

TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG
KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO CUỐI KỲ
HỌC PHẦN: HỆ THỐNG THỜI GIAN
THỰC

ĐỀ TÀI: SMART POWER OUTLET
BASED ON CAN BUS

Giảng viên hướng dẫn: TS. Đào Duy Tuấn

Nhóm: 5VAP

Sinh viên thực hiện: Nguyễn Văn An – 106210045

Phạm Thị Phương – 106210050

Dương Thị Thảo Vi - 106210259

Đà Nẵng, tháng 6 năm 2025

LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0 hiện nay, nhu cầu tự động hóa và điều khiển thông minh trong đời sống sinh hoạt hằng ngày ngày càng trở nên phổ biến. Các thiết bị gia dụng truyền thống đang dần được thay thế bởi những hệ thống thông minh, giúp con người tiết kiệm thời gian, năng lượng và nâng cao chất lượng cuộc sống. Một trong những ứng dụng tiêu biểu của xu hướng này là ô cắm điện thông minh sử dụng CAN bus, cho phép người dùng điều khiển và giám sát thiết bị điện từ xa thông qua điện thoại hoặc máy tính, đồng thời đảm bảo an toàn và tối ưu hóa việc sử dụng điện năng.

Dưới sự hướng dẫn của TS. Đào Duy Tuấn phụ trách học phần “Hệ thống thời gian thực”, nhóm đã thực hiện thành công đề tài **“Ô cắm điện thông minh sử dụng CAN bus”**. Báo cáo này trình bày toàn bộ quá trình nhóm đã thực hiện: từ giới thiệu, phân tích yêu cầu, phương pháp thực hiện, đến kiểm thử hệ thống trên tình huống thực tế. Thông qua đề tài này, nhóm mong muốn đóng góp một giải pháp khả thi và có tính ứng dụng cao cho các hệ thống thời gian thực trong thời gian tới.

MỤC LỤC

BẢNG PHÂN CÔNG NHIỆM VỤ.....	2
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	4
2.1 Giới thiệu về ESP32-C3:.....	4
2.2 Giới thiệu về STM32F103:.....	6
2.3 Giới thiệu về FreeRTOS:.....	9
2.4 Giới thiệu về Blynk:.....	10
2.5 Giới thiệu về giao tiếp CAN bus:.....	11
2.6 Giới thiệu về giao tiếp UART:.....	12
CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN.....	14
3.1 Sơ đồ kết nối thiết bị:.....	14
3.2 Các linh kiện sử dụng trong dự án:.....	14
3.2.1 Vi điều khiển ESP32 và STM32:.....	15
3.2.2 Cảm biến DHT11:.....	15
3.2.3 Màn hình OLED:.....	16
3.2.4 Relay:.....	18
3.2.5 Module MCP2551:.....	19
3.2.6 Các thiết bị hỗ trợ khác:.....	20
3.3 Sơ đồ mô tả hệ thống:.....	21
3.3.1 Lưu đồ thuật toán:.....	21
3.3.2 Sơ đồ Use Case:.....	22
3.3.3 Sơ đồ Activity:.....	23
3.3.4 Sơ đồ Sequence:.....	24
3.4 Nguyên lý làm việc của thiết bị:.....	24
3.5 Phần mềm sử dụng:.....	26
CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ, HƯỚNG PHÁT TRIỂN.....	27
4.1 Kết quả đạt được:.....	27
4.2 Hướng phát triển của đề tài:.....	31
TÀI LIỆU THAM KHẢO.....	32
PHỤ LỤC CODE.....	34

BẢNG PHÂN CÔNG NHIỆM VỤ

Họ và tên	Công việc	Tỷ lệ đóng góp
Nguyễn Văn An	Tìm hiểu đề tài, tổng hợp, thực hiện dự án, sửa lỗi	34%
Phạm Thị Phương	Lên ý tưởng, thực hiện dự án, viết chương trình	33%
Dương Thị Thảo Vi	Lên ý tưởng, thực hiện dự án, viết chương trình	33%

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

Trong xu thế phát triển của nhà thông minh (Smart Home), việc điều khiển và giám sát thiết bị điện từ xa trở thành một nhu cầu thiết yếu. Đề tài “Ô cảm điện thông minh sử dụng CAN bus” được xây dựng nhằm mục tiêu tạo ra một thiết bị ô cảm có khả năng kết nối WiFi, điều khiển qua ứng dụng Blynk và hoạt động theo thời gian thực. Ô cảm điện được tích hợp WiFi cho phép người dùng bật/tắt thủ công hoặc hẹn giờ qua ứng dụng, đồng thời có khả năng đo nhiệt độ, độ ẩm và tự động ngắt khi nhiệt độ vượt ngưỡng an toàn.

