

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỆN TỬ**  
-----o0o-----



**BÀI TẬP LỚN THIẾT KẾ HỆ THỐNG NHÚNG**

**TÊN ĐỀ TÀI**

**Hệ thống tưới tiêu tự động**

**(Smart Automatic Irrigation System)**

**GVHD: NGUYỄN PHAN HẢI PHÚ**

<b>SVTH</b>	<b>MSSV</b>
<b>Nguyễn Thành Nhân</b>	<b>2212368</b>
<b>Từ Vĩnh Minh</b>	<b>2114085</b>

**TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2025**

## TÓM TẮT ĐỀ TÀI

Đề tài Thiết kế hệ thống tưới nước tự động thông minh dựa trên ESP32 hướng đến việc xây dựng một giải pháp giám sát và điều khiển tưới tiêu đơn giản, tin cậy và tiết kiệm chi phí, phù hợp với các ứng dụng quy mô nhỏ như vườn rau gia đình, nhà kính hoặc phòng thí nghiệm. Trọng tâm của đề tài tập trung vào việc tích hợp vi điều khiển ESP32 với cảm biến độ ẩm đất và cảm biến mực nước, từ đó tạo nên một hệ thống tưới tự động thông minh có khả năng thu thập dữ liệu chính xác, xử lý thông tin nhanh chóng và đưa ra quyết định điều khiển hợp lý, đồng thời đảm bảo hoạt động ổn định liên tục trong thời gian dài với mức tiêu thụ năng lượng tối ưu.

Về phần cứng, hệ thống sử dụng vi điều khiển ESP32 làm trung tâm xử lý và kết nối với cảm biến độ ẩm đất, mực nước cùng với relay để điều khiển máy bơm. Nguồn cung cấp được ổn định ở mức phù hợp, các đường dây được bố trí gọn và ngăn để giảm nhiễu. Hệ thống có thể gắn kèm một màn hình LCD I2C để hiển thị trực tiếp nhiệt độ theo độ C và độ ẩm theo phần trăm, hoặc ghi lại số liệu vào bộ nhớ để xem lại khi cần.

Về phần mềm, chương trình khởi tạo hệ thống thời gian, chân vào ra và các ngắt cần thiết. Giao tiếp với cảm biến độ ẩm đất tuân theo chuỗi thời gian chuẩn. Chu kỳ đo được đặt ở mức một giây hoặc lâu hơn, có thể tăng lên nếu cần tiết kiệm năng lượng hoặc giảm xuống khi cần theo dõi biến động nhanh hơn.

Về tính năng điều khiển tự động, hệ thống sử dụng logic đơn giản dựa trên ngưỡng độ ẩm đất đã được thiết lập trước. Khi độ ẩm xuống dưới ngưỡng tối thiểu, relay được kích hoạt để bật máy bơm tưới nước. Khi độ ẩm đạt ngưỡng mong muốn, relay ngắt để dừng máy bơm. Cơ chế bảo vệ được tích hợp để ngăn máy bơm hoạt động khi mực nước trong bể chứa thấp hơn mức an toàn, tránh tình trạng chạy khô gây hư hỏng thiết bị. Thời gian hoạt động của máy bơm được ghi lại và hiển thị, giúp người dùng theo dõi lịch sử sử dụng và lên kế hoạch bảo trì.

Về giao diện giám sát, hệ thống cung cấp hai phương thức hiển thị thông tin. Thứ nhất là màn hình LCD I2C gắn trực tiếp trên thiết bị, hiển thị luân phiên các thông số như độ ẩm đất, mực nước, trạng thái máy bơm và thời gian hoạt động. Thứ hai là giao diện web được phát triển với HTML, CSS và JavaScript, cho phép người dùng truy cập từ xa qua WiFi. Giao diện web được thiết kế thân thiện, trực quan với các biểu đồ thanh tiến trình, màu sắc phân biệt trạng thái và cập nhật dữ liệu theo thời gian thực. Người dùng có thể chuyển đổi giữa chế độ tự động và thủ công, điều khiển máy bơm từ xa và xem lịch sử hoạt động của hệ thống.

## MỤC LỤC

1. GIỚI THIỆU .....	1
1.1 Tổng quan.....	1
1.2 Nhiệm vụ đề tài .....	1
1.2.1. Tìm hiểu về cảm biến độ ẩm đất và cảm biến mực nước: .....	1
1.2.2. Tìm hiểu về vi điều khiển ESP32 và kết nối WiFi: .....	2
1.2.3. Thiết kế và lập trình hệ thống điều khiển tự động:.....	2
2. LÝ THUYẾT .....	3
2.1. ESP 32: .....	3
2.2. Cảm biến độ ẩm đất: .....	4
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG: .....	5
3.1. Yêu cầu thiết kế:.....	5
3.2. Phân tích phương pháp thiết kế: .....	7
3.2.1. Phân tích lý do lựa chọn vi điều khiển: .....	7
3.2.2. Phân tích cách đọc độ ẩm: .....	8
3.2.3. Phân tích cách điều khiển động cơ bơm:.....	8
3.2.4. Phân tích cách hiển thị thông tin: .....	8
3.2.5. Phân tích cách thiết kế nguồn: .....	9
3.3. Sơ đồ khối:.....	9
3.4. Mô hình phần cứng: .....	11
3.4.1 Sơ đồ mạch nguồn: .....	11
3.4.2 Sơ đồ nối chân:.....	11
3.4.3 Mạch hàn:.....	12
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM: .....	12
4.1. Cấu hình Hệ thống:.....	12

4.2.	Triển khai đa nhiệm:.....	13
4.3.	Sơ đồ giải thuật .....	15
5.	KẾT QUẢ THỰC HIỆN.....	16
6.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	18
6.1	Kết luận: .....	18
6.1.1	Những kết quả đạt được:.....	18
6.1.2	Ưu điểm của hệ thống:.....	19
6.1.3	Hạn chế:.....	19
6.2	Hướng phát triển: .....	20
6.2.1	Nâng cấp hệ thống: .....	20
6.2.2	Ứng dụng thực tế: .....	20
7.	TÀI LIỆU THAM KHẢO.....	21
8.	PHỤ LỤC.....	21

## DANH SÁCH THÀNH VIÊN

Họ và tên	MSSV	Nhiệm vụ phân công	Tiến độ hoàn thành
Nguyễn Thành Nhân	2212368	Thiết kế phần mềm + hỗ trợ thiết kế hệ thống	100%
Từ Vĩnh Minh	2114085	Thiết kế phần cứng + hỗ trợ thiết kế phần mềm	100%

## 1. GIỚI THIỆU

### 1.1 Tổng quan

Trong bối cảnh nông nghiệp hiện đại đang chuyển mình theo hướng ứng dụng công nghệ cao và tự động hóa, việc quản lý nguồn nước tưới tiêu một cách hiệu quả không chỉ giúp tối ưu hóa năng suất cây trồng mà còn góp phần quan trọng trong việc tiết kiệm tài nguyên nước ngày càng khan hiếm. Hệ thống tưới nước truyền thống thường dựa vào kinh nghiệm và quan sát thủ công của người nông dân, dẫn đến nhiều hạn chế như khó kiểm soát lượng nước chính xác, tốn kém thời gian và công sức, đồng thời không thể đảm bảo sự đồng đều trong việc cung cấp nước cho cây trồng.

