**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**<LÊ HOÀNG KHANH>**

**ĐỒ ÁN MÔN HỌC**

**<HỆ THỐNG IOT QUẢN LÝ VƯỜN CÂY THÔNG MINH>**

**TP. HỒ CHÍ MINH, 2023**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**<LÊ HOÀNG KHANH> – <20521448>**

**ĐỒ ÁN MÔN HỌC**

**<** **HỆ THỐNG IOT QUẢN LÝ VƯỜN CÂY THÔNG MINH >**

**GIẢNG VIÊN HƯỚNG DẪN**

**PHAN ĐÌNH DUY**

**TP. HỒ CHÍ MINH, 2023**

MỤC LỤC

[Chương 1. TỔNG QUAN 3](#_Toc134979929)

[1.1. Đặt vấn đề 3](#_Toc134979930)

[1.2. Mục tiêu 3](#_Toc134979931)

[1.3. Nội dung nghiên cứu 4](#_Toc134979932)

[1.4. Giới hạn 4](#_Toc134979933)

[Chương 2. CƠ SỞ LÝ THUYẾT 5](#_Toc134979934)

[2.1. Giới thiệu mô hình vườn cây thông minh 5](#_Toc134979935)

[2.2. Tổng quan về ESP32 5](#_Toc134979936)

[2.3. Tổng quan IoTs 6](#_Toc134979937)

[2.4. Các chuẩn giao tiếp và các giao thức truyền thông. 7](#_Toc134979938)

[2.4.1. Chuẩn giao tiếp OneWire 7](#_Toc134979939)

[2.4.2. Giao thức HTTP 10](#_Toc134979940)

[2.4.3. Giao thức MQTT 11](#_Toc134979941)

[2.4.4. Giao thức WebSocket 12](#_Toc134979942)

[Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc134979943)

[3.1. Giới thiệu 14](#_Toc134979944)

[3.2. Thiết kế sơ đồ khối hệ thống. 14](#_Toc134979945)

[3.3. Tính toán, thiết kế và hiện thực mạch 15](#_Toc134979946)

[3.3.1. Khối trung tâm 15](#_Toc134979947)

[3.3.2. Khối server 17](#_Toc134979948)

[3.3.2.1. Firebase 17](#_Toc134979949)

[3.3.2.2. Node-RED 18](#_Toc134979950)

[3.3.2.3. Mosquitto 19](#_Toc134979951)

[3.3.3. Khối cảm biến 20](#_Toc134979952)

[3.3.3.1. Cảm biến DHT11 20](#_Toc134979953)

[3.3.3.2. Cảm Biến Độ Ẩm Đất 21](#_Toc134979954)

[3.3.3.3. Cảm biến ánh sáng 22](#_Toc134979955)

[3.3.4. Khối nguồn 23](#_Toc134979956)

[3.3.5. Khối chấp hành 23](#_Toc134979957)

[3.3.5.1. Máy bơm 24](#_Toc134979958)

[3.3.5.2. Đèn 25](#_Toc134979959)

[3.3.5.3. Mạch Relay 25](#_Toc134979960)

[3.4. Lập trình hệ thống 25](#_Toc134979961)

[3.4.1. Finite State Machine 26](#_Toc134979962)

[3.4.2. PlatformIO 27](#_Toc134979963)

[3.4.3. Node-Red 28](#_Toc134979964)

[Chương 4. KẾT QUẢ THỰC NGHIỆM 31](#_Toc134979965)

[4.1. Kết quả 31](#_Toc134979966)

[4.1.1. Tổng quan 31](#_Toc134979967)

[4.1.2. Các tính năng của hệ thống 31](#_Toc134979968)

[4.1.3. Giao diện người dùng 31](#_Toc134979969)

[4.1.3.1. Giao diện người dùng khi ở xa 31](#_Toc134979970)

[4.1.3.2. Giao diện người dùng khi ở gần 32](#_Toc134979971)

[4.1.4. Phần cứng 33](#_Toc134979972)

[4.1.5. Demo đồ án 33](#_Toc134979973)

[4.2. Nhận xét, đánh giá 33](#_Toc134979974)

[Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc134979975)

[5.1. Kết luận 34](#_Toc134979976)

[5.2. Hướng phát triển 34](#_Toc134979977)

DANH MỤC HÌNH VẼ

[Hình 2‑1: Hình ảnh và CPU ESP32 5](file:///D:\DoAn\Project1\Báo%20cáo\Học%20kì\Báo%20cáo%20giữa%20kì%20đồ%20án%20môn%20học-%2020521448-Lê%20Hoàng%20Khanh-Hệ%20thống%20IoT%20quản%20lý%20vườn%20cây%20thông%20minh\Báo%20cáo%20giữa%20kì%20đồ%20án%20môn%20học-%2020521448-Lê%20Hoàng%20Khanh-Hệ%20thống%20IoT%20quản%20lý%20vườn%20cây%20thông%20minh.docx#_Toc134982632)

[Hình 2‑2: Cấu trúc cơ bản của một mô hình IoT 6](#_Toc134982633)

[Hình 2‑3: Giao tiếp OneWire 8](#_Toc134982634)

[Hình 2‑4: Khung truyền của giao tiếp OneWire 8](#_Toc134982635)

[Hình 2‑5: Giao thức HTTP 11](#_Toc134982636)

[Hình 2‑6: Giao thức MQTT 12](#_Toc134982637)

[Hình 2‑7: Giao thức WebSocket 13](#_Toc134982638)

[Hình 3‑1: Cấu trúc hệ thống 14](#_Toc134982639)

[Hình 3‑2: Sơ đồ khối hệ thống 15](#_Toc134982640)

[Hình 3‑3: Hình ảnh và CPU ESP32-C3-32S 16](file:///D:\DoAn\Project1\Báo%20cáo\Học%20kì\Báo%20cáo%20giữa%20kì%20đồ%20án%20môn%20học-%2020521448-Lê%20Hoàng%20Khanh-Hệ%20thống%20IoT%20quản%20lý%20vườn%20cây%20thông%20minh\Báo%20cáo%20giữa%20kì%20đồ%20án%20môn%20học-%2020521448-Lê%20Hoàng%20Khanh-Hệ%20thống%20IoT%20quản%20lý%20vườn%20cây%20thông%20minh.docx#_Toc134982641)

[Hình 3‑4: Chức năng của Firebase 17](#_Toc134982642)

[Hình 3‑5: Chức năng Node-Red 19](#_Toc134982643)

[Hình 3‑6: Hình ảnh thực tế cảm biến DHT11 20](#_Toc134982644)

[Hình 3‑7: Kết nối cảm biến DHT11 với vi điều khiển 21](#_Toc134982645)

[Hình 3‑8: Hình ảnh thực tế cảm biến độ ẩm đất 22](#_Toc134982646)

[Hình 3‑9: Hình ảnh thực tế cảm biến ánh sáng 23](#_Toc134982647)

[Hình 3‑10: Hình ảnh thực tế máy bơm 24](#_Toc134982648)

[Hình 3‑11: Hình ảnh thực tế mạch relay 25](#_Toc134982649)

[Hình 3‑12: FSM hệ thống 26](#_Toc134982650)

[Hình 3‑13: Trang chủ PlatformIO trên VScode 27](#_Toc134982651)

[Hình 3‑14: Giao diện lập trình trên VScode trên nền tảng PlatformIO 28](#_Toc134982652)

[Hình 3‑15: Khởi chạy Node-Red bằng Terminal 29](#_Toc134982653)

[Hình 3‑16: Giao diện sử dụng Node-Red 30](#_Toc134982654)

[Hình 3‑17: Giao diện Dashboard của Node-Red 30](#_Toc134982655)

[Hình 4‑1: Giao diện người dùng khi ở xa 32](#_Toc134982656)

[Hình 4‑2: Giao diện người dùng khi ở gần 32](#_Toc134982657)

[Hình 4‑3: Phần cứng của hệ thống 33](#_Toc134982658)

DANH MỤC BẢNG

[Bảng 2‑1: Bảng giá trị thời gian OneWire 10](#_Toc134979896)

[Bảng 3‑1: Một vài thông số ESP32-C3-32S 17](#_Toc134979897)

[Bảng 3‑2: Ưu và nhược điểm Firebase 18](#_Toc134979898)

[Bảng 3‑3: Ưu và nhược điểm Mosquitto 20](#_Toc134979899)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Nội dung |
| 1 | IoTs | Internet of Things |
| 2 | ESP | Event Stream Processing |
| 3 | DHT | Digital Humidity and Temperature |
| 4 | CPU | Central Processing Unit |
| 5 | RF | Radio Frequency |
| 6 | TSMC | Taiwan Semiconductor Manufacturing Company |
| 7 | RFID | Radio Frequency Identification |
| 8 | NFC | Near-Field Communications |
| 9 | QR | Quick Response |
| 10 | IP | Internet Protocol |
| 11 | HTTP | Hypertext Transfer Protocol |
| 12 | API | Application Programming Interface |
| 13 | TCP | Transmission Control Protocol |
| 14 | UDP | User Datagram Protocol |
| 15 | MQTT | Message Queueing Telemetry Transport |
| 16 | ISO | International Organization for Standardization |
| 17 | OASIS | Organization for the Advancement of Structured Information Standards |
| 18 | IEC | International Electrotechnical Commission |
| 19 | M2M | Machine to Machine |
| 20 | RFC | Request for Comments |
| 21 | IDL | Interface Definition Language |
| 22 | W3C | World Wide Web Consortium |
| 23 | SoC | Security operations center |
| 24 | UART | Universal asynchronous receiver-transmitter |
| 25 | USB | Universal Serial Bus |
| 26 | WEP | Wired Equivalent Privacy |
| 27 | WPA | Wi-Fi Protected Access |
| 28 | PSK | Pre-Shared Key |
| 29 | STA | Station |
| 30 | AP | Access Point |
| 31 | IBM | International Business Machines |
| 32 | TTL | Time to live |
| 33 | RH | Relative Humidity |
| 34 | IC | Integrated Circuit |
| 35 | VDC | Voltage Direct Current |
| 36 | FSM | Finite State Machine |
| 37 | HTML | HyperText Markup Language |
| 38 | IDE | Integrated development environment |

TÓM TẮT ĐỒ ÁN

Đồ án này trình bày về cách thực hiện một ứng dụng hệ thống IoTs vào việc quản lý “vườn thông minh” sử dụng esp32. Trong đồ án này em sẽ đi chi tiết về phần cài đặt, thiết kế và thực hiện phần cứng, thiết kế và thực hiện phần mềm và cách thiết lập cũng như sử dụng đề tài. Đề tài sẽ cho người sử dụng theo dõi được các yếu tố môi trường và tình trạng của hệ thống. Đồng thời người sử dụng cũng có thể điều khiển hệ thống thông qua các thiết bị có kết nối mạng.

