

TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG
KHOA CÔNG NGHỆ THÔNG TIN, ROBOT
VÀ TRÍ TUỆ NHÂN TẠO

---o0o---

TIỂU LUẬN MÔN HỌC
LẬP TRÌNH HỆ THỐNG

Đề tài:

Thiết kế mô hình tưới cây tự động

Người hướng dẫn: Lê Duy Hùng

Sinh viên thực hiện : Vũ Duy Hoàng 22050038

Nguyễn Hoàng Vũ 22050072

Hồ Chí Minh, tháng 8 năm 2025

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN

1. Nhận xét:

a) Những kết quả đạt được:

.....

.....

.....

.....

.....

b) Những hạn chế:

.....

.....

.....

.....

.....

2. Điểm đánh giá

.....

.....

.....

.....

Bình Dương, ngày ... tháng ... năm 2025

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trước tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Lê Duy Hùng, giảng viên phụ trách môn *Lập trình Hệ thống*. Trong suốt quá trình học tập, Thầy đã luôn tận tình giảng dạy, truyền đạt cho chúng em những kiến thức chuyên môn quý báu cũng như những định hướng thực tế, giúp chúng em có cái nhìn toàn diện hơn về lập trình hệ thống – một môn học mang tính ứng dụng cao và đóng vai trò nền tảng trong lĩnh vực công nghệ thông tin.

Nhờ sự hướng dẫn chi tiết, sự kiên nhẫn giải đáp mọi thắc mắc và tinh thần nhiệt huyết của Thầy, chúng em không chỉ tiếp thu được nhiều kiến thức quan trọng, mà còn rèn luyện được tư duy phân tích, khả năng giải quyết vấn đề và tính cẩn trọng khi lập trình. Đây là những kỹ năng vô cùng cần thiết để chúng em có thể tự tin hơn trong học tập cũng như trong công việc sau này.

Bên cạnh đó, Thầy còn luôn tạo điều kiện thuận lợi, khuyến khích chúng em chủ động nghiên cứu, thảo luận và hợp tác nhóm, từ đó giúp chúng em học hỏi được nhiều kinh nghiệm quý giá trong việc làm việc nhóm và phát triển kỹ năng mềm. Quá trình thực hiện bài báo cáo/tiểu luận này chính là cơ hội để chúng em vận dụng và củng cố lại toàn bộ kiến thức đã được học, đồng thời rèn luyện tinh thần trách nhiệm và sự sáng tạo.

Chúng em xin chân thành cảm ơn sự tận tụy và nhiệt huyết của Thầy trong suốt học kỳ vừa qua. Một lần nữa, chúng em kính chúc Thầy luôn dồi dào sức khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy cũng như trong cuộc sống.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Thông tin chung.....	1
1.2. Bối cảnh và lý do chọn đề tài.....	1
1.3. Mục tiêu nghiên cứu.....	1
1.3.1. Mục tiêu tổng quát:	1
1.3.2. Mục tiêu cụ thể	2
1.3.3. Tiêu chí đánh giá kết quả	2
1.4. Phạm vi nghiên cứu và đối tượng áp dụng	2
1.4.1. Phạm vi nghiên cứu.....	2
1.4.2. Đối tượng áp dụng.....	2
1.5. Phương pháp nghiên cứu và công cụ hỗ trợ	3
1.5.1. Phương pháp nghiên cứu.....	3
1.5.2. Công cụ phần cứng.....	3
1.5.3. Công cụ phần mềm	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN.....	4
2.1. Khái niệm về hệ thống tưới cây tự động	4
2.2. Nguyên lý hoạt động chung của hệ thống điều khiển tự động	4
2.3. Tổng quan về các linh kiện phần cứng.....	5
2.3.1. Vi điều khiển ESP32.....	5
2.3.2. Cảm biến độ ẩm đất	5
2.3.3. Cảm biến mưa	5
2.3.4. Cảm biến DHT22 (nhiệt độ và độ ẩm không khí)	5
2.3.5. Cảm biến ánh sáng (LDR).....	5
2.3.6. Cảm biến mực nước	5
2.3.7. Relay điều khiển.....	5
2.3.8. Bơm nước	5
2.4. Các giao thức truyền thông và lập trình hệ thống: MQTT, Node-RED, FreeRTOS	6