Nhận thức được những vấn đề này, đề tài "Thiết kế hệ thống tưới nước tự động thông minh dựa trên ESP32" được triển khai nhằm mục đích xây dựng một giải pháp công nghệ tiên tiến, có khả năng tự động giám sát các thông số môi trường quan trọng như độ ẩm đất và mực nước, từ đó đưa ra quyết định tưới tiêu hợp lý mà không cần sự can thiệp liên tục của con người. Hệ thống không chỉ phù hợp với các ứng dụng quy mô nhỏ như vườn rau gia đình, ban công chung cư hay nhà kính thí nghiệm, mà còn có tiềm năng mở rộng cho các ứng dụng nông nghiệp quy mô lớn hơn trong tương lai, góp phần thúc đẩy xu hướng nông nghiệp thông minh và bền vững tại Việt Nam.

### 1.2 Nhiệm vụ đề tài

#### 1.2.1. Tìm hiểu về cảm biến độ ẩm đất và cảm biến mực nước:

**Yêu cầu:** Nghiên cứu nguyên lý hoạt động của cảm biến độ ẩm đất điện dung, cảm biến mực nước điện trở, khả năng đo lường trong môi trường ngoài trời, độ chính xác và độ bền.

**Kết quả cần đạt:** Hiểu rõ cách đọc giá trị ADC từ cảm biến, chuyển đổi thành phần trăm độ ẩm/mực nước, xác định ngưỡng khô/ướt phù hợp với từng loại cây trồng.

**Cách tiếp cận – Ý tưởng:** Sử dụng ADC 12-bit của ESP32 với kỹ thuật lấy mẫu trung bình (64 mẫu), hiệu chuẩn điểm khô (3500) và điểm ướt (1425) để ánh xạ tuyến tính sang phần trăm. Áp dụng ESP-IDF ADC calibration API để tăng độ chính xác.

### 1.2.2. Tìm hiểu về vi điều khiển ESP32 và kết nối WiFi:

**Yêu cầu:** Nghiên cứu kiến trúc ESP32, các chức năng GPIO, ADC, I2C, WiFi station mode, ESP-IDF framework và FreeRTOS.

**Kết quả cần đạt:** Lập trình ESP32 kết nối WiFi thành công, tạo HTTP server để phục vụ giao diện web và API RESTful cho ứng dụng IoT.

**Cách tiếp cận – Ý tưởng:** Sử dụng ESP HTTP Server component để tạo web server, thiết kế API endpoints: GET /api/sensor-data, POST /api/control-pump, POST /api/auto-mode.

### 1.2.3. Thiết kế và lập trình hệ thống điều khiển tự động:

**Yêu cầu:** Xây dựng logic điều khiển máy bơm tự động dựa trên ngưỡng độ ẩm (40%-70%) và mực nước (>15%), tích hợp màn hình LCD I2C 16x2 hiển thị thông tin, và giao diện web responsive để giám sát/điều khiển từ xa.

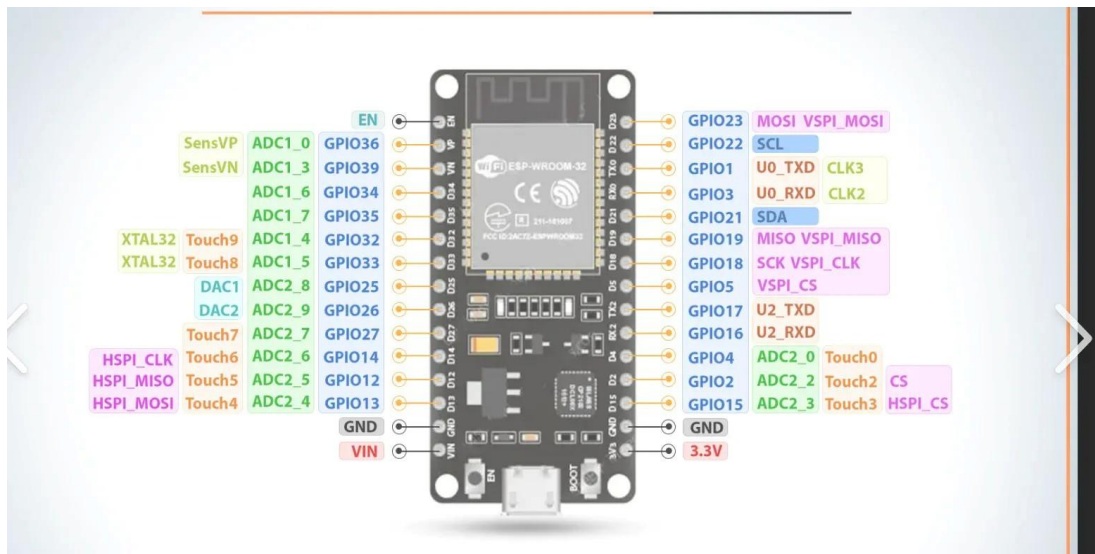
**Kết quả cần đạt:** Hệ thống hoạt động ổn định, tự động bật/tắt máy bơm khi cần, hiển thị dữ liệu real-time trên LCD và web dashboard, hỗ trợ chế độ tự động/thủ công, bảo vệ máy bơm khi mực nước thấp, thống kê thời gian hoạt động.

#### Cách tiếp cận – Ý tưởng:

- **Kiến trúc đa nhiệm với FreeRTOS:** Tạo 4 task độc lập (soil\_moisture\_task, water\_level\_task, pump\_control\_task, lcd\_display\_task) chạy song song, chu kỳ đọc cảm biến 2-3 giây.
- **Thuật toán điều khiển hysteresis:** Bật bơm khi độ ẩm < 40%, tắt khi >= 70% để tránh dao động.
- **Cơ chế bảo vệ:** Kiểm tra mực nước > 15% trước khi cho phép bơm hoạt động, tránh hư hỏng động cơ.
- **Giao diện LCD đa chế độ:** Xoay vòng hiển thị độ ẩm đất, mực nước, trạng thái bơm mỗi 2 giây.
- **Web dashboard responsive:** HTML5/CSS3 với gradient design, real-time update qua AJAX mỗi 2 giây, progress bar trực quan, toggle switch cho chế độ auto/manual.
- **JSON API:** Truyền dữ liệu lightweight giữa ESP32 và web client, hỗ trợ CORS để truy cập từ mọi thiết bị trong mạng LAN.

## 2. LÝ THUYẾT

### 2.1. ESP 32:



Vi điều khiển ESP32 là một hệ thống vi điều khiển trên chip (SoC) 32 bit của Espressif Systems, sử dụng bộ vi xử lý Tensilica Xtensa LX6 lõi kép với tần số hoạt động lên đến 240 MHz. Bên trong chip tích hợp bộ nhớ 520 KB SRAM, 448 KB ROM và bộ nhớ Flash ngoài 4 MB để chứa chương trình và dữ liệu. ESP32 hoạt động với điện áp 2.0-3.6V (khuyến nghị 3.3V), được trang bị mạch giám sát nguồn POR, PDR, PVD giúp khởi động ổn định và tránh treo máy khi nguồn không ổn định.

Đặc điểm nổi bật của ESP32 là tích hợp sẵn WiFi 802.11 b/g/n (2.4 GHz) với tốc độ lên đến 150 Mbps và Bluetooth v4.2 (Classic + BLE), cho phép thiết bị kết nối không dây với Internet và các thiết bị di động một cách dễ dàng. Chip có hai lõi CPU (PRO\_CPU và APP\_CPU) hoạt động độc lập, cho phép một lõi xử lý kết nối mạng trong khi lõi còn lại thực thi chương trình chính.

