HỌC VIỆN KỸ THUẬT MẬT MÃ



**KHOA CÔNG NGHỆ THÔNG TIN**

BÁO CÁO MÔN HỌC

**THIẾT KẾ HỆ THỐNG NHÚNG**

**Đề tài:**

**CHẬU CÂY THÔNG MINH**

Giảng viên hướng dẫn: ThS. Lê Thị Hồng Vân

Lê Thành Vinh

Sinh viên thực hiện: Lê Phi Hà - CT050215

Cấn Quang Ngọc - CT050237

Đào Anh Tuấn - CT050253

Đào Việt Hưng - CT050224

Hà Nội, 7-2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

Hà Nội, ngày 13 tháng 7 năm 2024 Giáo viên hướng dẫn

ThS. Lê Thị Hồng Vân

Lê Thành Vinh

# LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và vi điều kiển PIC ngày càng thông dụng và hoàn thiện hơn, nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người bắt đầu tìm tòi về vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi vê vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điệ tử.

Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Ngôn ngữ lập trình trên nền Java lại vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Trên cơ sở kiến thức đã học trong môn học: Tin học đại cương, Điện tử tương tự và số…cùng với những hiểu biết về các thiết bị điện tử, chúng em đã quyết định thực hiện đề tài: **Nghiên cứu phát triển hệ thống tưới cây tự động sử dụng hệ điều hành thời gian thực FREERTOS** do ThS. Lê Thị Thanh Vân và Lê Thành Vinh hướng dẫn. Đề tài gồm các nội dung sau:

Chương 1. Khảo sát và xây dựng bài toán

Chương 2. Xây dựng hệ thống (Thiết kế/ phát triển)

Chương 3. Thử nghiệm và đánh giá hệ thống

Nhóm sinh viên thực hiện:

Lê Phi Hà

Cấn Quang Ngọc

Đào Anh Tuấn

Đào Việt Hưng

Mục Lục

[LỜI NÓI ĐẦU 1](#_Toc171779341)

[DANH MỤC HÌNH VẼ 4](#_Toc171779342)

[CHƯƠNG 1. KHẢO SÁT VÀ XÂY DỰNG BÀI TOÁN 1](#_Toc171779343)

[1.1 Tổng quan 1](#_Toc171779344)

[1.1.1. Đặt vấn đề 1](#_Toc171779345)

[1.1.2. Mục tiêu 2](#_Toc171779346)

[1.2 Khảo sát các hệ thống hiện tại 3](#_Toc171779347)

[1.2.1 Đặc điểm của các hệ thống tưới cây hiện tại 3](#_Toc171779348)

[1.2.2 Thành phần trong một hệ thống tưới cây thông minh 5](#_Toc171779349)

[1.3 Khảo sát ứng dụng của hệ điều hành thời gian thực (FreeRTOS) trong các hệ thống IOT 7](#_Toc171779350)

[1.3.1. Giới thiệu về FreeRTOS 7](#_Toc171779351)

[1.3.2. FreeRTOS trong hệ thống IOT 8](#_Toc171779352)

[1.4 Công cụ sử dụng 11](#_Toc171779353)

[1.4.1 Proteus 11](#_Toc171779354)

[1.4.2 Arduino IDE 11](#_Toc171779355)

[1.4.3 ESP32 12](#_Toc171779356)

[1.4.4 Xampp và cơ sở dữ liệu phpMyAdmin 12](#_Toc171779357)

[1.5 Xây dựng bài toán 12](#_Toc171779358)

[1.5.1. Yêu cầu đặt ra 12](#_Toc171779359)

[1.5.2. Chức năng của dự án 13](#_Toc171779360)

[1.6 Tổng kết chương 14](#_Toc171779361)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ 15](#_Toc171779362)

[2.1 Mô hình tổng thể 15](#_Toc171779363)

[2.2 Thiết kế phần cứng 18](#_Toc171779364)

[2.3. Thiết kế phần mềm 20](#_Toc171779365)

[2.3.1. Lập trình freeRTOS Arduino 20](#_Toc171779366)

[2.3.2. Sơ đồ hoạt động của hệ thống 22](#_Toc171779367)

[2.4. Code Web 23](#_Toc171779368)

[2.5. Code ESP 25](#_Toc171779369)

[2.5.1 Khai báo các thư viện, chân tín hiệu 25](#_Toc171779370)

[2.5.2 Hàm setup() khởi tạo 26](#_Toc171779371)

[2.5.3. Hàm loop() để nhận và xử lý tín hiệu cho các cảm biến hệ thống 26](#_Toc171779372)

[2.5.4. Hàm gethttp() 27](#_Toc171779373)

[2.6 Code Arduino 30](#_Toc171779374)

[2.7 Tổng kết chương 2 33](#_Toc171779375)

[CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG 34](#_Toc171779376)

[3.1 Thử nghiệm tính năng 34](#_Toc171779377)

[3.2 Đánh giá hệ thống 38](#_Toc171779378)

[3.3 Nhận xét đánh giá 38](#_Toc171779379)

[3.4 Tổng kết chương 3 38](#_Toc171779380)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc171779381)

[4.1. Kết quả đạt được 39](#_Toc171779382)

[4.2. Hạn chế 39](#_Toc171779383)

[4.3. Hướng phát triển 39](#_Toc171779384)

# DANH MỤC HÌNH VẼ

[Hình 1. 1 Hệ thống tứi nước tự động cho cây trồng 1](#_Toc171777412)

[Hình 1. 2 Hệ thống tưới cây phun xương 5](#_Toc171777413)

[Hình 1. 3 So sánh sự khác biệt của cây trồng khi độ ẩm khác nhau 6](#_Toc171777414)

[Hình 1. 5 hệ thống tưới cây thông minh 18](#_Toc171777415)

[Hình 1. 6 Sơ đồ lập trình freeRTOS Arduino 19](#_Toc171777416)

[Hình 2.1 Lưu đồ thuật toán sơ đồ hệ thống tưới cây tự động 22](#_Toc171777417)

[Hình 3. 1 Mô hình thực tế 33](#_Toc171777418)

[Hình 3. 2 Mô hình thực tế 34](#_Toc171777419)

[Hình 3. 3 Giao diện ứng dụng 35](#_Toc171777420)

[Hình 3. 4 Giá trị thu được 36](#_Toc171777421)

# CHƯƠNG 1. KHẢO SÁT VÀ XÂY DỰNG BÀI TOÁN

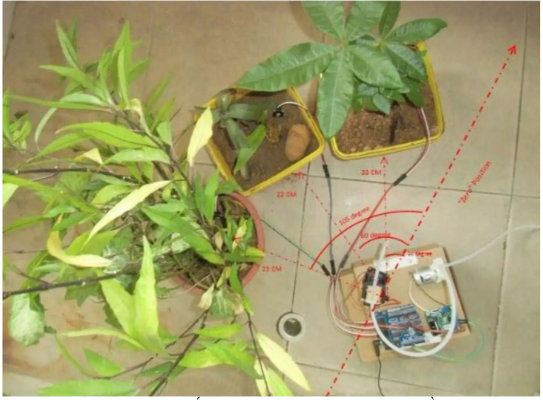
Chương 1 giới thiệu tổng quan về bài toán nhận diện đo độ ẩm, ánh sáng, các khảo sát về bài toán thông qua các chỉ số. Qua đó giới thiệu các công nghệ, công cụ và các thiết bị sử dụng để giải quyết bài toán “Chậu cây thông minh”

## 1.1. Tổng quan

### 1.1.1. Đặt vấn đề

Tại một số địa phương đã canh tác một số loại cây, hoa, rau có giá trị kinh tế cao, tuy nhiên hiện nay vẫn còn ít các đơn vị nào tiến hành nghiên cứu thiết kế các mô hình tự động đáp ứng điều kiện kinh tế, môi trường của nước ta.

Thực tế trong cuộc sống ngày càng bân rộn, nhiều người vẫn có thú vui là trồng những cây cảnh, vườn rau trong không gian trống của nhà mình như sân thượng, ban côn. Tuy nhiên, trong những lúc bạn bận các công việc hằng ngày thì những cây cảnh và vường hoa ử nhà sẽ không ai tưới nước.



Hình 1. 1 Hệ thống tứi nước tự động cho cây trồng

Ngoài phương pháp tưới thủ công, các phương pháp tưới còn lại đều cần được thiết kế, tính toán các thông số cho phù hợp. Nhìn chung, các phương pháp tưới có dung ống đều có nguyên lý tính toán trên cơ sở môn học cấp thoát nước và thuỷ lực đường ống, đó là: xác định diện thích tưới, nguồn nước, nhu cầu nước tưới phù hợp với từng loại cây trồng, diện tích, địa hình vùng tưới. Từ các thông số này, ta sẽ tính toán đường kính ống chính, ống phụ, ống nhánh, vận tốc nucows chảy trong ống, áp lực nước trong ống; tính toán chiều dài của các loại ống, các chi tiết nối (co, tê, van, lơi vv…), số lượng các bét phun, bét đế chân, ống dẫn dến gốc vv…và cuối cùng là lập bảng tổng hợp số lượng các loại vật tư, tính toán chí ít mua vật tư, tiền công xây lắp vv…

Từ những thực tiễn đó, kích thích sự tò mò, tìm hiểu của con người, là nguồn cảm hứng cho các ứng dụng thông minh ra đời. Là sinh viên khoa công nghệ thông tin của Học Viện Kỹ Thuật Mật Mã, với những kiến thức đã học cùng với mong muốn được tìm hiểu và được thử sức thiết kế một đề tài hay và thú vị trong việc thực thi các nhu cầu của con người thông qua mạng Internet. Đề tài “ **Nghiên cứu phát triển hệ thống tưới cây tự động sử dụng hệ điều hành thời gian thực FREERTOS** ” mà chúng em chọn cho đồ án môn học là sẽ giúp ích cho con người về việc quan sát, kiểm tra từ đó đưa ra tín hiệu điều khiển hoặc thực thi một nhu cầu nào đó của chính người sử dụng cụ thể áp dụng trong nông nghiệp có tính quy mô lớn.

### Mục tiêu

Hệ thống “Tưới cây tự động sử dụng hệ điều hành thời gian thực FREERTOS” là một hệ thống thu thập các giá trị của cảm biến. Khi hệ thống độ ẩm đất không đủ sẽ tiến hành bơm nước cho cây, có khả năng tự động điều khiển chậu cây xoay theo hướng sáng.

Hệ thống sẽ giúp một phần nào đó việc chăm sóc cây trông hàng ngày hoàn toàn tự động thông qua ứng dụng điều khiển trên điện thoại.

## Khảo sát các hệ thống hiện tại

Các hệ thống tưới cây thông minh hiện nay bao gồm tưới nhỏ giọt, tưới phun mưa và tưới ngầm. Mỗi loại hệ thống đều có những đặc điểm riêng biệt phục vụ cho các loại cây trồng và điều kiện đất khác nhau.