2.4.1. Giao thức MQTT.....	6
2.4.2. Node-RED	6
2.4.3. FreeRTOS trên ESP32.....	6
2.5. Các công trình, nghiên cứu, sản phẩm liên quan	6
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	7
3.1. Yêu cầu hệ thống	7
3.1.1. Yêu cầu chức năng	7
3.1.2. Yêu cầu phi chức năng.....	7
3.2. Kiến trúc tổng thể của hệ thống tưới cây tự động.....	7
3.3. Mô hình hoạt động (sơ đồ khối, sơ đồ use case, sơ đồ luồng dữ liệu).....	8
3.3.1. Sơ đồ khối hệ thống.....	8
3.3.2. Sơ đồ Use Case.....	8
3.3.3. Sơ đồ luồng dữ liệu (DFD mức khái quát).....	8
3.4. Thiết kế phần cứng.....	9
3.5. Thiết kế phần mềm.....	9
3.5.1. Cấu trúc chương trình	9
3.5.2. Các module chính.....	9
3.5.3. Giao diện giám sát (Node-RED Dashboard)	9
CHƯƠNG 4. TRIỂN KHAI VÀ THỰC NGHIỆM	10
4.1. Cài đặt phần mềm điều khiển (lập trình cho ESP32)	10
4.2. Cấu hình giao tiếp	10
4.3. Kết quả thử nghiệm	11
4.4. Đánh giá hệ thống.....	11
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	12
5.1. Tóm tắt những kết quả đạt được	12
5.2. Đánh giá hiệu quả hệ thống tưới cây tự động.....	12
5.3. Hướng cải tiến và phát triển	12
5.4. Kết luận chung.....	13
TÀI LIỆU KHAM KHẢO.....	14

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Thông tin chung

- Tên đề tài: thiết kế hệ thống tưới cây tự động
- Danh sách thành viên của nhóm: Vũ Duy Hoàng

Nguyễn Hoàng Vũ

- Ngày hoàn thành: 25/08/2025
- Hệ thống triển khai: Hệ thống tưới cây tự động
- Link github: Folder chứa dockerfile, docker-compose, file code ino để nhúng vào thiết bị,...
- Giới thiệu thiết bị sẽ sử dụng: bộ điều khiển (ESP32), cảm biến nhiệt độ và độ ẩm (DHT22), cảm biến ánh sáng (LDR), cảm biến độ ẩm đất, cảm biến mực nước, cảm biến mưa, relay điều khiển bơm.

1.2. Bối cảnh và lý do chọn đề tài

- Trong bối cảnh đô thị hóa và biến đổi khí hậu, nhu cầu quản lý nước hợp lý và hiệu quả cho việc trồng trọt, vườn nhà, sân vườn, và khu đô thị ngày càng tăng. Nhiều hộ gia đình, trường học và hộ nông nhỏ vẫn còn tưới theo phương pháp thủ công (tưới tay theo lịch cố định), dẫn đến các vấn đề như:
 - + Tưới thừa gây lãng phí nước, làm thiếu oxy trong đất và tăng nguy cơ bệnh cho cây.
 - + Tưới thiếu (hoặc không kịp thời) làm stress cây, giảm năng suất hoặc chết cây.
 - + Công lao động lớn, cần người trông coi, ít phù hợp khi chủ vườn đi xa.
 - + Thiếu dữ liệu và giám sát để kiểm soát điều kiện thực tế (độ ẩm đất, mưa, trạng thái bơm).
- Việc thiết kế một mô hình tưới cây tự động có khả năng cảm biến độ ẩm đất, quyết định tưới theo ngưỡng, điều khiển bơm, và cho phép giám sát từ xa sẽ giúp tối ưu việc dùng nước, giảm công lao động và tăng tính ổn định cho cây trồng. Đề tài này còn là môi trường thực hành tốt cho các kiến thức lập trình hệ thống, nhúng (embedded), truyền thông IoT và tích hợp phần cứng — phần mềm.

1.3. Mục tiêu nghiên cứu

1.3.1. Mục tiêu tổng quát:

- Thiết kế, triển khai và đánh giá một mô hình tưới cây thông minh dùng vi điều khiển ESP32, tích hợp các cảm biến (DHT22, LDR, cảm biến độ ẩm đất,

cảm biến mưa, cảm biến mực nước), relay điều khiển bơm và hệ thống giám sát/điều khiển từ xa qua MQTT, cho phép vận hành ở chế độ tự động (theo điểm số môi trường) hoặc chế độ thủ công.

1.3.2. Mục tiêu cụ thể

- Xây dựng nguyên mẫu phần cứng gồm: ESP32, cảm biến nhiệt độ & độ ẩm (DHT22), cảm biến ánh sáng (LDR), cảm biến độ ẩm đất, cảm biến mưa, cảm biến mực nước, relay + LED điều khiển bơm, và nút MODE/MANUAL.
- Phát triển firmware chạy FreeRTOS: các task đọc cảm biến, tính toán mức tưới, điều khiển bơm, xử lý nút bấm, và truyền/nhận dữ liệu qua MQTT.
- Thiết lập hệ thống giám sát từ xa: broker MQTT + Node-RED Dashboard hiển thị dữ liệu (telemetry JSON, trạng thái, cảnh báo) và cho phép điều khiển thủ công.
- Thực nghiệm mô hình trên 1–3 chậu cây trong điều kiện thực tế (hoặc mô phỏng) để thu thập dữ liệu trong ít nhất một chu kỳ vận hành.
- Đánh giá hiệu năng theo các chỉ số: độ ổn định hệ thống, tính chính xác của cảm biến, độ tin cậy khi kích hoạt bơm, phản hồi MQTT, và khả năng tiết kiệm nước (so sánh với tưới thủ công).

