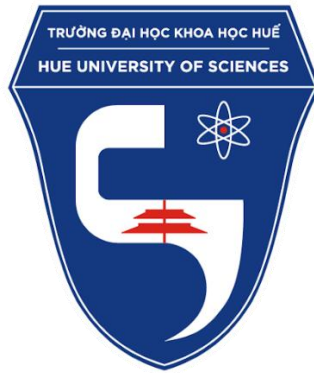


**TRƯỜNG ĐẠI HỌC KHOA
HỌC KHOA CNTT**



**THIẾT KẾ HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ VÀ
ĐỘ ẨM DỰA TRÊN ESP32**

THỰC TẬP VIẾT TIỂU LUẬN

2024-2025.2.TIN4024.006

Sinh viên thực hiện: Dương Duy Quốc

Giảng viên hướng dẫn: Ths.Võ Việt Dũng

HUẾ, 12 THÁNG 4 NĂM 2025.

DANH MỤC CÁC TỪ VIẾT TẮT

- **ESP32**: Tên vi điều khiển do Espressif phát triển
 - **DHT22**: Cảm biến nhiệt độ - độ ẩm
 - **TM1637**: Driver/IC điều khiển LED 7 thanh 4 chữ số
 - **WiFi**: Công nghệ mạng không dây (Wireless Fidelity)
 - **BLINK**: Nền tảng IoT (Internet of Things) hỗ trợ giám sát và điều khiển thiết bị qua mạng
 - **IoT**: Internet of Things - Mạng lưới vạn vật kết nối
-

MỤC LỤC

1. Phần Mở đầu
2. Phần nội dung
 - 2.1. Tổng quan về hệ thống giám sát nhiệt độ - độ ẩm sử dụng ESP32
 - 2.2. Phần cứng và nguyên lý hoạt động
 - 2.3. Sơ đồ mô phỏng Wokwi
 - 2.4. Gửi thông tin bằng Blynk / ThingSpeak / Telegram
 - 2.5. Phân tích ưu nhược điểm và đề xuất cải tiến
 - 2.6. Đề xuất cải tiến
3. Phần Kết luận
4. Tài liệu tham khảo

1. PHẦN MỞ ĐẦU

Nhu cầu giám sát nhiệt độ và độ ẩm ngày càng trở nên quan trọng trong nhiều lĩnh vực, từ nông nghiệp thông minh (smart farming), quản lý kho bãi, đến tự động hóa trong môi trường nhà thông minh (smart home). Sự phát triển của các nền tảng IoT giúp việc thu thập và phân tích dữ liệu môi trường trở nên thuận tiện hơn. Trong đó, vi điều khiển ESP32 được đánh giá cao bởi khả năng kết nối WiFi tích hợp, giá thành hợp lý và cộng đồng hỗ trợ rộng rãi.

Bài tiểu luận này tập trung vào việc xây dựng và mô phỏng một hệ thống **giám sát nhiệt độ - độ ẩm** sử dụng **ESP32**, **cảm biến DHT22**, và hiển thị giá trị lên **màn hình LED 4 chữ số** (TM1637). Hệ thống có khả năng gửi dữ liệu lên nền tảng Blynk hoặc các nền tảng IoT tương tự như ThingSpeak/Telegram để người dùng có thể giám sát dữ liệu từ xa.

Nội dung chi tiết bao gồm phân tích phần cứng, **sơ đồ mô phỏng với Wokwi**, quá trình cấu hình Blynk (hoặc các nền tảng thay thế), cũng như đánh giá ưu nhược điểm và đưa ra một số đề xuất cải tiến, mở rộng.