Về ngoại vi, ESP32 tích hợp 30-36 chân GPIO có thể cấu hình linh hoạt, 18 kênh ADC 12-bit để đọc tín hiệu analog, 2 kênh DAC 8-bit xuất analog, 16 kênh PWM điều khiển LED và động cơ. Chip hỗ trợ các giao tiếp chuẩn: 3 UART, 4 SPI, 2 I2C, 2 I2S, phục vụ kết nối với cảm biến, màn hình, thẻ nhớ và các module khác. Ngoài ra còn có 10 cảm biến cảm ứng điện dung (capacitive touch), cảm biến Hall Effect và cảm biến nhiệt độ tích hợp sẵn.

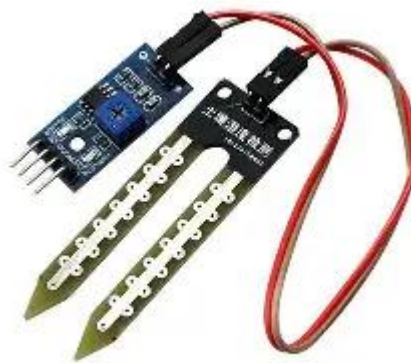
Hệ thống quản lý năng lượng của ESP32 rất linh hoạt với các chế độ: Active Mode (hoạt động đầy đủ), Modem Sleep (tắt WiFi/Bluetooth), Light Sleep (tạm dừng CPU), Deep

Sleep (chỉ giữ RTC, tiêu thụ dưới  $10\mu A$ ) và Hibernation (tắt hầu hết chức năng). Các chế độ này giúp ESP32 phù hợp cho thiết bị IoT chạy pin, có thể hoạt động hàng tháng mà không cần sạc.

ESP32 có thể lập trình bằng nhiều công cụ: ESP-IDF (framework chính thức dùng C/C++ với FreeRTOS), Arduino IDE (dễ học cho người mới), MicroPython (lập trình Python), và PlatformIO (IDE chuyên nghiệp). Chip tích hợp bảo mật phần cứng AES, SHA-2, RSA, Secure Boot và Flash Encryption để bảo vệ dữ liệu.

Với hiệu năng mạnh, WiFi/Bluetooth tích hợp, ngoại vi phong phú, quản lý năng lượng tốt và giá thành rẻ, ESP32 là lựa chọn phổ biến cho các ứng dụng IoT, nhà thông minh, thiết bị đeo, tự động hóa công nghiệp, và hệ thống nhúng kết nối không dây.

## 2.2. Cảm biến độ ẩm đất:



Cảm biến độ ẩm đất (Soil Moisture Sensor) là thiết bị điện tử dùng để đo lượng nước có trong đất, hoạt động dựa trên nguyên lý đo điện trở hoặc điện dung của môi trường đất. Cảm biến thường gồm hai phần chính: đầu dò (probe) có hai điện cực kim loại được cắm vào đất, và mạch xử lý tín hiệu chuyển đổi giá trị điện trở thành tín hiệu điện áp analog hoặc digital. Thiết bị này cung cấp dữ liệu độ ẩm đất theo thời gian thực, giúp theo dõi tình trạng đất và hỗ trợ các ứng dụng trong nông nghiệp thông minh, hệ thống tưới tiêu tự động và nghiên cứu môi trường.

Nguyên lý hoạt động của cảm biến điện trở dựa trên sự thay đổi điện trở của đất theo độ ẩm. Khi đất ẩm, nước trong đất chứa các ion và khoáng chất làm tăng khả năng dẫn điện, điện trở giữa hai điện cực giảm xuống. Ngược lại, khi đất khô, không khí chiếm nhiều



khoảng trống, điện trở tăng lên do khả năng dẫn điện kém. Mạch xử lý đo điện trở này và chuyển đổi thành tín hiệu điện áp tương ứng: điện áp cao khi đất khô, điện áp thấp khi đất ẩm. Tín hiệu analog này được đưa vào vi điều khiển qua chân ADC để đọc và xử lý.

Cấu tạo phần cứng của cảm biến gồm hai điện cực kim loại (thường là thép không gỉ hoặc đồng mạ vàng) được cố định song song trên tấm mạch PCB màu vàng, tạo thành đầu dò cắm vào đất. Module xử lý tín hiệu (thường màu xanh) chứa mạch so sánh LM393 hoặc IC tương đương, cung cấp cả tín hiệu analog (AO) và digital (DO). Chân VCC kết nối nguồn 3.3V-5V, chân GND nối mass, chân AO xuất tín hiệu analog 0-3.3V hoặc 0-5V tùy nguồn cấp, chân DO xuất tín hiệu digital 0/1 dựa trên ngưỡng cài đặt qua biến trở. Module thường có LED chỉ thị nguồn và LED chỉ thị trạng thái khi độ ẩm vượt ngưỡng.

Để kết nối với vi điều khiển như ESP32 hoặc Arduino, chân VCC nối với 3.3V hoặc 5V, GND nối mass chung, chân AO nối với chân ADC của vi điều khiển để đọc giá trị analog (GPIO34-39 trên ESP32), chân DO nối với GPIO thường để đọc tín hiệu digital khi cần báo động ngưỡng. Trong code, sử dụng hàm `analogRead()` để đọc giá trị từ 0-4095 (ESP32) hoặc 0-1023 (Arduino), sau đó chuyển đổi thành phần trăm độ ẩm qua công thức calibration. Cần lưu ý calibrate cảm biến bằng cách đo giá trị trong không khí (khô hoàn toàn) và trong nước (ẩm tối đa) để xác định ngưỡng min-max.

Để sử dụng hiệu quả và kéo dài tuổi thọ, nên chỉ cấp nguồn cho cảm biến khi đo, tắt nguồn khi không dùng để giảm ăn mòn điện cực. Sử dụng transistor hoặc relay điều khiển nguồn cảm biến qua GPIO của vi điều khiển. Bảo vệ điện cực bằng sơn cách điện hoặc epoxy ở phần mạch, chỉ để lộ phần kim loại cần đo. Đọc giá trị nhiều lần rồi lấy trung bình để giảm nhiễu và dao động.

### 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG:

#### 3.1. Yêu cầu thiết kế:

Thiết kế và xây dựng hệ thống tưới cây tự động thông minh sử dụng ESP32, có khả năng giám sát độ ẩm đất và điều khiển bơm nước tự động hoặc thủ công thông qua giao diện web.

**Yêu cầu chức năng:**

1. Giám sát độ ẩm đất:
  - Đọc giá trị cảm biến độ ẩm đất liên tục
  - Hiển thị độ ẩm đất theo thời gian thực.
  - Chu kỳ đọc: 1.5 giây
2. Điều khiển bơm nước
  - Chế độ tự động (AUTO):
    - + Bật bơm khi độ ẩm  $< 40\%$ , Tắt bơm khi độ ẩm  $> 50\%$ .
    - + Tránh bật/tắt liên tục, Kiểm tra chu kỳ: 2 giây.
  - Chế độ thủ công (MANUAL):
    - + Người dùng bật/tắt bơm qua giao diện web.
    - + Chuyển đổi giữa AUTO/MANUAL linh hoạt.
3. Hiện thị thông minh:
  - Màn hình LCD I2C (16x2):
    - + Dòng 1: Hiển thị độ ẩm đất (%)
    - + Dòng 2: Trạng thái bơm (ON/OFF)
    - + Cập nhật mỗi 1.5 giây
  - Giao diện Web:
    - + Hiển thị độ ẩm đất theo thời gian thực
    - + Hiển thị trạng thái bơm, trạng thái hoạt động.
    - + Nút điều khiển bơm thủ công
4. Kết nối mạng:
  - Kết nối WiFi tự động khi khởi động.
  - Tự động kết nối lại khi mất kết nối.
  - Hoạt động offline nếu không có WiFi (hiển thị LCD).