1.3.3. Tiêu chí đánh giá kết quả

- Hệ thống có thể tự động quyết định tưới (ít/vừa/nhiều) dựa trên dữ liệu cảm biến và logic điểm số.
- Hệ thống có thể ngừng tưới khi trời mưa hoặc bồn hết nước.
- Dữ liệu telemetry và cảnh báo được hiển thị ổn định trên dashboard.
- Hệ thống chạy liên tục nhiều ngày mà không reset hoặc treo do lỗi chương trình.

1.4. Phạm vi nghiên cứu và đối tượng áp dụng

1.4.1. Phạm vi nghiên cứu

- Tập trung vào mô hình prototype, không triển khai thương mại.
- Giao tiếp qua Wi-Fi + MQTT broker (HiveMQ Cloud).

1.4.2. Đối tượng áp dụng

- Hộ gia đình trồng cây nhỏ lẻ (chậu, ban công).
- Trường học, phòng lab, câu lạc bộ IoT.
- Sinh viên, học viên muốn học IoT, hệ nhúng, FreeRTOS, MQTT..

1.5. Phương pháp nghiên cứu và công cụ hỗ trợ

1.5.1. Phương pháp nghiên cứu

- Nghiên cứu tài liệu: các giải pháp tưới thông minh, FreeRTOS, MQTT, Node-RED.
- Phân tích yêu cầu: chế độ tự động, thủ công, giám sát, cảnh báo.
- Thiết kế hệ thống: kiến trúc phần cứng, sơ đồ luồng dữ liệu, phân chia task FreeRTOS.
- Triển khai: lắp ráp phần cứng, viết firmware ESP32 (đa nhiệm, MQTT, logic tưới theo điểm số).
- Thực nghiệm: chạy thử, ghi log MQTT/Serial, quan sát hoạt động.
- Đánh giá: so sánh kết quả thực tế với mục tiêu đặt ra.

1.5.2. Công cụ phần cứng

- ESP32.
- Cảm biến: DHT22, LDR, cảm biến ẩm đất, cảm biến mưa (analog), cảm biến mực nước.
- Relay + LED báo trạng thái bơm.
- Bơm mini 5V.
- Nút MODE và MANUAL.
- Nguồn: adapter 5V–9V.

1.5.3. Công cụ phần mềm

- Arduino IDE (lập trình ESP32).
- Ngôn ngữ: C/C++.
- Giao thức: MQTT (HiveMQ Cloud hoặc Mosquitto).
- Dashboard: Node-RED + Node-RED Dashboard.
- Công cụ giám sát: Serial Monitor, log MQTT, file CSV.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

2.1. Khái niệm về hệ thống tưới cây tự động

- Hệ thống tưới cây tự động là giải pháp ứng dụng công nghệ điện tử và IoT để kiểm soát quá trình cung cấp nước cho cây trồng. Dữ liệu từ các cảm biến (độ ẩm đất, mưa, ánh sáng, nhiệt độ, độ ẩm không khí, mực nước) được thu thập và xử lý bởi ESP32, từ đó đưa ra quyết định bật/tắt bơm.
- Hệ thống có thể hoạt động theo hai chế độ chính:
 - + Tự động (AUTO): Bộ điều khiển sử dụng thuật toán chấm điểm (scoring) để đánh giá tình trạng đất và môi trường. Khi điểm đạt mức cần tưới (ít/vừa/nhiều), hệ thống sẽ tự động kích hoạt bơm; khi đạt điều kiện đủ ẩm hoặc trời mưa thì dừng.
 - + Thủ công (manual): Người dùng có thể điều khiển trực tiếp bơm thông qua nút bấm, công tắc, hoặc ứng dụng giám sát từ xa.
- Hệ thống không chỉ giúp tiết kiệm nước, giảm công sức mà còn tăng độ chính xác nhờ giám sát thời gian thực, phù hợp xu hướng Smart Agriculture và IoT.

2.2. Nguyên lý hoạt động chung của hệ thống điều khiển tự động

- Nguyên lý cơ bản của một hệ thống điều khiển tự động gồm các thành phần sau:
 - + Khối cảm biến: thu thập dữ liệu (độ ẩm đất, ánh sáng, mưa, nhiệt độ, độ ẩm không khí, mực nước).
 - + Khối xử lý (ESP32): chạy FreeRTOS để thực hiện đa nhiệm – task đọc cảm biến, task điều khiển bơm, task xử lý nút bấm, task MQTT. ESP32 tính điểm số tưới và quyết định chế độ hoạt động.
 - + Khối chấp hành (Relay + Bơm): relay nhận tín hiệu từ ESP32 để bật/tắt bơm.
 - + Khối truyền thông: dữ liệu telemetry (đóng gói JSON) được publish qua MQTT; dashboard (Node-RED) subscribe và hiển thị, đồng thời gửi lệnh điều khiển về ESP32.
 - + Khối giao diện người dùng: dashboard hiển thị thông số môi trường, trạng thái bơm, cảnh báo (hết nước, trời mưa, thay đổi chế độ). Người dùng cũng có thể chuyển chế độ AUTO/MANUAL qua nút MODE.
 - + Khối nguồn: cung cấp điện áp ổn định cho ESP32, cảm biến và bơm.

