

DANH MỤC CÁC TỪ VIẾT TẮT

IoT: Internet of Things (Mạng lưới vạn vật kết nối Internet)

ESP32: Espressif Systems 32 (Tên vi điều khiển)

API: Application Programming Interface (Giao diện lập trình ứng dụng)

OLED: Organic Light-Emitting Diode (Đi-ốt phát quang hữu cơ)

HTTP: HyperText Transfer Protocol (Giao thức truyền tải siêu văn bản)

SSL: Secure Sockets Layer (Lớp công bảo mật)

I2C: Inter-Integrated Circuit (Chuẩn giao tiếp giữa các vi mạch)

SPI: Serial Peripheral Interface (Giao tiếp ngoại vi nối tiếp)

VS Code: Visual Studio Code.

MỤC LỤC

PHẦN MỞ ĐẦU	1
PHẦN NỘI DUNG.....	2
I. Tổng quan lý thuyết	2
1. ESP32	2
2. Cảm biến BME280.....	2
3. Dữ liệu mô phỏng.....	3
4. Màn hình OLED SSD1306.....	3
5. Telegram Bot API.....	4
II. Thiết kế hệ thống.....	4
1. Sơ đồ khối.....	4
2. Sơ đồ nguyên lý (Wokwi)	5
3. Giao diện Telegram Bot	6
III. Cài đặt phần mềm	7
1. Cài đặt Visual Studio Code và các tiện ích mở rộng.....	7
2. Cài đặt phần cứng mô phỏng trên Wokwi.....	7
IV. Kết quả và đánh giá.....	8
1. Kết quả mô phỏng trên Wokwi	8
2. Kết quả tin nhắn gửi qua Telegram	8
3. Đánh giá hệ thống.....	9
PHẦN KẾT LUẬN	10

PHẦN MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, việc ứng dụng các thiết bị điện tử vào đời sống thường ngày ngày càng trở nên phổ biến và thiết thực. Một trong những lĩnh vực ứng dụng nổi bật chính là Internet of Things (IoT) – nơi các thiết bị được kết nối với nhau thông qua mạng Internet để thu thập, chia sẻ và xử lý dữ liệu một cách thông minh và tự động. Trong bối cảnh đó, việc xây dựng các trạm thời tiết mini không chỉ giúp con người chủ động theo dõi các yếu tố môi trường xung quanh mà còn là tiền đề để phát triển các hệ thống tự động hóa trong nông nghiệp, nhà thông minh, giáo dục và nghiên cứu khoa học.

Đề tài "Xây dựng trạm thời tiết mini với ESP32" được thực hiện với mục tiêu thiết kế và triển khai một hệ thống đơn giản, chi phí thấp nhưng hiệu quả trong việc giám sát các thông số môi trường như nhiệt độ, độ ẩm và áp suất khí quyển. Hệ thống sử dụng ESP32 – một vi điều khiển mạnh mẽ với khả năng kết nối Wi-Fi làm trung tâm điều khiển. Kết hợp với cảm biến BME280, hệ thống có thể đo lường chính xác các chỉ số môi trường kể trên. Các dữ liệu sau đó được hiển thị trực quan trên màn hình OLED, đồng thời được gửi định kỳ tới người dùng thông qua Telegram Bot, giúp việc theo dõi từ xa trở nên tiện lợi và linh hoạt hơn bao giờ hết.

Thông qua đề tài này, sinh viên không chỉ có cơ hội tiếp cận với các công nghệ mới trong lĩnh vực IoT, mà còn rèn luyện kỹ năng lập trình, giao tiếp với cảm biến, xử lý dữ liệu, truyền thông qua mạng Internet và tích hợp các dịch vụ bên ngoài như Telegram. Đây là một bước khởi đầu quan trọng cho quá trình nghiên cứu và phát triển các ứng dụng thực tiễn trong tương lai.