**Yêu cầu phần cứng:**

1. Vi điều khiển: ESP32 DevKit: Xử lý trung tâm, WiFi, đa nhiệm
2. Cảm biến độ ẩm đất Capacitive. Điện áp hoạt động: 3.3V – 5V, Đầu ra: Analog (ADC)

## 3. Thiết bị điều khiển:

- Module Relay 1 kênh: Điện áp kích: 3.3V – 5V, tải tối đa: 10A/250VAC hoặc 10A/30VDC
- Bơm nước mini DC: Điện áp: 5V - 12V DC, lưu lượng: 120-200 L/h

## 4. Hiển thị: LCD I2C 16x2

## 5. Nguồn cấp: Adapter 5V/2A.

**3.2. Phân tích phương pháp thiết kế:**

## 3.2.1. Phân tích lý do lựa chọn vi điều khiển:

Tiêu chí	Arduino Uno (ATmega328P)	ESP32 DevKit (Đã chọn)	Kit STM32F103C8T6
Kiến trúc & tốc độ	8-bit AVR, 16 MHz	32-bit Tensilica Xtensa dual-core, tối đa ~240 MHz	32-bit ARM Cortex-M3, tối đa ~72 MHz
Bộ nhớ	SRAM: 2 KB; Flash: 32 KB; EEPROM: 1 KB	SRAM: 520 KB; Flash: 4 MB; PSRAM: tùy chọn	SRAM: 20 KB; Flash: 64 KB; Không EEPROM
ADC & số chân I/O	10-bit ADC, ít chân I/O (14 digital, 6 analog)	12-bit ADC, nhiều chân I/O (34 GPIO, 18 kênh ADC), WiFi/Bluetooth tích hợp	12-bit ADC, khá nhiều chân I/O, thư viện phong phú
Khả năng kết nối & tính năng bổ sung	Cơ bản – không WiFi/Bluetooth tích hợp (cần thêm ESP8266/module ngoài)	Tích hợp WiFi/Bluetooth – không cần phần cứng thêm, tiết kiệm chi phí tổng	Không WiFi, cần module riêng nếu muốn IoT
Giá thành & đơn giản	~50–80k VND (nhưng +50k nếu cần WiFi module)	~60–100k VND (đã bao gồm WiFi/BT, tổng rẻ hơn)	~30–70k VND nhưng khó lập trình hơn
Hỗ trợ phần mềm & thư viện	Rất tốt cho người mới, cộng đồng lớn, Arduino IDE	ESP-IDF chuyên nghiệp + Arduino IDE, thư viện IoT phong phú, tài liệu tiếng Việt nhiều	STM32CubeIDE phức tạp, đường cong học tập dốc
Đa nhiệm (RTOS)	Không hỗ trợ – chạy tuần tự	FreeRTOS tích hợp – xử lý song song nhiều task	Hỗ trợ FreeRTOS nhưng cần cấu hình thủ công
Web Server	Không đủ RAM cho HTTP server	HTTP server ổn định, đủ RAM xử lý nhiều request	Đủ RAM nhưng thiếu WiFi, cần module ESP-01
Tiêu thụ điện	~40–50 mA (tiết kiệm nhất)	~160–260 mA (cao hơn nhưng phù hợp dự án có nguồn AC)	~30–100 mA

**Chọn ESP32 DevKit vì:**

- WiFi + Bluetooth đã có sẵn, không tốn thêm tiền mua module riêng và không tốn công đấu nối phức tạp
- Ram đủ lớn: 520KB đủ chạy web server ổn định + xử lý nhiều request đồng thời, Arduino Uno chỉ 2KB không đủ.
- Đa nhiệm: FreeRTOS cho phép đọc cảm biến + điều khiển bơm + xử lý web request song song không bị treo

**3.2.2. Phân tích cách đọc độ ẩm:**

Hệ thống sử dụng cảm biến độ ẩm đất được kết nối với bộ chuyển đổi tín hiệu analog sang digital của vi điều khiển. Cảm biến hoạt động dựa trên nguyên lý đo điện trở hoặc điện dung của đất - khi đất khô thì tín hiệu điện áp cao, khi đất ướt thì tín hiệu điện áp thấp. Vi điều khiển đọc giá trị điện áp này và chuyển đổi thành thông số phần trăm độ ẩm dựa trên các ngưỡng được hiệu chuẩn trước. Quá trình đọc được lặp lại liên tục theo chu kỳ để cập nhật trạng thái độ ẩm thời gian thực. Giá trị độ ẩm thu được sẽ được hiển thị trên màn hình và dùng để điều khiển hệ thống tưới tự động, đảm bảo cây trồng luôn được cung cấp đủ nước theo nhu cầu.

**3.2.3. Phân tích cách điều khiển động cơ bơm:**

Hệ thống điều khiển máy bơm thông qua relay được kết nối với chân GPIO của vi điều khiển. Máy bơm hoạt động theo hai chế độ: tự động và thủ công. Ở chế độ tự động, vi điều khiển liên tục theo dõi giá trị độ ẩm đất và so sánh với các ngưỡng đã đặt trước - khi độ ẩm xuống quá thấp thì tự động bật bơm, khi độ ẩm đạt mức cao hơn thì tắt bơm để tránh tưới thừa. Ở chế độ thủ công, người dùng có thể chủ động bật/tắt máy bơm thông qua giao diện web bất kể giá trị độ ẩm hiện tại. Trạng thái của máy bơm (bật/tắt) được hiển thị trực quan trên màn hình LCD và giao diện web, giúp người dùng dễ dàng giám sát. Hệ thống còn cung cấp API để chuyển đổi giữa hai chế độ hoặc điều khiển máy bơm từ xa qua mạng WiFi.

**3.2.4. Phân tích cách hiển thị thông tin:**

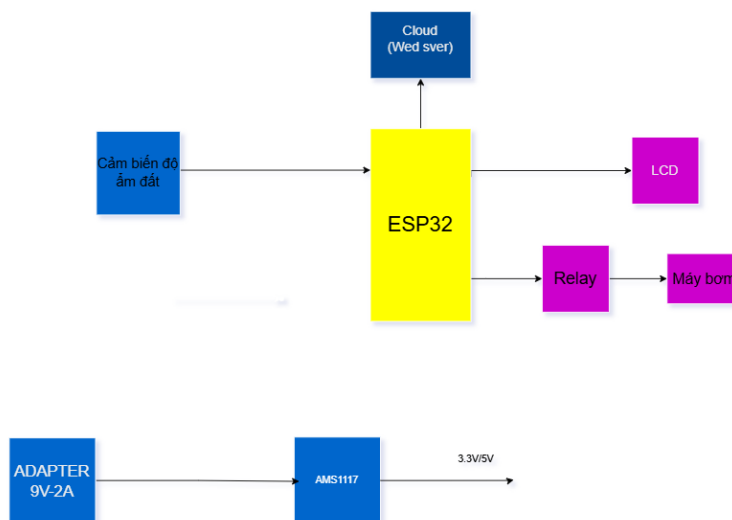
Hệ thống hiển thị thông tin qua hai kênh chính: màn hình LCD cục bộ và giao diện web từ xa. Màn hình LCD được kết nối qua giao thức I2C, hiển thị trực tiếp các thông số quan trọng như giá trị độ ẩm đất theo phần trăm và trạng thái máy bơm đang bật hay tắt. Thông tin trên LCD được cập nhật liên tục theo chu kỳ để phản ánh trạng thái thời gian thực của hệ thống. Đồng thời, hệ thống tích hợp web server cho phép người dùng truy cập thông