- Luồng hoạt động: Cảm biến → ESP32 xử lý (FreeRTOS + scoring logic) → quyết định điều khiển bơm → relay bật/tắt bơm → đồng thời gửi dữ liệu JSON qua MQTT → Node-RED hiển thị.

2.3. Tổng quan về các linh kiện phần cứng

2.3.1. Vi điều khiển ESP32

- Tích hợp Wi-Fi, hỗ trợ FreeRTOS, nhiều GPIO, phù hợp cho IoT.
- Trong hệ thống: ESP32 đóng vai trò trung tâm, đọc cảm biến, xử lý thuật toán điểm số, điều khiển relay, quản lý MQTT, và giám sát chế độ hoạt động..

2.3.2. Cảm biến độ ẩm đất

- Đo độ ẩm theo sự thay đổi điện dung, bền hơn loại điện trở.
- Dữ liệu được sử dụng trong thuật toán điểm số tưới.

2.3.3. Cảm biến mưa

- Dựa trên thay đổi điện trở khi có nước.
- Khi trời mưa, hệ thống tự động hủy bỏ lệnh tưới để tránh ngập úng.

2.3.4. Cảm biến DHT22 (nhiệt độ và độ ẩm không khí)

- Đo nhiệt độ (-40 ~ +80 °C) và độ ẩm không khí (0–100%).
- Dữ liệu hỗ trợ đánh giá môi trường và hiển thị trên dashboard.

2.3.5. Cảm biến ánh sáng (LDR)

- Điện trở thay đổi theo cường độ ánh sáng.
- Được dùng trong logic chấm điểm, ví dụ không tưới vào giữa trưa nắng gắt.

2.3.6. Cảm biến mực nước

- Theo dõi mức nước trong bồn chứa.
- Nếu cạn, hệ thống tự động dừng bơm và gửi cảnh báo qua MQTT.

2.3.7. Relay điều khiển

- Relay 1 kênh 5V, đóng/ngắt bơm dựa trên tín hiệu từ ESP32.
- Có LED báo trạng thái bơm đi kèm.

2.3.8. Bơm nước

- Bơm 5V phù hợp cho mô hình nhỏ, tưới 1–3 chậu cây.

2.4. Các giao thức truyền thông và lập trình hệ thống: MQTT, Node-RED, FreeRTOS

2.4.1. Giao thức MQTT

- Giao thức publish/subscribe nhẹ, tối ưu cho IoT.
- Trong hệ thống:
 - + ESP32 publish telemetry (JSON: nhiệt độ, độ ẩm, mực nước, trạng thái bơm, chế độ).
 - + Dashboard (Node-RED) subscribe và hiển thị dữ liệu.
 - + Người dùng publish command (bật/tắt bơm, đổi chế độ AUTO/MANUAL).

2.4.2. Node-RED

- Công cụ lập trình trực quan, dễ tích hợp với MQTT.
- Hiển thị dashboard gồm: biểu đồ độ ẩm, trạng thái bơm, cảnh báo, nút điều khiển.

2.4.3. FreeRTOS trên ESP32

- Hệ điều hành thời gian thực, hỗ trợ đa nhiệm.
- Trong code:
 - + Task đọc cảm biến (định kỳ).
 - + Task MQTT (gửi telemetry, nhận lệnh).
 - + Task điều khiển bơm.
 - + Task xử lý nút MODE & MANUAL.
- FreeRTOS giúp hệ thống chạy mượt, không bị treo khi vừa đọc cảm biến, vừa điều khiển, vừa truyền dữ liệu..

2.5. Các công trình, nghiên cứu, sản phẩm liên quan

- Nghiên cứu học thuật: nhiều đề tài đã áp dụng cảm biến ẩm đất + vi điều khiển ESP để điều khiển bơm. Một số nâng cao tích hợp dự báo thời tiết hoặc AI. Tuy nhiên đa phần dùng logic ngưỡng, chưa có scoring.
- Sản phẩm thương mại: ví dụ Gardena Smart System, Xiaomi Smart Garden, có app di động nhưng giá cao, khó tùy biến.
- Cộng đồng Maker (DIY): nhiều dự án open-source với ESP32/ESP8266 + MQTT + Node-RED. Điểm khác biệt của đề tài này là:
 - + Sử dụng logic điểm số (scoring) thay vì chỉ ngưỡng đơn giản.
 - + Có nút MODE/MANUAL để thay đổi chế độ tại chỗ.
 - + Hệ thống chạy trên FreeRTOS, hỗ trợ đa nhiệm ổn định hơn.
 - + Dữ liệu telemetry đóng gói JSON, dễ tích hợp về sau.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu hệ thống

