

TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ
KHOA CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN

PHÁT TRIỂN ỨNG DỤNG IOT

XÂY DỰNG TRẠM THỜI TIẾT MINI VỚI ESP32

HUẾ, 2024

DANH MỤC CÁC TỪ VIẾT TẮT

IoT	Internet of Things – Internet vạn vật
ESP32	Tên vi điều khiển tích hợp WiFi và Bluetooth (do Espressif Systems phát triển)
WiFi	Wireless Fidelity – Kết nối không dây
BLE	Bluetooth Low Energy – Bluetooth năng lượng thấp
I2C	Inter-Integrated Circuit – Giao tiếp nối tiếp nội bộ
SPI	Serial Peripheral Interface – Giao tiếp nối tiếp ngoại vi
ADC	Analog to Digital Converter – Bộ chuyển đổi tín hiệu tương tự sang số
DAC	Digital to Analog Converter – Bộ chuyển đổi tín hiệu số sang tương tự
UART	Universal Asynchronous Receiver Transmitter – Bộ thu phát nối tiếp không đồng bộ
LCD	Liquid Crystal Display – Màn hình tinh thể lỏng
LCD2004	Loại màn hình LCD hiển thị 20 ký tự \times 4 dòng
TLS	Transport Layer Security – Giao thức bảo mật tầng truyền tải
API	Application Programming Interface – Giao diện lập trình ứng dụng
BME280	Tên cảm biến môi trường (do Bosch sản xuất), đo nhiệt độ, độ ẩm và áp suất
RH	Relative Humidity – Độ ẩm tương đối

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

PHẦN MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Mục tiêu nghiên cứu.....	1
3. Phương pháp nghiên cứu.....	2
PHẦN NỘI DUNG.....	4
Chương I: Tổng quan về IoT và đề tài	4
1. Khái niệm IOT và vai trò của IOT trong cuộc sống hiện nay	
2. Vai trò của IoT trong đời sống hiện nay: 4	
Chương II: Các thành phần phần cứng và phần mềm sử dụng	6
1. Giới thiệu những thiết bị sử dụng.....	6
1.1 ESP32.....	6
1.2 Cảm biến BME280 và màn hình lcd2004.....	6
1.3 Tổng quan về Telegram.....	7
2. Thiết kế hệ thống trạm thời tiết Mini.....	9
3. Triển khai phần mềm và xử lý dữ liệu.....	10
3.1. Phần mềm sử dụng.....	10
3.2. Giải pháp thay thế cảm biến BME280.....	10
3.3. Kết nối Wifi mô phỏng:	11
3.4. Xử lý và hiển thị dữ liệu.....	11
3.5. Dữ liệu gửi về Telegram.....	11
3.6. Xử lý lệnh và cập nhật dữ liệu.....	11
3.7. Mã nguồn chương trình TramThoiTietMini.....	12
Chương III: Thiết kế và xây dựng hệ thống trạm thời tiết mini	17
1. Kết quả mô phỏng.....	17
2. Mã nguồn và mô phỏng dự án.....	19

PHẦN KẾT LUẬN	20
1. Nhận xét và đánh giá.....	20
2. Kiến nghị.....	20

TÀI LIỆU THAM KHẢO

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài:

Hiện nay, thời tiết ảnh hưởng trực tiếp đến đời sống chúng ta như trong nông nghiệp và nhiều lĩnh vực khác. Ngày xưa ông cha ta nhờ vào quan sát các hiện tượng tự nhiên nên có thể dự đoán được thời tiết, ví dụ như câu :” Chuồn chuồn bay thấp thì mưa, bay cao thì nắng, bay vừa thì mưa” hay là câu “Đông Nam có chớp chéo nhau, Thấp sát mặt biển hôm sau bão về”. Nhưng hiện nay chúng ta đang ở thời kỳ công nghiệp 4.0, tiệm cận 5.0 vì thế thời tiết có thể được theo dõi một cách khoa học và chính xác hơn nhờ các thiết bị cảm biến và hệ thống giám sát thời tiết thông minh. Với sự phát triển mạnh mẽ của Internet of Things (IoT), con người có thể thu thập dữ liệu thời tiết theo thời gian thực, phân tích và dự báo thông qua các công nghệ hiện đại.

Để xây dựng một hệ thống trạm thời tiết đơn giản và hiệu quả, ESP32 kết hợp với cảm biến BME280 là một lựa chọn phù hợp nhờ các ưu điểm sau:

- ESP32 có khả năng kết nối WiFi, hỗ trợ việc gửi dữ liệu lên nền tảng giám sát trực tuyến như Blynk, ThingSpeak hoặc hiển thị trên ứng dụng di động.
- Cảm biến BME280 có độ chính xác cao, hỗ trợ đo nhiều thông số thời tiết trong một thiết bị nhỏ gọn.
- Hệ thống có chi phí thấp, dễ triển khai, phù hợp cho các ứng dụng thực tế và nghiên cứu.