MỞ ĐẦU

Hiện nay cùng với sự phát triển của xã hội, cuộc sống ngày càng được nâng cao

thì việc áp dụng công nghệ khoa học kỹ thuật vào đời sống công việc ngày càng cần

thiết. Cùng với sự phát triển của các ngành khoa học kỹ thuật, công nghệ kỹ thuật điện tử mà trong đó đặc biệt là kỹ thuật điều khiển tự động đóng vai trò quan trọng trong mọi lĩnh vực khoa học kỹ thuật, quản lý, công nghiệp, nông nghiệp, đời sống, quản lý thông tin,...

Nước ta là một đất nước nông nghiệp, tuy nhiên trong nhiều năm quy mô cũng như chất lượng và sản lượng nông nghiệp của nước ta luôn thấp hơn so với các nước khác mà nguyên nhân chính là việc công nghệ sản xuất của nước ta quá lạc hậu, chủ yếu dựa vào tay chân. Do đó, IoTs đã và đang dẫn đầu trong việc cải thiện chất lượng cũng như năng suất nuôi trồng nông nghiệp nước ta hiện nay. Tất cả được điều chỉnh và điều khiển hoàn toàn tự động và áp dụng công nghệ khoa học kỹ thuật vào quy trình giám sát và sản xuất. Việc sử dụng hệ thống tự động giúp chúng ta có thể tiết kiệm nhân lực, tăng độ chính xác trong giám sát và điều khiển môi trường nhằm nâng cao chất lượng sản phẩm.

Với mục đích muốn tiếp cận với các công nghệ đang phát triển trên. Vì vậy, nhóm

thực hiện đồ án với mong muốn chế tạo ra mô hình hệ thống quản lý vườn cây thông minh sử dụng esp32 được điều khiển, giám sát bằng các thiết bị có kết nối mạng

# TỔNG QUAN

## Đặt vấn đề

Hiện nay cùng với sự phát triển của xã hội, cuộc sống ngày càng được nâng cao thì việc áp dụng công nghệ khoa học kỹ thuật vào đời sống công việc ngày càng cần thiết. Cùng với sự phát triển của các ngành khoa học kỹ thuật, công nghệ kỹ thuật điện tử mà trong đó đặc biệt là kỹ thuật điều khiển tự động đóng vai trò quan trọng trong mọi lĩnh vực khoa học kỹ thuật, quản lý, công nghiệp, nông nghiệp, đời sống, quản lý thông tin,...

Nước ta là một đất nước nông nghiệp, tuy nhiên trong nhiều năm quy mô cũng như chất lượng và sản lượng nông nghiệp của nước ta luôn thấp hơn so với các nước khác mà nguyên nhân chính là việc công nghệ sản xuất của nước ta quá lạc hậu, chủ yếu dựa vào tay chân. Mô hình nhà kính là nền tảng cho tiêu chuẩn về chất lượng, công năng và giá trị của sản phẩm trong việc sản xuất nông nghiệp theo hướng nông nghiệp công nghệ cao. Tính linh hoạt của nhà kính giúp cho người trồng trọt có thể trồng trọt trên bất cứ môi trường nào, diện tích trồng trọt có thể từ vài trăm mét vuông đến hàng chục héc-ta.

Trên cơ sở và yêu cầu từ thực tế, những đòi hỏi ngày càng cao của phát triển nông nghiệp công nghệ cao, cộng với sự phát triển mạnh của khoa học công nghệ, đặc biệt là công nghệ thông tin, kỹ thuật điện-điện tử. Phát triển kỹ thuật điều khiển tự động từ khoảng cách xa trong nông nghiệp đang là xu thế phát triển nông nghiệp cao nói chung và nhà kính tự động nói riêng. Em đề xuất đề tài “Hệ Thống Iot Quản Lý Vườn Cây Thông Minh”.

## Mục tiêu

* Tìm hiểu được sự sinh trưởng của cây trồng mong muốn để từ đó nắm bắt được những điều kiện nhiệt độ, độ ẩm, ánh sáng thích hợp với sự phát triển của loại cây đó.
* Tìm hiểu được cơ sở lý thuyết của việc thiết kế và thi công một vườn cây trồng.
* Thiết kế được hệ thống tự động chăm sóc thông qua việc giám sát các thông số nhiệt độ, độ ẩm, cường độ ánh sáng.
* Thiết kế được giao diện Web giám sát, điều khiển các thiết bị thông qua mạng Internet.
* Căn chỉnh, hoàn thiện mô hình môi trường.

## Nội dung nghiên cứu

* **Nội dung 1**: Tìm hiểu và nghiên cứu về cấu tạo phần cứng, nguyên lý hoạt động, tính năng của module ESP32, DHT11, cảm biến ánh sáng, cảm biến độ ẩm đất.
* **Nội dung 2**: Tìm hiểu và nghiên cứu về thiết kế Web, cloud database.
* **Nội dung 3**: Thiết kế hệ thống điều khiển, lưu đồ giải thuật và chương trình điều khiển mô hình.
* **Nội dung 4**: Tìm hiểu và nghiên cứu về cây trồng.

## Giới hạn

* Chọn cây đậu làm đối tượng nghiên cứu.
* Hệ thống có chức năng tưới nước và bật đèn.
* Dừng ở mức độ mô hình học tập chưa đưa vào thực tế để sử dụng.

# CƠ SỞ LÝ THUYẾT

## Giới thiệu mô hình vườn cây thông minh

## Tổng quan về ESP32

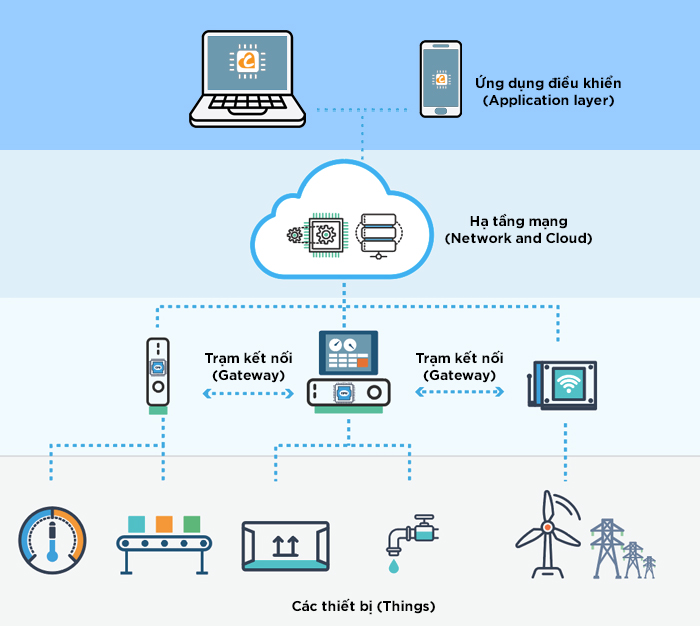
Hình 2‑1: Hình ảnh và CPU ESP32

- ESP32 là một vi điều khiển giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth. 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.

- Một số loại esp32 phổ biến trên thị trường có thể kể đến là: ESP-WROOM-32, ESP32-WROVER, ESP32-PICO-D4, ESP-32S,…

## Tổng quan IoTs



Hình 2‑2: Cấu trúc cơ bản của một mô hình IoT

- IoT (Internet of Things) là thuật ngữ dùng để chỉ các đối tượng có thể được nhận biết cũng như chỉ sự tồn tại của chúng trong một kiến trúc mang tính kết nối. Cụm từ này được đưa ra bởi Kevin Ashton vào năm 1999. Ông là một nhà khoa học đã sáng lập ra Trung tâm Auto-ID ở đại học MIT, nơi thiết lập các quy chuẩn toàn cầu cho RFID (một phương thức giao tiếp không dây dùng sóng radio) cũng như một số loại cảm biến khác. IoT sau đó cũng được dùng nhiều trong các ấn phẩm đến từ các hãng và nhà phân tích.

- Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng. Nếu mọi đối tượng, kể cả con người, được "đánh dấu" để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lí được nó thông qua máy tính. Việc đánh dấu có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID, NFC, mã vạch, mã QR, watermark kĩ thuật số... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại...

- Ngoài những kĩ thuật nói trên, nếu nhìn từ thế giới web, chúng ta có thể sử dụng các địa chỉ độc nhất để xác định từng vật, chẳng hạn như địa chỉ IP. Mỗi thiết bị sẽ có một IP riêng biệt không nhầm lẫn. Sự xuất hiện của IPv6 với không gian địa chỉ cực kì rộng lớn sẽ giúp mọi thứ có thể dễ dàng kết nối vào Internet cũng như kết nối với nhau.

- Một trong những vấn đề với IoT đó là khả năng tạo ra một ứng dụng IoT nhanh chóng. Để khắc phục, hiện nay nhiều hãng, công ty, tổ chức trên thế giới đang nghiên cứu các nền tảng giúp xây dựng nhanh ứng dụng dành cho IoT. Đại học British Columbia ở Canada hiện đang tập trung vào một bộ toolkit cho phép phát triển phần mềm IoT chỉ bằng các công nghệ/tiêu chuẩn Web cũng như giao thức phổ biến. Công ty như ioBridge thì cung cấp giải pháp kết nối và điều khiển hầu như bất kì thiết bị nào có khả năng kết nối Internet.

## Các chuẩn giao tiếp và các giao thức truyền thông.

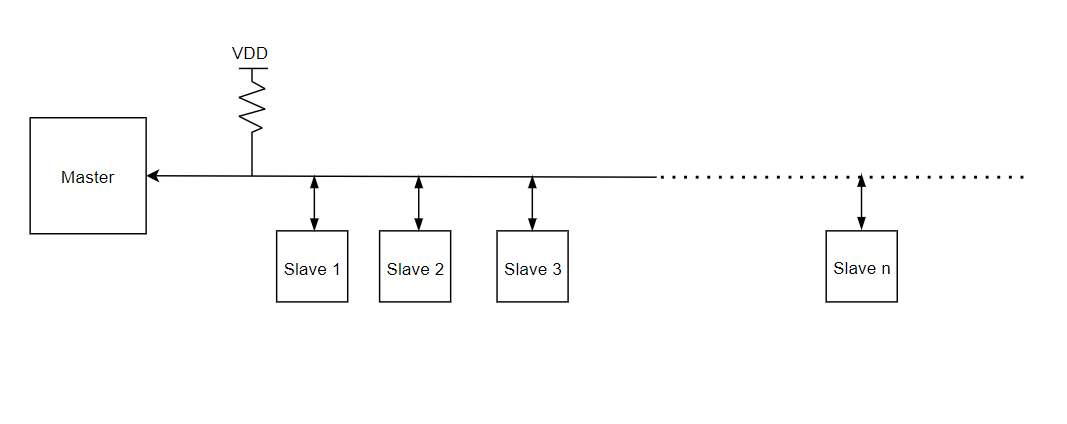
### Chuẩn giao tiếp OneWire

OneWire là hệ thống bus giao tiếp được thiết kế bởi Dallas Semiconductor Corp. Giống như tên gọi, hệ thống bus này chỉ sử dụng 1 dây để truyền nhận dữ liệu.