3.1.1. Yêu cầu chức năng

- Thu thập dữ liệu cảm biến:
 - + Cảm biến độ ẩm đất → xác định tình trạng khô/ẩm.
 - + Cảm biến mưa → phát hiện mưa, dừng tưới.
 - + Cảm biến DHT22 → đo nhiệt độ và độ ẩm không khí.
 - + Cảm biến ánh sáng (LDR) → phân biệt ngày/đêm, hỗ trợ điều chỉnh lịch tưới.
 - + Cảm biến mực nước → theo dõi mức nước trong bồn, cảnh báo khi gần cạn.
- Điều khiển tưới:
 - + Bật/tắt bơm khi độ ẩm đất thấp hơn ngưỡng cài đặt.
 - + Ngừng tưới khi có mưa hoặc bồn cạn nước.
 - + Cho phép điều khiển thủ công (Manual Mode) từ Node-RED Dashboard qua MQTT.
- Truyền thông & giám sát:
 - + ESP32 gửi dữ liệu cảm biến qua Wi-Fi (MQTT publish).
 - + Node-RED hiển thị dữ liệu theo thời gian thực.
 - + Người dùng gửi lệnh (MQTT subscribe) để điều khiển bơm.

3.1.2. Yêu cầu phi chức năng

- Ổn định: chạy liên tục \geq vài ngày mà không reset.
- Chính xác: dữ liệu cảm biến sai số nhỏ, đủ để quyết định tưới hợp lý.
- An toàn: relay cách ly mạch bơm với ESP32.
- Mở rộng: dễ thêm cảm biến hoặc nhiều zone tưới.
- Trực quan: dashboard hiển thị dễ hiểu, có nút bật/tắt thủ công.

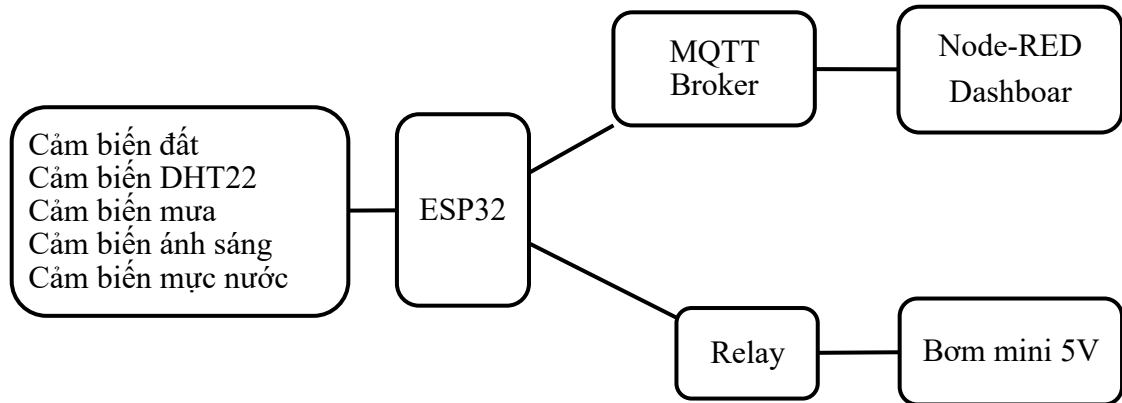
3.2. Kiến trúc tổng thể của hệ thống tưới cây tự động

- Khối cảm biến: Soil Moisture, DHT22, Rain Sensor, LDR, Water Level.
- Khối điều khiển (ESP32):
 - + Đọc cảm biến.
 - + So sánh với ngưỡng cấu hình.
 - + Gửi dữ liệu lên MQTT Broker.
 - + Nhận lệnh điều khiển từ Node-RED Dashboard.
- Khối chấp hành: Relay điều khiển bơm mini 5V.

- Khối truyền thông: ESP32 ↔ MQTT Broker ↔ Node-RED.
- Khối giám sát/điều khiển: Node-RED Dashboard (biểu đồ + nút bấm).