PHẦN NỘI DUNG

I. Tổng quan lý thuyết

1. ESP32

- ESP32 là một bộ vi điều khiển thuộc danh mục vi điều khiển trên chip công suất thấp và tiết kiệm chi phí. Hầu hết tất cả các biến thể ESP32 đều tích hợp Bluetooth và Wi-Fi chế độ kép, làm cho nó có tính linh hoạt cao, mạnh mẽ và đáng tin cậy cho nhiều ứng dụng.
- Nó là sự kế thừa của vi điều khiển NodeMCU ESP8266 phổ biến và cung cấp hiệu suất và tính năng tốt hơn. Bộ vi điều khiển ESP32 được sản xuất bởi Espressif Systems và được sử dụng rộng rãi trong nhiều ứng dụng khác nhau như IoT, robot và tự động hóa.
- ESP32 cũng được thiết kế để tiêu thụ điện năng thấp, lý tưởng cho các ứng dụng chạy bằng pin. Nó có hệ thống quản lý năng lượng cho phép nó hoạt động ở chế độ ngủ và chỉ thức dậy khi cần thiết, điều này có thể kéo dài tuổi thọ pin rất nhiều.



2. Cảm biến BME280

- BME280 là cảm biến môi trường đa năng được sản xuất bởi Bosch Sensortec. Cảm biến này có khả năng đo ba thông số quan trọng: nhiệt độ, độ ẩm và áp suất khí quyển. Nhờ độ chính xác cao, kích thước nhỏ gọn và tiêu thụ năng lượng thấp, BME280 rất thích hợp cho các ứng dụng trong thời tiết, nông nghiệp thông minh và thiết bị đeo thông minh.

- Thông số kỹ thuật cơ bản của BME280:
 - + Nhiệt độ: -40°C đến +85°C, độ chính xác $\pm 1.0^{\circ}\text{C}$.
 - + Độ ẩm: 0% đến 100%, độ chính xác $\pm 3\%$.
 - + Áp suất: 300 hPa đến 1100 hPa, độ chính xác ± 1 hPa.
 - + Giao tiếp: I2C hoặc SPI.
- BME280 sẽ được sử dụng để đo liên tục các thông số thời tiết, cung cấp dữ liệu đầu vào cho ESP32 xử lý và hiển thị lên màn hình OLED hoặc gửi lên Telegram.

3. Dữ liệu mô phỏng

- Nếu không sử dụng cảm biến BME280, việc lấy dữ liệu môi trường được thay thế bằng cách mô phỏng các giá trị ngẫu nhiên. Vì điều khiển ESP32, thông qua lập trình trong môi trường Visual Studio Code kết hợp với Wokwi và PlatformIO, sử dụng hàm `random()` để sinh ra các giá trị đại diện cho nhiệt độ, độ ẩm và áp suất khí quyển.

```
temp = random(200, 350) / 10.0; // 20.0 - 35.0 °C
hum = random(300, 900) / 10.0; // 30% - 90%
pres = random(9900, 10200) / 10.0; // 990 - 1020 hPa
```

4. Màn hình OLED

- Màn hình OLED SSD1306 là một loại màn hình hiển thị nhỏ gọn sử dụng công nghệ đi-ốt phát quang hữu cơ (Organic Light-Emitting Diode), có khả năng hiển thị rõ ràng ngay cả trong điều kiện ánh sáng yếu. Màn hình phổ biến có kích thước 0.96 inch, độ phân giải 128x64 pixel.
- Các đặc điểm chính:
 - + Giao tiếp thông qua I2C hoặc SPI
 - + Hiển thị văn bản, biểu tượng, hình ảnh đơn sắc
 - + Tiêu thụ năng lượng thấp, phù hợp cho các thiết bị di động
 - + Sử dụng driver SSD1306 – hoạt động tốt trong môi trường lập trình Visual Studio Code kết hợp với nền tảng mô phỏng như Wokwi.

- OLED SSD1306 sẽ được dùng trong hệ thống để hiển thị thông tin nhiệt độ, độ ẩm và áp suất trực tiếp tại thiết bị, giúp người dùng dễ dàng quan sát mà không cần kết nối mạng.