Chính vì chỉ sử dụng 1 dây nên giao tiếp này có tốc độ truyền thấp nhưng dữ liệu lại truyền được khoảng cách xa hơn.

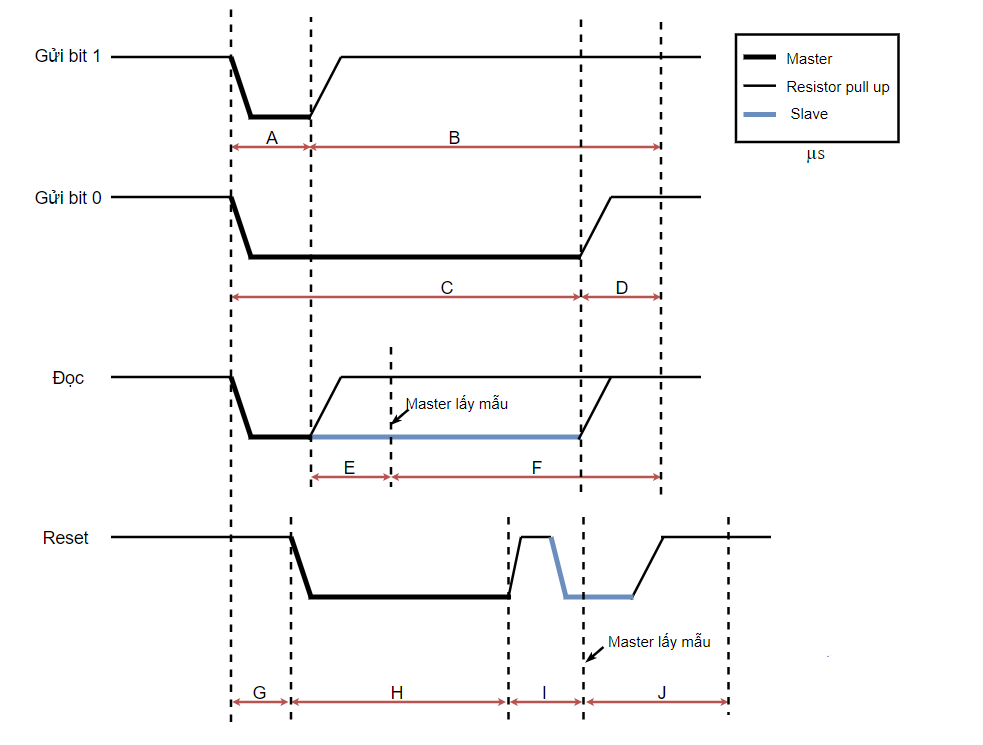
OneWire chủ yếu sử dụng để giao tiếp với các thiết bị nhỏ, thu thập và truyền nhận dữ liệu thời tiết, nhiệt độ,… các công việc không yêu cầu tốc độ cao.

Giống như các chuẩn giao tiếp khác, 1-Wire cho phép truyền nhận dữ liệu với nhiều Slave trên đường truyền. Tuy nhiên chỉ có thể có 1 Master ( điểm này giống với SPI).



Hình 2‑3: Giao tiếp OneWire

Các thao tác hoạt động cơ bản của bus sẽ được quy định bởi thời gian kéo đường truyền xuống mức thấp (Low) như hình vẽ dưới. Có 4 thao tác cơ bản như sau:



Hình 2‑4: Khung truyền của giao tiếp OneWire

Giải thích ý nghĩa :

- Gửi bit 1: Khi muốn gửi đi bit 1, thiết bị Master sẽ kéo bus xuống mức 0 trong một khoảng thời gian A (µs) và trở về mức 1 trong khoảng B (µs).

- Gửi bit 0: Thiết bị Master kéo bus xuống mức 0 trong một khoảng thời gian C (µs) và trở về mức 1 trong khoảng D (µs).

- Đọc bit: Thiết bị Master kéo bus xuống 1 khoảng A (µs). Trong khoảng thời gian E (µs) tiếp theo, thiết bị master sẽ tiến hành lấy mẫu. Có nghĩa trong E (µs) này, nếu bus ở mức 1, thiết bị master sẽ đọc bit 1. Ngược lại, nếu bus ở mức 0 thì master sẽ đọc được bit 0.

- Reset: Thiết bị Master kéo bus xuống 1 khoảng thời gian H (µs) và sau đó về mức 1. Khoảng thời gian này gọi là tín hiệu reset. Trong khoảng thời gian I (µs) tiếp theo, thiết bị master tiến hành lấy mẫu. Nếu thiết bị slave gắn với bus gửi về tín hiệu 0, (tức bus ở mức 0), master sẽ hiểu rằng slave vẫn có mặt và quá trình trao đổi dữ liệu lại tiếp tục. Ngược lại nếu slave gửi về tin hiệu 1 ( bus ở mức 1) thì master hiểu rằng không có thiết bị slave nào tồn tại và dừng quá trình.

|  |  |  |
| --- | --- | --- |
| Parameter | Speed | Recommended (µs) |
| A | Standard | 6 |
| Overdrive | 1 |
| B | Standard | 64 |
| Overdrive | 7.5 |
| C | Standard | 60 |
| Overdrive | 7.5 |
| D | Standard | 10 |
| Overdrive | 2.5 |
| E | Standard | 9 |
| Overdrive | 1 |
| F | Standard | 55 |
| Overdrive | 7 |
| G | Standard | 0 |
| Overdrive | 2.5 |
| H | Standard | 480 |
| Overdrive | 70 |
| I | Standard | 70 |
| Overdrive | 8.5 |
| J | Standard | 410 |
| Overdrive | 40 |

Bảng 2‑1: Bảng giá trị thời gian OneWire

Chế độ hoạt động:

- Chế độ Standard - Chế độ tiêu chuẩn:

* 15.4 Kb/s
* 65 µs bit

- Chế độ Overdrive – Chế độ tốc độ nhanh:

* 125 Kb/s
* 8 µs bit

### Giao thức HTTP

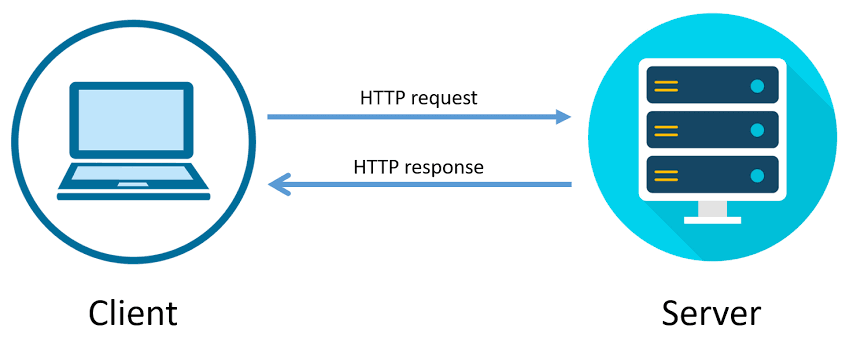
HTTP (Hypertext Transfer Protocol) là một giao thức (quy tắc truyền tin) để trao đổi thông tin giữa máy chủ Web và trình duyệt Web. Khi chúng ta thu thập thông tin trên trang chủ của Web hoặc đọc các blog, HTTP được sử dụng để giao tiếp giữa máy chủ và máy khách (người dùng).

Một trong những tính năng của HTTP là nó rất đơn giản để hoạt động. Bất cứ khi nào thông tin được trao đổi, máy khách (chẳng hạn như trình duyệt Web) đưa ra yêu cầu và máy chủ sẽ phản hồi.

Quy tắc một yêu cầu (request) thì chỉ trả lại một phản hồi (response) duy nhất. Và response cho một request sẽ luôn giống nhau trong cùng một điều kiện.

Vì HTTP có một đặc tính hoàn chỉnh và đơn giản nên bên cạnh sự tương tác giữa máy chủ Web và trình duyệt Web, nó còn được sử dụng rộng rãi để gọi các chức năng máy chủ từ điện thoại thông minh và ứng dụng hay gọi các dịch vụ giữa các máy chủ với nhau. Nó chủ yếu liên quan đến "REST API" và được sử dụng để gọi các chức năng của chương trình khi phát triển một ứng dụng.

Trong nhiều trường hợp, HTTP được sử dụng kết hợp với TCP và hiếm khi kết hợp với UDP. Số cổng mà máy chủ nhận được giao tiếp HTTP thường là cổng 80. Đối với các mục đích đặc biệt, chẳng hạn như proxy HTTP (bộ lọc nội dung hiệu suất cao, v.v.), hãy sử dụng các cổng khác với cổng 80.



Hình 2‑5: Giao thức HTTP

### Giao thức MQTT

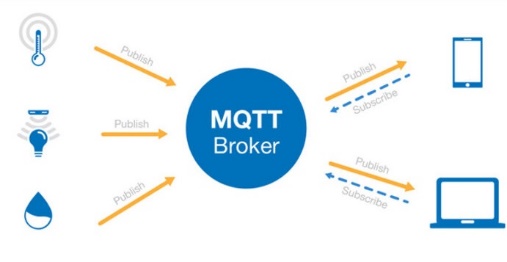
MQTT (Message Queueing Telemetry Transport) là một giao thức mạng kích thước nhỏ (lightweight), hoạt động theo cơ chế publish – subscribe (tạm dịch: xuất bản – đăng ký) theo tiêu chuẩn ISO (ISO/IEC 20922) và OASIS mở để truyền tin nhắn giữa các thiết bị.

Giao thức này hoạt động trên nền tảng TCP/IP. MQTT được thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị ở xa, các thiết bị hay vi điều khiển nhỏ có tài nguyên hạn chế hoặc trong các ứng dụng có băng thông mạng bị hạn chế.

MQTT là lựa chọn lý tưởng trong các môi trường như:

* Những nơi mà giá mạng viễn thông đắt đỏ hoặc băng thông thấp hay thiếu tin cậy.
* Khi chạy trên thiết bị nhúng bị giới hạn về tài nguyên tốc độ và bộ nhớ.
* Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M (Machine to Machine).

MQTT cũng là giao thức sử dụng trong Facebook Messenger và Amazon IoT.