2. PHẦN NỘI DUNG

2.1. Tổng quan về hệ thống giám sát nhiệt độ - độ ẩm sử dụng ESP32

Hệ thống giám sát môi trường tích hợp các thành phần chính sau:

1. **Vi điều khiển ESP32:** Đảm nhận vai trò điều khiển, kết nối WiFi, giao tiếp với cảm biến DHT22 và điều khiển LED hiển thị.
2. **Cảm biến DHT22:** Đọc giá trị nhiệt độ (°C) và độ ẩm (%) của môi trường.
3. **Màn hình LED 4 chữ số (TM1637):** Hiển thị giá trị nhiệt độ, độ ẩm hoặc thời gian chạy (uptime) của hệ thống.
4. **Nút bấm (Push Button):** Điều khiển bật/tắt đèn LED xanh, hoặc chuyển chế độ hiển thị.
5. **Blynk/ThingSpeak (hoặc Telegram):** Giúp người dùng giám sát và điều khiển hệ thống từ xa qua internet.

Các tính năng chính:

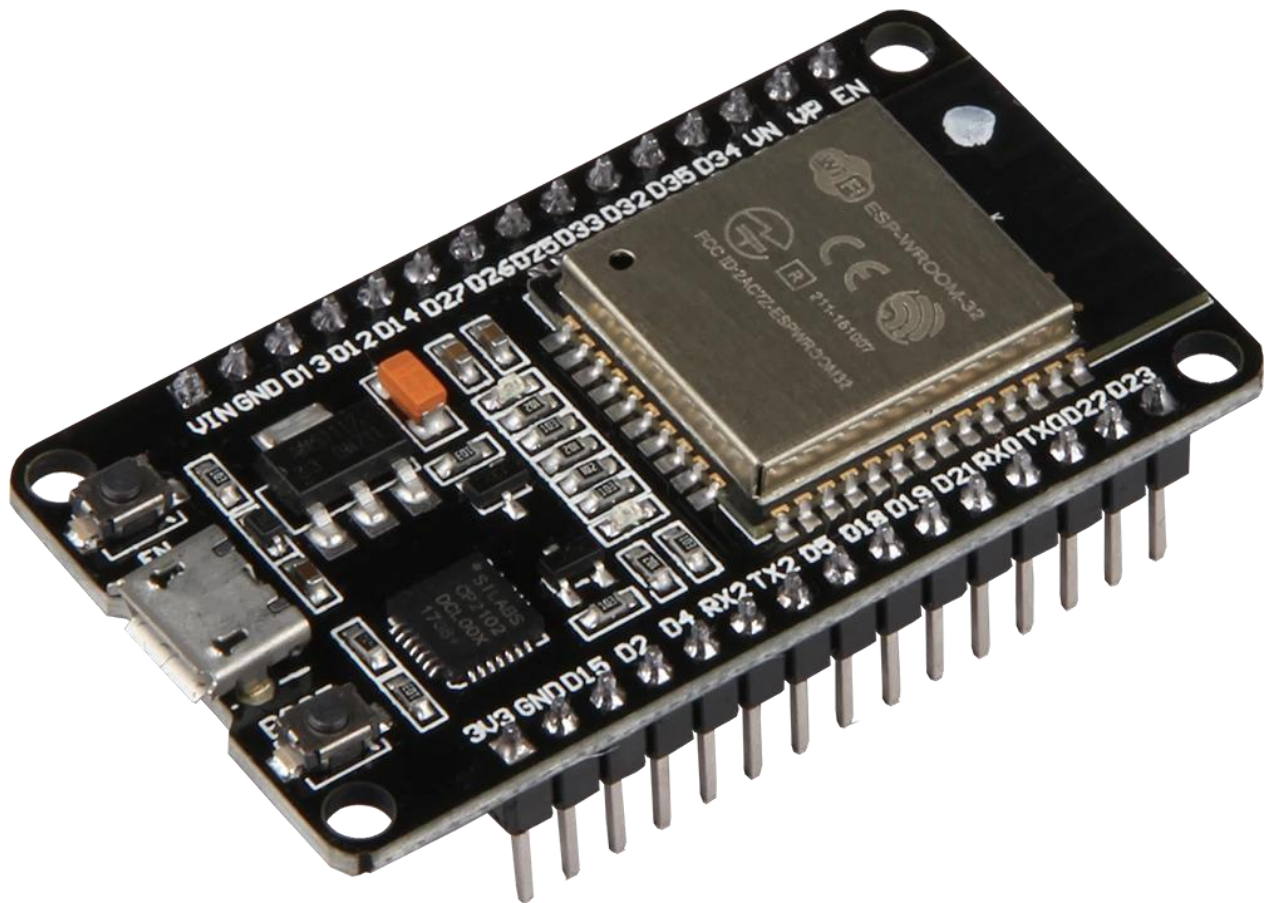
- Đo nhiệt độ/độ ẩm và hiển thị cục bộ.
- Kết nối internet để gửi dữ liệu.
- Nút bấm để tương tác thủ công, qua đó tắt/mở đèn LED hoặc thay đổi chế độ hiển thị.
- Tùy chọn giám sát từ xa với Blynk, ThingSpeak hay Telegram.