5. Telegram Bot API

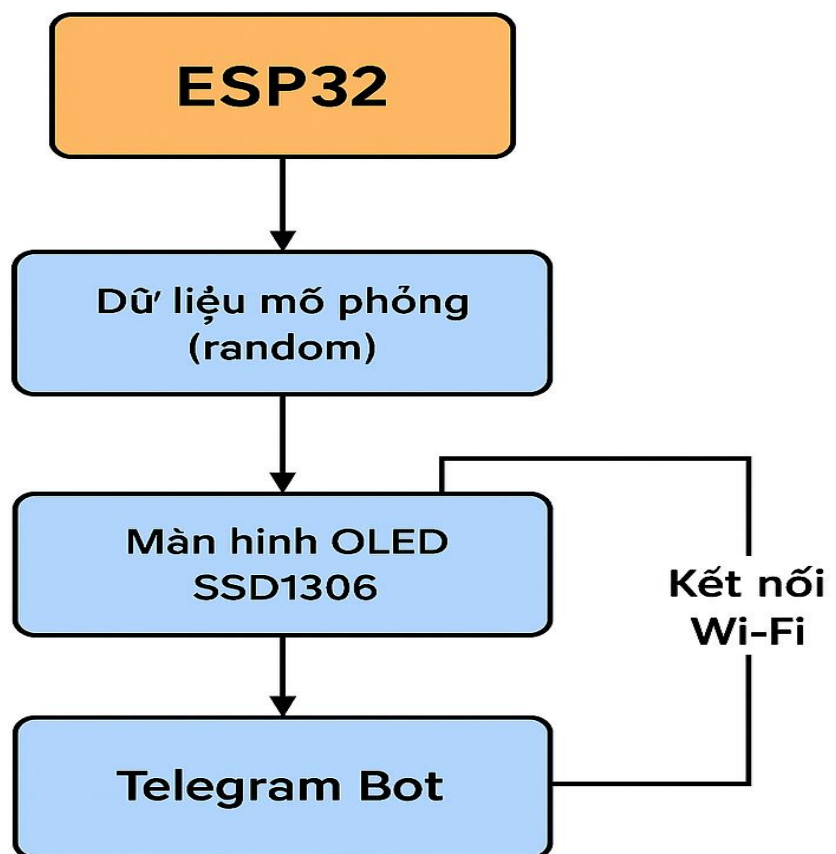
- Telegram Bot API là một giao diện lập trình ứng dụng do Telegram cung cấp, cho phép lập trình viên tạo và điều khiển các bot trên nền tảng nhắn tin Telegram. Bot có thể gửi, nhận tin nhắn, phản hồi lệnh người dùng, gửi hình ảnh, dữ liệu... thông qua HTTP requests.
- Trong ứng dụng IoT, Telegram bot thường được dùng để:
 - + Gửi thông báo từ thiết bị đến người dùng.
 - + Nhận lệnh điều khiển từ xa.
 - + Theo dõi dữ liệu thời gian thực.
- Telegram Bot sẽ đóng vai trò trung gian để ESP32 gửi dữ liệu thời tiết đo được đến người dùng thông qua Internet. Điều này giúp người dùng có thể theo dõi tình trạng môi trường ở bất cứ đâu, bất kỳ lúc nào.

II. Thiết kế hệ thống

1. Sơ đồ khối

- Sơ đồ khối thể hiện các thành phần chính trong hệ thống và cách chúng tương tác với nhau. Hệ thống không sử dụng cảm biến mà thay vào đó là dữ liệu ngẫu nhiên để mô phỏng thông số thời tiết. Các khối chức năng gồm:
 - + ESP32: Vi điều khiển trung tâm, xử lý dữ liệu, hiển thị và giao tiếp Internet.