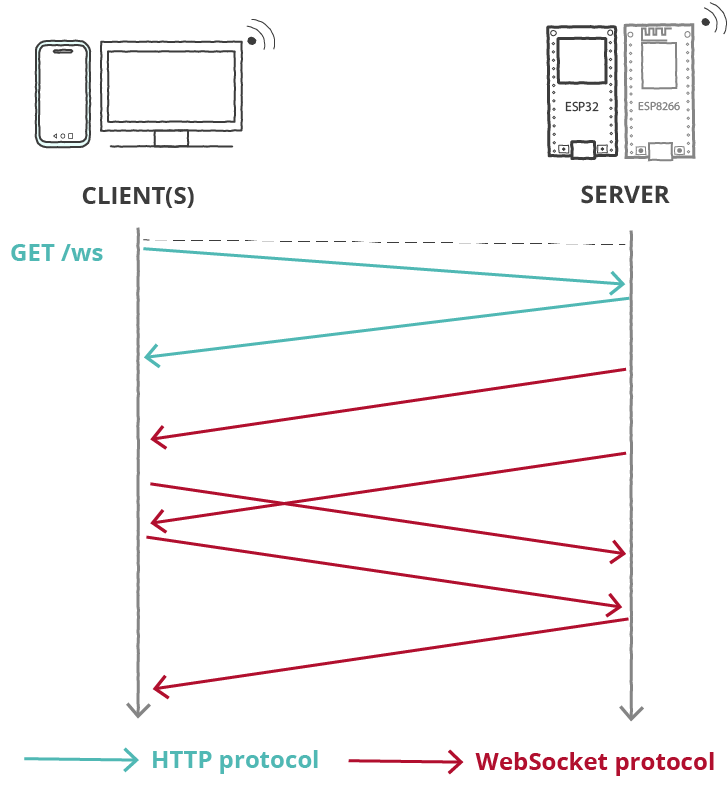


Hình 2‑6: Giao thức MQTT

### Giao thức WebSocket

WebSocket là một giao thức giao tiếp máy tính, hỗ trợ các channel giao tiếp full-duplex qua một kết nối TCP. Giao thức WebSocket được IETF chuẩn hóa RFC 6455 vào năm 2011. Hiện nay, API WebSocket trong Web IDL cũng đang được chuẩn hóa bởi W3C.

Sử dụng WebSockets, bạn có thể tạo một ứng dụng real-time đúng nghĩa như ứng dụng chat, phối hợp soạn thảo văn bản, giao dịch chứng khoán hay game online nhiều người chơi cùng lúc.



Hình 2‑7: Giao thức WebSocket

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

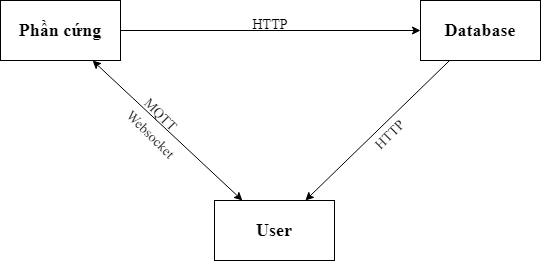
## Giới thiệu

Đề tài “hệ thống iot quản lý vườn cây thông minh” bao gồm:

* Hệ thống chính là hệ thống tự động tưới nước cho cây dựa vào điều kiện nhiệt độ, độ ẩm môi trường. Người dùng có thể điều khiển hệ thống thông qua các thiết bị có kết nối mạng.
* Hệ thống mở rộng là hệ thống bật tắt đèn dựa vào môi trường.

## Thiết kế sơ đồ khối hệ thống.

Cấu trúc hệ thống được chia làm 3 phần bao gồm Phần cứng, database, User. Ba phần này giao tiếp với nhau như hình bên dưới:

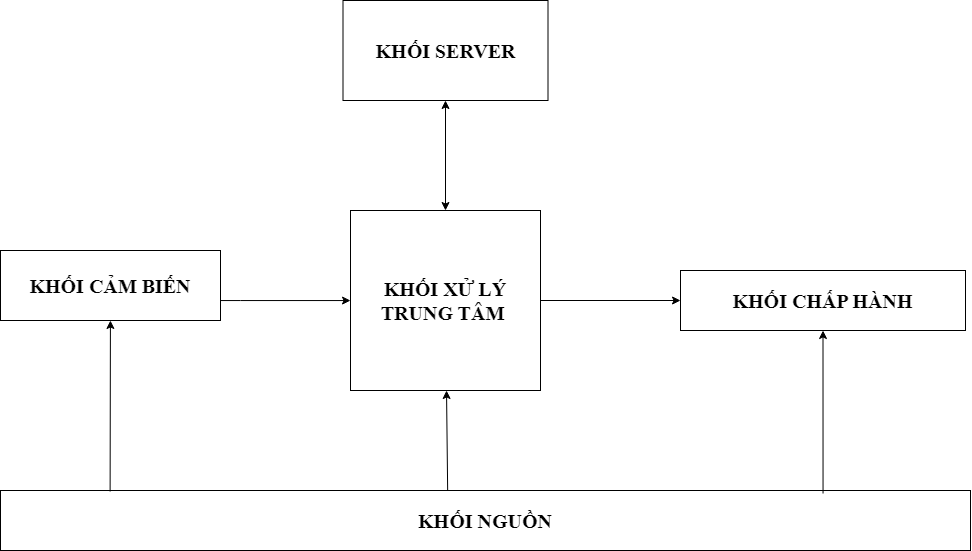


Hình ‑: Cấu trúc hệ thống

Ý nghĩa từng phần:

* Phần cứng: Phần vật lý của hệ thống
* Database: Phần server của hệ thống
* User: Phần của người dùng tương tác với hệ thống.

Hệ thống gồm 5 khối kết hợp với nhau tạo nên một hệ thống hoạt động ổn định được trình bày bên dưới:



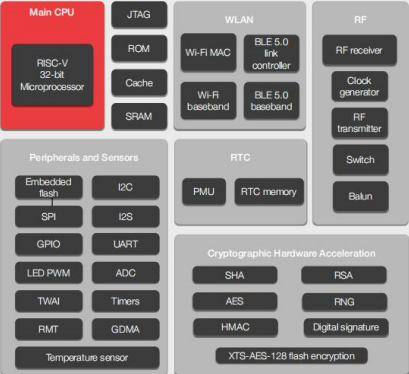
Hình 3‑2: Sơ đồ khối hệ thống

Chức năng từng khối:

* **Khối nguồn**: Cung cấp nguồn cho hệ thống bao gồm: khối cảm biến, khối xử lý trung tâm, khối chấp hành.
* **Khối xử lý trung tâm**: Thu thập dữ liệu từ các thiết bị, xử lý chúng và điều khiển khối chấp hành. Gửi và nhận các giá trị từ khối server.
* **Khối server**: Giao diện Web/Web App để hiển thị, lưu trữ dữ liệu, đòng thời cho phép người dùng thao tác, điều khiển gián tiếp hệ thống thông qua Wifi.
* **Khối cảm biến**: bao gồm các cảm biến có nhiệm vụ thu thập các thông số môi trường.
* **Khối chấp hành**: bao gồm các thiết bị mà sẽ giúp điều chỉnh các thông số,cơ cấu của khu vườn để giữ cho khu vườn luôn ở điều kiện thích hợp nhất.

## Tính toán, thiết kế và hiện thực mạch

### Khối trung tâm

Khối xử lý trung tâm bao gồm các nhiệm vụ nhận các giá trị và điều khiển các thiết bị, đồng thời còn hỗ trợ kết nối mạng, gửi và nhận các tín hiệu lên server. Vì vậy, để đáp ứng yêu cầu trên em quyết định sử dụng ESP32-C3-32S-Kit.

Hình 3‑3: Hình ảnh và CPU ESP32-C3-32S

ESP32-C3-32S-Kit được phát triển với Vi điều khiển trung tâm là ESP32-C3 SoC với công nghệ Wifi, Bluetooth 5 kết hợp với kiến trúc RISC-V mới nhất hiện nay, kit có thiết kế phần cứng, firmware và cách sử dụng tương tự Kit NodeMCU ESP8266, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CH340, thích hợp với các nghiên cứu, ứng dụng về RISC-V, Wifi, Bluetooth 5, IoT và điều khiển, thu thập dữ liệu qua mạng

Một vài thông số ESP32-C3-32S:

|  |  |
| --- | --- |
| Vi điều khiển | ESP32-C3 |
| Điện áp hoạt động | 5VDC từ cổng Micro USB |
| SPI Flash | 4MByte |
| Kích thước | 48.2 x 25.4 x 12.9mm |
| Số cổng I/O | 20 cổng |
| Bluetooth | BLE 5.0 |
| Bảo mật | WEP/WPA-PSK/WPA2-PSK |
| Wifi | STA/AP/STA+AP |

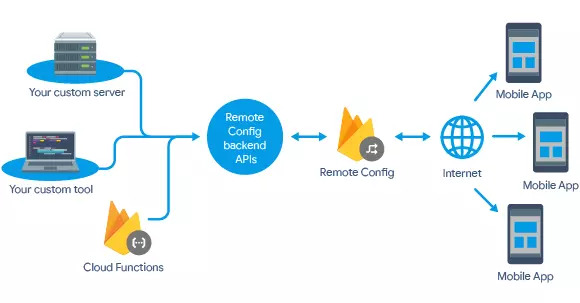
Bảng 3‑1: Một vài thông số ESP32-C3-32S

### Khối server

Khối server là một khối hoạt động độc lập, song song với khối xử lý trung tâm. Nó có nhiệm vụ theo dõi hoạt động của khu vườn và điều khiển khu vườn.

#### Firebase

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.



Hình 3‑4: Chức năng của Firebase

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.

Các tính năng chính của Firebase:

* Realtime Database
* Authentication
* Cloud Storage
* Cloud Firestore
* Hosting
* Cloud Functions
* Analytics

Ưu và nhược điểm Firebase:

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Sử dụng miễn phí và thuận tiện  Dễ sử dụng và tích hợp  Đáp ứng nhu cầu của người dùng  Cập nhật liên tục và đa nền tảng | Giới hạn về quy mô ứng dụng  Khả năng tùy chỉnh hạn chế  Bảo mật còn yếu |

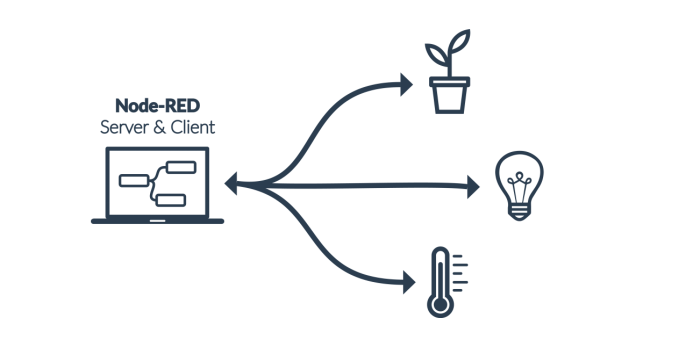
Bảng 3‑2: Ưu và nhược điểm Firebase

#### Node-RED

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 3‑5: Chức năng Node-Red

#### 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

Mosquitto có ưu và nhược điểm sau:

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Ưu điểm nổi bật của Mosquitto là 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). | 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. |

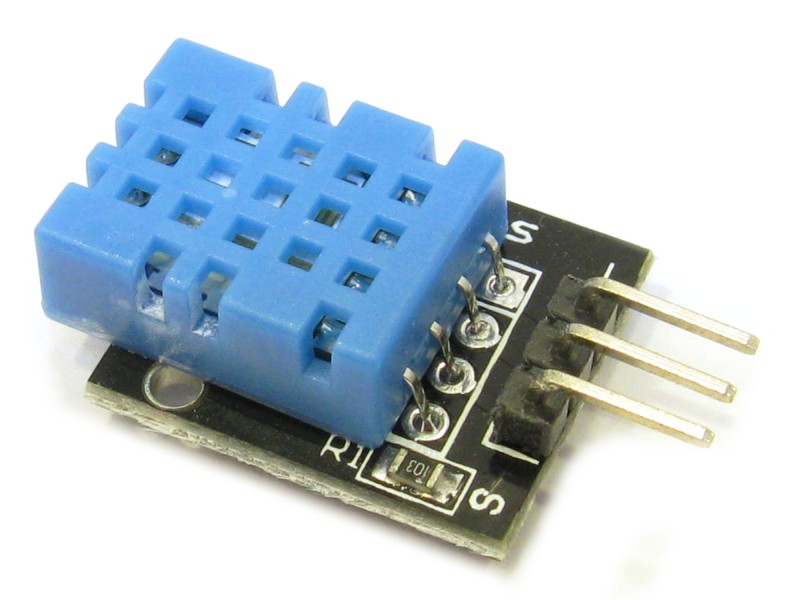
Bảng 3‑3: Ưu và nhược điểm Mosquitto

### Khối cảm biến

Khối cảm biến có nhiệm vụ thu thập các thông số của môi trường để cung cấp cho chúng cho khối xử lý trung tâm để có thông số để từ đó có những xử lý, điều chỉnh để phù hợp nhất với sự phát triển và sinh trưởng của các cây trồng trong khu vườn.