2.2. Phần cứng và nguyên lý hoạt động

2.2.1. Vi điều khiển ESP32

- ESP32 là dòng vi điều khiển tích hợp WiFi/Bluetooth do hãng Espressif phát triển.
- Có kiến trúc 32-bit, hoạt động ở tần số lên đến 240 MHz, sở hữu bộ nhớ flash và SRAM lớn.

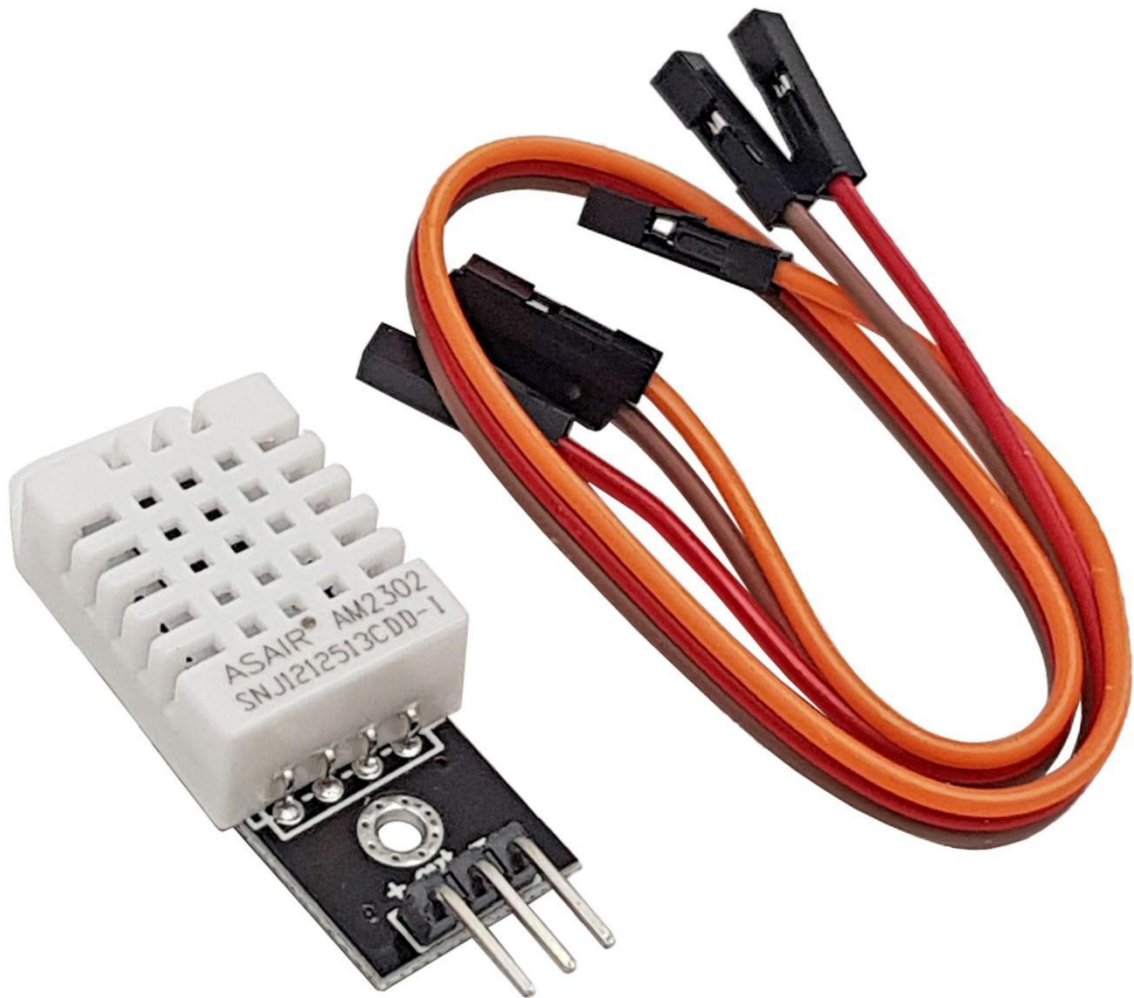
- Tích hợp nhiều cổng giao tiếp (GPIO, ADC, DAC, I2C, SPI, UART...) thuận tiện cho các ứng dụng IoT.