Do đó, việc nghiên cứu và phát triển một trạm thời tiết mini sử dụng ESP32 sẽ giúp tận dụng tối đa những lợi ích của công nghệ hiện đại, mang lại nhiều giá trị thực tiễn cho đời sống của chúng ta.

2. Mục tiêu nghiên cứu:

Trong bối cảnh hiện nay, công nghệ IOT đang ngày càng phát triển và việc xây dựng một trạm thời tiết mini giúp theo dõi các yếu tố của môi trường như nhiệt độ, độ

ẩm và áp suất khí quyển là một vấn đề rất cần thiết. Mục tiêu nghiên cứu này của em nhằm thiết kế một hệ thống giám sát thời tiết đơn giản, nhỏ gọn với chi phí thấp, dễ dùng sử dụng. Vì không có thiết bị esp32 thật nên em sẽ sử dụng mô phỏng trên wokwi.

Các nghiên cứu của em hướng đến các mục tiêu sau:

- ✚ Xây dựng một hệ thống trạm thời tiết sử dụng mô phỏng trên Wokwi
 - Thiết kế một hệ thống đơn giản, hoạt động ổn định trên nền tảng Wokwi.
 - Mô phỏng kết nối WiFi trên Wokwi để giả lập việc truyền dữ liệu lên nền tảng giám sát từ xa.
 - Xây dựng chương trình điều khiển ESP32 mô phỏng, đọc dữ liệu từ cảm biến BME280 giả lập và xử lý thông tin thu thập được.
- ✚ Lấy dữ liệu nhiệt độ, độ ẩm và áp suất từ cảm biến mô phỏng trên Wokwi
 - Sử dụng cảm biến BME280 mô phỏng trên Wokwi thay thế cho cảm biến thực tế trong giai đoạn demo.
 - Lập trình để ESP32 mô phỏng đọc dữ liệu từ BME280 trong môi trường giả lập thông qua giao tiếp I2C hoặc SPI.
 - Kiểm tra tính chính xác và độ ổn định của dữ liệu mô phỏng trước khi triển khai lên phần cứng thực tế.
- ✚ Hiển thị dữ liệu và gửi về Telegram để giám sát từ xa
 - Truyền dữ liệu thời tiết (nhiệt độ, độ ẩm, áp suất) qua Telegram Bot, cho phép người dùng nhận thông báo ngay trên điện thoại hoặc máy tính.
 - Lập trình để ESP32 mô phỏng tự động gửi dữ liệu theo chu kỳ hoặc gửi theo yêu cầu của người dùng trên Telegram.

3. Phương pháp nghiên cứu:

3.1. Lý thuyết

- Tìm hiểu về ESP32, cảm biến BME280, Telegram Bot, màn hình OLED và cách truyền dữ liệu qua Internet.
- Nghiên cứu cách sử dụng nền tảng Wokwi để mô phỏng hệ thống mà không cần phần cứng thực tế.
- Tìm hiểu phương thức kết nối WiFi giả lập và gửi dữ liệu lên Telegram trong môi trường mô phỏng.

3.2. Mô phỏng trên Wokwi

- Sử dụng Wokwi để xây dựng và kiểm thử mô hình trạm thời tiết ảo trước khi triển khai thực tế.
- Lập trình ESP32 mô phỏng để đọc dữ liệu từ cảm biến BME280 giả lập.
- Hiển thị dữ liệu thời tiết lên màn hình OLED mô phỏng trong Wokwi.
- Mô phỏng quá trình kết nối WiFi và gửi dữ liệu lên Telegram.
- Kiểm tra và đánh giá tính chính xác, ổn định của dữ liệu mô phỏng.

3.3. Đánh giá hệ thống

- Đánh giá tốc độ và độ ổn định của việc gửi dữ liệu từ ESP32 mô phỏng lên Telegram.
- Đề xuất cải tiến và hướng phát triển hệ thống trong tương lai.

PHẦN NỘI DUNG

CHƯƠNG I: TỔNG QUAN VỀ IOT VÀ ĐỀ TÀI

1. Khái niệm IOT và vai trò của IOT trong cuộc sống hiện nay:

“Internet of Things (IoT) hay Internet vạn vật là mạng lưới các thiết bị được kết nối internet để truyền tải dữ liệu với nhau và với đám mây mà không cần đến sự tương tác trực tiếp giữa người với người, hay giữa người với máy tính. Các thiết bị này bao gồm mọi thứ từ đồ gia dụng hàng ngày đến các công cụ công nghiệp phức tạp và được trang bị các công nghệ như WiFi, Bluetooth,... để giao tiếp và điều khiển từ xa”[1].

2. Vai trò của IoT trong đời sống hiện nay:

Với tính linh hoạt cao, Internet of Things (IoT) đã và đang trở thành công cụ quan trọng đối với nhiều doanh nghiệp, tổ chức và cả các cơ quan chính phủ.