#### Cảm biến DHT11

Cảm biến DHT11 là cảm biến có chức năng đo nhiệt độ và độ ẩm. Cảm biến DHT11 sử dụng giao tiếp OneWire.



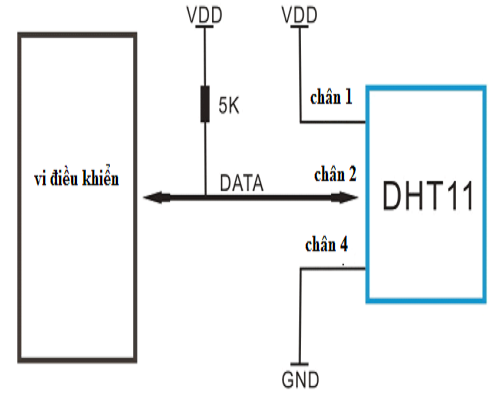
Hình 3‑6: Hình ảnh thực tế cảm biến DHT11

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.

**Thông số kỹ thuật:**

* Điện áp hoạt động : 5VDC
* Chuẩn giao tiếp: TTL, 1 wire.
* Khoảng đo độ ẩm: 20%-90% RH (sai số 5%RH)
* Khoảng đo nhiệt độ: 0-50°C (sai số 2°C)
* Tần số lấy mẫu tối đa: 1Hz (1 giây / lần)
* Kích thước : 28mm x 12mm x10mm

Cách kết nối cảm biến DHT11 với vi điều khiển:

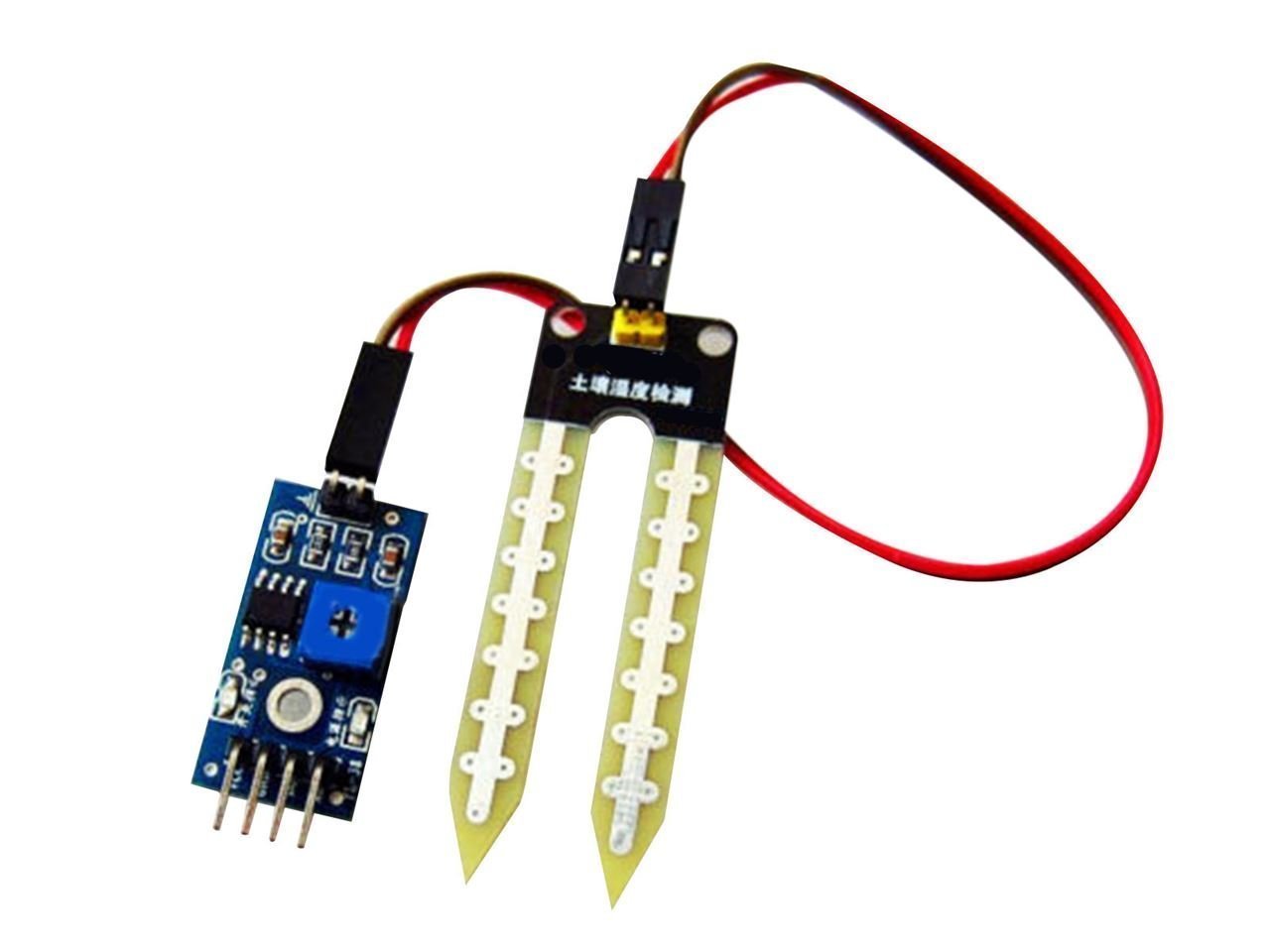


Hình 3‑7: Kết nối cảm biến DHT11 với vi điều khiển

#### Cảm Biến Độ Ẩm Đất

Cảm biến độ ẩm đất dùng để đo độ ẩm trong đất. Hai đầu đo của cảm biến được cắm vào đất để phát hiện độ ẩm. Dùng dây nối giữa cảm biến và module chuyển đổi. Thông tin về độ ẩm đất sẽ được đọc về và gửi tới module chuyển đổi.

Module chuyển đổi có cấu tạo chính gồm một IC so sánh LM393, một biến trở, 4 điện trở dán 100 ohm và 2 tụ dán. Biến trở có chức năng định ngưỡng so sánh với tín hiệu độ ẩm đất đọc về từ cảm biến. Ngưỡng so sánh và tín hiệu cảm biến sẽ là 2 đầu vào của IC so sánh LM393. Khi độ ẩm thấp hơn ngưỡng định trước, ngõ ra của IC là mức cao (1), ngược lại là mức thấp (0).



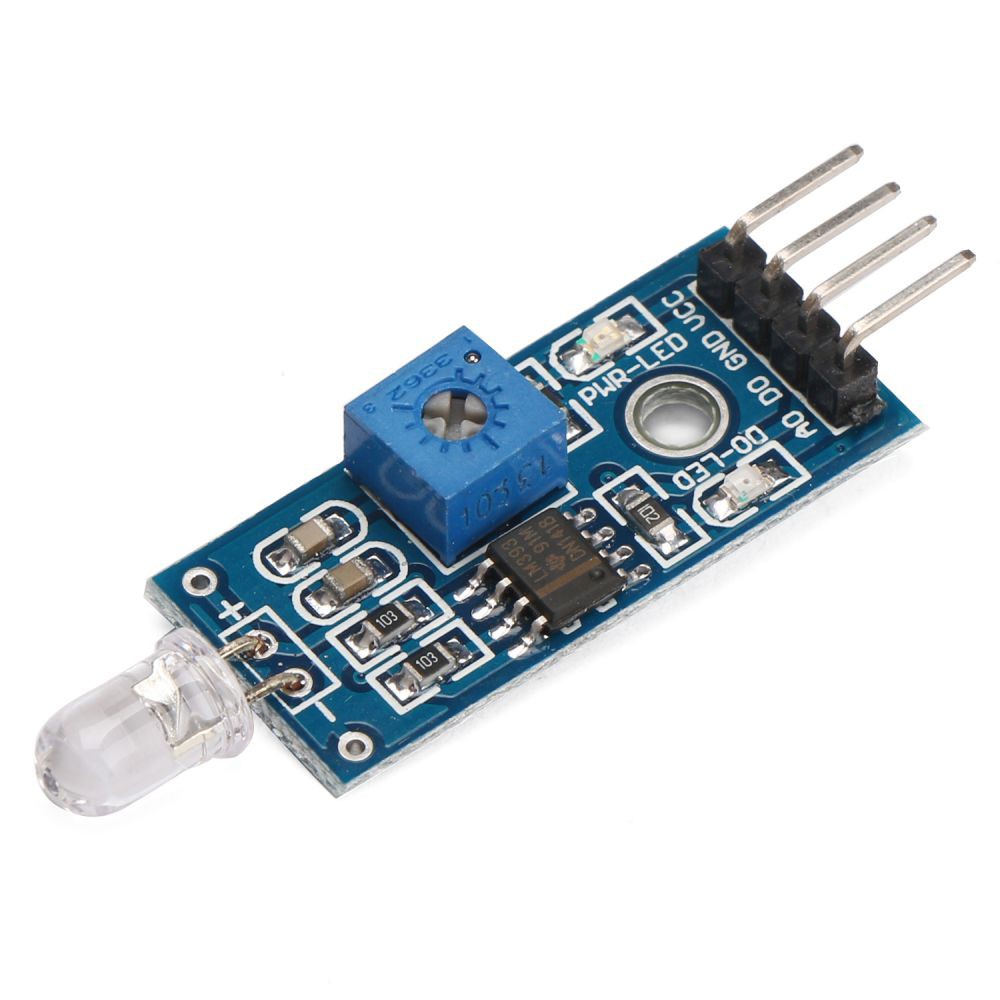
Hình 3‑8: Hình ảnh thực tế cảm biến độ ẩm đất

**Thông số kỹ thuật:**

* Điện áp hoạt động: 3.3~5VDC
* Tín hiệu đầu ra:
  + Analog: theo điện áp cấp nguồn tương ứng.
  + Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trở thông qua mạch so sánh LM393 tích hợp.
* Kích thước: 3 x 1.6cm.