### Đặc điểm của các hệ thống tưới cây hiện tại

Hiện nay, có ba loại hệ thống tưới cây chính được sử dụng rộng rãi: hệ thống tưới nhỏ giọt, hệ thống tưới phun mưa, và hệ thống tưới ngầm.

**Hệ thống tưới nhỏ giọt:**

* **Mô tả:** Hệ thống tưới nhỏ giọt cung cấp nước trực tiếp đến gốc cây thông qua các ống dẫn nhỏ và các đầu tưới nhỏ giọt. Nước được phát ra từ từ, đảm bảo rằng cây nhận được lượng nước cần thiết mà không bị lãng phí.
* **Ưu điểm:**
  + Tiết kiệm nước: Nước được cung cấp trực tiếp đến gốc cây, giảm thiểu sự bốc hơi và thấm mất nước.
  + Hiệu quả cao: Đảm bảo rằng cây luôn nhận được lượng nước ổn định và đủ để phát triển.
  + Giảm cỏ dại: Bởi vì nước chỉ được cung cấp cho cây trồng, cỏ dại xung quanh không nhận được nước và không phát triển mạnh.
* **Nhược điểm:**
  + Chi phí ban đầu cao: Hệ thống cần nhiều ống dẫn và đầu tưới, làm tăng chi phí lắp đặt ban đầu.
  + Bảo trì phức tạp: Đầu tưới nhỏ giọt có thể bị tắc nghẽn do cặn bẩn trong nước, cần bảo trì thường xuyên.

**Hệ thống tưới phun mưa:**

* **Mô tả:** Hệ thống tưới phun mưa phun nước lên không trung dưới dạng những hạt mưa nhỏ, mô phỏng giống như mưa tự nhiên. Nước sau đó rơi xuống và thấm vào đất.
* **Ưu điểm:**
  + Dễ lắp đặt: Hệ thống phun mưa có thể được lắp đặt nhanh chóng và dễ dàng trên các diện tích rộng.
  + Phân phối nước đồng đều: Nước được phân phối đều trên toàn bộ diện tích cây trồng.
* **Nhược điểm:**
  + Mất nước do bốc hơi: Khi nước được phun lên không trung, một phần nước sẽ bốc hơi trước khi đến đất.
  + Tiêu tốn năng lượng: Hệ thống cần áp suất cao để phun nước, do đó tiêu tốn nhiều năng lượng hơn.
  + Ảnh hưởng bởi gió: Gió có thể làm thay đổi hướng và phạm vi phun nước, dẫn đến phân phối nước không đều.

**Hệ thống tưới ngầm:**

* **Mô tả:** Hệ thống tưới ngầm sử dụng các ống dẫn nước được đặt dưới bề mặt đất để cung cấp nước trực tiếp vào vùng rễ cây.
* **Ưu điểm:**
  + Tiết kiệm nước: Nước được cung cấp trực tiếp đến rễ cây, giảm thiểu sự bốc hơi và thấm mất nước.
  + Hiệu quả cao: Đảm bảo rằng rễ cây luôn nhận được lượng nước ổn định và đủ để phát triển.
  + Giảm cỏ dại: Giống như hệ thống tưới nhỏ giọt, tưới ngầm chỉ cung cấp nước cho cây trồng, hạn chế sự phát triển của cỏ dại.
* **Nhược điểm:**
  + Chi phí ban đầu cao: Lắp đặt hệ thống tưới ngầm đòi hỏi kỹ thuật cao và chi phí lắp đặt ban đầu lớn.
  + Bảo trì phức tạp: Hệ thống ngầm khó phát hiện và sửa chữa khi xảy ra sự cố.

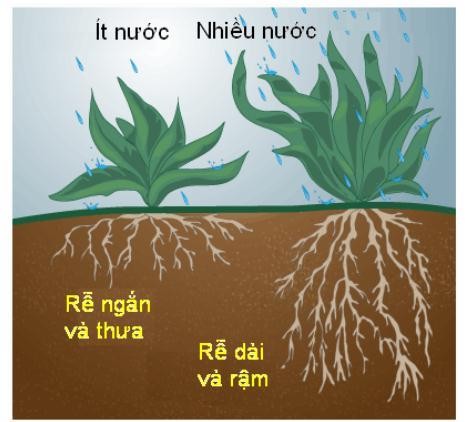


Hình 1. 2 Hệ thống tưới cây phun xương

### Thành phần trong một hệ thống tưới cây thông minh

Các hệ thống tưới cây thông minh hiện nay thường bao gồm các thành phần chính như sau:

* **Cảm biến độ ẩm đất:** Được đặt trong đất để đo lường độ ẩm của đất. Dữ liệu từ cảm biến này giúp hệ thống quyết định khi nào cần tưới nước.
* **Cảm biến nhiệt độ:** Đo lường nhiệt độ môi trường xung quanh. Nhiệt độ có thể ảnh hưởng đến lượng nước cần thiết cho cây, do đó cảm biến này giúp điều chỉnh hệ thống tưới cho phù hợp.
* **Cảm biến ánh sáng:** Đo cường độ ánh sáng mặt trời. Ánh sáng ảnh hưởng trực tiếp đến quá trình quang hợp của cây, và từ đó ảnh hưởng đến nhu cầu nước của cây.
* **Bộ điều khiển trung tâm:** Đây là "bộ não" của hệ thống, nhận và xử lý dữ liệu từ các cảm biến, sau đó điều khiển các thiết bị tưới nước.
* **Hệ thống tưới nước:** Bao gồm các ống dẫn nước và các đầu tưới nước (như đầu tưới nhỏ giọt hoặc béc phun mưa), giúp cung cấp nước đến cây trồng

[](https://tincay.com/wp-content/uploads/2017/02/t%C6%B0%E1%BB%9Bi-n%C6%B0%E1%BB%9Bc-c%C3%A2y-tr%E1%BB%93ng.jpg)Cây trồng cung cấp đầy đủ nước (độ ẩm thích hợp) sẽ có bộ rễ dài và sâu, vươn ra theo các chiều trong đất. Ngược lại cây nếu thiếu nước, bộ rễ cây sẽ ngắn và thưa.

Hình 1. 3 So sánh sự khác biệt của cây trồng khi độ ẩm khác nhau

Lượng nước tưới ảnh hưởng đến sự phát triển của bộ rễ – Ảnh hưởng của độ ẩm đất đến cây trồng

Cây yêu cầu đất phải có độ ẩm thích hợp. Đảm bảo sức giữ nước của đất luôn luôn bé hơn sức hút nước của cây và đất có tính thấm nước tốt để độ ẩm đó nhanh chóng chuyển đến cung cấp cho cây trồng. Độ ẩm đất thích hợp trong tầng đất bộ rễ hoạt động thay đổi theo yêu cầu sinh lý của từng loại cây trồng. Qua các thời kỳ sinh trưởng khác nhau. đối với cây trồng cạn.

Giới hạn trên của độ ẩm thích hợp thường trùng với độ chứa ẩm tối đa của đất. Phụ thuộc vào thành phần cơ giới và kết cấu đất, nằm trong phạm vi từ 70 – 85%. Giới hạn dưới thích hợp dao động xung quanh độ ẩm 60 – 70% độ chứa ẩm tối đa của đất.

**Ưu nhược điểm của các hệ thống tưới cây thông minh hiện tại**

**Ưu điểm:**

* **Tự động hóa hoàn toàn:** Hệ thống có thể tự động tưới nước mà không cần sự can thiệp của con người, giúp tiết kiệm thời gian và công sức.
* **Tiết kiệm nước:** Nhờ vào việc đo lường và kiểm soát chính xác lượng nước cung cấp, hệ thống giúp tiết kiệm nước một cách hiệu quả.
* **Tăng hiệu suất trồng trọt:** Cây trồng được cung cấp nước đầy đủ và ổn định, giúp tăng năng suất và chất lượng sản phẩm.
* **Giảm công lao động:** Hệ thống tự động hóa giúp giảm thiểu công việc thủ công, giảm chi phí lao động.
* **Khả năng giám sát và điều khiển từ xa:** Nhiều hệ thống hiện nay được kết nối với ứng dụng di động hoặc web, cho phép người dùng giám sát và điều khiển hệ thống từ xa.

**Nhược điểm:**

* **Chi phí đầu tư ban đầu cao:** Hệ thống tưới cây thông minh đòi hỏi các thiết bị và công nghệ cao, dẫn đến chi phí lắp đặt ban đầu lớn.
* **Yêu cầu bảo trì và vận hành kỹ thuật:** Hệ thống cần được bảo trì định kỳ để đảm bảo hoạt động ổn định. Các thiết bị cảm biến và điều khiển có thể gặp trục trặc và cần sự can thiệp của kỹ thuật viên.
* **Phụ thuộc vào nguồn điện và kết nối mạng:** Hệ thống cần nguồn điện ổn định và kết nối mạng để hoạt động hiệu quả. Khi gặp sự cố về điện hoặc mạng, hệ thống có thể không hoạt động đúng chức năng.
* **Khả năng bị tác động bởi yếu tố môi trường:** Các thiết bị cảm biến và ống dẫn nước có thể bị hư hỏng do các yếu tố môi trường như nhiệt độ cao, độ ẩm, và côn trùng.

## Khảo sát ứng dụng của hệ điều hành thời gian thực (FreeRTOS) trong các hệ thống IOT

### 1.3.1. Giới thiệu về FreeRTOS

Hệ điều hành thời gian thực (RTOS) đóng một vai trò quan trọng trong việc quản lý các nhiệm vụ và tài nguyên trong hệ thống IoT. FreeRTOS là một trong những RTOS phổ biến nhất, được thiết kế để hoạt động trên các vi điều khiển nhỏ với tài nguyên hạn chế. Trong phần này, chúng ta sẽ tìm hiểu về FreeRTOS và cách nó được ứng dụng trong các hệ thống IoT, đặc biệt là hệ thống tưới cây thông minh.

FreeRTOS là một hệ điều hành thời gian thực mã nguồn mở, được thiết kế đặc biệt cho các hệ thống nhúng và IoT. FreeRTOS cung cấp các tính năng cần thiết để quản lý tác vụ, bộ nhớ, và tài nguyên một cách hiệu quả. Một số đặc điểm chính của FreeRTOS bao gồm:

* **Nhẹ và nhanh:** FreeRTOS có dung lượng nhỏ và yêu cầu tài nguyên thấp, phù hợp với các vi điều khiển có bộ nhớ hạn chế.
* **Quản lý đa nhiệm:** FreeRTOS cho phép chạy nhiều tác vụ cùng một lúc, với cơ chế lập lịch ưu tiên đảm bảo các tác vụ quan trọng được thực hiện kịp thời.
* **Đồng bộ hóa và giao tiếp:** FreeRTOS cung cấp các cơ chế đồng bộ hóa và giao tiếp giữa các tác vụ, như hàng đợi (queue), semaphore, và mutex.
* **Hỗ trợ đa nền tảng:** FreeRTOS hỗ trợ nhiều loại vi điều khiển và kiến trúc khác nhau, giúp dễ dàng triển khai trên các thiết bị khác nhau.