- Trong đời sống cá nhân và sinh hoạt hàng ngày
 - **Nhà thông minh (Smart Home):** IoT cho phép người dùng điều khiển đèn, quạt, máy lạnh, camera an ninh và các thiết bị điện khác thông qua điện thoại thông minh hoặc giọng nói. Điều này không chỉ nâng cao sự tiện nghi mà còn giúp tiết kiệm năng lượng.
 - **Thiết bị đeo thông minh:** Các thiết bị như đồng hồ thông minh, vòng đeo tay theo dõi sức khỏe có khả năng đo nhịp tim, lượng calo tiêu thụ, chất lượng giấc ngủ và gửi cảnh báo đến người dùng hoặc bác sĩ.
 - **Tiết kiệm và tối ưu hóa năng lượng:** Hệ thống IoT có thể tự động điều chỉnh nhiệt độ phòng, tắt thiết bị khi không sử dụng, từ đó giúp tiết kiệm điện năng.
- Trong lĩnh vực Y tế (IoT trong y tế - IoMT)
 - **Giúp theo dõi bệnh nhân từ xa:** IoT giúp bác sĩ theo dõi tình trạng bệnh nhân liên tục thông qua cảm biến đo các chỉ số y tế. Các dữ liệu này được cập nhật theo thời gian thực, giúp kịp thời xử lý nếu có dấu hiệu bất thường.

- **Thiết bị y tế thông minh:** Máy trợ thở, máy theo dõi đường huyết, huyết áp,... có thể tự động thu thập dữ liệu và gửi đến hệ thống lưu trữ.
- **Quản lý thiết bị và vật tư:** IoT hỗ trợ quản lý vị trí, trạng thái sử dụng và bảo trì các thiết bị trong bệnh viện một cách hiệu quả.
- Trong nông nghiệp
 - **Giám sát môi trường canh tác:** Cảm biến đo độ ẩm, nhiệt độ đất, ánh sáng,... giúp nông dân hiểu rõ tình trạng cây trồng để điều chỉnh tưới tiêu, bón phân hợp lý.
 - **Tưới tiêu tự động:** Hệ thống tưới có thể tự kích hoạt khi độ ẩm đất thấp, giúp tiết kiệm nước và tăng năng suất.
 -
 - **Theo dõi vật nuôi:** Cảm biến được gắn trên vật nuôi để giám sát sức khỏe, định vị hoặc phát hiện bệnh sớm.
- Trong hệ thống giao thông:
 - **Giám sát và bảo trì thiết bị:** Cảm biến phát hiện nhiệt độ, độ rung, độ ẩm,... giúp phát hiện sớm sự cố trong máy móc công nghiệp, từ đó tránh thiệt hại lớn.
 - **Tự động hóa sản xuất:** IoT kết hợp với robot và AI giúp tự động hóa dây chuyền sản xuất, nâng cao hiệu suất và giảm chi phí nhân công.
 - **Quản lý kho và chuỗi cung ứng:** Cảm biến theo dõi hàng tồn kho, quản lý vận chuyển hàng hóa, chống thất thoát.
- Trong thương mại và bán lẻ
 - **Quản lý hàng hóa thông minh:** Thiết bị IoT giúp kiểm tra số lượng hàng tồn, hạn sử dụng, nhiệt độ bảo quản,... một cách chính xác.
 - **Trải nghiệm khách hàng cá nhân hóa:** Dữ liệu thu thập từ thiết bị IoT giúp cửa hàng hiểu được hành vi mua sắm của khách để đưa ra các gợi ý phù hợp.

CHƯƠNG II: CÁC THÀNH PHẦN CỨNG VÀ PHẦN MỀM SỬ DỤNG

1. Giới thiệu những thiết bị sử dụng:

1.1. ESP32:

- “ESP32 là một vi điều khiển tích hợp WiFi và Bluetooth do Espressif Systems phát triển. Đây là phiên bản nâng cấp của dòng ESP8266, nổi bật với khả năng xử lý mạnh mẽ, tiêu thụ điện năng thấp, phù hợp với các ứng dụng IoT hiện đại”[2].

- Một số đặc điểm nổi bật của ESP32:

- Vi xử lý lõi kép (dual-core) 32-bit Tensilica LX6
- Tích hợp WiFi chuẩn 802.11 b/g/n và Bluetooth 4.2 BLE
- Hỗ trợ nhiều giao thức giao tiếp: UART, SPI, I2C, PWM, ADC, DAC,...
- Bộ nhớ Flash và RAM lớn hơn so với ESP8266
- Có thể chạy các chương trình phức tạp, kết nối Internet, giao tiếp cảm biến, hiển thị dữ liệu và điều khiển thiết bị từ xa.