- + Dữ liệu mô phỏng (random): Tạo dữ liệu giả lập nhiệt độ, độ ẩm và áp suất.
- + Màn hình OLED SSD1306: Hiển thị thông tin thời tiết mô phỏng theo thời gian thực.
- + Kết nối Wi-Fi: Kết nối Internet để gửi thông tin đến người dùng thông qua Telegram.
- + Telegram Bot: Gửi tin nhắn thông báo chứa dữ liệu thời tiết đến người dùng.
- Luồng hoạt động hệ thống:
- + ESP32 khởi tạo dữ liệu mô phỏng bằng cách sử dụng hàm random() để sinh ra các giá trị giả định.
- + Các giá trị này được hiển thị lên màn hình OLED.
- + Đồng thời, dữ liệu được gửi đến người dùng qua Telegram Bot bằng cách sử dụng API của Telegram thông qua kết nối Wi-Fi.



2. Sơ đồ nguyên lý (Wokwi)

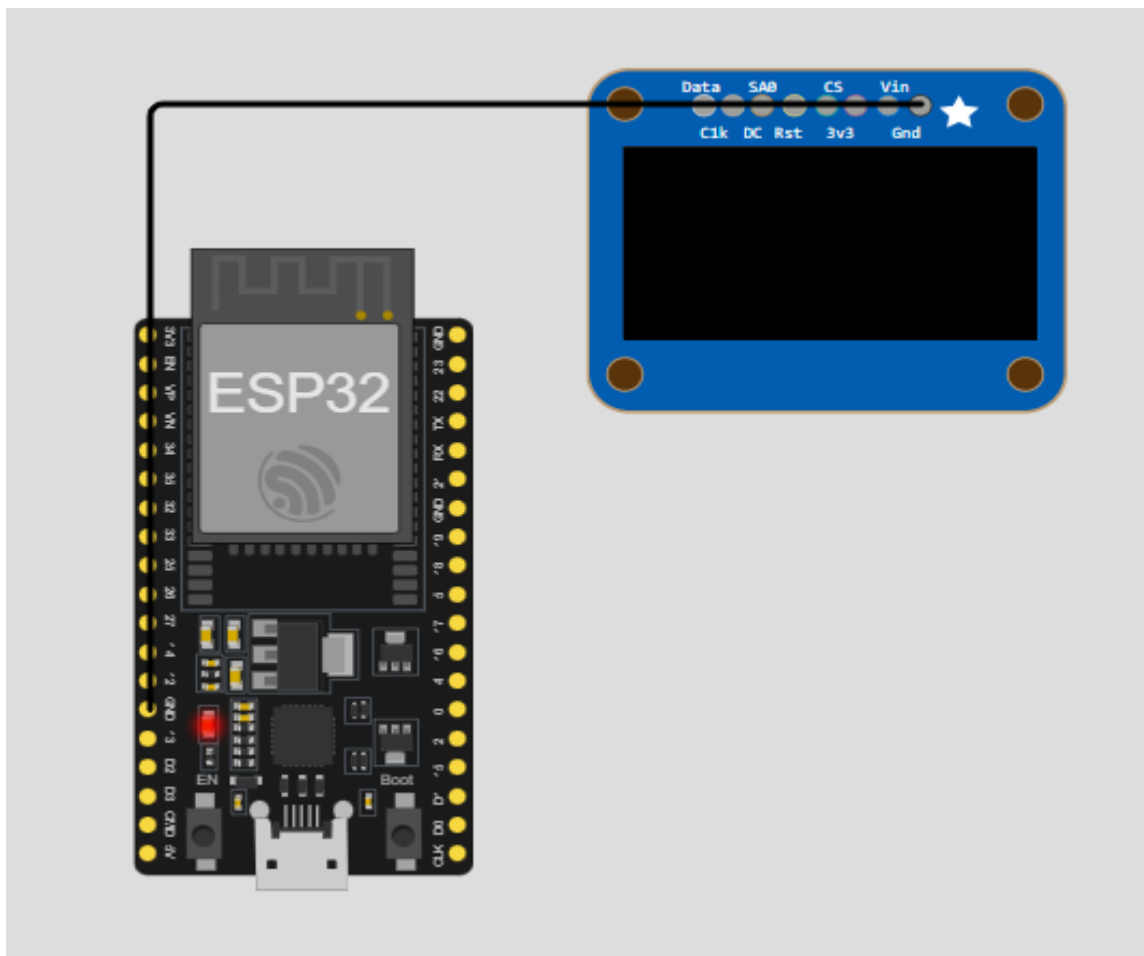
- Sơ đồ nguyên lý thể hiện kết nối phần cứng ảo giữa ESP32 và màn hình OLED.
- + ESP32 DevKit v1: Là vi điều khiển chính của hệ thống.
- + BME280: Có thể bỏ qua khi sử dụng mô phỏng bằng random()