Trong các hệ thống IoT, FreeRTOS thường được sử dụng để quản lý các nhiệm vụ và tài nguyên, đảm bảo hệ thống hoạt động ổn định và hiệu quả. Một số ứng dụng của FreeRTOS trong IoT bao gồm:

* **Quản lý tác vụ:** FreeRTOS cho phép chia hệ thống IoT thành nhiều tác vụ nhỏ, mỗi tác vụ thực hiện một chức năng cụ thể. Ví dụ, trong hệ thống tưới cây thông minh, có thể có các tác vụ như đo độ ẩm, điều khiển bơm, giám sát hệ thống, và gửi dữ liệu lên đám mây.
* **Đồng bộ hóa và giao tiếp:** FreeRTOS cung cấp các công cụ đồng bộ hóa và giao tiếp giữa các tác vụ, giúp các thành phần trong hệ thống IoT hoạt động phối hợp nhịp nhàng. Ví dụ, dữ liệu đo độ ẩm từ cảm biến có thể được gửi qua hàng đợi đến tác vụ điều khiển bơm.
* **Quản lý tài nguyên:** FreeRTOS giúp quản lý hiệu quả các tài nguyên như CPU, bộ nhớ, và thiết bị ngoại vi. Điều này đặc biệt quan trọng trong các hệ thống IoT, nơi tài nguyên thường rất hạn chế.

### 1.3.2. FreeRTOS trong hệ thống IOT

**Ứng dụng của FreeRTOS trong hệ thống IoT**

Trong các hệ thống IoT, FreeRTOS thường được sử dụng để quản lý các nhiệm vụ và tài nguyên, đảm bảo hệ thống hoạt động ổn định và hiệu quả. Một số ứng dụng của FreeRTOS trong IoT bao gồm:

* **Quản lý tác vụ:** FreeRTOS cho phép chia hệ thống IoT thành nhiều tác vụ nhỏ, mỗi tác vụ thực hiện một chức năng cụ thể. Ví dụ, trong hệ thống tưới cây thông minh, có thể có các tác vụ như đo độ ẩm, điều khiển bơm, giám sát hệ thống, và gửi dữ liệu lên đám mây.
* **Đồng bộ hóa và giao tiếp:** FreeRTOS cung cấp các công cụ đồng bộ hóa và giao tiếp giữa các tác vụ, giúp các thành phần trong hệ thống IoT hoạt động phối hợp nhịp nhàng. Ví dụ, dữ liệu đo độ ẩm từ cảm biến có thể được gửi qua hàng đợi đến tác vụ điều khiển bơm.
* **Quản lý tài nguyên:** FreeRTOS giúp quản lý hiệu quả các tài nguyên như CPU, bộ nhớ, và thiết bị ngoại vi. Điều này đặc biệt quan trọng trong các hệ thống IoT, nơi tài nguyên thường rất hạn chế.

**FreeRTOS trong hệ thống tưới cây thông minh**

Trong hệ thống tưới cây thông minh, FreeRTOS được sử dụng để đảm bảo các nhiệm vụ được thực hiện đúng thời gian và theo thứ tự ưu tiên. Dưới đây là một số ứng dụng cụ thể của FreeRTOS trong hệ thống này:

* **Nhiệm vụ đo độ ẩm:** Cảm biến độ ẩm đất liên tục đo độ ẩm của đất và gửi dữ liệu này đến bộ điều khiển trung tâm. Nhiệm vụ này có thể được cấu hình với mức ưu tiên cao, đảm bảo rằng dữ liệu độ ẩm luôn được cập nhật.
* **Nhiệm vụ điều khiển bơm:** Dựa trên dữ liệu từ cảm biến độ ẩm, hệ thống quyết định khi nào cần bơm nước. Nhiệm vụ điều khiển bơm có thể được kích hoạt khi độ ẩm đất dưới mức ngưỡng định trước.
* **Nhiệm vụ giám sát:** Hệ thống cần giám sát các thành phần và hoạt động của hệ thống tưới cây, đảm bảo rằng tất cả các tác vụ đều hoạt động bình thường. Nhiệm vụ giám sát có thể bao gồm kiểm tra trạng thái của cảm biến, bơm, và kết nối mạng.
* **Nhiệm vụ gửi dữ liệu:** Dữ liệu từ hệ thống tưới cây thông minh có thể được gửi lên đám mây hoặc ứng dụng di động để người dùng có thể theo dõi và điều khiển từ xa. Nhiệm vụ này có thể được thực hiện theo định kỳ hoặc khi có sự thay đổi quan trọng trong hệ thống.

**Lợi ích của việc sử dụng FreeRTOS trong hệ thống tưới cây thông minh**

Việc sử dụng FreeRTOS trong hệ thống tưới cây thông minh mang lại nhiều lợi ích:

* **Độ tin cậy cao:** FreeRTOS đảm bảo rằng các nhiệm vụ quan trọng được thực hiện đúng thời gian, giúp hệ thống hoạt động ổn định và đáng tin cậy.
* **Hiệu quả tài nguyên:** FreeRTOS quản lý tài nguyên một cách hiệu quả, giảm thiểu lãng phí và tối ưu hóa hiệu suất hệ thống.
* **Dễ dàng mở rộng:** FreeRTOS hỗ trợ nhiều loại vi điều khiển và thiết bị, giúp hệ thống dễ dàng mở rộng và nâng cấp.
* **Tích hợp dễ dàng:** FreeRTOS có thể tích hợp dễ dàng với các thư viện và công cụ khác, giúp tăng cường chức năng và khả năng của hệ thống.

**Thách thức và giải pháp khi sử dụng FreeRTOS trong hệ thống tưới cây thông minh**

Mặc dù FreeRTOS mang lại nhiều lợi ích, việc triển khai và sử dụng FreeRTOS cũng gặp phải một số thách thức:

* **Thiết kế hệ thống:** Việc thiết kế một hệ thống sử dụng FreeRTOS yêu cầu kiến thức chuyên sâu về lập trình nhúng và quản lý thời gian thực.
* **Quản lý ưu tiên:** Xác định và quản lý mức ưu tiên của các nhiệm vụ là một thách thức quan trọng để đảm bảo hệ thống hoạt động hiệu quả.
* **Bảo trì và cập nhật:** Hệ thống cần được bảo trì và cập nhật định kỳ để đảm bảo hoạt động ổn định và bảo mật.

Giải pháp cho các thách thức này bao gồm:

* **Đào tạo và nâng cao kỹ năng:** Đảm bảo rằng đội ngũ phát triển có đủ kiến thức và kỹ năng về FreeRTOS và lập trình nhúng.
* **Thiết kế linh hoạt:** Xây dựng hệ thống với thiết kế linh hoạt, cho phép dễ dàng điều chỉnh và nâng cấp.
* **Sử dụng công cụ và tài liệu hỗ trợ:** Tận dụng các công cụ và tài liệu hỗ trợ từ cộng đồng FreeRTOS để giải quyết các vấn đề kỹ thuật.

## Công cụ sử dụng

### 1.4.1 Proteus

− Phần mềm Proteus là một bộ phần mềm thiết kế mạch in được phát minh bởi công ty Labcenter

Electronics. Phần mềm này được sử dụng để thiết kế ra các dạng mạch trên PCB ( bo vi mạch )

và mô phỏng lại các mạch khác nhau

− Proteus sở hữu rất nhiều mặt ưu điểm để thu hút các nhà lập trình nhúng chắc chắn không thể bỏ

qua :

• Có khả năng mô phỏng hầu hết các trình điều khiển cho vi điều khiển

• Chọn đối tượng và thiết lập thông số cho đối tượng dễ dàng

• Xuất ra File Netlist tương thích với các chương trình mạch in thông dụng

• Xuất file thống kê linh kiện cho mạch

• ISIS tích hợp nhiều công cụ giúp cho việc quản lý mạch điện lớn, mạch điện có thể lên đến hàng

nghìn linh kiện phục vụ cho thiết kế mạch chuyên nghiệp

• Thiết kế theo cấu trúc ( hierarchical design)

• Khả năng tự động đánh số linh kiện

### 1.4.2 Arduino 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 lế sẵn với các cảm biến, linh kiện. Phần mềm có thể sử dụng các cảm biến

, linh kiện ấy của Arduino một cách linh hoạt và phù hợp với mục đích sử dụng

− Một trong những lý do bạn nên sử dụng Arduino IDE

• IDE trong Arduino IDE là phần có nghĩa là mã nguồn mở. Nghĩa là phần mềm này miễn phí cả

về phần tải lẫn phần bản quyền. Người sử dụng 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 và Arduino

IDE cũng có khả năng bảo mật tuyệt vời

• Sử dụng ngôn ngữ lập trình C/C++ rất phổ biến trong giới lập trình. Bất kì 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

• 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 cập

nhật để các dữ liệu có sẵn tron 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

• Thư viện hỗ trợ phong phú với hơn 700 thư viện, được viết và chia sẻ bởi nhà phát hành Arduino

Software và thành viên trong cộng đồng Arduino

• Giao diện đơn giản, dễ sử dụng

• Hỗ trợ đa nền tảng như Windows, MacOS, Linux

### 1.4.3 ESP32

ESP32-WROOM-32 là mô đun MCU đa dụng, mạnh mẽ và được sử dụng rộng rãi trong thiết

kế mạch PCB Wifi- Bluetooth, BLE được ứng dụng rất phổ biến cho nhiều ứng dụng về IoT

hiện nay. Phạm vi ứng dụng từ mạng sensor tiết kiệm năng lượng đến những ứng dụng với tác

vụ phức tạp nhất, như mã hóa âm thanh, âm nhạc trực tuyến đến giải mã MP3.

Hệ điều hành chạy được trên ESP32 là FreeRTOS vơi LwIP, TLS 1.2. Hỗ trợ update

firmware qua OTA mã hóa, điều này cho phép nhà phát triển sản phẩm có thể nâng cấp phần

mềm sản phẩm ngay cả khi thiết bị đang được sử dụng một cách tiết kiệm tiền bạc và nhân

lực.

Chip ESP32 đi kèm với 48 chân với nhiều chức năng trên cùng 1 chân có thể được cấu

hình theo từng trường hợp sử dụng. Không phải tất cả các chân đều được đưa ra Via kết nối

trong tất cả các bảng phát triển ESP32, và có một số chân không thể được sử dụng.