3.3. Mô hình hoạt động (sơ đồ khối, sơ đồ use case, sơ đồ luồng dữ liệu)

3.3.1. Sơ đồ khối hệ thống



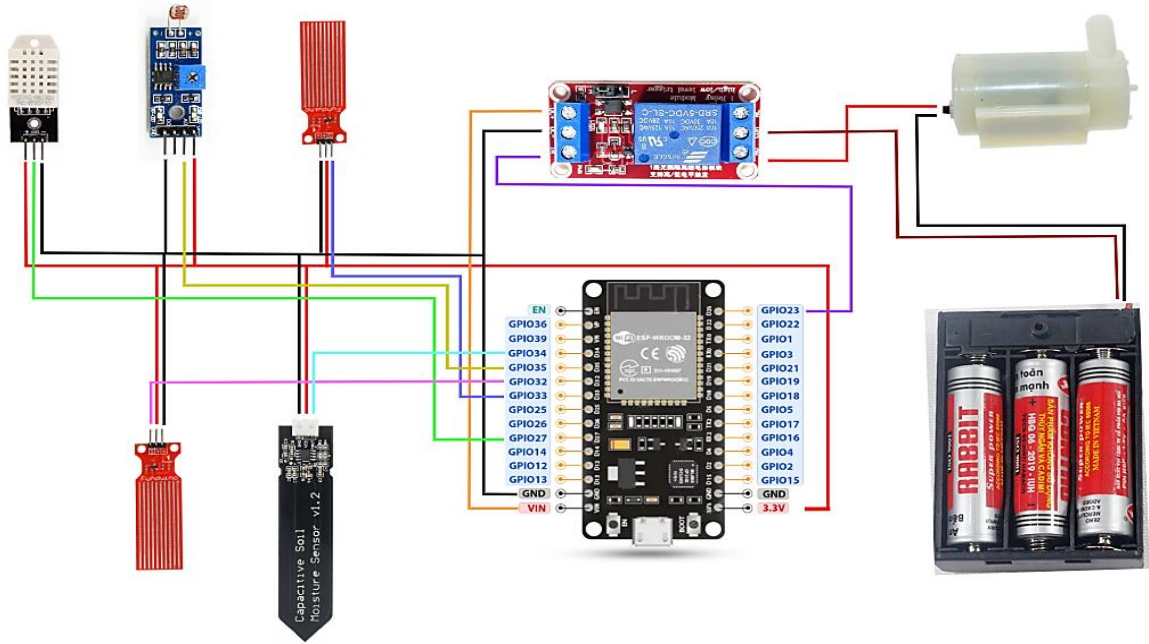
3.3.2. Sơ đồ Use Case

- Tác nhân (Actor): Người dùng.
- Use case chính:
 - + Xem dữ liệu cảm biến.
 - + Nhận cảnh báo (cạn nước, trời mưa).
 - + Bật/tắt bơm thủ công.
 - + Cấu hình ngưỡng độ ẩm.

3.3.3. Sơ đồ luồng dữ liệu (DFD mức khái quát)

- Người dùng ↔ Node-RED Dashboard: xem dữ liệu, gửi lệnh.
- Dashboard ↔ MQTT Broker: nhận dữ liệu cảm biến, truyền lệnh điều khiển.
- ESP32 ↔ MQTT Broker: publish dữ liệu cảm biến, subscribe lệnh điều khiển.
- ESP32 ↔ Relay ↔ Bơm: điều khiển bật/tắt bơm.
- ESP32 ↔ Cảm biến: thu thập dữ liệu môi trường.

3.4. Thiết kế phần cứng



3.5. Thiết kế phần mềm

3.5.1. Cấu trúc chương trình

- Setup: Khởi tạo Serial, kết nối Wi-Fi, MQTT, các cảm biến, relay.
- Task trong FreeRTOS:
 - + Đọc dữ liệu cảm biến (soil, DHT22, LDR, rain, water level).
 - + Publish dữ liệu lên MQTT.
 - + Subscribe lệnh từ Node-RED (bật/tắt bơm).
 - + Nếu Auto Mode: so sánh ngưỡng → bật relay (bơm).
 - + Ghi log ra Serial Monitor.

3.5.2. Các module chính

- Module cảm biến: đọc Soil, DHT22, Rain, LDR, Water Level.
- Module xử lý logic tưới: kiểm tra ngưỡng, quyết định tưới.
- Module MQTT: publish (dữ liệu), subscribe (lệnh).
- Module relay: bật/tắt bơm.
- Module dashboard (Node-RED): hiển thị dữ liệu, nhận lệnh từ người dùng.

3.5.3. Giao diện giám sát (Node-RED Dashboard)

- Biểu đồ độ ẩm đất theo thời gian.
- Chỉ báo nhiệt độ, độ ẩm không khí, ánh sáng, mưa.
- Thanh hiển thị mực nước bồn chứa.
- Nút bật/tắt bơm thủ công.
- Đèn trạng thái bơm (On/Off).