+ Màn hình OLED SSD1306: Giao tiếp I2C.

+ Kết nối I2C giữa ESP32 và OLED:

- VCC → 3.3V
- GND → GND
- SDA (OLED) → GPIO21 (ESP32)
- SCL (OLED) → GPIO22 (ESP32)

- Cảm biến BME280 không được kết nối trong mô hình, dữ liệu được tạo ra bằng hàm random().



3. Giao diện Telegram Bot

- Telegram Bot là phần mềm trung gian giúp hệ thống gửi dữ liệu thời tiết đến người dùng theo cách trực quan và tiện lợi.

- Tính năng của Bot:

+ Gửi định kỳ dữ liệu mô phỏng từ ESP32.

+ Tin nhắn bao gồm: Nhiệt độ, độ ẩm, áp suất.

+ Dễ dàng tích hợp thêm các chức năng mở rộng như cảnh báo thời tiết hoặc phản hồi theo lệnh người dùng.

- Cách hoạt động:
- + ESP32 sử dụng thư viện UniversalTelegramBot để gửi dữ liệu tới Telegram thông qua HTTP request đến api.telegram.org. Để đảm bảo bảo mật, sử dụng kết nối SSL qua WiFiClientSecure.
- + Để chạy bot, cần tạo một Telegram Bot qua BotFather và lấy bot token. Ngoài ra, cần xác định chat_id để bot biết nơi gửi tin nhắn.

III. Cài đặt phần mềm

1. Cài đặt Visual Studio Code và các tiện ích mở rộng

- Visual Studio Code là môi trường lập trình đa năng, được sử dụng trong đề tài để viết mã và mô phỏng phần cứng thông qua nền tảng Wokwi.
- Các bước thực hiện:
- + Cài đặt Visual Studio Code từ trang chính thức: <https://code.visualstudio.com>
- + Cài đặt tiện ích mở rộng từ Extension.
 - Wokwi for VS Code: Hỗ trợ mô phỏng phần cứng ESP32.
 - PlatformIO IDE: Hỗ trợ quản lý dự án nhúng, biên dịch, và nạp chương trình.
 - C/C++ (Microsoft): Hỗ trợ ngôn ngữ lập trình C/C++ cho ESP32.
- + Sau khi cài đặt xong, khởi động lại VS Code để hoàn tất thiết lập.