Bo mạch ESP32 DevKit V1

2.2.2. Cảm biến DHT22

- Là cảm biến đo nhiệt độ - độ ẩm với độ chính xác khá cao.
- Gồm 4 chân (VCC, GND, DATA, NC), thường chỉ sử dụng 3 chân (VCC, GND và DATA).
- Dải đo nhiệt độ: -40°C đến 125°C; Độ ẩm: 0 - 100% RH.
- Tín hiệu digital dễ dàng giao tiếp với vi điều khiển (sử dụng thư viện hỗ trợ sẵn).



Cảm biến nhiệt độ – độ ẩm DHT22

2.2.3. Màn hình LED 4 chữ số (TM1637)

- IC TM1637 đảm nhiệm chức năng quét và điều khiển hiển thị LED 7 thanh 4 digit.
- Giao tiếp qua 2 dây: CLK (clock) và DIO (data).
- Hiển thị được các giá trị số nguyên, đồng thời có thể hiển thị một vài ký tự đặc biệt.



Mạch hiển thị 4 Led 7 đoạn TM1637

2.2.4. Nút bấm (Push Button) và LED xanh

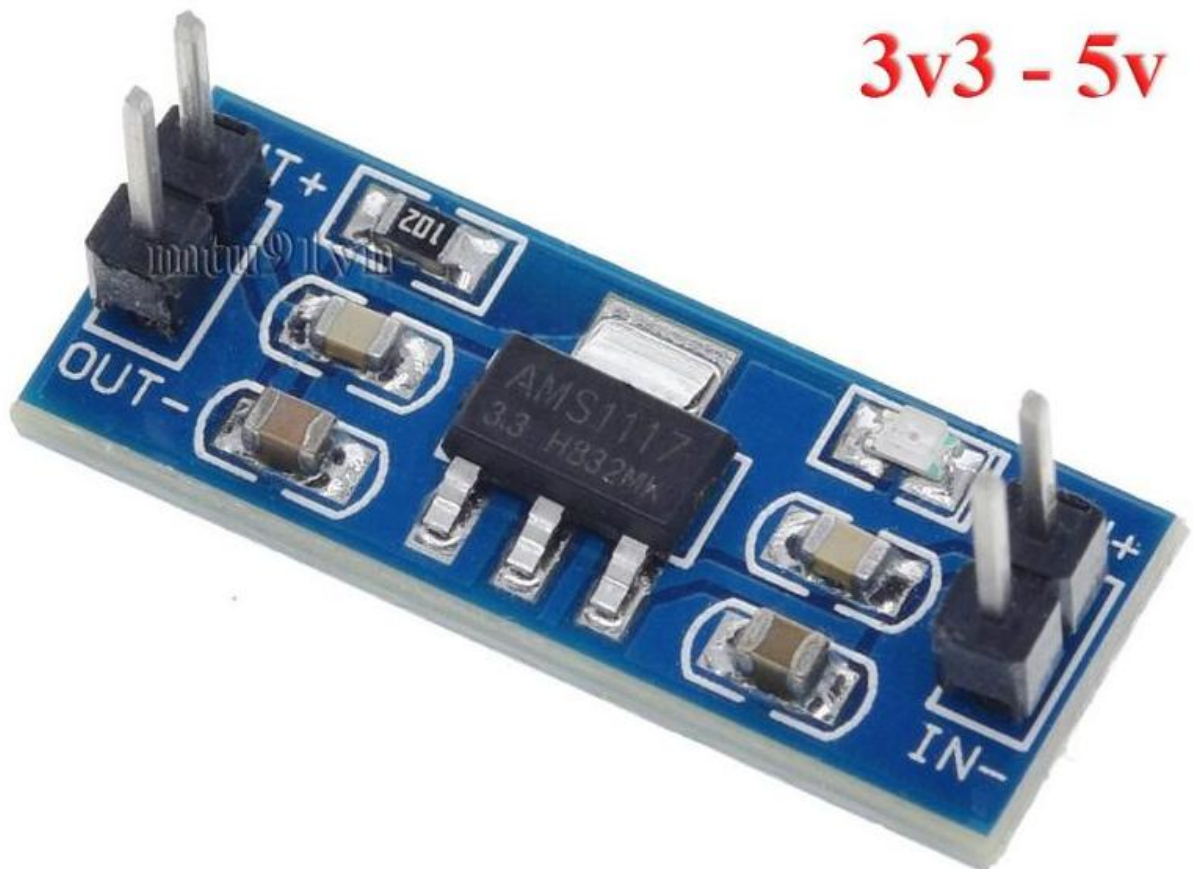
- Nút bấm được cấu hình ở chế độ INPUT_PULLUP, đọc mức LOW khi được nhấn, HIGH khi không nhấn.
- LED xanh (blue LED) được điều khiển thông qua 1 chân GPIO.