tin từ xa qua mạng WiFi bằng trình duyệt. Giao diện web hiển thị đầy đủ các thông số cảm biến, trạng thái thiết bị và các nút điều khiển tương tác. Dữ liệu được truyền tải qua các API endpoint dưới dạng JSON, cho phép giao diện web tự động làm mới thông tin mà không cần tải lại trang. Cách tiếp cận này giúp người dùng có thể giám sát và điều khiển hệ thống linh hoạt cả ở gần và từ xa.

### 3.2.5. Phân tích cách thiết kế nguồn:

Hệ thống sử dụng nguồn adapter 9V 2A làm nguồn cung cấp chính từ lưới điện. Điện áp 9V này được đưa vào mạch ổn áp ASM1117 để chuyển đổi xuống các mức điện áp phù hợp cho từng linh kiện. Mạch ASM1117 là IC ổn áp tuyến tính (linear regulator) có khả năng giảm và ổn định điện áp đầu ra, cung cấp nguồn 5V cho các thiết bị như relay, màn hình LCD và cảm biến, đồng thời cấp nguồn 3.3V cho vi điều khiển ESP32 và các module hoạt động ở điện áp thấp. Việc sử dụng IC ổn áp giúp bảo vệ các linh kiện điện tử khỏi biến động điện áp, lọc nhiễu và đảm bảo hệ thống hoạt động ổn định. Dòng điện tối đa 2A từ adapter đủ để cung cấp cho tất cả các thiết bị trong hệ thống bao gồm cả máy bơm nước khi hoạt động. Thiết kế nguồn này đơn giản, hiệu quả và đáng tin cậy cho các ứng dụng IoT trong nông nghiệp.

### 3.3. Sơ đồ khối:



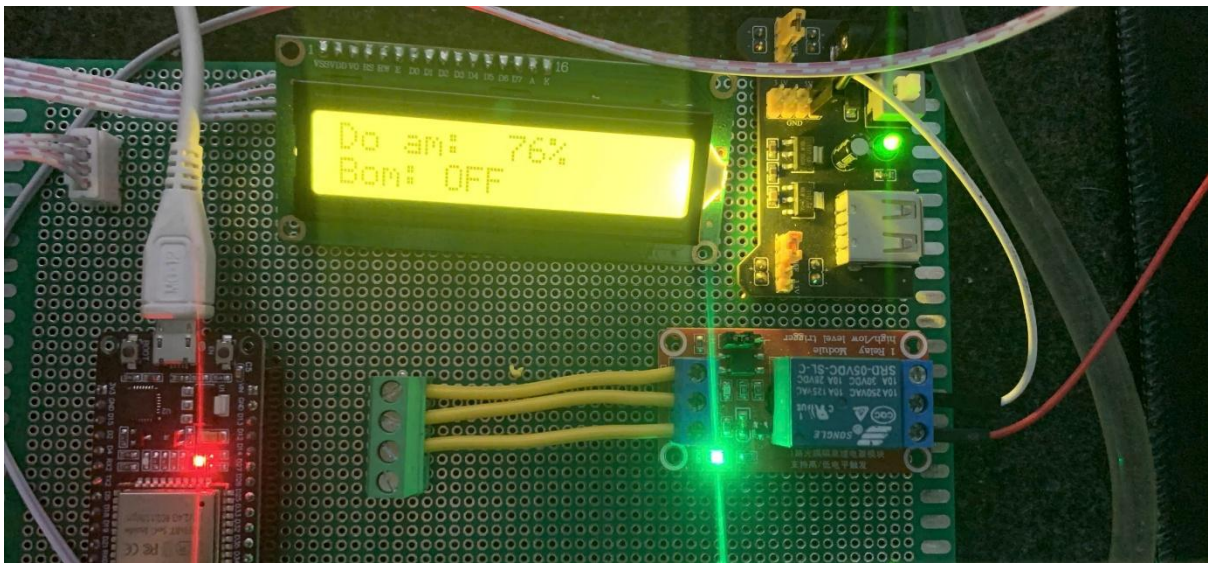
Hệ thống tưới cây tự động xây dựng dựa trên Kit Esp 32. Bao gồm các khối với các chức năng:

#### 1. Khối nguồn:

- Adapter cấp nguồn 9V và thông qua IC AMS1117 để hạ áp cung cấp điện áp 3.3V/5V dùng để cấp nguồn cho esp32, cũng như các khối khác trong mạch.
2. Khối Vi điều khiển ESP32:
- Là bộ não điều khiển hệ thống.
  - Kết nối WiFi.
  - Đọc dữ liệu từ cảm biến độ ẩm đất và cảm biến nước.
  - Xử lý logic điều khiển máy bơm (tự động/thủ công).
  - Hiển thị thông tin lên LCD.
  - Chạy web server để điều khiển từ xa qua HTTP API.
3. Khối Cảm biến độ ẩm đất:
- Đo độ ẩm trong đất bằng phương pháp điện trở.
  - Trả về tín hiệu analog (điện áp) tỷ lệ với độ ẩm.
  - Kết nối với ESP32 qua ADC (Analog-to-Digital Converter).
4. Khối LCD I2C (16x2):
- Hiển thị thông tin độ ẩm đất theo thời gian thực.
  - Hiển thị trạng thái máy bơm (ON/OFF).
  - Giao tiếp với ESP32 qua chuẩn I2C.
5. Khối Relay + Máy bơm:
- Relay đóng vai trò công tắc điện tử.
  - Điều khiển bật/tắt máy bơm nước.
  - Ngăn cách mạch điều khiển (3.3V) và mạch động lực (220V/12V)
  - ESP32 (GPIO23) xuất tín hiệu HIGH (3.3V) → Relay đóng → Máy bơm hoạt động.
  - ESP32 xuất tín hiệu LOW (0V) → Relay mở → Máy bơm ngừng.
6. Khối Cloud (Web Server):
- ESP32 tạo HTTP server cục bộ trên mạng WiFi.
  - Cung cấp giao diện web để giám sát và điều khiển



## 3.4.3 Mạch hàn:



## 4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM:

Phần mềm của hệ thống được xây dựng dựa trên mô hình đa nhiệm song song (Multi-tasking) sử dụng FreeRTOS. Hệ thống bao gồm các tác vụ (task) độc lập chạy đồng thời, đảm bảo việc đọc cảm biến, điều khiển bơm và cập nhật giao diện diễn ra liên tục không bị gián đoạn.

## 4.1. Cấu hình Hệ thống:

## 1. WIFI:

- Khởi tạo: Thiết lập driver WiFi, khởi tạo esp\_netif, tạo vòng lặp sự kiện mặc định (esp\_event\_loop\_create\_default) và kết nối tới SSID/PASS đã cấu hình.
- Event Handling: Sử dụng s\_wifi\_event\_group và bit WIFI\_CONNECTED\_BIT để quản lý trạng thái kết nối. Tự động thử kết nối lại khi mất kết nối (WIFI\_EVENT\_STA\_DISCONNECTED).