### 1.4.4 Xampp và cơ sở dữ liệu phpMyAdmin

**XAMPP** là một phần mềm mã nguồn mở miễn phí, được sử dụng phổ biến để thiết lập một máy chủ web cục bộ. Nó bao gồm các thành phần chủ yếu như Apache (máy chủ HTTP), MySQL (cơ sở dữ liệu), PHP (ngôn ngữ lập trình máy chủ), và Perl. XAMPP được thiết kế để dễ dàng cài đặt và sử dụng, giúp các nhà phát triển web nhanh chóng triển khai các ứng dụng web trên máy tính cá nhân trước khi đưa lên môi trường thực tế

**phpMyAdmin** là một công cụ quản lý cơ sở dữ liệu MySQL/MariaDB được viết bằng PHP, cung cấp giao diện web thân thiện cho người dùng. Nó cho phép quản lý cơ sở dữ liệu, bảng, cột, quan hệ, chỉ mục, người dùng, quyền hạn, v.v., một cách dễ dàng mà không cần phải sử dụng dòng lệnh SQL

## Xây dựng bài toán

### 1.5.1. Yêu cầu đặt ra

Việc xây dựng một hệ thống tưới cây thông minh cần phải đáp ứng được nhiều yêu cầu khác nhau để đảm bảo hệ thống hoạt động hiệu quả, bền vững và đáng tin cậy. Các yêu cầu này bao gồm:

* **Tự động hóa việc tưới nước:** Hệ thống cần tự động tưới cây khi độ ẩm đất dưới mức cho phép và ngừng tưới khi độ ẩm đạt ngưỡng tối ưu.
* **Giám sát từ xa:** Hệ thống phải có khả năng giám sát từ xa qua ứng dụng di động hoặc web, cho phép người dùng kiểm tra và điều khiển hệ thống bất cứ lúc nào và ở bất cứ đâu.
* **Cảnh báo:** Hệ thống cần gửi cảnh báo khi gặp sự cố hoặc khi độ ẩm đất quá thấp hoặc quá cao, giúp người dùng có thể can thiệp kịp thời.
* **Tiết kiệm nước:** Hệ thống phải tối ưu hóa việc sử dụng nước, tránh lãng phí và bảo vệ tài nguyên nước.
* **Dễ dàng lắp đặt và bảo trì:** Hệ thống cần được thiết kế sao cho dễ dàng lắp đặt và bảo trì, giảm thiểu chi phí và thời gian bảo trì.
* **Tính mở rộng:** Hệ thống phải có khả năng mở rộng để kết nối thêm nhiều cảm biến và thiết bị khác khi cần thiết.

### 1.5.2. Chức năng của dự án

Dựa trên các yêu cầu đặt ra, hệ thống tưới cây thông minh cần có các chức năng chính sau đây:

**Chức năng đo lường:**

* **Đo độ ẩm đất:** Sử dụng cảm biến độ ẩm đất để liên tục đo lường độ ẩm của đất. Cảm biến sẽ gửi dữ liệu này đến bộ điều khiển trung tâm.
* **Đo nhiệt độ:** Cảm biến nhiệt độ giúp theo dõi nhiệt độ môi trường, điều chỉnh hoạt động của hệ thống dựa trên nhiệt độ.
* **Đo độ ẩm không khí:** Cảm biến độ ẩm không khí phát hiện độ ẩm trong không khí.

**Chức năng điều khiển:**

* **Điều khiển bơm nước:** Bộ điều khiển trung tâm nhận dữ liệu từ cảm biến độ ẩm đất và quyết định khi nào cần tưới nước. Khi độ ẩm đất dưới mức ngưỡng định trước, hệ thống sẽ bật bơm nước. Khi độ ẩm đất đạt ngưỡng tối ưu, hệ thống sẽ ngừng bơm nước.
* **Điều chỉnh lượng nước:** Dựa trên dữ liệu từ các cảm biến, hệ thống có thể điều chỉnh lượng nước tưới phù hợp với nhu cầu của cây.

**Chức năng giám sát và báo cáo:**

* **Giám sát từ xa:** Hệ thống kết nối với ứng dụng di động hoặc web, cho phép người dùng giám sát tình trạng của hệ thống từ xa. Người dùng có thể kiểm tra độ ẩm đất, nhiệt độ, ánh sáng và trạng thái của bơm nước.

**Chức năng tiết kiệm nước:**

* **Tối ưu hóa tưới nước:** Hệ thống sử dụng dữ liệu từ các cảm biến để tối ưu hóa việc tưới nước, đảm bảo cung cấp đủ nước cho cây mà không lãng phí.
* **Tự động ngắt nước:** Khi trời mưa hoặc độ ẩm đất đủ cao, hệ thống sẽ tự động ngắt bơm nước để tránh lãng phí nước.

**Chức năng bảo trì và mở rộng:**

* **Dễ dàng lắp đặt:** Hệ thống được thiết kế với các thành phần module, giúp dễ dàng lắp đặt và thay thế khi cần.
* **Dễ dàng bảo trì:** Các thành phần của hệ thống có thể được kiểm tra và bảo trì một cách dễ dàng, giảm thiểu thời gian và chi phí bảo trì.
* **Khả năng mở rộng:** Hệ thống có thể kết nối thêm nhiều cảm biến và thiết bị khác khi cần, giúp nâng cao khả năng giám sát và điều khiển.

## Tổng kết chương

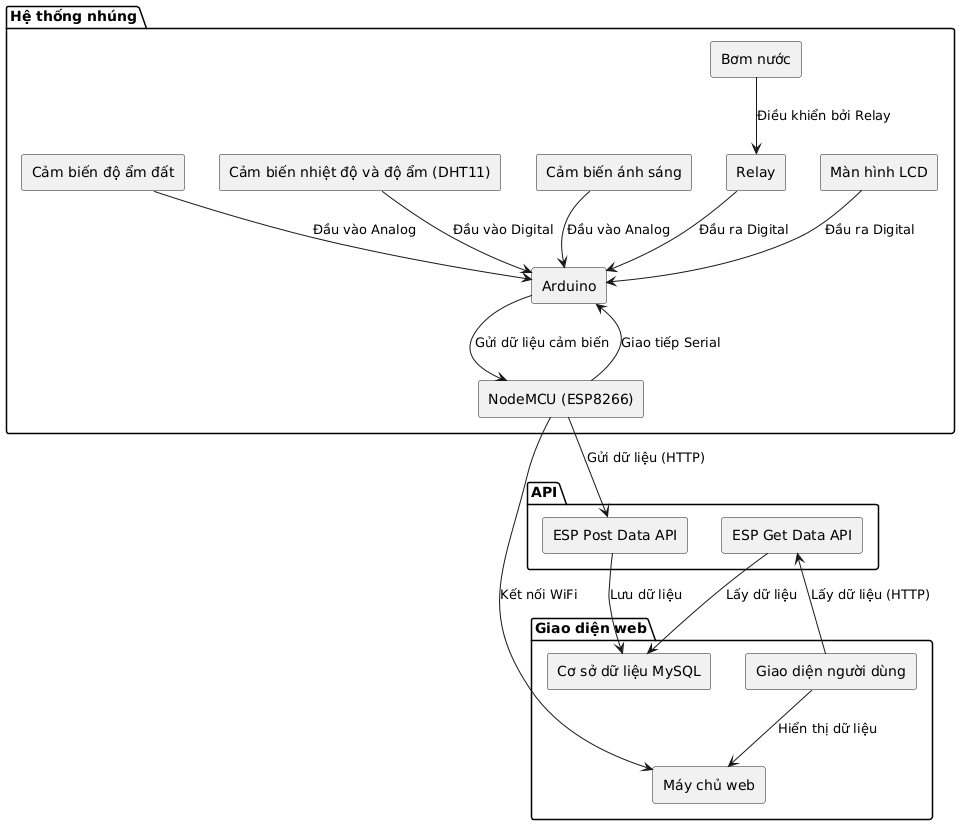
Chương 1 giới thiệu tổng quan về bài toán và các công nghệ được sử dụng, các vấn để tổng quát và tầm quan trọng của nó. Các công nghệ chính được sử dụng để giải quyết vấn đề bao gồm đọc dữ liệu độ ẩm, nhiệt độ, phân tích dữ liệu, tạo một hệ thống báo chăm sóc cây đơn giản.

Khi tiến vào các chương tiếp theo, chúng ta sẽ đi sâu vào hơn những công nghệ này để hiểu rõ hơn về nguyên tắc cơ bản và ứng dụng của chúng trong giải quyết bài toán điều khiển chậu cây thông minh.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ

Trong chương này, sẽ phân tích bài toán “ Chậu cây thông minh” và thiết kế các module tương ứng để giải quyết bài toán. Chương cũng đi sâu hơn về các công nghệ sử dụng để tạo nên sự hoạt động của toàn bộ hệ thống. Cuối chương là cách thức hoạt động và luồng xử lý để giải quyết bài toán

## Mô hình tổng thể



***Hình 1. 4 Mô hình tổng thể***

Sơ đồ này mô tả các thành phần chính của hệ thống tưới cây thông minh và cách chúng kết nối với nhau. Dưới đây là mô tả chi tiết từng thành phần và kết nối:

1. **Nguồn cấp:**
   * **Chức năng:** Cung cấp nguồn điện cho toàn bộ hệ thống bao gồm Arduino, NodeMCU, các cảm biến, relay, bơm nước và màn hình LCD.
2. **Arduino:**
   * **Chức năng:** Là bộ điều khiển trung tâm, nhận và xử lý dữ liệu từ các cảm biến, điều khiển bơm nước và các thiết bị khác.
   * **Kết nối:**
     + **Cảm biến độ ẩm đất:** Kết nối với Arduino qua đầu vào analog.
     + **Cảm biến nhiệt độ và độ ẩm (BME280):** Kết nối với Arduino qua đầu vào I2c.
     + **Cảm biến ánh sáng:** Kết nối với Arduino qua đầu vào digital .
     + **Relay:** Kết nối với Arduino qua đầu ra digital để điều khiển bơm nước.
     + **Màn hình LCD:** Kết nối với Arduino qua giao tiếp I2C để hiển thị thông tin.
     + **NodeMCU (ESP8266):** Giao tiếp với Arduino qua giao tiếp Serial để gửi và nhận dữ liệu.
3. **NodeMCU (ESP8266):**
   * **Chức năng:** Kết nối WiFi để gửi dữ liệu lên máy chủ web và nhận lệnh điều khiển từ xa.
   * **Kết nối:**
     + **Giao tiếp với Arduino:** Qua giao tiếp Serial.
     + **Kết nối với máy chủ web:** Qua giao tiếp HTTP.
4. **Cảm biến độ ẩm đất:**
   * **Chức năng:** Đo lường độ ẩm của đất và gửi dữ liệu về Arduino.
   * **Kết nối:** Kết nối với Arduino qua đầu vào analog.
5. **Cảm biến nhiệt độ và độ ẩm (DHT11):**
   * **Chức năng:** Đo lường nhiệt độ và độ ẩm không khí và gửi dữ liệu về Arduino.
   * **Kết nối:** Kết nối với Arduino qua đầu vào digital.
6. **Cảm biến ánh sáng:**
   * **Chức năng:** Đo lường cường độ ánh sáng và gửi dữ liệu về Arduino.
   * **Kết nối:** Kết nối với Arduino qua đầu vào analog.
7. **Relay:**
   * **Chức năng:** Điều khiển bơm nước dựa trên tín hiệu từ Arduino.
   * **Kết nối:** Kết nối với Arduino qua đầu ra digital và điều khiển bơm nước.
8. **Bơm nước:**
   * **Chức năng:** Bơm nước tưới cây khi nhận được tín hiệu từ Relay.
   * **Kết nối:** Kết nối với Relay để nhận tín hiệu điều khiển.
9. **Màn hình LCD:**
   * **Chức năng:** Hiển thị thông tin về độ ẩm đất, nhiệt độ, độ ẩm không khí và trạng thái của hệ thống.
   * **Kết nối:** Kết nối với Arduino qua đầu ra digital để nhận dữ liệu hiển thị.
10. **Máy chủ web:**
    * **Chức năng:** Nhận và hiển thị dữ liệu từ hệ thống, cung cấp giao diện điều khiển từ xa.
    * **Kết nối:** Giao tiếp với NodeMCU qua giao tiếp HTTP.