#### Cảm biến ánh sáng

Cảm biến ánh sáng dùng để đo cường độ ánh sáng của môi trường. Sử dụng quang trở đề phát hiện cường độ ánh sáng.



Hình 3‑9: Hình ảnh thực tế cảm biến ánh sáng

**Thông số kỹ thuật:**

* Nguồn: 3.3 -> 5VDC
* Sử dụng Photodiod cho độ chính xác cao.
* Xuất tín hiệu Digital tinh chỉnh bằng biến trở trên mạch hoặc Analog rất dễ sử dụng.
* Kích thước: 30 x 16mm

### Khối nguồn

### Khối chấp hành

Khối cơ cấu chấp hành có nhiệm vụ là khi các thông số của môi trường đọc được từ cảm biến không phù hợp với sự sinh trưởng và phát triển của cây trồng, khối xử lý trung tâm sẽ tác động đến khối cơ cấu chấp hành để điều chỉnh các thông số của khu vườn thông qua hoạt động của các thiết bị trong khối này.

#### Máy bơm



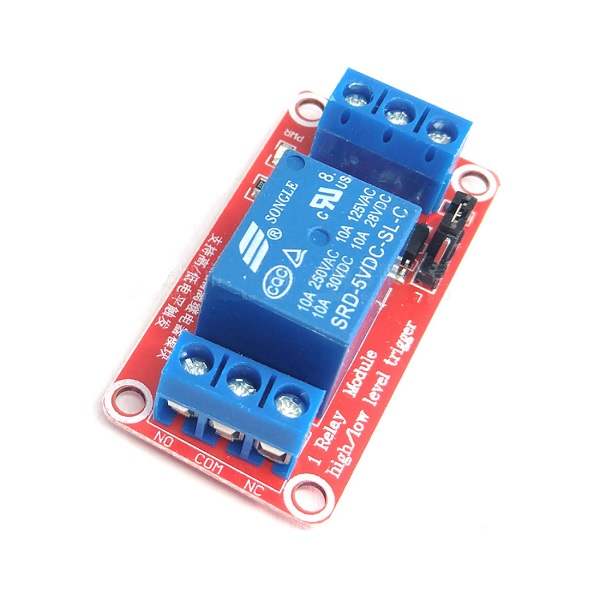
Hình ‑: Hình ảnh thực tế máy bơm

**Thông số kĩ thuật**:

* Điện áp sử dụng: 3~5VDC.
* Dòng điện sử dụng: 100~200mA.
* Lưu lượng bơm: 1.2~1.6L / 1 phút.
* Đường kính ngoài ống dẫn: 7.5mm
* Kích thước: 34 x 43 mm
* Trọng lượng: 28g

#### Đèn

#### Mạch Relay



Hình ‑: Hình ảnh thực tế mạch relay

**Thông số kỹ thuật:**

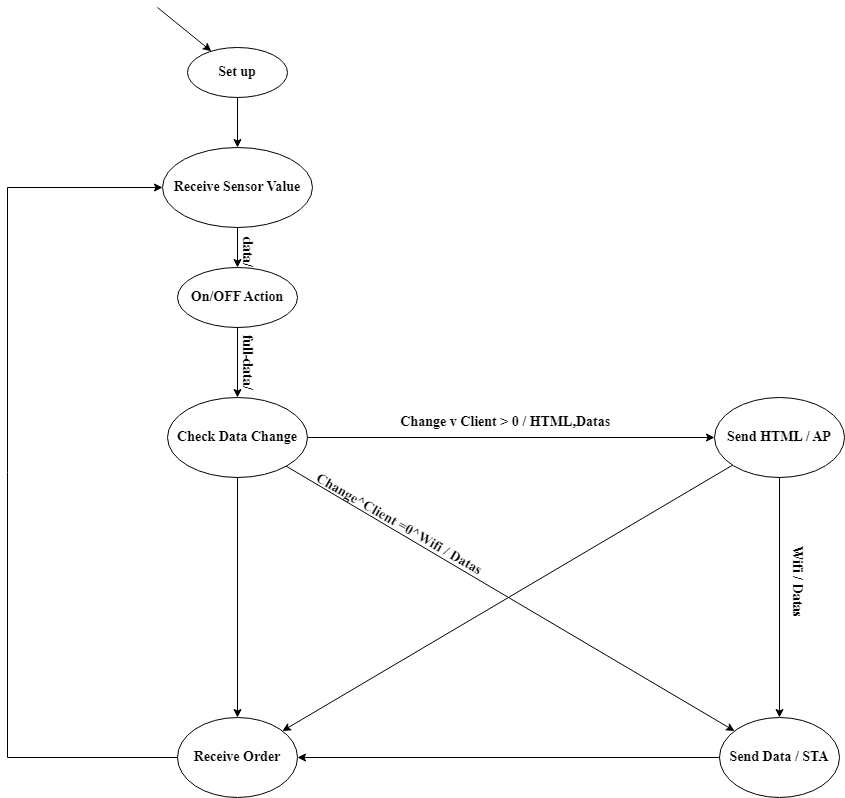
* Điện áp sử dụng: 5VDC.
* Tín hiệu kích: TTL 3.3~5VDC, mức cao High Relay đóng, mức thấp Low Relay ngắt.
* Relay tiêu thụ dòng khoảng 80mA.
* Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A (Để an toàn nên dùng cho tải có công suất <100W).
* Tích hợp Diod chống nhiễu và đèn báo tín hiệu kích.
* Kích thước: 34 x 27 x 20mm

## Lập trình hệ thống

Đề tài “Hệ Thống Iot Quản Lý Vườn Cây Thông Minh” bao gồm các tính năng:

* Tự động tưới tự động khi môi trường đạt đủ điều kiện.
* Cập nhập các thông số môi trường lên app.
* Điều khiển hệ thống bằng điện thoại (có hay không có Internet)
* Cho phép thay đổi Wifi.

### Finite State Machine



Hình ‑: FSM hệ thống

**Giải thích các state**:

* Set up: Khởi động các cảm biến, thử kết nối server.
* Receive Sensor Value: Đọc các giá trị từ cảm biến
* On/Off Action: Dựa vào data nhận từ Receive Sensor Value mà quyết định bật tắt máy bơm hoặc đèn.
* Check Data Change: So sánh full-data nhận từ On/Off Action với full-data nhận từ trước đó mà quyết định có gửi dữ liệu thay không.
* Send HTML /AP: Nếu có Client thì gửi file html theo giao thức http và duy trì kết nối bằng giao thức websocket.
* Send Data/ Sta: Nếu có kết nối Wifi, thì gửi Datas lên Cloud Database.
* Receive Order: Nhận các tính hiệu điều khiển của người dùng thông qua giao thức giao tiếp MQTT, hoặc Websocket.

**Giải thích Signal:**

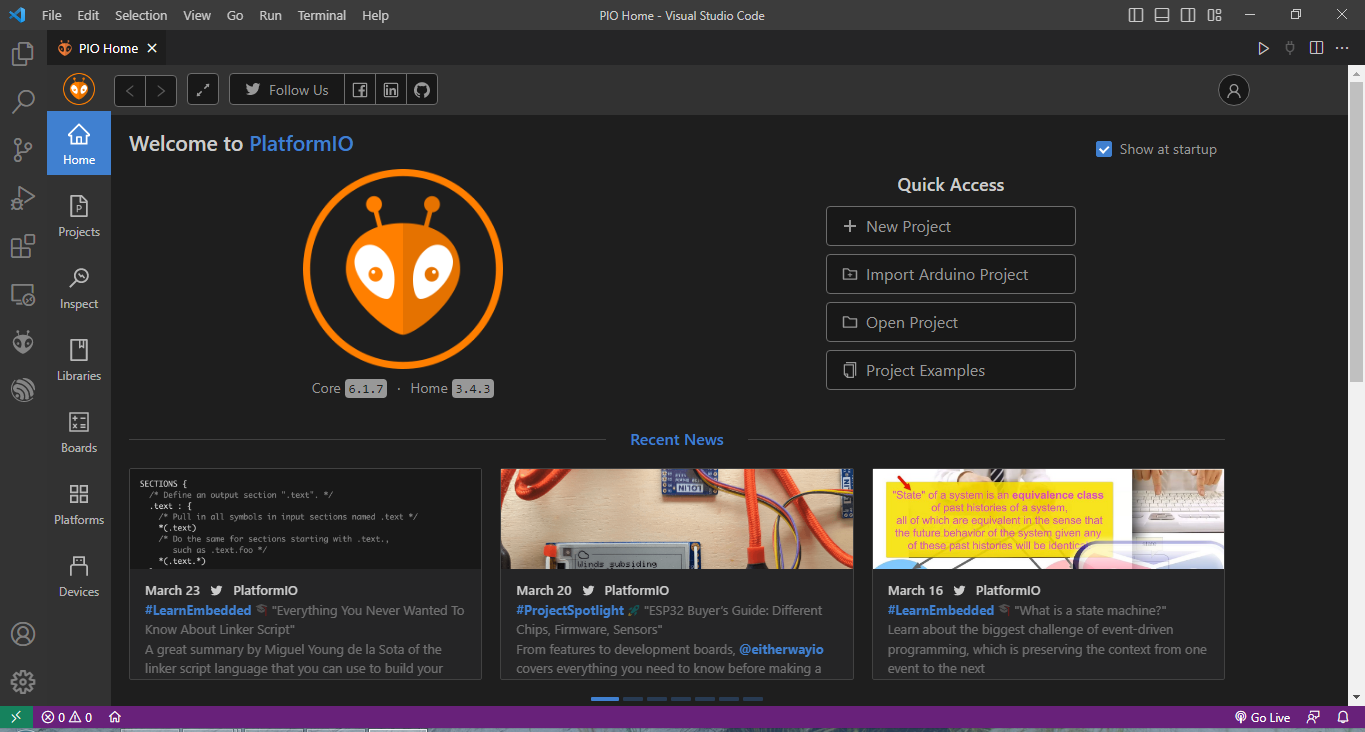
* data: Các giá trị đọc từ sensor và các tín hiệu điều khiển (nếu có)
* full-data: Các giá trị đọc từ sensor và trạng thái hiện tại máy bơm, đèn.
* Change: Tín hiệu cho sự so sánh của full-data hiện tại với full-data trước.
* Wifi: Tín hiệu cho biết có kết nối Wifi
* Datas: Các giá trị đã thay đổi của full-data hiện tại so với full-data trước.
* HTML: Tệp file html.
* Client: Đại diện cho số người đang sử dụng chế độ AP

### PlatformIO

PlatformIO là một hệ sinh thái mã nguồn mở được viết trên Python để phát triển IoT và là một IDE đa nền tảng với trình gỡ lỗi hợp nhất chạy trên Windows, Mac và Linux.