CHƯƠNG 4. TRIỂN KHAI VÀ THỰC NGHIỆM

4.1. Cài đặt phần mềm điều khiển (lập trình cho ESP32)

- Môi trường lập trình: Arduino IDE cài đặt board ESP32 Dev Module.
- Thư viện sử dụng:
 - + WiFi.h → quản lý kết nối Wi-Fi.
 - + PubSubClient.h → giao tiếp MQTT.
 - + DHT.h → đọc cảm biến DHT22.
 - + ArduinoJson.h (nếu có) → đóng gói dữ liệu JSON gửi lên MQTT.
 - + Driver analog/digital → cảm biến mực nước, mưa, ánh sáng, độ ẩm đất.
- Thuật toán điều khiển:
 - + Kết nối Wi-Fi và MQTT Broker.
 - + Đọc dữ liệu cảm biến: Soil Moisture, DHT22, Rain, LDR, Water Level.
 - + Publish dữ liệu lên các topic MQTT mỗi 5 giây.
 - + Subscribe các lệnh điều khiển từ Node-RED (ví dụ: pump/control, mode/control, threshold/control).
 - + Xử lý logic tưới:
 - Chế độ Auto:
 - Nếu độ ẩm đất < ngưỡng và không mưa và còn nước → bật bơm.
 - Nếu độ ẩm đất > ngưỡng trên hoặc phát hiện mưa hoặc hết nước → tắt bơm.
 - Chế độ Manual: bật/tắt bơm theo lệnh người dùng.
 - + Cập nhật trạng thái bơm lên Dashboard.

4.2. Cấu hình giao tiếp

- MQTT Broker:
 - + Có thể dùng Mosquitto trên localhost hoặc HiveMQ public broker.
 - + Các topic chính:
 - sensor/data → publish dữ liệu cảm biến (JSON).
 - pump/control → nhận lệnh bật/tắt bơm.
 - mode/control → nhận lệnh chuyển chế độ Auto/Manual.
 - threshold/control → nhận ngưỡng tưới do người dùng cài đặt.
 - pump/status → publish trạng thái bơm (ON/OFF).

- Node-RED:
 - + Kết nối MQTT In/Out để trao đổi dữ liệu với ESP32.
 - + Dashboard hiển thị dữ liệu real-time (biểu đồ, gauge, đèn trạng thái).
- Dashboard (UI):
 - + Biểu đồ độ ẩm đất, nhiệt độ/độ ẩm không khí.
 - + Chỉ báo mưa/nắng, mức nước.
 - + Nút bấm bật/tắt bơm thủ công.
 - + Công tắc chọn chế độ Auto/Manual.
 - + Đèn LED báo trạng thái bơm.

4.3. Kết quả thử nghiệm

- Độ ẩm đất:
 - + Hệ thống bật bơm khi độ ẩm < ngưỡng (ví dụ 40%).
 - + Hệ thống tắt bơm khi độ ẩm > ngưỡng trên (ví dụ 70%).
- Trạng thái bơm: hiển thị đúng trên Dashboard và điều khiển thủ công được.
- Chế độ hoạt động:
 - + Auto: chạy theo cảm biến và logic trong code.
 - + Manual: người dùng điều khiển qua Dashboard.
- Dữ liệu Dashboard: hiển thị liên tục (5 giây/lần), có thể theo dõi biểu đồ lịch sử ngắn hạn.

4.4. Đánh giá hệ thống

- Ưu điểm:
 - + Tự động hóa tưới cây → tiết kiệm công sức.
 - + Linh hoạt → chuyển đổi Auto/Manual dễ dàng.
 - + Giám sát từ xa qua Node-RED Dashboard.
 - + Mô hình nhỏ gọn, chi phí thấp.
 - + Code sử dụng MQTT nên có thể mở rộng nhiều node ESP32 cùng hoạt động.
- Hạn chế:
 - + Cảm biến giá rẻ nên độ chính xác chưa cao (độ ẩm đất và mực nước có sai số).
 - + Cần Wi-Fi ổn định để duy trì kết nối MQTT.
 - + Chưa có database lưu dữ liệu dài hạn (chỉ xem real-time).
 - + Chưa có cảnh báo qua SMS/Zalo/Telegram (chỉ hiện trong Dashboard).

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Tóm tắt những kết quả đạt được

- Đã thiết kế và lập trình thành công hệ thống tưới cây tự động sử dụng ESP32 làm bộ điều khiển trung tâm.
- Hệ thống thu thập được dữ liệu từ nhiều cảm biến: độ ẩm đất, nhiệt độ/độ ẩm không khí (DHT22), mưa, ánh sáng và mực nước.
- Xây dựng giao tiếp MQTT giúp truyền dữ liệu và lệnh điều khiển ổn định giữa ESP32 và máy chủ.
- Tích hợp Node-RED Dashboard để hiển thị dữ liệu real-time, cho phép điều khiển bơm ở chế độ Auto/Manual.
- Thử nghiệm thực tế: hệ thống tưới hoạt động đúng như thiết kế → tự động bật bơm khi đất khô, dừng khi đủ ẩm hoặc khi có mưa, đồng thời có thể bật/tắt thủ công từ Dashboard.