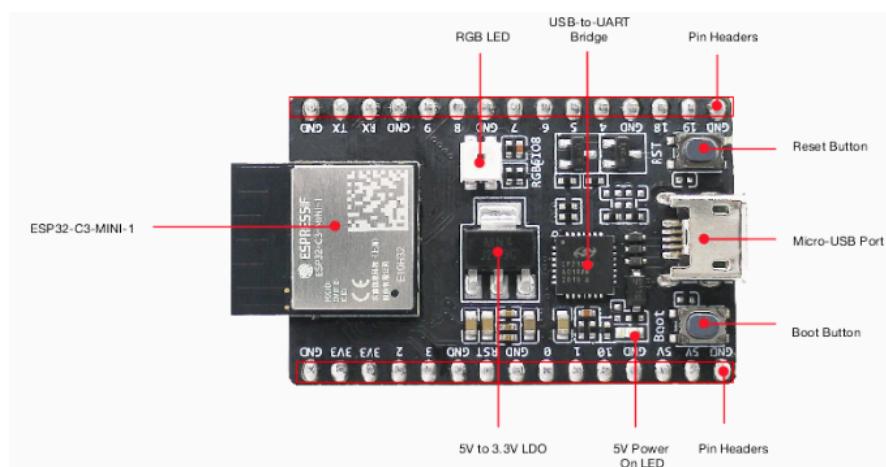
Bên cạnh đó, thiết bị còn có chức năng hiển thị thông số môi trường trên màn hình OLED, tự động chuyển đổi chế độ online/offline tùy vào tình trạng kết nối Internet, cũng như gửi cảnh báo đến người dùng trong các tình huống nguy hiểm. Đặc biệt, người dùng có thể cấu hình thông tin kết nối WiFi cho ô cảm một cách dễ dàng.

Qua việc kết hợp phần cứng và phần mềm điều khiển từ xa, đề tài không chỉ nâng cao tính tiện ích và an toàn trong sử dụng điện mà còn thể hiện được khả năng ứng dụng của hệ thống nhúng và Internet of Things (IoT) vào đời sống thực tiễn.

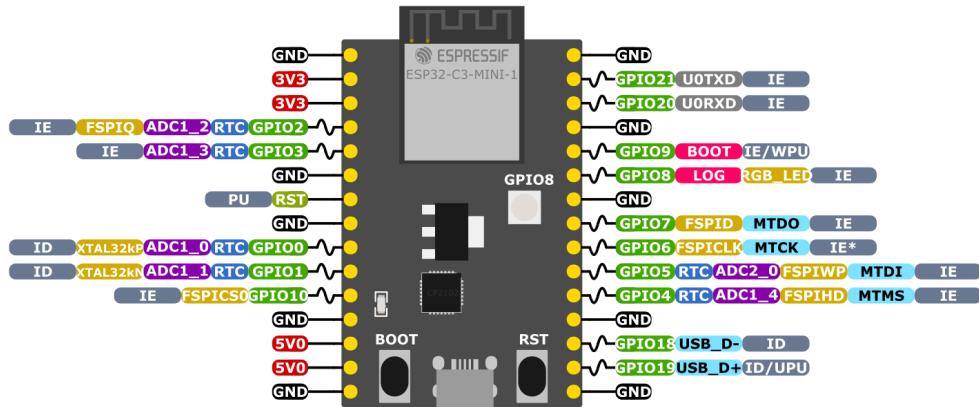
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về ESP32-C3:

ESP32-C3 là một vi điều khiển 32-bit do Espressif Systems phát triển, thuộc dòng chip ESP32 với kiến trúc RISC-V mạnh mẽ và hiện đại. Chip được tích hợp WiFi 2.4GHz và Bluetooth 5 (LE), cho phép giao tiếp không dây linh hoạt trong các ứng dụng Internet of Things (IoT). Với bộ nhớ Flash và RAM tích hợp đủ lớn, ESP32-C3 hỗ trợ các tác vụ kết nối mạng, điều khiển thiết bị, xử lý dữ liệu cảm biến một cách hiệu quả.



Hình 2.1: Vị trí các chân của vi điều khiển ESP32-C3

**ESP32-C3 Specs**

32-bit RISC-V single-core @160MHz
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz
 Bluetooth LE 5
 400 KB SRAM (16 KB for cache)
 384 KB ROM
 22 GPIOs, 3x SPI, 2x UART, I2C,
 I2S, RMT, LED PWM, USB Serial/JTAG,
 GDMA, TWAI®, 12-bit ADC

GPIOX: PWM Capable Pin
GPIO: GPIO Input and Output
JTAG/USB: JTAG for Debugging and USB
FLASH: External Flash Memory (SPI)
ADC_X_C: Analog-to-Digital Converter
OTHER: Other Related Functions
SERIAL: Serial for Debug/Programming
STRAP: Strapping Pin Functions

RTC: RTC Power Domain (VDD3P3_RTC)
GND: Ground
PWR: Power Rails (3V3 and 5V)

GPIO STATE

UPU : USB Weak Pull-up (Internal)	WPU : Weak Pull-up (Internal)
WPD : Weak Pull-down (Internal)	PUD : Pull-up (External)
IE : Input Enable (After Reset)	ID : Input Disabled (After Reset)
OE : Output Enable (After Reset)	OD : Output Disabled (After Reset)