**Tóm tắt cách hoạt động của hệ thống:**

* Cảm biến độ ẩm đất, cảm biến nhiệt độ và độ ẩm (DHT11), cảm biến ánh sáng liên tục đo lường các thông số môi trường và gửi dữ liệu về Arduino.
* Arduino xử lý dữ liệu từ các cảm biến và quyết định khi nào cần tưới nước.
* Khi độ ẩm đất dưới mức ngưỡng định trước, Arduino gửi tín hiệu để bật relay, kích hoạt bơm nước.
* Khi độ ẩm đất đạt ngưỡng tối ưu, Arduino gửi tín hiệu để tắt relay, dừng bơm nước.
* NodeMCU kết nối với WiFi và gửi dữ liệu lên máy chủ web để hiển thị trên trang web. Người dùng có thể kiểm tra và giám sát tình trạng của hệ thống từ xa.
* Màn hình LCD hiển thị thông tin về độ ẩm đất, nhiệt độ, độ ẩm không khí và trạng thái của hệ thống để người dùng dễ dàng theo dõi.

## Thiết kế phần cứng

Dựa trên phát biểu về bài toán chậu cây thông minh, nghiên cứu và khảo sát thực tế, hệ thống đặt ra những yêu cầu sau :

#### 2.1. Về phía người dùng:

Theo dõi thông số độ ẩm, ánh sáng theo chức năng đã được đề ra, điều khiển bơm hoạt động theo chế độ bằng tay hoặc tự động.

#### 2.2. Về phía hệ thống:

Đo các thông số môi trường : truyền dữ liệu lên Web điều khiển. Điều khiển tự động tưới nước khi độ ẩm không đạt yêu cầu.

Điều khiển chậu cây theo hướng sáng.

Ngoài các yêu cầu chức năng, hệ thống cần đáp ứng các yêu cầu phi chức năng :

* Có khả năng chống nhiễu tốt
* Điều khiển đáp ứng nhanh.
* Không bị ảnh hưởng bởi các hệ thống khác được lắp đặt chung hoặc riêng rẽ
* Hệ thống đảm bảo độ tin cậy, thực hiện đúng các chức năng đặt ra.
* Những tác động bên ngoài gây ra sự cố cho một bộ phận của hệ thống không được gây ra những sự cố tiếp trong hệ thống

A circuit board with wires

Description automatically generated

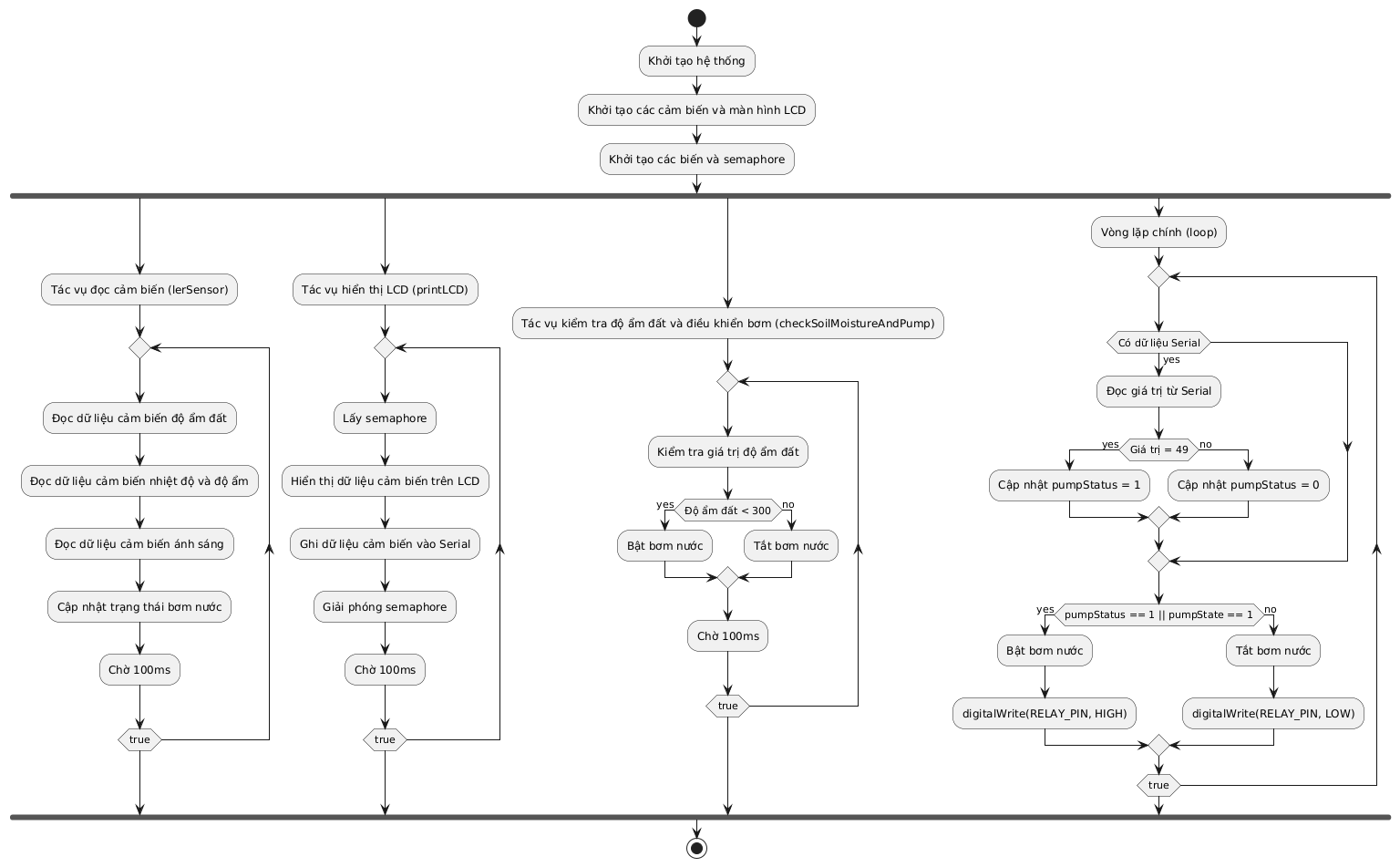
Hình 1. 5 hệ thống tưới cây thông minh

Sơ đồ khối chức năng của hệ thống tưới cây thông minh bao gồm các thành phần chính như sau:

* **Arduino:** Đóng vai trò là bộ điều khiển trung tâm, xử lý dữ liệu từ các cảm biến và điều khiển các thiết bị khác.
* **NodeMCU (ESP8266):** Kết nối WiFi để gửi dữ liệu lên ứng dụng di động hoặc web và nhận lệnh điều khiển từ xa.
* **Cảm biến độ ẩm đất (Soil Moisture Sensor):** Đo lường độ ẩm của đất và gửi dữ liệu đến Arduino.
* **Cảm biến nhiệt độ và độ ẩm (DHT11):** Đo lường nhiệt độ và độ ẩm không khí và gửi dữ liệu đến Arduino.
* **Cảm biến ánh sáng (Light Sensor):** Đo lường cường độ ánh sáng và gửi dữ liệu đến Arduino.
* **Relay:** Điều khiển bơm nước dựa trên tín hiệu từ Arduino.
* **Bơm nước (Water Pump):** Bơm nước tưới cây khi nhận được tín hiệu từ Arduino qua Relay.
* **Màn hình LCD:** Hiển thị thông tin về độ ẩm đất, nhiệt độ, độ ẩm không khí và trạng thái của hệ thống.
* **Nguồn cấp:** Cung cấp điện cho toàn bộ hệ thống.

## 2.3. Thiết kế phần mềm

### 2.3.1. Lập trình freeRTOS Arduino



Hình 1. 6 Sơ đồ lập trình freeRTOS Arduino

Sơ đồ này mô tả các thành phần chính của hệ thống tưới cây thông minh và cách chúng kết nối với nhau, bao gồm các tác vụ song song như đọc cảm biến, hiển thị dữ liệu lên màn hình LCD, kiểm tra độ ẩm đất và điều khiển bơm nước, cũng như vòng lặp chính trong hàm loop.

**1. Khởi tạo hệ thống**

* **Khởi tạo các cảm biến và màn hình LCD:** Thiết lập các cảm biến cần thiết và khởi động màn hình LCD.
* **Khởi tạo các biến và semaphore:** Thiết lập các biến và semaphore để điều khiển truy cập dữ liệu chia sẻ.

**2. Tác vụ đọc cảm biến (lerSensor)**

* **Đọc dữ liệu cảm biến độ ẩm đất:** Đo lường giá trị độ ẩm của đất.
* **Đọc dữ liệu cảm biến nhiệt độ và độ ẩm:** Đo lường giá trị nhiệt độ và độ ẩm không khí.
* **Đọc dữ liệu cảm biến ánh sáng:** Đo lường giá trị cường độ ánh sáng.
* **Cập nhật trạng thái bơm nước:** Kiểm tra trạng thái của cảm biến chạm để xác định xem bơm nước có đang bật hay tắt.
* **Chờ 100ms:** Chờ một khoảng thời gian ngắn trước khi lặp lại quy trình.