PlatformIO đi kèm với trình quản lý thư viện cho các nền tảng như Arduino hay MBED cùng với kiểm thử phần mềm và cập nhật firmware.

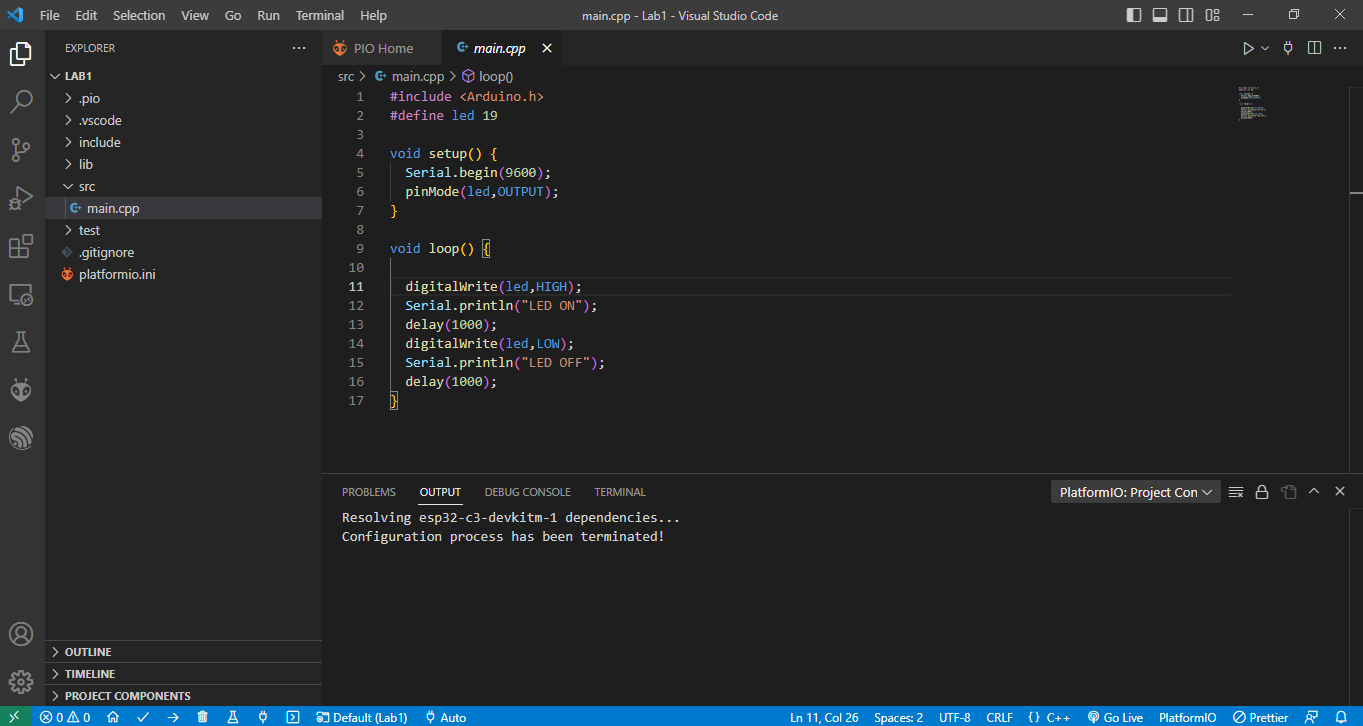
PlatformIO hỗ trợ một số platform, framework, board như Arduino, ESP32, ESP8266 và đi kèm với một số ví dụ và thư viện. Nó độc lập với nền tảng mà nó đang chạy và chỉ yêu cầu Python được cài đặt trên máy tính.



Hình ‑: Trang chủ PlatformIO trên VScode

Một số lợi thế của việc sử dụng VS Code với PlatformIO IDE so với Arduino IDE truyền thống:

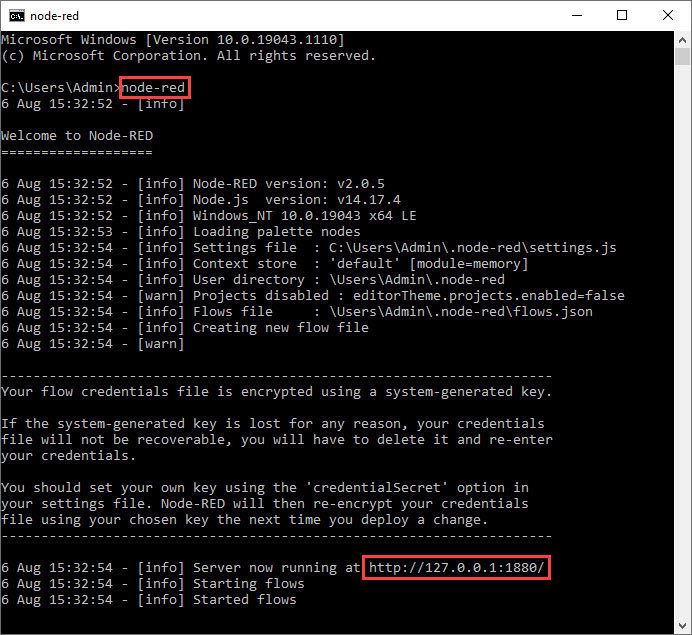
* Tự động phát hiện cổng COM của board mạch
* IntelliSense: Tự động hoàn thành code khi chúng ta đang gõ. VS Code sẽ cố gắng đoán những gì bạn muốn viết, hiển thị các khả năng khác nhau và cung cấp thông tin chi tiết về các tham số của hàm đang được gọi
* Tự động phát hiện và gạch chân các lỗi có trong mã trước khi biên dịch
* Mở nhiều tab và file cùng một lúc
* Bạn có thể ẩn đi từng phần của code



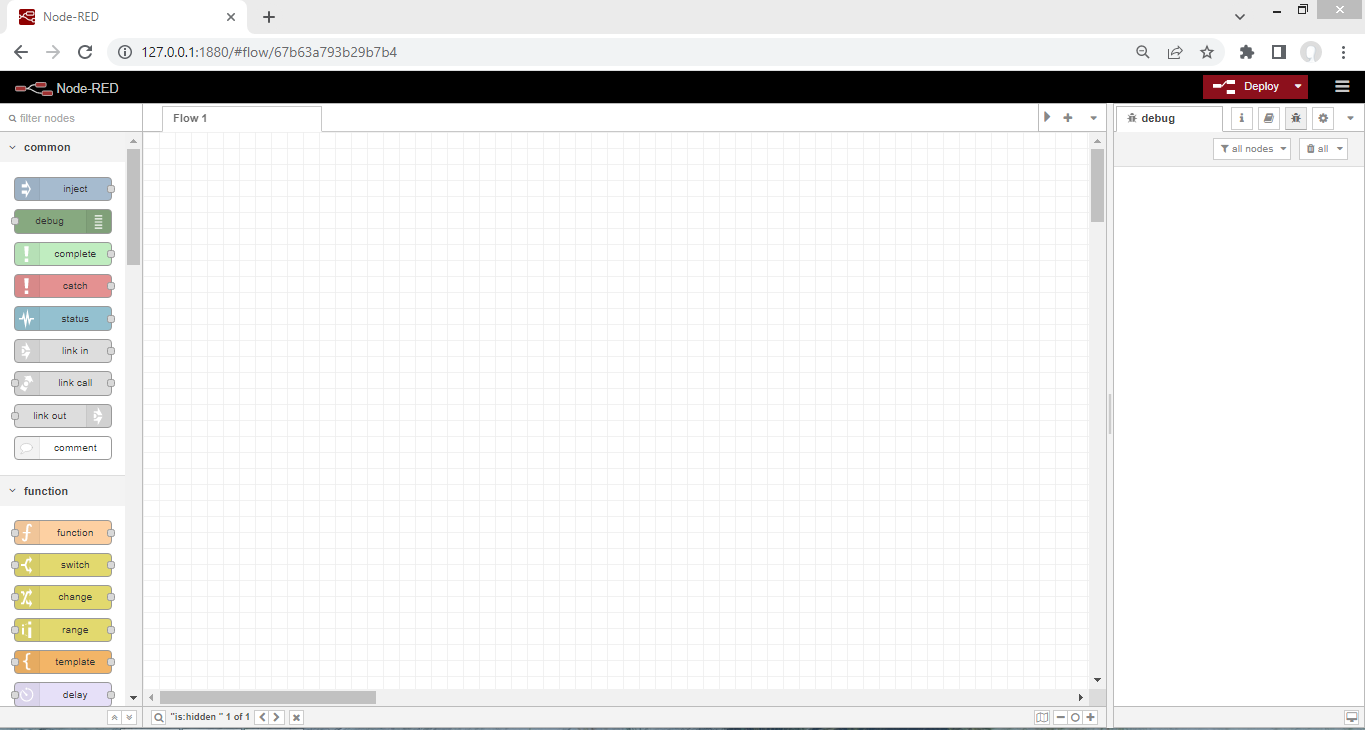
Hình ‑: Giao diện lập trình trên VScode trên nền tảng PlatformIO