1.2. Cảm biến BME280 và màn hình lcd2004:

a) Cảm biến BME280:

- “Cảm biến BME280 là một module cảm biến môi trường do hãng Bosch Sensortec phát triển”[3]. Cảm biến này tích hợp ba chức năng đo trong một:

- Nhiệt độ (Temperature)
- Độ ẩm (Humidity)
- Áp suất khí quyển (Barometric Pressure)

- Thông số kỹ thuật chính:

- Dải đo nhiệt độ: từ -40°C đến $+85^{\circ}\text{C}$ ($\pm 1.0^{\circ}\text{C}$)
- Dải đo độ ẩm: từ 0% đến 100% RH ($\pm 3\%$ RH)
- Dải đo áp suất: từ 300 hPa đến 1100 hPa (± 1 hPa)
- Giao tiếp: I2C hoặc SPI
- Điện áp hoạt động: 1.8V – 3.6V

- Ưu điểm:
 - Kích thước nhỏ gọn, tích hợp ba chức năng trong một.
 - Độ chính xác cao.
 - Tiêu thụ năng lượng thấp.
 - Dễ tích hợp với vi điều khiển như ESP32 qua giao tiếp I2C.

b) Màn hình LCD2004:

- “LCD 2004 là loại màn hình tinh thể lỏng có khả năng hiển thị 20 ký tự trên 4 dòng, thường được sử dụng để hiển thị dữ liệu trong các hệ thống nhúng”[4].
- Đặc điểm nổi bật:
 - Có thể hiển thị dữ liệu dạng chữ và số
 - Sử dụng module I2C, giúp:
 - Tiết kiệm chân kết nối (chỉ cần SDA và SCL)
 - Giao tiếp dễ dàng với ESP32 qua thư viện LiquidCrystal_I2C
 - Điện áp hoạt động: 5V.
 - Có thể điều chỉnh độ tương phản và bật/tắt đèn nền.
 - Lợi ích khi dùng trong dự án:
 - Hiển thị đồng thời nhiều thông tin như: nhiệt độ, độ ẩm, áp suất
 - Phù hợp cho cả dự án mô phỏng và triển khai thực tế.
 - Thân thiện với người mới học lập trình nhúng.
 - Kết nối với ESP32:
 - SDA (Data) → Chân D21 (mặc định)
 - SCL (Clock) → Chân D22 (mặc định)

1.3. Tổng quan về Telegram:

1.3.1. Khái niệm về Telegram?

- “Telegram là một ứng dụng nhắn tin nhanh, mã nguồn mở, hoạt động đa nền tảng, nổi bật nhờ:
 - Bảo mật cao.
 - Giao diện thân thiện.

- Tính năng tạo bot để tự động hóa hoặc kết nối với hệ thống bên ngoài.
- Với khả năng tạo Telegram Bot, người dùng có thể dễ dàng:
 - Gửi/nhận thông báo từ các hệ thống IoT.
 - Tương tác điều khiển thiết bị từ xa.
 - Giám sát dữ liệu từ cảm biến theo thời gian thực.”[5]

1.3.2. Khái niệm về Telegram Bot:

- “**Telegram Bot** là một tài khoản đặc biệt hoạt động thông qua API. Nó có thể:
 - Trả lời tin nhắn
 - Gửi dữ liệu dạng văn bản, hình ảnh, file
 - Tự động xử lý các lệnh như /getdata, /start,...” [6]
- Việc sử dụng bot cho dự án IoT giúp truyền nhận dữ liệu dễ dàng mà không cần giao diện web phức tạp.

1.3.3. Tích hợp Telegram vào ESP32:

- Các bước ESP32 giao tiếp Telegram:

Bước 1: Tạo Telegram Bot

- Mở ứng dụng Telegram, tìm @BotFather
- Dùng lệnh /newbot để tạo bot mới
- Lưu lại Token API do BotFather cấp

Bước 2: Tạo nhóm, thêm bot và lấy chatId nhóm

- Gửi tin nhắn bất kỳ cho bot
- Truy cập <https://api.telegram.org/bot<TOKEN>/getUpdates>
→ Thay <TOKEN> bằng token bot bạn vừa tạo
- Lấy chat_id trong kết quả JSON