**3. Tác vụ hiển thị LCD (printLCD)**

* **Lấy semaphore:** Đảm bảo rằng dữ liệu được truy cập một cách an toàn bằng cách lấy semaphore.
* **Hiển thị dữ liệu cảm biến trên LCD:** Hiển thị các giá trị cảm biến lên màn hình LCD.
* **Ghi dữ liệu cảm biến vào Serial:** Ghi dữ liệu cảm biến vào giao tiếp Serial để theo dõi.
* **Giải phóng semaphore:** Giải phóng semaphore sau khi hoàn thành việc truy cập dữ liệu.
* **Chờ 100ms:** Chờ một khoảng thời gian ngắn trước khi lặp lại quy trình.

**4. Tác vụ kiểm tra độ ẩm đất và điều khiển bơm (checkSoilMoistureAndPump)**

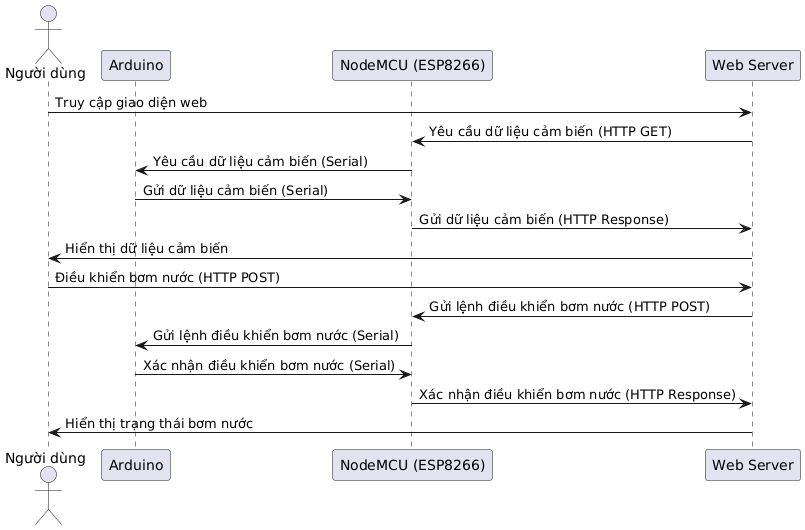
* **Kiểm tra giá trị độ ẩm đất:** Đo lường giá trị độ ẩm của đất.
* **Bật bơm nước:** Nếu độ ẩm đất thấp hơn ngưỡng 300, bật bơm nước.
* **Tắt bơm nước:** Nếu độ ẩm đất cao hơn hoặc bằng ngưỡng 300, tắt bơm nước.
* **Chờ 100ms:** Chờ một khoảng thời gian ngắn trước khi lặp lại quy trình.

**5. Vòng lặp chính (loop)**

* **Có dữ liệu Serial:** Kiểm tra xem có dữ liệu mới từ Serial không.
* **Đọc giá trị từ Serial:** Đọc giá trị từ giao tiếp Serial.
* **Cập nhật pumpStatus:** Nếu giá trị từ Serial là 49, cập nhật pumpStatus là 1, ngược lại cập nhật pumpStatus là 0.
* **Kiểm tra trạng thái bơm:** Nếu pumpStatus hoặc pumpState là 1, bật bơm nước, ngược lại tắt bơm nước.
* **Bật bơm nước:** Điều khiển chân RELAY\_PIN để bật bơm nước.
* **Tắt bơm nước:** Điều khiển chân RELAY\_PIN để tắt bơm nước.

### 2.3.2. Sơ đồ hoạt động của hệ thống

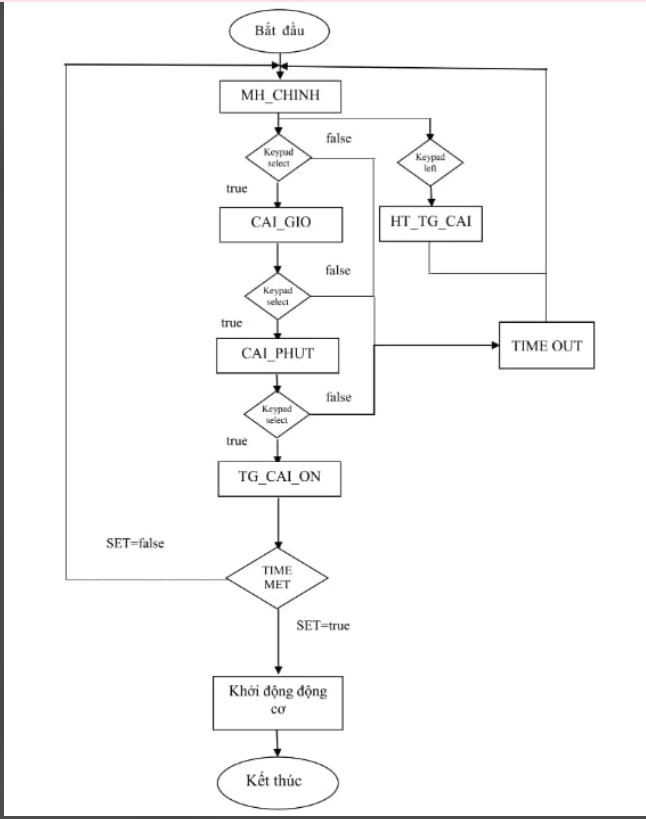
Tổng quan hệ thống gồm hai phần chính:



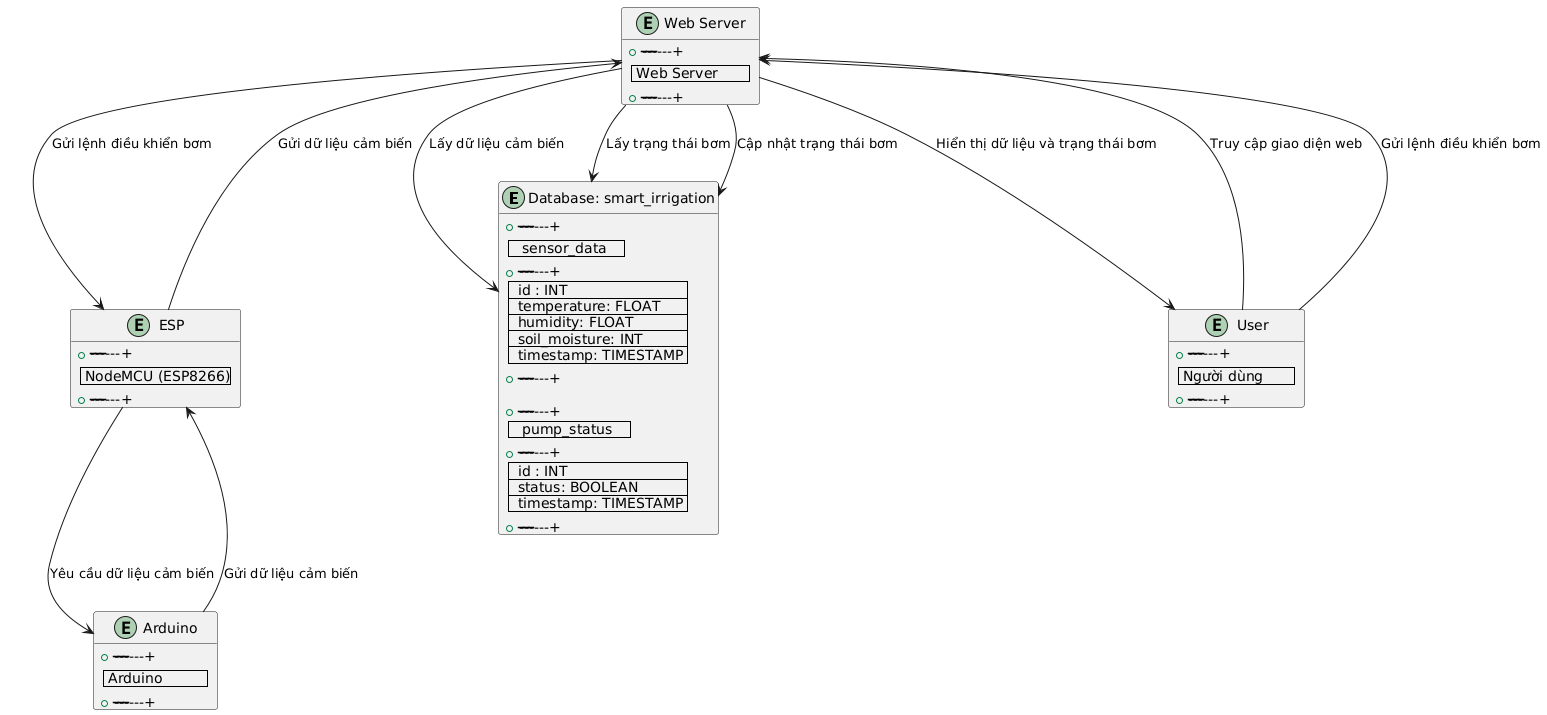
**Giải thích sơ đồ:**

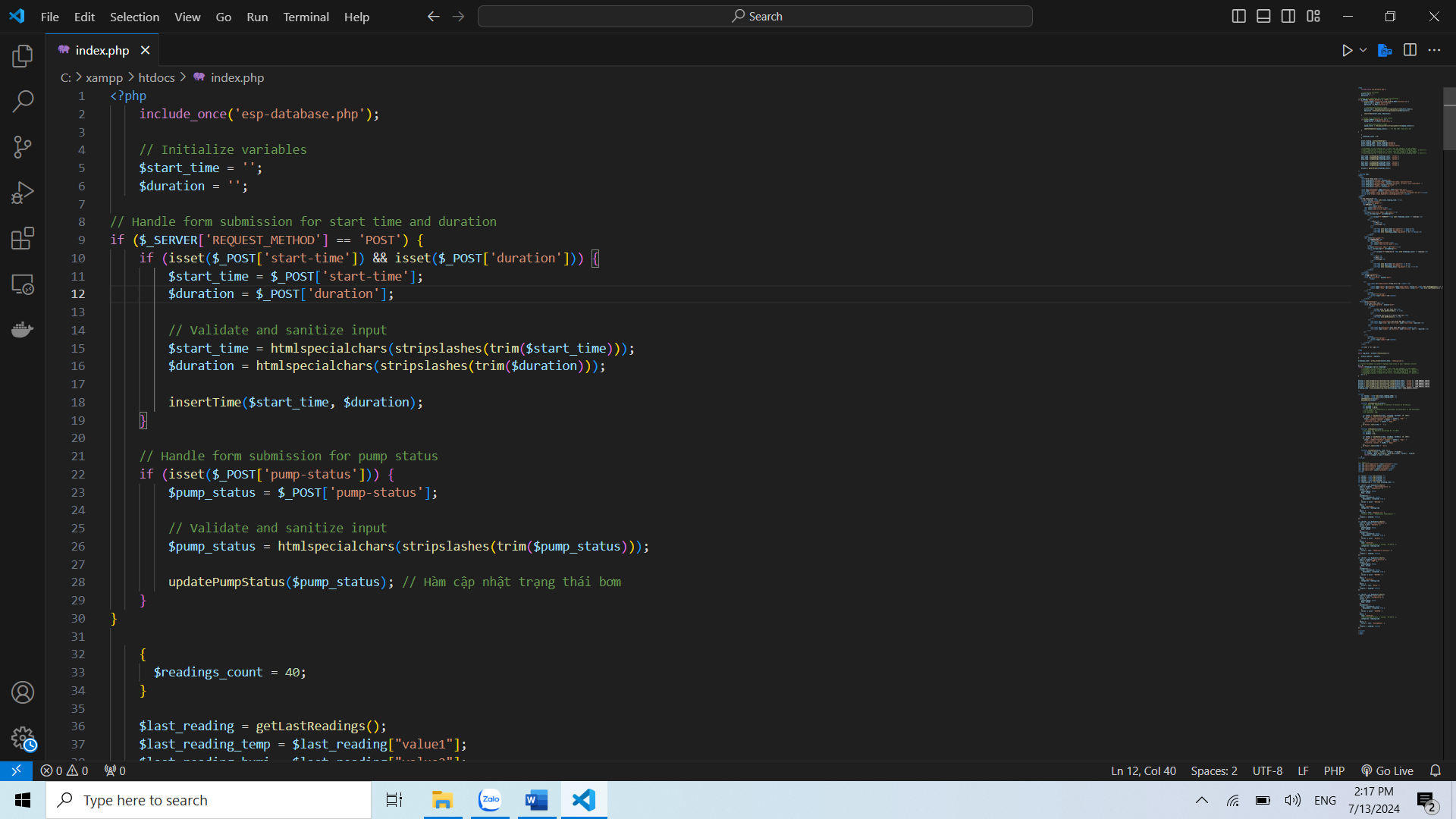
* **Người dùng truy cập giao diện web:**
  1. Người dùng truy cập giao diện web để xem dữ liệu cảm biến và điều khiển hệ thống tưới cây.
* **Yêu cầu dữ liệu cảm biến:**
  1. Web Server gửi yêu cầu HTTP GET đến NodeMCU để lấy dữ liệu cảm biến.
  2. NodeMCU gửi yêu cầu qua giao tiếp Serial đến Arduino để lấy dữ liệu cảm biến.
  3. Arduino đọc dữ liệu cảm biến và gửi lại cho NodeMCU qua giao tiếp Serial.
  4. NodeMCU gửi dữ liệu cảm biến đến Web Server qua HTTP Response.
  5. Web Server hiển thị dữ liệu cảm biến cho người dùng.
* **Điều khiển bơm nước:**
  1. Người dùng gửi lệnh điều khiển bơm nước thông qua giao diện web (HTTP POST).
  2. Web Server gửi lệnh điều khiển bơm nước đến NodeMCU (HTTP POST).
  3. NodeMCU gửi lệnh điều khiển bơm nước đến Arduino qua giao tiếp Serial.
  4. Arduino thực hiện lệnh và gửi xác nhận điều khiển bơm nước trở lại NodeMCU qua giao tiếp Serial.
  5. NodeMCU gửi xác nhận điều khiển bơm nước đến Web Server qua HTTP Response.
  6. Web Server hiển thị trạng thái bơm nước cho người dùng.

## 2.4. Code Web



**Hình 2.1 Lưu đồ thuật toán sơ đồ hệ thống tưới cây tự động**



Hình 2.2 màn hình code web hiển thị

## 2.5. Code ESP

### 2.5.1 Khai báo các thư viện, chân tín hiệu

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <WiFiUdp.h>

#include <NTPClient.h>

#include <ArduinoJson.h>

// Define DEBUG\_SERIAL to enable or disable serial debug

// #define DEBUG\_SERIAL

#ifdef DEBUG\_SERIAL

#define DEBUG\_PRINT(x) Serial.print(x)

#define DEBUG\_PRINTLN(x) Serial.println(x)

#else

#define DEBUG\_PRINT(x)

#define DEBUG\_PRINTLN(x)

#endif

// WiFi credentials

const char\* ssid = "Xuong BI-A 2.4G";

const char\* password = "66668888";

WiFiClient client;

HTTPClient http;

// NTP Server

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org", 3600\*7, 60000); // 3600 là GMT offset, 60000 là thời gian update mỗi 60 giây

// Variables

const String apiKeyValue = "tPmAT5Ab3j7F9";

const String sensorName = "sensor";

const String sensorLocation = "location";

const char\* serverTime = "http://192.168.6.208/esp-time-duration.php"; // Replace with your server URL

const char\* serverUrl = "http://192.168.6.208/esp-post-data.php?api\_key=";

const char\* serverPumpStatusUrl = "http://192.168.6.208/esp-get-pump-status.php"; // Replace with your server URL

float Humidity = 60;

float Temperature = 32;

int soilMoistureValue = 300;

bool pumpState = false;

long distanceCm = 10;

bool statusRelay = false;

### 2.5.2 Hàm setup() khởi tạo

void setup() {

  // Start serial communication

    Serial.begin(115200);

  // Connect to Wi-Fi

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    DEBUG\_PRINTLN("Connecting to WiFi...");

  }

  DEBUG\_PRINTLN("Connected to WiFi");

  // Start NTP client

  timeClient.begin();

}

### 2.5.3. Hàm loop() để nhận và xử lý tín hiệu cho các cảm biến hệ thống

void loop() {

  // Update NTP client to get the latest time

  timeClient.update();

  // Read data from Serial JSON

  if (Serial.available() > 0) {

    String jsonData = Serial.readStringUntil('\n');

    DEBUG\_PRINTLN("Received JSON: " + jsonData);

    StaticJsonDocument<200> doc;

    DeserializationError error = deserializeJson(doc, jsonData);

    if (error) {

      DEBUG\_PRINT(F("deserializeJson() failed: "));

      DEBUG\_PRINTLN(error.f\_str());

      return;

    }

    Humidity = doc["Humidity"];

    Temperature = doc["Temperature"];

    soilMoistureValue = doc["soilMoistureValue"];

    pumpState = doc["pumpState"];

    DEBUG\_PRINT("Humidity: ");

    DEBUG\_PRINTLN(Humidity);

    DEBUG\_PRINT("Temperature: ");

    DEBUG\_PRINTLN(Temperature);

    DEBUG\_PRINT("Soil Moisture Value: ");

    DEBUG\_PRINTLN(soilMoistureValue);

    DEBUG\_PRINT("Pump State: ");

    DEBUG\_PRINTLN(pumpState);

  }

  // Send pumpState string to Arduino Uno via Serial

  Serial.print(String(statusRelay));

  // Get the current time

  String formattedTime = timeClient.getFormattedTime();

  DEBUG\_PRINTLN("Current time: " + formattedTime);

  // Send data to the server

  getHttp();

  // Get scheduled time and duration

  getScheduledTimeAndControlPump();

  // Get remote pump status and control pump

  getRemotePumpStatus();

  // Delay between each loop iteration

  delay(50);

}

### 2.5.4. Hàm gethttp()

void getHttp() {

  // Check WiFi connection status

  if (WiFi.status() == WL\_CONNECTED) {

    String url = serverUrl + apiKeyValue +

                 "&sensor=" + sensorName + "&location=" + sensorLocation +

                 "&value1=" + String(Humidity) + "&value2=" + String(Temperature) +

                 "&value3=" + String(pumpState) + "&value4=" + String(soilMoistureValue) +

                 "&value5=" + String(pumpState);

    DEBUG\_PRINTLN(url);

    http.begin(client, url);

    int httpCode = http.GET();

    if (httpCode == HTTP\_CODE\_OK) {

      DEBUG\_PRINTLN("Data pushed successfully");

    } else {

      DEBUG\_PRINTLN("Push error: " + String(httpCode));

    }

    http.end();

    delay(50);

  } else {

    DEBUG\_PRINTLN("WiFi Disconnected");

  }

}

void getScheduledTimeAndControlPump() {

  if (WiFi.status() == WL\_CONNECTED) {

    http.begin(client, serverTime); // Specify the URL

    int httpCode = http.GET(); // Make the request

    if (httpCode > 0) { // Check for the returning code

      String payload = http.getString();

      DEBUG\_PRINTLN("Response: " + payload);

      // Split the data

      int separatorIndex = payload.indexOf('/');

      if (separatorIndex != -1) {

        String time = payload.substring(0, separatorIndex);

        String durationStr = payload.substring(separatorIndex + 1);

        DEBUG\_PRINTLN("Time: " + time);

        DEBUG\_PRINTLN("Duration: " + durationStr);

        // Further split the time into hours and minutes

        int colonIndex1 = time.indexOf(':');

        int colonIndex2 = time.indexOf(':', colonIndex1 + 1);

        if (colonIndex1 != -1 && colonIndex2 != -1) {

          String hoursStr = time.substring(0, colonIndex1);

          String minutesStr = time.substring(colonIndex1 + 1, colonIndex2);

          int duration = durationStr.toInt();

          int hours = hoursStr.toInt();

          int minutes = minutesStr.toInt();

          DEBUG\_PRINT("Hours: ");

          DEBUG\_PRINTLN(hours);

          DEBUG\_PRINT("Minutes: ");

          DEBUG\_PRINTLN(minutes);

          DEBUG\_PRINT("Duration: ");

          DEBUG\_PRINTLN(duration);

          // Get current time from NTP client

          int currentHours = timeClient.getHours();

          int currentMinutes = timeClient.getMinutes();

          DEBUG\_PRINT("Current Hours: ");

          DEBUG\_PRINTLN(currentHours);

          DEBUG\_PRINT("Current Minutes: ");

          DEBUG\_PRINTLN(currentMinutes);

          // Calculate the end time

          int endMinutes = minutes + duration;

          int endHours = hours + (endMinutes / 60);

          endMinutes = endMinutes % 60;

          DEBUG\_PRINT("End Hours: ");

          DEBUG\_PRINTLN(endHours);

          DEBUG\_PRINT("End Minutes: ");

          DEBUG\_PRINTLN(endMinutes);

          bool isWithinTimeFrame = true;

          // Check if current time is within start time and end time

          if (currentHours < hours

              || currentHours > endHours

              || (currentHours == hours && currentMinutes < minutes)

              || (currentHours == endHours && currentMinutes > endMinutes)) {

            isWithinTimeFrame = false;

          }

          if (isWithinTimeFrame) {

            DEBUG\_PRINTLN("Current time is within the scheduled time frame.");

            statusRelay = true;

          } else {

            DEBUG\_PRINTLN("Current time is not within the scheduled time frame.");

            statusRelay = false;

          }

        } else {

          DEBUG\_PRINTLN("Invalid time format");

        }

      } else {

        DEBUG\_PRINTLN("Invalid format");