Nút bấm (Push Button) và LED xanh

2.2.5. Nguồn cung cấp

- Trong mô phỏng Wokwi, dùng các chân 3.3V hoặc 5V có sẵn trên ESP32.
- Trong thực tế, có thể sử dụng nguồn USB 5V hoặc nguồn ổn áp 3.3V tùy board ESP32.



Module – Mạch Nguồn Giảm Áp AMS1117 đầu ra 3.3v 5v

2.2.6: Dây Cắm Breadboard

1. **Chức năng:** Dùng để kết nối các linh kiện điện tử trên Breadboard hoặc giữa Breadboard và vi điều khiển (ESP32, Arduino...).
2. **Loại dây:**

-**Male to Male (Đực - Đực):** dùng phổ biến để kết nối các chân trên Breadboard.

-**Male to Female (Đực - Cái):** dùng để kết nối ESP32 với cảm biến có chân cắm cái.

-**Female to Female (Cái - Cái):** dùng trong một số trường hợp đặc biệt hoặc board dạng



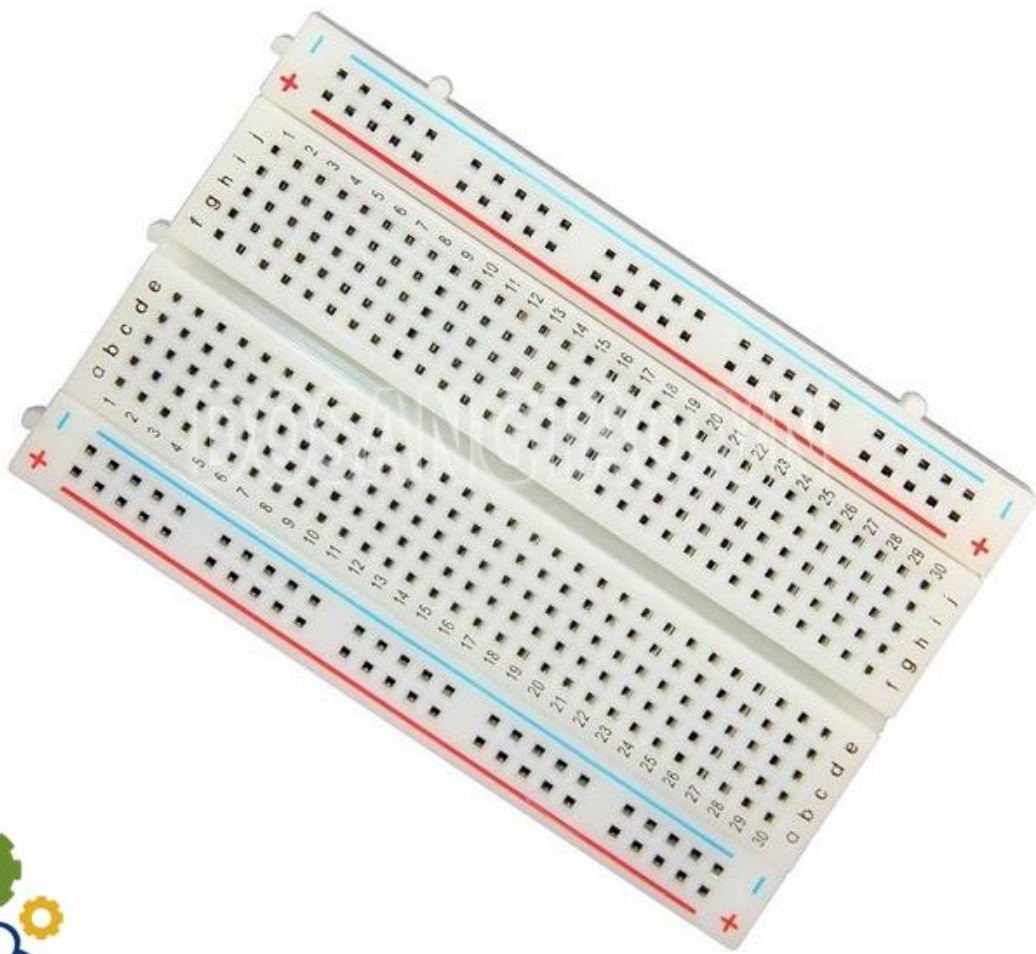
Dây Cắm Breadboard

Bảng mạch Breadboard

1.Chức năng: Là bảng mạch không hàn dùng để cắm thử các linh kiện và thiết kế mạch tạm thời.

2.Cấu tạo chính:

- Gồm các **hàng và cột tiếp điểm** bên trong nối với nhau theo quy luật.
- Có 2 thanh nguồn chạy dọc hai bên (thường dùng cho GND và VCC).



Bảng Mạch Breadboard 400 Lỗ

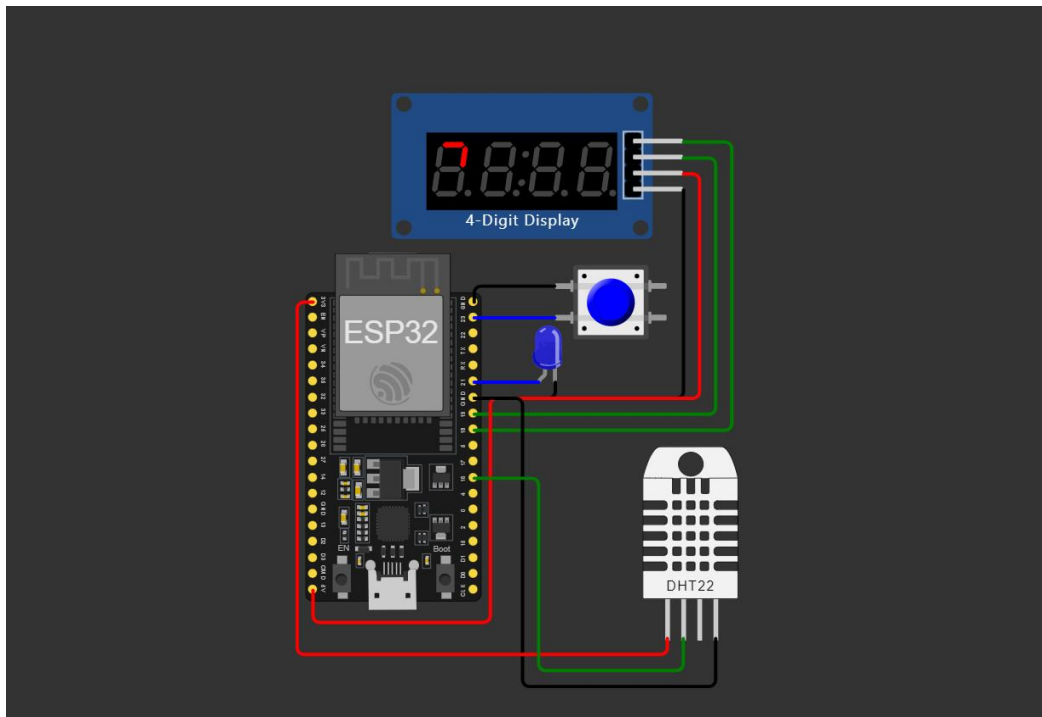
*Tổng quát nguyên lý hoạt động:

- ESP32 liên tục đọc giá trị nhiệt độ, độ ẩm từ DHT22 với chu kỳ 1-2 giây/lần.
- Giá trị đọc được hiển thị cục bộ lên màn hình LED TM1637 (nếu được bật chế độ hiển thị).
- Thông qua kết nối WiFi, ESP32 gửi dữ liệu nhiệt độ, độ ẩm lên Blynk/ThingSpeak/Telegram.
- Nút bấm giúp người dùng bật/tắt đèn LED xanh hoặc chuyển chế độ hiển thị.

2.3. Sơ đồ mô phỏng Wokwi

Dưới đây là **hình ảnh sơ đồ kết nối** trong Wokwi (trích xuất từ tệp diagram.json). Lưu ý các chân được **tạm quy ước** như sau:

- **CLK** của TM1637 nối với GPIO 18 (ESP32)
- **DIO** của TM1637 nối với GPIO 19 (ESP32)
- **Nút bấm** nối với GPIO 23, dùng **Pullup** nội bộ.
- **LED xanh** nối với GPIO 21.
- **DHT22** nối DATA vào GPIO 16, nguồn VCC vào 3.3V, GND vào GND.



Hình ảnh mô phỏng minh họa ESP32 ở trung tâm, kết nối với các linh kiện.

2.4. Gửi thông tin bằng ThingSpeak, Blynk hay Telegram

Trong phạm vi bài này, mã nguồn ví dụ sử dụng **Blynk** để gửi dữ liệu. Có thể thay thế bằng **ThingSpeak** hoặc **Telegram** theo nhu cầu:

1. **Blynk:**

- Tạo một template trên Blynk Cloud, lấy **Auth Token**.
- Cập nhật **ssid**, **pass**, **BLYNK_AUTH_TOKEN** trong mã nguồn.
- Dữ liệu (nhiệt độ, độ ẩm) được đẩy lên các Virtual Pin (V2, V3), hiển thị trên ứng dụng Blynk.

2. **ThingSpeak:**

- Tạo tài khoản, tạo kênh mới, lấy **Channel ID** và **Write API Key**.
- Gửi HTTP request hoặc dùng thư viện ThingSpeak.h để gửi dữ liệu (độ ẩm, nhiệt độ).

3. **Telegram:**

- Tạo bot thông qua BotFather, lấy **Bot Token** và **Chat ID**.
- Mỗi khi có dữ liệu mới, ESP32 gửi tin nhắn (nhiệt độ, độ ẩm) về Telegram.

Trong file main.cpp minh họa:

- Thư viện WiFi.h và BlynkSimpleEsp32.h hỗ trợ kết nối.
- Khởi lệnh Blynk.virtualWrite(V2, temperature); để gửi dữ liệu nhiệt độ.
- Khởi lệnh Blynk.virtualWrite(V3, humidity); để gửi dữ liệu độ ẩm.