## 2. Cảm biến Độ ẩm Đất (Soil Moisture ADC):

- ADC\_CHANNEL\_0: đọc dữ liệu DC
- SOIL\_ADC\_DRY: Giá trị ADC đại diện cho Đất khô.
- SOIL\_ADC\_WET: Giá trị ADC đại diện cho Đất ướt.
- MOISTURE\_LOW\_THRESHOLD: Ngưỡng độ ẩm bật bơm.
- MOISTURE\_HIGH\_THRESHOLD: Ngưỡng độ ẩm tắt bơm.

## 3. Giao tiếp I2C (cho LCD):



- SCL
- SDA

#### 4. ADC (Analog-to-Digital Converter):

- Thực hiện việc Hiệu chuẩn (Calibration) bằng phương pháp line\_fitting nếu được hỗ trợ để chuyển đổi giá trị Raw ADC sang mili-volt chính xác hơn.
- Chức năng soil\_moisture\_percent(uint32\_t mV) chuyển đổi giá trị mV sang phần trăm độ ẩm bằng công thức nội suy tuyến tính giữa giá trị mV của đất khô và đất ướt

$$\text{Phan tram do am} = \frac{(\text{SOIL\_ADC\_DRY-mV}) \times 100}{(\text{SOIL\_ADC\_DRY-SOIL\_ADC\_WET})}$$

#### 5. HTTP Web Server:

- Khởi tạo và chạy sau khi kết nối WiFi.
- Các API:
  - + /api/sensor-data: trả về dữ liệu JSON chứa độ ẩm đất (soilMoisture), trạng thái bơm (pumpStatus), và chế độ tự động (autoMode).
  - + /api/control-pump: API điều khiển bơm thủ công (bật/tắt).
  - + /api/auto-mode: API bật/tắt chế độ tự động.

### 4.2. Triển khai đa nhiệm:

#### 1. soil\_task (Đọc ADC và cập nhật LCD):

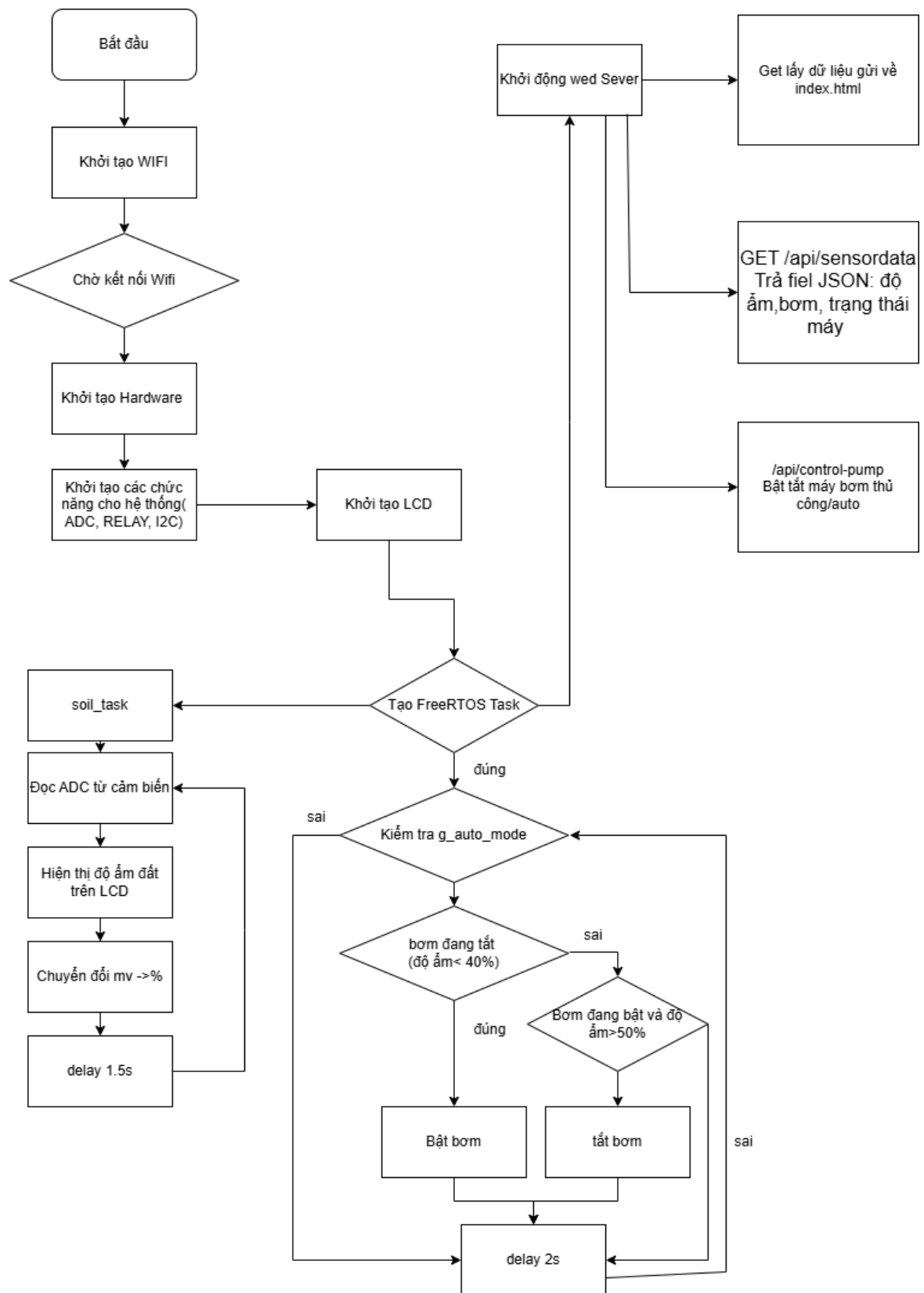
- Chức năng: Thực hiện việc đọc giá trị độ ẩm đất từ cảm biến ADC, chuyển đổi sang đơn vị mili-volt (mV) , sau đó tính toán ra phần trăm độ ẩm.
- Cập nhật Giao diện: Hiển thị phần trăm độ ẩm và trạng thái bơm lên màn hình LCD thông qua giao tiếp I2C.
- Tần suất: Chạy lặp lại sau mỗi 1500 ms.

#### 2. pump\_task (Tự động điều khiển Bơm):

- Chức năng: Là logic điều khiển tự động của hệ thống.
- Điều kiện Bật Bơm: Nếu hệ thống đang ở chế độ Tự động (g\_auto\_mode = true) và Bơm đang Tắt (!g\_pump\_status) và Độ ẩm thấp hơn Ngưỡng thấp (g\_soil\_percent < MOISTURE\_LOW\_THRESHOLD (40%)), thì Bật Bơm (pump\_control(true)).

