua-mqtt-trong-iot-V3m5WL3bKO7. [Accessed 21 5 2022]. |
| [10] | "iot4beginners.com," [Online]. Available: https://iot4beginners.com/mosquitto-mqtt-broker-introduction/. [Accessed 21 5 2022]. |
| [11] | "myphammioskin.com.vn," [Online]. Available: https://myphammioskin.com.vn/mosquitto-la-gi/. [Accessed 21 5 2022]. |
| [12] | "dientuviet.com," [Online]. Available: https://dientuviet.com/node-red-la-gi/. [Accessed 21 5 2022]. |
| [13] | "vn.got-it.ai," [Online]. Available: https://vn.got-it.ai/blog/phan-mem-arduino-ide-la-gi-chi-tiet-nhat. [Accessed 21 5 2022]. |
| [14] | "github.com," [Online]. Available: https://github.com/knolleary/pubsubclient/. [Accessed 21 5 2022]. |
| [15] | "htelectronics.vn," [Online]. Available: https://htelectronics.vn/huong-dan-su-dung-esp8266-trong-cac-ung-dung-internet-of-things-phan-9/. [Accessed 21 5 2022]. |
| [16] | "khuyennong.lamdong.gov," [Online]. Available: http://khuyennong.lamdong.gov.vn/ky-thuat-trong-trot/ki-thuat-trong-cay-an-qua/276-quy-trinh-k-thu-t-tr-ng-cay-dau-tay. [Accessed 20 5 2022]. |
| [17] | "tainangviet," [Online]. Available: https://www.tainangviet.vn/index.php/ky-thuat-trong-ot-theo-vietgap-dar965/. [Accessed 20 5 2022]. |
| [18] | "vietaseeds," [Online]. Available: http://vietaseeds.com.vn/ky-thuat-trong-ot.html. [Accessed 20 5 2022]. |
| [19] | "khuyennong.lamdong.gov," [Online]. Available: http://khuyennong.lamdong.gov.vn/ky-thuat-trong-trot/ki-thuat-trong-rau/864-quy-trinh-k-thut-trng-u-co-ve. [Accessed 20 5 2022]. |