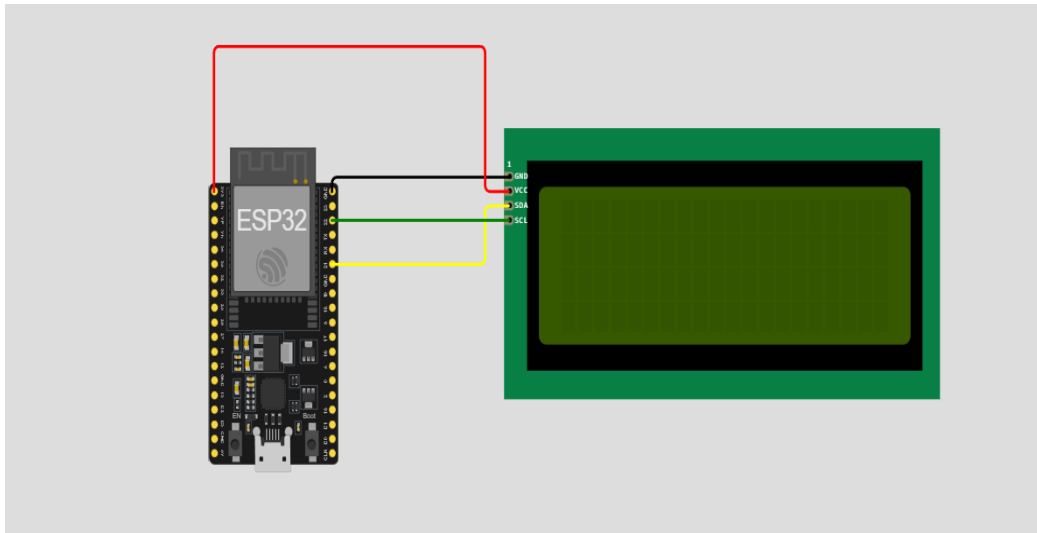
Bước 3: Kết nối ESP32 với bot

- Dùng thư viện WiFi, WiFiClientSecure, UniversalTelegramBot
- Cài thư viện:
 - UniversalTelegramBot
 - WiFiClientSecure
- Cấu hình ESP32:
 - WiFiClientSecure client;
 - UniversalTelegramBot bot(BOT_TOKEN, client);
- Sử dụng các hàm:
 - bot.sendMessage(chat_id, message, ""); để gửi tin nhắn
 - bot.getUpdates(...) để kiểm tra lệnh từ người dùng

2. Thiết kế hệ thống trạm thời tiết Mini:

- Thành phần sử dụng
 - ESP32 DevKit V4: Bộ vi điều khiển chính, có tích hợp WiFi giúp dễ dàng kết nối Internet để gửi dữ liệu về Telegram.
 - Màn hình LCD 20x4 I2C (LCD2004): Dùng để hiển thị dữ liệu cảm biến. Giao tiếp I2C giúp kết nối đơn giản hơn.
 - Giả lập cảm biến BME280: Do không có cảm biến thực tế, dữ liệu được thay thế bằng các giá trị được sinh ngẫu nhiên mô phỏng thông số của BME280 (nhiệt độ, độ ẩm, áp suất).
 - Mạng WiFi mô phỏng Wokwi-GUEST: Giúp ESP32 kết nối Internet trong môi trường giả lập.
- Sơ đồ kết nối (mô phỏng trên Wokwi)
 - SDA của LCD → GPIO21 (SDA trên ESP32)
 - SCL của LCD → GPIO22 (SCL trên ESP32)
 - VCC của LCD → 3V3
 - GND của LCD → GND

❖ Dưới đây là sơ đồ mô phỏng trạm thời tiết:



3. Triển khai phần mềm và xử lý dữ liệu:

3.1. Phần mềm sử dụng:

- Trong quá trình thực hiện đề tài, em sử dụng công cụ Wokwi – một nền tảng mô phỏng phần cứng trực tuyến – để triển khai và kiểm tra chương trình. Việc sử dụng Wokwi giúp em dễ dàng mô phỏng hệ thống mà không cần đến thiết bị phần cứng thực tế, phù hợp với điều kiện học tập hiện tại.

Về mặt lập trình, em sử dụng Arduino IDE kết hợp với thư viện cần thiết như:

- WiFi.h để kết nối mạng không dây,
- Wire.h để giao tiếp I2C,
- LiquidCrystal_I2C.h để điều khiển màn hình LCD,
- UniversalTelegramBot.h và WiFiClientSecure.h để gửi/nhận tin nhắn qua Telegram.

3.2. Giải pháp thay thế cảm biến BME280:

- Vì không có cảm biến BME280 thực tế, em đã thay thế việc lấy dữ liệu từ cảm biến bằng cách sử dụng hàm `esp_random()` trong ESP32 để tạo ra các giá trị ngẫu nhiên. Cụ thể:
- Nhiệt độ được tạo ngẫu nhiên trong khoảng 15.00 – 30.00 °C
- Độ ẩm trong khoảng 30.0 – 100.0 %
- Áp suất khoảng 980 – 1050 hPa.

- Việc này đảm bảo hệ thống vẫn có thể hoạt động bình thường và phục vụ cho việc mô phỏng xử lý dữ liệu, hiển thị và truyền thông qua Telegram.