2. Cài đặt phần cứng mô phỏng trên Wokwi

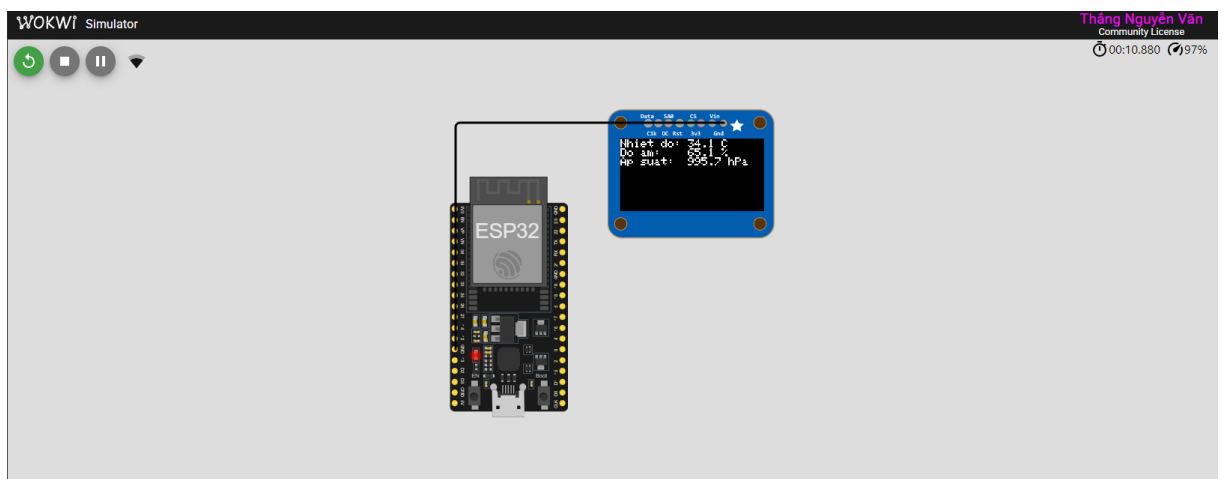
- Wokwi là nền tảng mô phỏng phần cứng được tích hợp trực tiếp vào VS Code thông qua Extension, cho phép người dùng tạo và chạy mô phỏng các bo mạch ESP32 mà không cần phần cứng thật. Việc sử dụng Wokwi giúp tiết kiệm chi phí, dễ dàng kiểm thử chương trình và trực quan hóa quá trình vận hành hệ thống.
- Các bước thực hiện:
- + Tạo dự án mới: Nhấn tổ hợp phím Ctrl + Shift + P, chọn lệnh Wokwi: New Project để khởi tạo một dự án ESP32 mới.
- + Thiết lập sơ đồ phần cứng: Dự án sẽ bao gồm file diagram.json – đây là nơi định nghĩa cấu trúc phần cứng như bo mạch ESP32, cảm biến BME280, màn hình OLED...
- + Viết mã điều khiển: Mã nguồn chính được viết trong tệp main.cpp, nơi chứa toàn bộ chương trình điều khiển thiết bị.
- + Thêm các thư viện cần thiết, các thư viện cần dùng bao gồm:

- Wire.h - Giao tiếp I2C để kết nối với màn hình OLED SSD1306.
- Adafruit_BME280.h – (Nếu có dùng cảm biến BME280 thực tế).
- Adafruit_SSD1306.h – Điều khiển màn hình OLED SSD1306.
- Adafruit_GFX.h – Xử lý hiển thị văn bản/đồ họa trên màn hình OLED.
- UniversalTelegramBot.h – Thư viện hỗ trợ gửi tin nhắn từ ESP32 đến người dùng qua Telegram.
- WiFi.h – Kết nối ESP32 với Wi-Fi
- WiFiClientSecure.h – Gửi dữ liệu qua giao thức HTTPS (Telegram API)