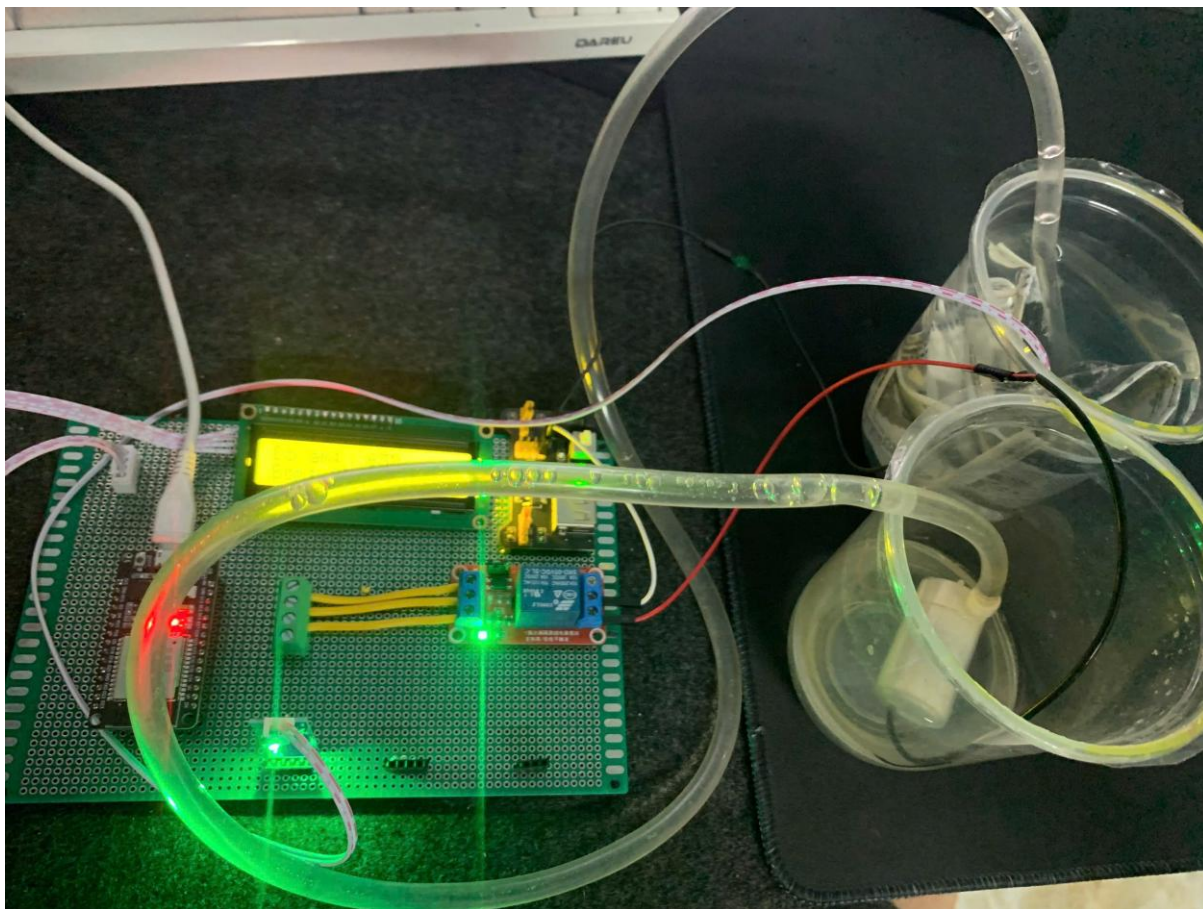
- Điều kiện Tắt Bơm: Nếu hệ thống đang ở chế độ Tự động (`g_auto_mode = true`) và Bơm đang Bật (`g_pump_status`) và Độ ẩm cao hơn Ngưỡng cao (`g_soil_percent > MOISTURE_HIGH_THRESHOLD (50\%)`), thì Tắt Bơm (`pump_control(false)`).
- Tần suất: Chạy lặp lại sau mỗi 2000 ms.

### 4.3. Sơ đồ giải thuật

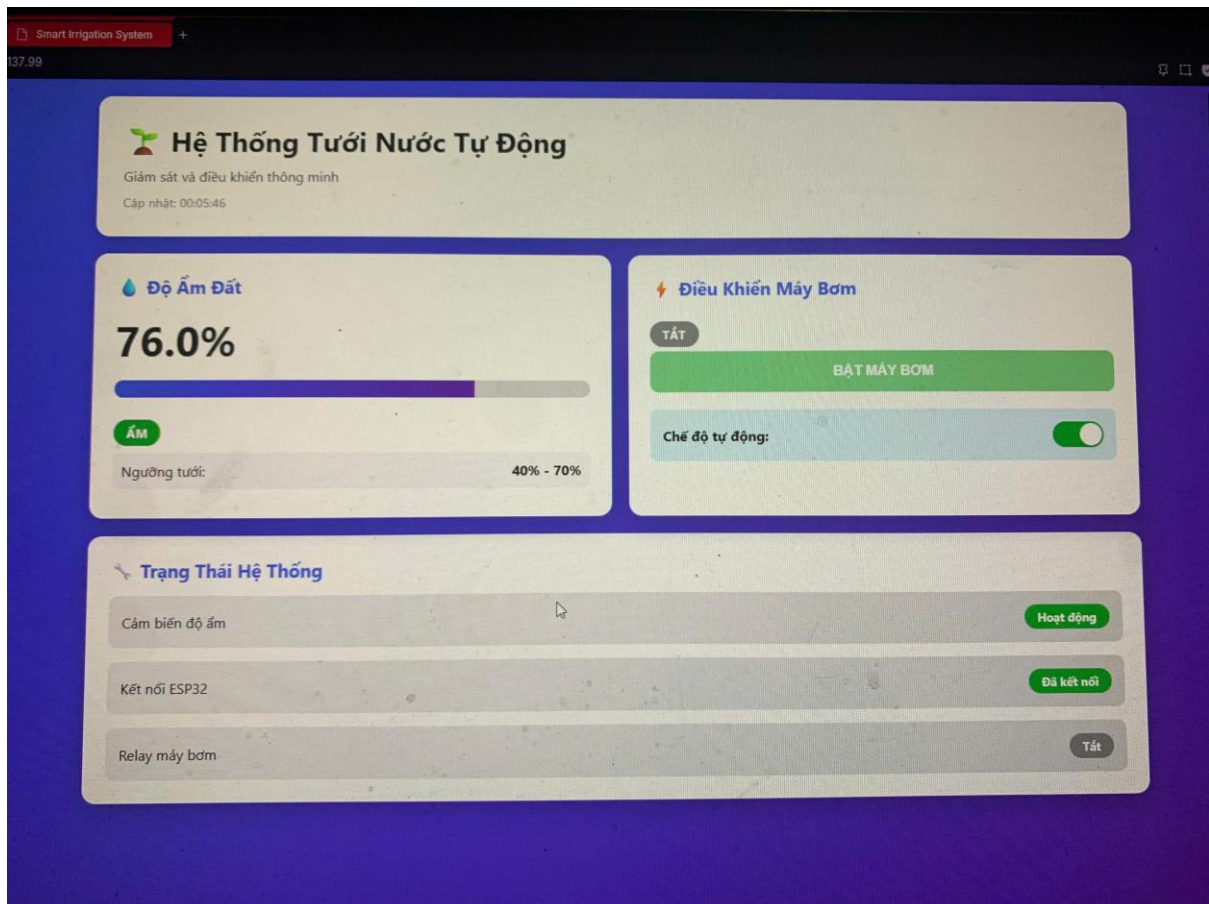


## 5. KẾT QUẢ THỰC HIỆN

### 5.1. Phần cứng:



### 5.2. Phần mềm:



## 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận:

Qua quá trình nghiên cứu và thực hiện đề tài "Hệ thống tưới cây tự động dựa trên ESP32", sinh viên đã đạt được những kết quả quan trọng sau:

#### 6.1.1 Những kết quả đạt được:

Về phần cứng:

Thiết kế và lắp ráp thành công hệ thống phần cứng hoàn chỉnh bao gồm: vi điều khiển ESP32, cảm biến độ ẩm đất (kết nối ADC GPIO36), module relay điều khiển bơm nước (GPIO23), và màn hình LCD 16x2 I2C (địa chỉ 0x27) hiển thị thông tin thời gian thực.