3.3. Kết nối Wifi mô phỏng:

- ESP32 được lập trình để kết nối với mạng WiFi mô phỏng “Wokwi-GUEST” mà không cần mật khẩu. Quá trình kết nối được thực hiện trong hàm setup() bằng lệnh:
 - `WiFi.begin("Wokwi-GUEST", "");`

3.4. Xử lý và hiển thị dữ liệu:

- Dữ liệu ngẫu nhiên sau khi được tạo sẽ được hiển thị trực tiếp trên màn hình LCD 20x4 sử dụng giao tiếp I2C với địa chỉ 0x27. Thông tin hiển thị bao gồm:
 - Nhiệt độ (°C),
 - Độ ẩm (%),
 - Áp suất (hPa).
- Mỗi 4 giây, màn hình sẽ được cập nhật lại với dữ liệu mới. Lệnh `lcd.clear()` được sử dụng để làm mới nội dung trước khi hiển thị lại.

3.5. Dữ liệu gửi về Telegram:

- Hệ thống sử dụng Telegram Bot để giao tiếp với người dùng từ xa. Khi người dùng gửi lệnh `/getdata` đến bot, chương trình sẽ phản hồi bằng một chuỗi tin nhắn chứa các giá trị nhiệt độ, độ ẩm, áp suất tại thời điểm đó.
- Việc này được thực hiện thông qua hàm `bot.sendMessage(chat_id, message, "")`, cho phép hệ thống phản hồi nhanh chóng và tiện lợi với người dùng trên ứng dụng Telegram.

3.6. Xử lý lệnh và cập nhật dữ liệu:

- Chương trình liên tục kiểm tra có tin nhắn mới từ Telegram hay không bằng cách gọi `bot.getUpdates(bot.last_message_received + 1)`. Nếu có tin nhắn, chương trình duyệt qua từng tin và nếu người gọi đúng lệnh `“/getdata”` chương trình sẽ trả về dữ liệu tương ứng.

3.7. Mã nguồn chương trình TramThoiTietMini:

```
#include <WiFi.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <UniversalTelegramBot.h>

#include <WiFiClientSecure.h>

#include <secret.h>


#define SDA_PIN 21 // Chân SDA của ESP32

#define SCL_PIN 22 // Chân SCL của ESP32


// Kết nối WiFi

const char *WIFI_SSID = "Wokwi-GUEST";

const char *WIFI_PASSWORD = ""; // Không có mật khẩu


// Thông tin Telegram Bot


WiFiClientSecure client;

UniversalTelegramBot bot(BOT_TOKEN, client);


LiquidCrystal_I2C lcd(0x27, 20, 4); // Địa chỉ I2C của LCD

unsigned long lastCheck = 0;    // Thời gian kiểm tra tin nhắn
```



```

void setup()

{

  Serial.begin(115200);

  Wire.begin(SDA_PIN, SCL_PIN); // Khởi tạo giao tiếp I2C

  lcd.begin(20, 4);

  lcd.backlight();


  // Kết nối WiFi

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("Đang kết nối WiFi");

  while (WiFi.status() != WL_CONNECTED)

  {

    Serial.print(".");

    delay(500);

  }

  Serial.println("\nĐã kết nối WiFi");


  // Cấu hình kết nối TLS cho Telegram

  client.setInsecure();


  lcd.setCursor(0, 0);

  lcd.print("ESP32 Ready!");

}

```

```

void loop()

{
    // Sinh giá trị ngẫu nhiên

    float temperature = (esp_random() % 1501 + 1500) / 100.0;

    float humidity = (esp_random() % 701 + 300) / 10.0;

    float pressure = (esp_random() % 71 + 980);


    // Hiển thị lên LCD

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Nhiệt Độ: ");

    lcd.print(temperature);

    lcd.print(" C");


    lcd.setCursor(0, 1);

    lcd.print("Độ Ẩm: ");

    lcd.print(humidity);

    lcd.print(" %");


    lcd.setCursor(0, 2);

    lcd.print("Áp Suất: ");

    lcd.print(pressure);

    lcd.print(" hPa");

```

```
delay(4000); // Cập nhật mỗi 4 giây
```

```
// Kiểm tra tin nhắn Telegram mỗi 5 giây
```

```
if (millis() - lastCheck > 5000)
```

```
{
```

```
    lastCheck = millis();
```

```
    int newMessages = bot.getUpdates(bot.last_message_received + 1);
```

```
    while (newMessages)
```

```
    {
```

```
        for (int i = 0; i < newMessages; i++)
```

```
        {
```

```
            String chat_id = bot.messages[i].chat_id;
```

```
            String text = bot.messages[i].text;
```

```
            Serial.println("Tin nhắn từ Telegram: " + text);
```

```
            if (text == "/getdata")
```

```
            {
```

```
                String message = "📊 Dữ liệu thời tiết:\n";
```

```
                message += "🌡️ Nhiệt độ: " + String(temperature) + " °C\n";
```

```
                message += "💧 Độ ẩm: " + String(humidity) + " %\n";
```

```
                message += "🌪️ Áp suất: " + String(pressure) + " hPa";
```