2.5. Phân tích ưu nhược điểm và đề xuất cải tiến

1. **Ưu điểm**

- ESP32 có WiFi tích hợp, dễ kết nối nền tảng IoT.

- Cảm biến DHT22 có dải đo rộng, tương đối chính xác.
- Màn hình 4 chữ số TM1637 gọn gàng, hiển thị rõ ràng.
- Blynk (hoặc ThingSpeak/Telegram) có giao diện thân thiện, dễ cấu hình, phù hợp người mới bắt đầu.
- Việc mô phỏng trên Wokwi giúp kiểm tra nhanh mạch và mã trước khi triển khai thực tế.

2. Nhược điểm

- DHT22 có thời gian đáp ứng chậm hơn một số cảm biến khác (ví dụ: SHT31).
- Độ chính xác của DHT22 suy giảm nếu môi trường quá ẩm hoặc quá khô, cần hiệu chuẩn định kỳ.
- ESP32 chạy WiFi đôi khi không ổn định nếu tín hiệu mạng yếu.
- Màn hình 4 chữ số bị hạn chế khi muốn hiển thị đồng thời cả nhiệt độ và độ ẩm chi tiết (ví dụ: 25.7°C - 65.3% RH).

2.6. Đề xuất cải tiến

- Sử dụng cảm biến có tốc độ phản hồi nhanh hơn hoặc độ chính xác cao hơn (như SHT3x, BME280).
- Thêm tính năng lưu trữ dữ liệu lên thẻ nhớ (SD Card) hay vào flash.
- Cấu hình sleep mode (tiết kiệm năng lượng) cho ESP32 trong trường hợp chỉ cập nhật dữ liệu định kỳ.
- Xây dựng giao diện webserver trực tiếp trên ESP32 để thay thế hoặc bổ sung cho Blynk/ThingSpeak.
- Tận dụng **OTA (Over-The-Air)** để cập nhật firmware từ xa.

3. PHẦN KẾT LUẬN

Bài tiểu luận này đã trình bày phương pháp **mô phỏng và triển khai** một hệ thống **giám sát nhiệt độ - độ ẩm** bằng **ESP32**, **DHT22**, và hiển thị dữ liệu lên màn hình **TM1637**.

Việc tích hợp với nền tảng IoT như **Blynk** giúp người dùng dễ dàng giám sát và điều khiển từ xa qua internet. Hệ thống có tính linh hoạt cao, dễ tùy biến, đồng thời chi phí thấp.

Trong tương lai, để đáp ứng tốt hơn nhu cầu thực tế, người dùng có thể cải tiến về phần cứng (cảm biến, hiển thị), tối ưu phần mềm (thuật toán lọc giá trị đo, tiết kiệm năng lượng) hoặc triển khai thêm nhiều nền tảng IoT khác nhau. Hy vọng bài tiểu luận này sẽ mang lại kiến thức hữu ích, giúp sinh viên, người học tiếp cận nhanh chóng và hiệu quả lĩnh vực IoT và tự động hóa.

4. TÀI LIỆU THAM KHẢO

- Espressif Systems. (2023). *ESP32 Technical Reference Manual*.
Truy cập tại: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- Espressif Systems. (n.d.). *ESP32 Hardware Reference*.
Truy cập tại: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/index.html>
- Wokwi. (n.d.). *Online ESP32 Simulator*.
Truy cập tại: <https://wokwi.com>
- Blynk IoT Platform. (n.d.). *Blynk Documentation*.
Truy cập tại: <https://docs.blynk.io>
- ThingSpeak (MathWorks). (n.d.). *ThingSpeak Documentation*.
Truy cập tại: <https://thingspeak.com/docs>
- Telegram Bot API. (n.d.). *Official Telegram Bot API Documentation*.
Truy cập tại: <https://core.telegram.org/bots/api>
- Random Nerd Tutorials. *ESP32 Projects and Tutorials*. Truy cập từ: <https://randomnerdtutorials.com/>

→ Website hướng dẫn rất nhiều dự án thực tế với ESP32, cảm biến, MQTT, Blynk, v.v.