      }

    } else {

      DEBUG\_PRINTLN("Error on HTTP request");

    }

    http.end(); // Free the resources

  } else {

    DEBUG\_PRINTLN("WiFi not connected");

  }

}

void getRemotePumpStatus() {

  if (WiFi.status() == WL\_CONNECTED) {

    http.begin(client, serverPumpStatusUrl); // Specify the URL

    int httpCode = http.GET(); // Make the request

    if (httpCode > 0) { // Check for the returning code

      String payload = http.getString();

      DEBUG\_PRINTLN("Pump Status Response: " + payload);

      if (payload == "1") {

        DEBUG\_PRINTLN("Turning pump ON remotely");

        statusRelay = true;

      } else if (payload == "0") {

        DEBUG\_PRINTLN("Turning pump OFF remotely");

        statusRelay = false;

      } else {

        DEBUG\_PRINTLN("Invalid pump status received");

      }

    } else {

      DEBUG\_PRINTLN("Error on HTTP request");

    }

    http.end(); // Free the resources

  } else {

    DEBUG\_PRINTLN("WiFi not connected");

  }

}

## 2.6 Code Arduino

#include <Arduino\_FreeRTOS.h>

#include <semphr.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <BME280I2C.h>

BME280I2C bme;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#define SOIL\_PIN A0

#define SOIL\_D\_PIN 3

#define RELAY\_PIN 2

#define TOUCH\_PIN 7

bool pumpState = false;

bool soilState = false;

int soilMoistureValue = 0;

bool pumpStatus = false;

SemaphoreHandle\_t mutex\_v;

BME280::TempUnit tempUnit(BME280::TempUnit\_Celsius);

BME280::PresUnit presUnit(BME280::PresUnit\_Pa);

float temp(NAN), hum(NAN), pres(NAN);

String jsonString;

void setup() {

  Serial.begin(115200);

  lcd.init();

  lcd.backlight();

  lcd.clear();

  while(!bme.begin()) {

    Serial.println("Could not find BME280 sensor!");

    delay(1000);

  }

  switch(bme.chipModel()) {

    case BME280::ChipModel\_BME280:

      Serial.println("Found BME280 sensor! Success.");

      break;

    case BME280::ChipModel\_BMP280:

      Serial.println("Found BMP280 sensor! No Humidity available.");

      break;

    default:

      Serial.println("Found UNKNOWN sensor! Error!");

  }

  mutex\_v = xSemaphoreCreateMutex();

  if (mutex\_v == NULL) {

    Serial.println("Mutex can not be created");

  }

  pinMode(RELAY\_PIN, OUTPUT);

  pinMode(TOUCH\_PIN, INPUT);

  pinMode(SOIL\_D\_PIN, INPUT);

  xTaskCreate(lerSensor, "SENSOR", 128, NULL, 1, NULL);

  xTaskCreate(printLCD, "LCD", 256, NULL, 2, NULL);

  // xTaskCreate(checkSoilMoistureAndPump, "CHECK\_SOIL\_PUMP", 128, NULL, 1, NULL);

  vTaskStartScheduler();

}

void loop() {

  if (pumpStatus == 1 || pumpState == 1) {

    digitalWrite(RELAY\_PIN, HIGH);  // Turn on pump

  } else {

    digitalWrite(RELAY\_PIN, LOW);   // Turn off pump

  }

    Serial.print(F("{\"Humidity\":"));

    Serial.print(round(hum));

    Serial.print(F(",\"Temperature\":"));

    Serial.print(round(temp));

    Serial.print(F(",\"soilMoistureValue\":"));

    Serial.print(soilMoistureValue);

    Serial.print(F(",\"pumpState\":"));

    Serial.print(pumpState);

    Serial.println(F("}"));

  if (Serial.available() > 0) {

    byte k = Serial.read();

    Serial.println(k);

    if (k == 49) {

      pumpStatus = 1;

    } else {

      pumpStatus = 0;

    }

  }

}

void lerSensor(void \*parametro) {

  while(true) {

    soilState = digitalRead(SOIL\_D\_PIN) == LOW;

    soilMoistureValue = analogRead(SOIL\_PIN);

    if (digitalRead(TOUCH\_PIN) == HIGH) {

      pumpState = 0;

    } else {

      vTaskDelay(20 / portTICK\_PERIOD\_MS);

      if (digitalRead(TOUCH\_PIN) == LOW) {

        pumpState = 1;

      }

    }

    vTaskDelay(100 / portTICK\_PERIOD\_MS);

  }

}

void printLCD(void \*parametro) {

  while(true) {

    xSemaphoreTake(mutex\_v, portMAX\_DELAY);

    bme.read(pres, temp, hum, tempUnit, presUnit);

    lcd.setCursor(0, 0);

    lcd.print(F("H:"));

    lcd.print(round(hum));

    lcd.print(F("%"));

    lcd.setCursor(8, 0);

    lcd.print(F("T:"));

    lcd.print(round(temp));

    lcd.print(F("oC"));

    lcd.setCursor(0, 1);

    lcd.print(F("PUMP:"));

    lcd.print((pumpState));

    lcd.setCursor(7, 1);

    lcd.print(F("Soid:"));

    lcd.print(soilMoistureValue);

    lcd.print(F(" "));

    vTaskDelay(100 / portTICK\_PERIOD\_MS);

    xSemaphoreGive(mutex\_v);

  }

}

void checkSoilMoistureAndPump(void \*parametro) {

  while(true) {

    if (soilMoistureValue > 900) {

      digitalWrite(RELAY\_PIN, HIGH);  // Turn on pump

    } else {

      digitalWrite(RELAY\_PIN, LOW);   // Turn off pump

    }

    vTaskDelay(100 / portTICK\_PERIOD\_MS);

  }

}

## 2.7 Tổng kết chương 2

Tổng kết chương này, chúng ta đã tập chung vào phân tích, thiết kế hệ thống, nêu ra được các yêu cầu chức năng và phi chức năng cho sản phẩm. Trong phần phân tích hệ thống, chúng ta đã đưa ra lưu đồ thuật toán của hệ thống, chúng ta cũng đã đã xây dựng sơ đồ ghép nối mạch để mô tả các thành phần kết nối với nhau, đảm bảo sự liên kết mạch lạc

Tóm lại, quá trình phân tích và thiết kế hệ thống đã giúp chúng ta hiểu rõ hơn về cấu trúc và hoạt động của dự án. Điều này là cơ sở quan trọng để tiến hành bước tiếp theo trong quá trình phát triển là phần thực nghiệm và đánh giá hệ thống.

# CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG

## Thử nghiệm tính năng

***A close up of a device

Description automatically generated***

Hình 3. 1 Mô hình thực tế

A close up of a circuit board

Description automatically generated

Hình 3. 2 Mô hình thực tế

A screenshot of a computer

Description automatically generated

Hình 3. 3 Giao diện ứng dụng

A diagram of a graph

Description automatically generated with medium confidence

Hình 3. 4 Giá trị thu được

## Đánh giá hệ thống

-Demo trực tiếp tại buổi thuyết trình

## Nhận xét đánh giá

* + - Ưu điểm:
      * Mô hình đã có thể đo được cường độ ánh sáng
      * Xử lý tự động bơm nước khi độ ẩm không đạt yêu cầu
      * Dễ quan sát, dễ sử dụng
      * Hệ thống tiêu thụ điện năng ít
      * Có nguồn cung cấp cho mạch
      * Xử lý tự động quay khi cường độ ánh sáng, độ ẩm đất không đủ
    - Nhược điểm:
      * Hệ thống bơm nước còn chưa được tối ưu về đường dẫn
      * Hệ thống dây điện của sản phẩm còn nhiều sự chồng chéo, rối mắt
      * Mô hình còn thô sơ chưa có khả năng phát triển với quy mô lớn
      * Chưa thể theo dõi trực tiếp cây trồng 24/7

## Tổng kết chương 3

Trong chương này, đã nêu việc hoàn thành các thành phần bài toán và cũng là hoàn thành bài toán và cũng là hoàn thành bài toán với các nội dung chính: yêu cầu cài đặt, xây dựng các chức năng, chăm sóc cây thông minh thông qua cảm độ ẩm và cảm biến quang

Chương này cũng đánh dấu kết thúc hoàn thành cho bài toán “Hệ thống tưới cây tự động”.

Phần thực nghiệm đã hoàn thành mục tiêu và giải quyết bài toán đưa ra, hệ thống có thể tưới nước và điều chỉnh ánh sáng cho cây một cách trơn và suôn sẻ.

# CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1. Kết quả đạt được

Sau một thời gian nghiên cứu, tìm hiểu và thực hiện đề tài “ Chậu cây thông minh” sản phẩm cơ bản đã được hoàn thành. Bằng sự nỗ lực cố gắng , đoàn kết của cả nhóm, sản phẩm đã được hoàn thành đúng thời gian như đã định và đã đạt được những kết quả sau:

* Chế tạo thành công mô hình chậu cây thông minh
  + Hệ thống hoạt động trơn chu, ổn định
* Đối với nhóm áp dụng được nhiều kiến thức bổ ích như sau:
  + Nguyên tắc hoạt động của mạch
  + Hiểu rõ hơn về các linh kiện và vi điểu khiển
  + Cấu tạo, cách thức hoạt động của chậu cây thông minh
  + Có thêm kiến thức về lập trình Arduino

## 4.2. Hạn chế

Bên cạnh đó cũng tránh khỏi những khuyết điểm, hạn chế cần khắc phục như sau:

* Đề tài mới chỉ có thể khảo sát ở điều kiện cho phép
* Chưa ứng dụng vào một trang trại hay vườn cụ thể
* Chưa giải quyết được bào toán về kinh phí vì giá thành cấu tạo sản phẩm còn chưa hợp lý

## 4.3. Hướng phát triển

Với những hạn chế của đề tài, để đề tài được mở rộng và áp dụng vào đời sống sau này, nhóm đã đưa ra một số hướng phát triển của đề tài :

Tích hợp thêm nhiều chức năng :

* Sử dụng camera tích hợp theo dõi trực tiếp cây trồng
* Nâng cấp ứng dụng để hoạt động một cách hiệu quả và tối ưu hơn
* Thiết kế giao diện ứng dụng thân thiện và dễ tiếp cận hơn cho người dùng

Trong báo cáo này, chúng em đã trình bày khá đầy đủ về các yêu cầu chức năng, phi chức năng, cấu trúc của từng linh kiện. Như vậy, giúp người đọc có thể nắm bắt, đọc hiểu được chức năng của từng linh kiện một cách dễ dàng. Bên cạnh đó, nội dung của đề tài được trình bày khá chi tiết rõ ràng bằng cách sử dụng những ngôn từ thông dụng, các hình ảnh đi kèm giúp người đọc có thể áp dụng một cách tương tự, đạt hiệu quả như mong muốn