Hệ thống hoạt động ổn định với nguồn cấp 5V, tiêu thụ điện năng thấp phù hợp cho ứng dụng thực tế lâu dài.

Cảm biến độ ẩm đất sử dụng ADC 12-bit với khả năng đo chính xác giá trị điện áp từ 1425mV (ướt) đến 3500mV (khô), cho phép tính toán độ ẩm theo phần trăm với độ chính xác cao.

Về phần mềm:

Phát triển thành công firmware cho ESP32 sử dụng ESP-IDF framework với FreeRTOS, tận dụng khả năng xử lý đa nhiệm để vận hành các tác vụ độc lập:

- soil\_task: Đọc cảm biến và cập nhật LCD mỗi 1.5 giây
- pump\_task: Điều khiển tự động bơm nước mỗi 2 giây
- HTTP server: Xử lý các yêu cầu API từ giao diện web

Xây dựng thuật toán điều khiển thông minh dựa trên ngưỡng độ ẩm hysteresis (40% - 50%):

Bật bơm khi độ ẩm < 40% (ngưỡng thấp)

Tắt bơm khi độ ẩm > 50% (ngưỡng cao)

Cơ chế này tránh được hiện tượng bật/tắt liên tục, kéo dài tuổi thọ thiết bị

Về chức năng:

Chế độ tự động: Hệ thống tự động bật/tắt bơm dựa trên độ ẩm đất đo được, không cần can thiệp của người dùng, phù hợp cho việc chăm sóc cây trồng liên tục.

Chế độ thủ công: Cho phép người dùng điều khiển bơm nước trực tiếp qua giao diện web khi cần tưới ngoài lịch trình tự động.

Hiển thị thời gian thực: Độ ẩm đất và trạng thái bơm được cập nhật liên tục trên màn hình LCD 16x2 và giao diện web, giúp người dùng theo dõi trực quan.

Khả năng hoạt động offline: Hệ thống vẫn duy trì chức năng tự động tưới ngay cả khi mất kết nối WiFi, đảm bảo cây trồng luôn được chăm sóc.

#### 6.1.2 Ưu điểm của hệ thống:

Tiết kiệm nước: Tưới chính xác theo nhu cầu thực tế của cây, tránh lãng phí nước so với tưới thủ công hoặc theo lịch cố định.

Tiết kiệm thời gian và công sức: Không cần theo dõi và tưới cây thủ công hàng ngày, đặc biệt hữu ích khi người dùng đi vắng.

Chi phí thấp: Sử dụng linh kiện phổ biến và giá rẻ (ESP32, cảm biến capacitive, relay), tổng chi phí dưới 500.000 VNĐ.

Độ tin cậy cao: Sử dụng FreeRTOS đảm bảo các tác vụ chạy độc lập, không ảnh hưởng lẫn nhau.

#### 6.1.3 Hạn chế:

Mặc dù đạt được nhiều kết quả tích cực, hệ thống vẫn còn một số hạn chế:

Chỉ hỗ trợ một cảm biến độ ẩm: Hiện tại chỉ đo độ ẩm tại một điểm, chưa phản ánh chính xác toàn bộ khu vực trồng cây.

Chưa có lưu trữ dữ liệu: Dữ liệu độ ẩm chưa được ghi nhận theo thời gian để phân tích xu hướng và tối ưu hóa lịch tưới.

Chưa có cảnh báo: Chưa có thông báo qua email/SMS khi có sự cố (cảm biến lỗi, bơm không hoạt động).



Phụ thuộc WiFi cho giám sát: Khi mất WiFi, không thể giám sát từ xa mặc dù chức năng tự động vẫn hoạt động.

Chưa tối ưu năng lượng: Chưa có chế độ tiết kiệm pin cho ứng dụng sử dụng pin mặt trời.

## 6.2 Hướng phát triển:

### 6.2.1 Nâng cấp hệ thống:

Tích hợp cảm biến nhiệt độ và độ ẩm không khí (DHT22 hoặc BME280) để đánh giá môi trường tổng thể.

Gửi email/SMS khi phát hiện bất thường (cảm biến lỗi, bơm không hoạt động, độ ẩm quá thấp/cao).

Thêm tính năng xem lịch sử dữ liệu, biểu đồ thống kê.

Tích hợp điều khiển giọng nói qua Google Assistant hoặc Alexa.

Thêm tính năng xem lịch sử dữ liệu, biểu đồ thống kê.

Áp dụng chế độ Deep Sleep của ESP32 để giảm tiêu thụ điện, chỉ thức dậy khi cần đo hoặc tưới

Tích hợp pin lithium 18650 và module sạc năng lượng mặt trời.

Áp dụng machine learning để học thói quen tiêu thụ nước của từng loại cây, tự động điều chỉnh ngưỡng tưới.

### 6.2.2 Ứng dụng thực tế:

Áp dụng trong nông nghiệp công nghệ cao: nhà kính thông minh, vườn rau thủy canh.

Xây dựng bộ cài đặt dễ dàng (Easy Setup) qua WiFi hotspot.

Phát triển hệ thống đa node: một gateway ESP32 kết nối với nhiều node cảm biến qua ESP-NOW hoặc LoRa để giám sát vườn lớn.



## 7. TÀI LIỆU THAM KHẢO

- [1] Espressif Systems, "ESP-IDF Programming Guide," Espressif Systems, 2024. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/>
- [2] Espressif Systems, "ESP32 Technical Reference Manual," Version 4.8, Espressif Systems, 2024.
- [3] Real Time Engineers Ltd, "FreeRTOS Reference Manual," 2024. [Online]. Available: [https://www.freertos.org/Documentation/RTOS\\_book.html](https://www.freertos.org/Documentation/RTOS_book.html)
- [4] Espressif Systems, "ESP32 ADC OneShot Mode Example," ESP-IDF Examples Repository, 2024. [Online]. Available: <https://github.com/espressif/esp-idf/tree/master/examples/peripherals/adc>
- [5] Espressif Systems, "ESP32 I2C Driver Documentation," ESP-IDF API Reference, 2024.
- [6] Espressif Systems, "ESP32 HTTP Server Documentation," ESP-IDF API Reference, 2024.
- [7] "Capacitive Soil Moisture Sensor v1.2 User Manual," DFRobot, 2023.
- [8] "LCD1602 with I2C Interface Datasheet," PCF8574 I2C Adapter, 2023.
- [9] [11] Espressif Systems, "ESP32 WiFi Driver Documentation," ESP-IDF API Reference, 2024.
- [10] Wi-Fi Alliance, "Wi-Fi Protected Access (WPA2) Specification," Version 2.0, 2022.
- [11] N.V. Hiếu và Đ.T. Minh, "Ứng dụng IoT trong nông nghiệp thông minh: Thực trạng và triển vọng," Tạp chí Khoa học và Công nghệ, Đại học Bách Khoa TP.HCM, 2023.
- [12]

## 8. PHỤ LỤC