### Node-Red

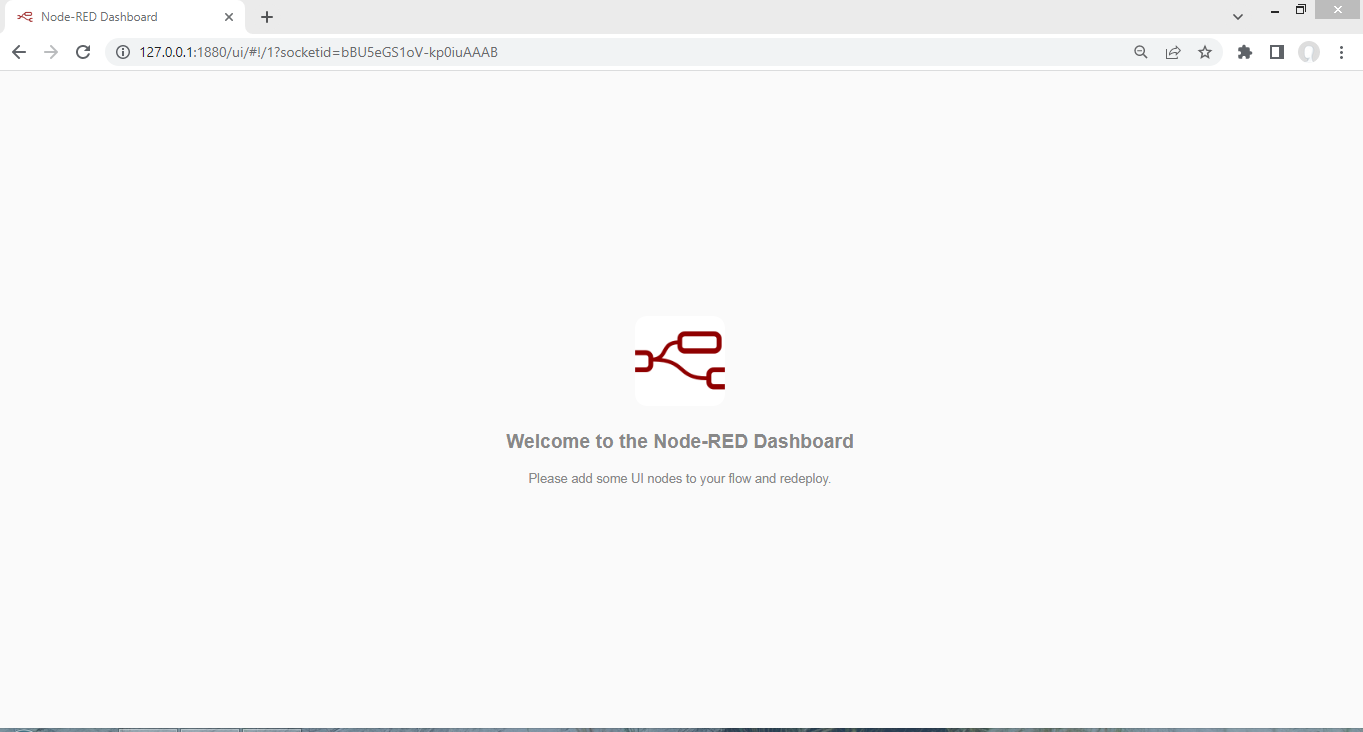
Node-RED là một công cụ lập trình kéo-thả để kết nối các thiết bị phần cứng, API và online services với nhau. Nó cung cấp một trình soạn thảo dựa trên trình duyệt giúp dễ dàng kết nối các luồng với nhau bằng cách sử dụng một loạt các Node trong bảng màu (palette) có thể được triển khai chỉ bằng một cú nhấp chuột.



Hình ‑: Khởi chạy Node-Red bằng Terminal



Hình ‑: Giao diện sử dụng Node-Red



Hình ‑: Giao diện Dashboard của Node-Red

# KẾT QUẢ THỰC NGHIỆM

## Kết quả

### Tổng quan

Sau một thời gian tìm hiểu, nghiên cứu các đề tài, tài liệu liên quan, tìm hiểu qua mạng, tổng hợp lại các kiến thức đã học. Em đã làm được 1 phần của đồ án.

Sau đề tài này, em đã nghiên cứu là tích luỹ thêm nhiều hiểu biết, kiến thức mới như:

* Hiểu biết sâu hơn về sử dụng và các tính năng của ESP32 như giao tiếp, kết nối mạng, giao tiếp với loại module khác như: dht11, cảm biến ánh sáng, cảm biến độ ẩm đất, động cơ bơm nước, đèn.
* Tìm hiểu và nắm bắt được cách lập trình server – giao diện người dùng.
* Có khả năng phát hiện cảm biến lỗi.

### Các tính năng của hệ thống

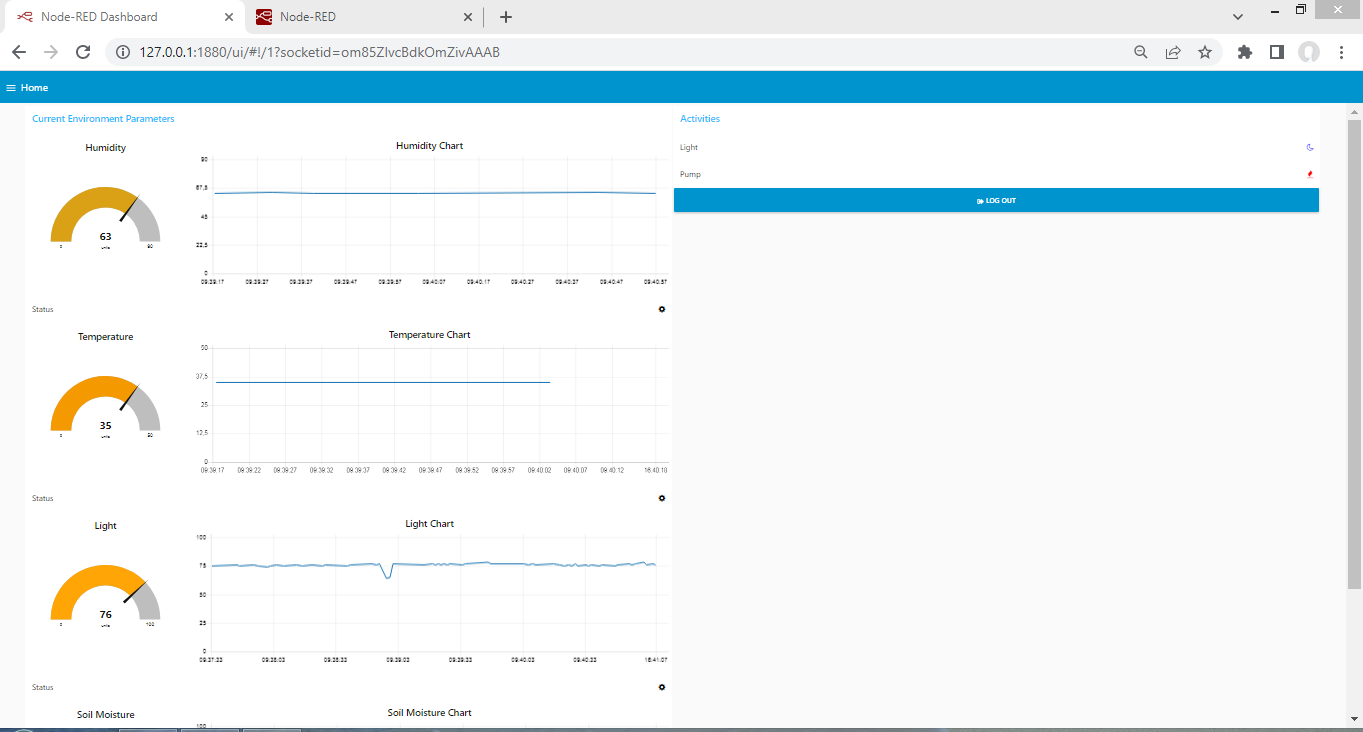
Các tính năng của hệ thống bao gồm:

* Tự động tưới tự động khi môi trường đạt đủ điều kiện.
* Cập nhập các thông số môi trường lên app ở mọi khoảng cách
* Điều khiển hệ thống bằng điện thoại (có hay không có Internet)
* Cho phép thay đổi Wifi.

### Giao diện người dùng

#### Giao diện người dùng khi ở xa

Giao diện này được thiết kế để cho người dùng có thể sử dụng khi không có ở gần khu vườn, sử dụng kết nối Internet và Realtime database làm trung gian.



Hình ‑: Giao diện người dùng khi ở xa

#### Giao diện người dùng khi ở gần

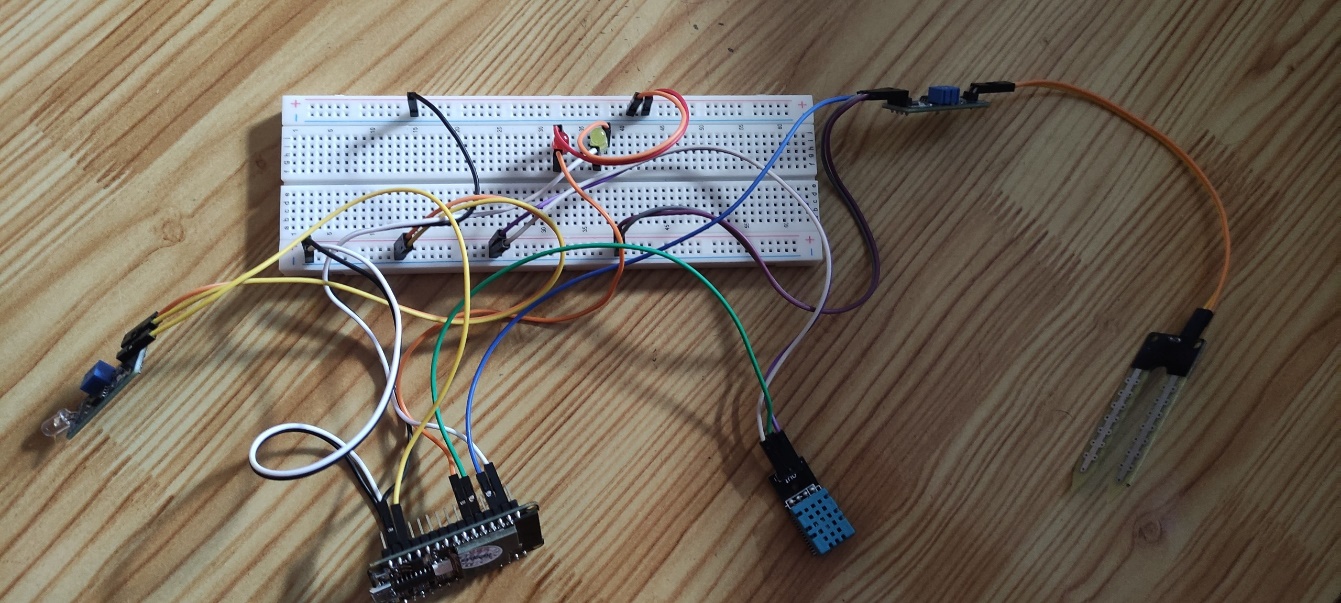
Giao diện này được thiết kế khi người dùng ở gần khu vườn. Sử dụng chính ESP32 làm cục phát Wifi, người dùng sẽ truy cập vào IP của ESP32 để sử dụng



Hình ‑: Giao diện người dùng khi ở gần

### Phần cứng

Hiện tại phần cứng đồ án đang sử dụng là mạch cắm, máy bơm và đèn được đại diện bằng led, cụ thể hơn led vàng đại diện cho máy bơm, led đỏ đại diện cho đèn.



Hình ‑: Phần cứng của hệ thống

### Demo đồ án

Demo đồ án (YouTube): <https://youtu.be/roktIgvt7V8>

## Nhận xét, đánh giá

* Mô hình hoạt động tương đối ổn định, hiếm khi bị crash.
* Chưa có hệ thống nguồn độc lập.
* Chỉ có thể tương tác với hệ thống thông qua thiết bị có khả năng kết nối mạng.
* Mệnh lệnh của hệ thống tự động cao ngang với mệnh lên của người dùng.
* Đôi lúc có tình trạng không đồng bộ dữ liệu giữa hệ thống và người dùng.
* Mô hình chưa có tính thẩm mĩ.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

## Hướng phát triển

TÀI LIỆU THAM KHẢO

Theo chuẩn IEEE