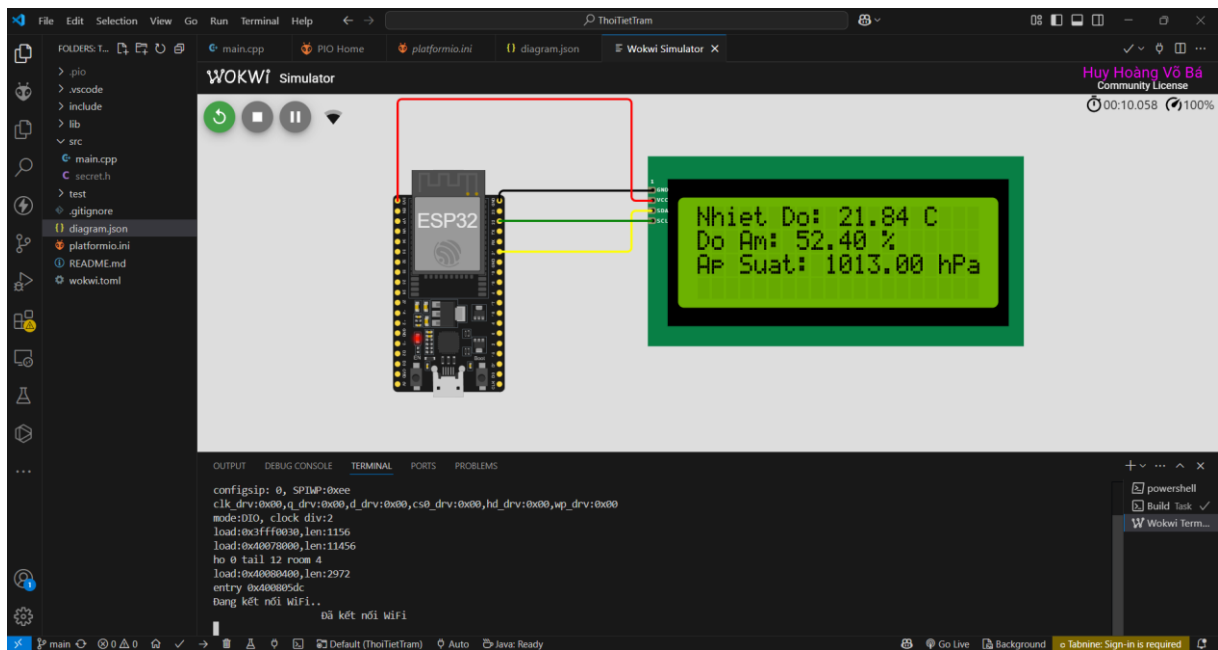
```
        bot.sendMessage(chat_id, message, "");  
    }  
}  
newMessages = bot.getUpdates(bot.last_message_received + 1);  
}  
}}
```

CHƯƠNG III: KẾT QUẢ MÔ PHỎNG

1. Kết quả mô phỏng:

Bước 1 : Build mã nguồn của chương trình ở mục 5.7

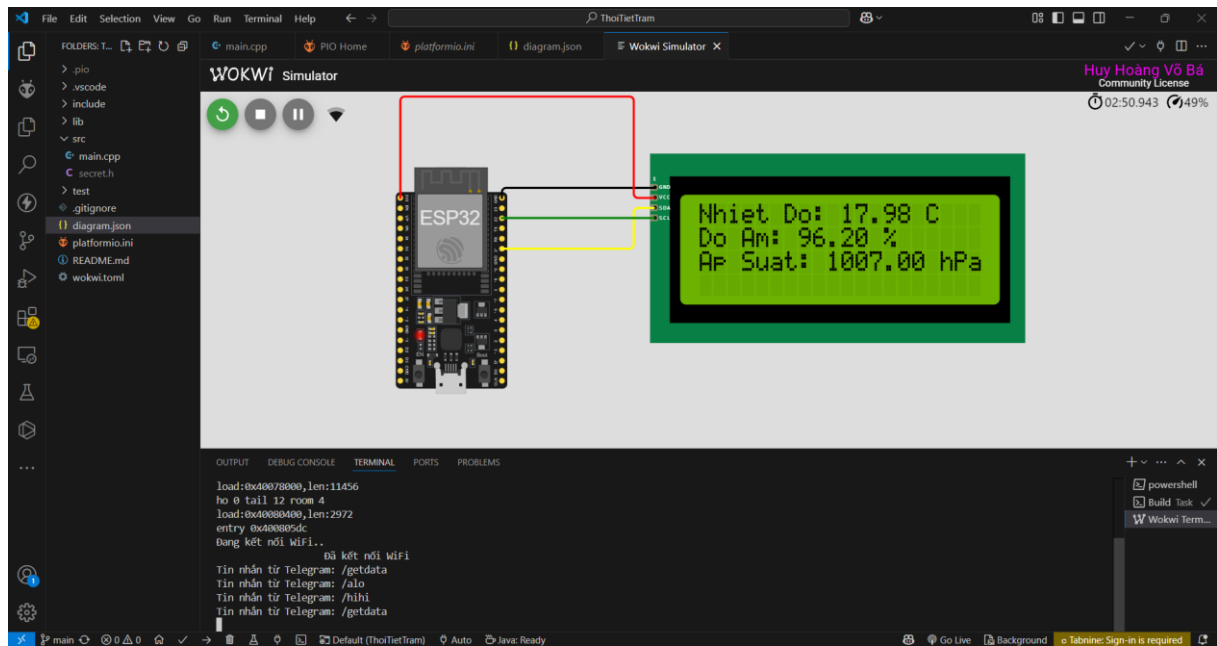
Bước 2: Sau khi hoàn thành B1 chạy mô phỏng wokwi, hình ảnh mô phỏng như sau:



Bước 3: Từ màn hình giao diện ứng dụng Tele tại nhóm đã tạo, gõ lệnh “/getdata”, hình ảnh mô phỏng như sau:



Bước 4: Sau khi gõ lệnh chương trình sẽ nhận lệnh, nếu đúng lệnh sẽ thông báo gửi tin nhắn về thông tin nhiệt độ, độ ẩm và áp suất ở nhóm chat.



2. Mã nguồn và mô phỏng dự án:

- ✓ Dự án được công khai tại:

https://github.com/Bahuy24503/TramThoiTietMiNi_IOT

PHẦN KẾT LUẬN

1. Nhận xét và đánh giá:

Trong đề tài này, em đã tìm hiểu và mô phỏng thành công một hệ thống trạm thời tiết mini sử dụng vi điều khiển ESP32 và cảm biến BME280 có khả năng đo nhiệt độ, độ ẩm và áp suất (bằng cách sinh ngẫu nhiên giá trị thay thế cho cảm biến thực tế), hiển thị dữ liệu trên màn hình LCD, đồng thời gửi thông tin về Telegram khi người dùng yêu cầu.

Việc mô phỏng giúp em chủ động hơn trong kiểm thử chương trình, đặc biệt khi không có sẵn phần cứng. Em cũng học được cách sinh dữ liệu giả lập (random) để kiểm tra chức năng hệ thống, một kỹ năng rất quan trọng khi làm việc trong môi trường thực tế hoặc nghiên cứu.

Ngoài ra, quá trình cấu hình WiFi, kết nối đến Telegram Bot và xử lý lệnh từ người dùng qua chat là một trải nghiệm mới mẻ, giúp em hiểu rõ hơn về giao tiếp qua mạng và sử dụng API.

Tuy nhiên, em cũng nhận thấy một số hạn chế:

- Không có thiết bị thực tế nên chưa thể kiểm tra tính chính xác của dữ liệu đo.
- Một số thành phần mô phỏng không hoàn toàn giống với thực tế, nhất là cảm biến hoặc độ trễ mạng.
- Giao tiếp với Telegram đôi khi bị giới hạn do tốc độ phản hồi và yêu cầu bảo mật TLS.

2. Kiến nghị:

Nếu có điều kiện tốt hơn để tiếp cận phần cứng:

- Có thể tích hợp cảm biến BME280 thật để thu thập dữ liệu chính xác hơn.
- Gửi dữ liệu thời tiết theo thời gian thực lên ThingSpeak, Blynk, v.v..., giúp dễ dàng theo dõi xu hướng nhiệt độ, độ ẩm, áp suất theo thời gian.
- Tích hợp thêm cảm biến ánh sáng, cảm biến mưa, hoặc cảm biến bụi mịn để mở rộng chức năng trạm thời tiết

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] Viettel IDC, "IoT là gì? Những ứng dụng nổi bật của IoT"

<https://viettelidc.com.vn/tin-tuc/iot-la-gi-nhung-ung-dung-noi-bat-cua-iot>

[2] IoT Zone, "Giới thiệu ESP32 là gì?"

https://www.iotzone.vn/esp32/esp32-co-ban/gioi-thieu-esp32-la-gi/?utm_source=chatgpt.com

[3] Mecsu.vn, "Cảm biến áp suất và độ ẩm BME280"

https://mecsu.vn/ho-tro-ky-thuat/cam-bien-ap-suat-va-do-am-bme280.Dg5?utm_source=chatgpt.com

[4] Icdayroi.com, "LCD 2004 tích hợp sẵn module chuyển đổi I2C"

https://icdayroi.com/lcd-2004-tich-hop-san-module-chuyen-doi-i2c?utm_source=chatgpt.com

[5] Memart.vn, "Tổng quan về bot Telegram là gì và cách sử dụng để tiện lợi cho cuộc sống"

https://memart.vn/tin-tuc/blog/tong-quan-ve-bot-telegram-la-gi-va-cach-su-dung-de-tien-loi-cho-cuoc-song-vi-cb.html?utm_source=chatgpt.com

Tiếng Anh:

[6] Telegram, "Telegram Bot API"

<https://core.telegram.org/bots/api>