IV. Kết quả và đánh giá

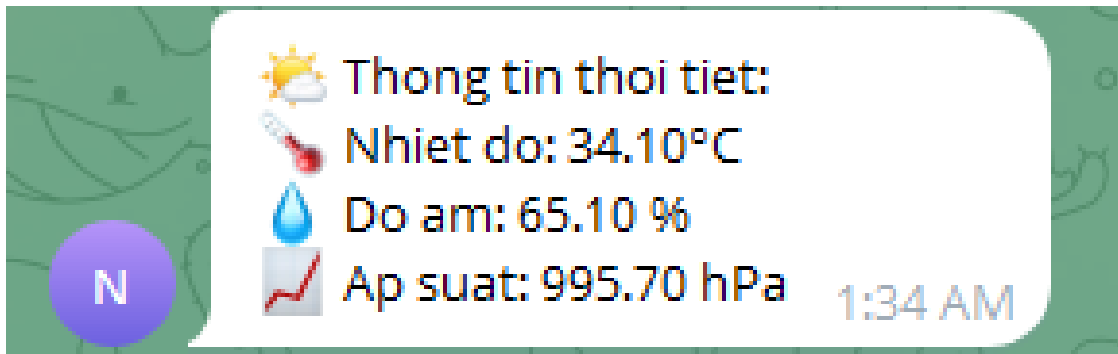
1. Kết quả mô phỏng trên Wokwi

- Hệ thống được lập trình và chạy mô phỏng trên Visual Studio Code sử dụng Extension Wokwi for VS Code.
- Vi điều khiển ESP32 được kết nối với màn hình OLED SSD1306 thông qua giao thức I2C.
- Dữ liệu thời tiết (nhiệt độ, độ ẩm, áp suất) được sinh ra ngẫu nhiên bằng hàm random() trong chương trình.
- Màn hình OLED hiển thị các thông số mô phỏng một cách rõ ràng.



2. Kết quả tin nhắn gửi qua Telegram

- ESP32 kết nối WiFi, sử dụng Telegram Bot API để gửi dữ liệu thời tiết cho người dùng.
- Dữ liệu bao gồm nhiệt độ, độ ẩm, áp suất được định dạng dạng văn bản.
- Người dùng có thể kiểm tra thông tin từ xa qua ứng dụng Telegram.



3. Đánh giá hệ thống

- Qua quá trình lập trình và mô phỏng, hệ thống hoạt động đúng theo mục tiêu đề ra. Giao tiếp giữa ESP32 và màn hình OLED ổn định, dữ liệu mô phỏng hiển thị rõ ràng. Chức năng gửi thông tin qua Telegram hoạt động hiệu quả, giúp người dùng nhận được dữ liệu thời tiết từ xa.
- Tuy nhiên, hệ thống vẫn còn một số hạn chế như:
 - + Chưa tích hợp cảm biến thật (BME280), dữ liệu hiện tại là mô phỏng.
 - + Chưa hỗ trợ điều khiển hai chiều (từ người dùng về thiết bị).
- Nhìn chung, hệ thống đã hoàn thiện phần lõi và có thể mở rộng thành ứng dụng thực tế trong tương lai.

PHẦN KẾT LUẬN

Qua quá trình tìm hiểu, thiết kế và mô phỏng, đề tài “Xây dựng trạm thời tiết mini với ESP32” đã đạt được những mục tiêu đề ra ban đầu là xây dựng một hệ thống giám sát môi trường đơn giản, chi phí thấp, nhưng hiệu quả và có tính ứng dụng cao. Hệ thống sử dụng vi điều khiển ESP32 kết hợp với cảm biến BME280 (hoặc dữ liệu mô phỏng) để đo nhiệt độ, độ ẩm và áp suất khí quyển, sau đó hiển thị thông tin lên màn hình OLED SSD1306 và gửi dữ liệu đến người dùng thông qua Telegram Bot API.

Việc triển khai mô phỏng trên nền tảng Wokwi cùng môi trường Visual Studio Code giúp sinh viên tiếp cận các công nghệ hiện đại mà không cần phần cứng thực tế, đồng thời nâng cao kỹ năng lập trình, giao tiếp I2C/SPI, xử lý dữ liệu và truyền thông qua mạng Internet.

Từ kết quả đạt được, có thể thấy hệ thống có tiềm năng lớn để phát triển thành một sản phẩm hoàn chỉnh phục vụ trong các mô hình như nhà thông minh, nông nghiệp công nghệ cao hoặc giáo dục.

TÀI LIỆU THAM KHẢO

1. <https://randomnerdtutorials.com>
2. Tài liệu kỹ thuật của cảm biến BME280 - Bosch
3. Tài liệu sử dụng ESP32 - Espressif Systems
4. Hướng dẫn lập trình VS Code và PlatformIO cho ESP32
5. <https://www.codientu.online/2022/04/gioi-thieu-module-esp32-voi-ung-dung-wifi.html>
6. <https://lastminuteengineers.com/bme280-arduino-tutorial/>
7. <https://docs.wokwi.com/>
8. <https://code.visualstudio.com>
9. api.telegram.org