5.2. Đánh giá hiệu quả hệ thống tưới cây tự động

- Ưu điểm:
 - + Tiết kiệm nước, giảm công sức so với tưới truyền thống.
 - + Cho phép giám sát và điều khiển từ xa qua Node-RED Dashboard.
 - + Hoạt động real-time, độ phản hồi nhanh.
 - + Linh hoạt: có thể chuyển đổi Auto/Manual dễ dàng.
 - + Chi phí thấp, dễ lắp đặt, phù hợp hộ gia đình và nông trại nhỏ.
- Hạn chế:
 - + Phụ thuộc vào mạng Wi-Fi, khi mất kết nối việc giám sát từ xa bị gián đoạn.
 - + Cảm biến giá rẻ có sai số, ảnh hưởng một phần đến độ chính xác.
 - + Chưa có cơ chế lưu dữ liệu dài hạn (database) để phân tích.
 - + Nguồn điện vẫn dựa vào lưới, chưa ứng dụng năng lượng tái tạo.

5.3. Hướng cải tiến và phát triển

- Tích hợp AI: áp dụng machine learning để dự đoán nhu cầu tưới dựa trên dữ liệu cảm biến và đặc tính cây trồng.
- Kết nối IoT Cloud: lưu trữ dữ liệu trên các nền tảng như Firebase, AWS IoT, Azure IoT để theo dõi lâu dài.
- Mở rộng cảm biến: bổ sung cảm biến pH đất, ánh sáng chuyên dụng, cảm biến EC để tối ưu chăm sóc cây.

- Năng lượng tái tạo: sử dụng pin mặt trời và ắc quy để cấp nguồn cho ESP32 và bơm.
- Ứng dụng di động: phát triển app Android/iOS để điều khiển và giám sát trực tiếp thay vì chỉ qua Node-RED.
- Cảnh báo thông minh: gửi thông báo qua Telegram/Zalo khi phát hiện bất thường (cạn nước, lỗi cảm biến).

5.4. Kết luận chung

- Hệ thống tưới cây tự động dựa trên ESP32 + MQTT + Node-RED đã chứng minh được tính khả thi và hiệu quả thực tế.
- Kết quả thử nghiệm cho thấy hệ thống có thể hoạt động ổn định, tiết kiệm nước, giảm công lao động, và nâng cao tính hiện đại trong sản xuất nông nghiệp.
- Dù còn một số hạn chế, nhưng với tiềm năng mở rộng (AI, IoT cloud, năng lượng tái tạo), hệ thống có thể phát triển thành giải pháp nông nghiệp thông minh và bền vững, phù hợp cho cả quy mô hộ gia đình và trang trại lớn.

TÀI LIỆU KHAM KHẢO

- [1]. Tổng quan về hệ thống tưới thông minh sử dụng IoT
Nsoh, B., Katimbo, A., Guo, H., Heeren, D. M., Nakabuye, H. N., Qiao, X., ... & et al. (2024). Internet of Things-Based Automated Solutions Utilizing Machine Learning for Smart and Real-Time Irrigation Management: A Review. *Sensors*, 24(23), 7480. <https://doi.org/10.3390/s24237480> MDPI
- [2]. Hệ thống tưới thông minh trong nông nghiệp
Zhang, Y., Wang, X., Jin, L., Ni, J., Zhu, Y., Cao, W., & Jiang, X. (2025). Research and Development of an IoT Smart Irrigation System for Farmland Based on LoRa and Edge Computing. *Agronomy*, 15(2), 366. <https://doi.org/10.3390/agronomy15020366> MDPI
- [3]. Ứng dụng IoT trong quản lý tưới
An overview of smart irrigation systems using IoT. (n.d.). Retrieved from ScienceDirect ScienceDirect
- [4]. Nghiên cứu hệ thống điều khiển dựa trên Fuzzy và DNN
Intelligent and automatic irrigation system based on Internet of Things... (2025). *Nature*. <https://doi.org/10.1038/s41598-025-98137-2> Nature
- [5]. Tổng quan hệ thống tưới tự động với IoT
Singh, H. K., Jyoti, Munendra K., & Chaudhary, S. (n.d.). Reviewing the Smart Irrigation System Based on IoT. IIP Series. IIP Series
- [6]. Tổng hợp các phương pháp tưới thông minh – IoT
IoT-Based Smart Irrigation Systems: An Overview on the Recent... (n.d.). National Center for Biotechnology Information. PMC
- [7]. Nghiên cứu sử dụng IoT và AI cho tưới tự động
Kunt, Y. E. (2025). Development of a Smart Autonomous Irrigation System Using IoT and AI (Preprint). arXiv. <https://doi.org/10.48550/arXiv.2506.11835> arXiv
- [8]. Hệ thống IoT + AI cho tưới cây tự động
Aydin, Ö., Kandemir, C. A., Kiraç, U., & Dalkılıç, F. (2021). An Artificial Intelligence and Internet of Things Based Automated Irrigation System (Preprint). arXiv. <https://doi.org/10.48550/arXiv.2104.04076> arXiv