IE*: Input Enable (Depends of FUSE_DIS_PAD_JTAG)

Hình 2.2: Sơ đồ kết nối các chân của ESP32-C3

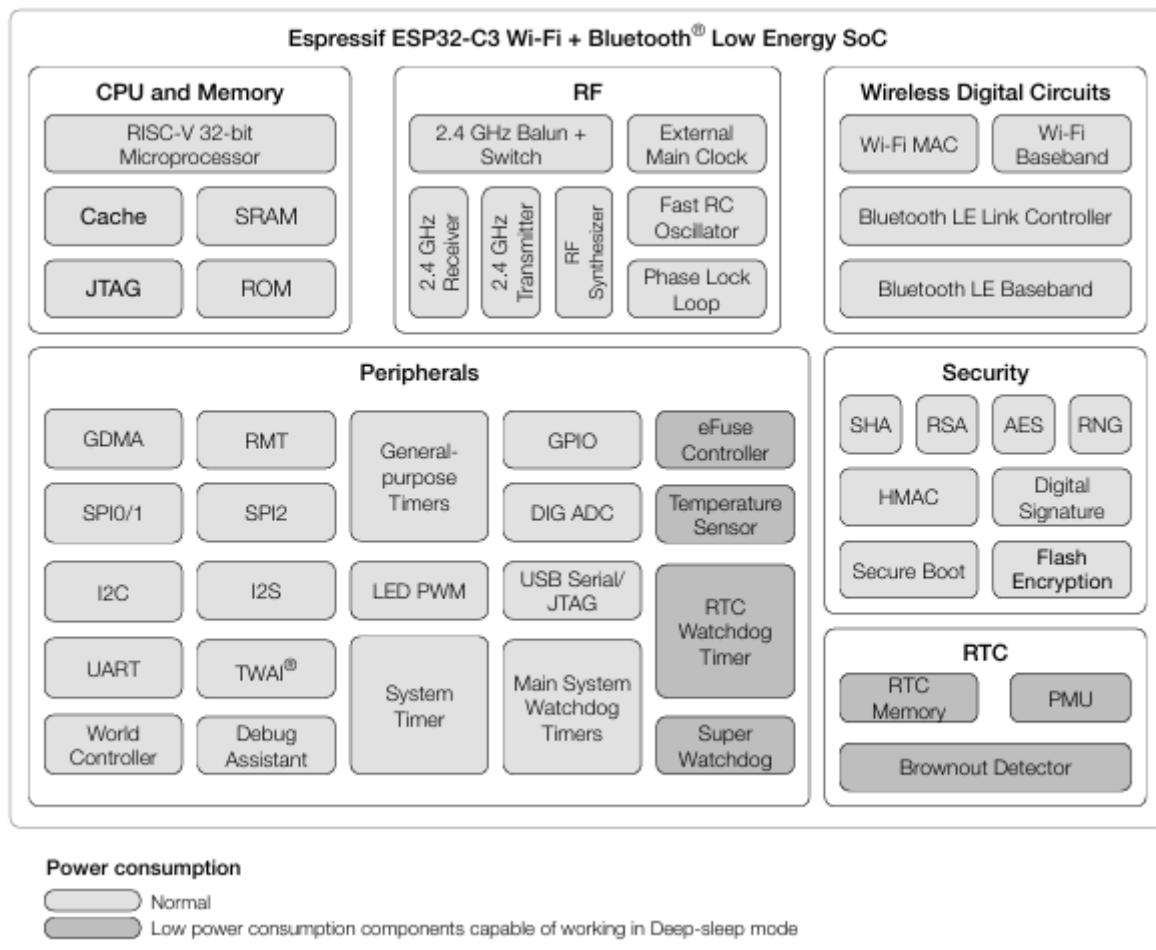
ESP32-C3 là một hệ thống trên chip (SoC), đi kèm bộ nhớ ROM, SRAM, cache và hỗ trợ giao tiếp JTAG cho mục đích lập trình và debug.

Về kết nối không dây, ESP32-C3 tích hợp đầy đủ các khối mạch RF bao gồm bộ khuếch đại công suất, bộ dao động nội/ngoại, và mạch điều chế/dải tần giúp hỗ trợ Wi-Fi 2.4 GHz và Bluetooth 5 (LE) ổn định. Các khối Wi-Fi MAC, Bluetooth Link Controller và Baseband xử lý dữ liệu không dây hiệu quả, hỗ trợ đa giao thức.

Ngoài ra, ESP32-C3 cung cấp nhiều ngoại vi phong phú như SPI, I2C, UART, PWM, ADC, GPIO, I2S, USB Serial, cùng với hệ thống Timer và Watchdog giúp điều khiển thời gian và bảo vệ hệ thống khỏi lỗi treo. Chip còn tích hợp bộ điều khiển nhiệt độ, eFuse, và các công cụ hỗ trợ phát triển như Debug Assistant.

Về bảo mật, ESP32-C3 trang bị đầy đủ các chức năng như SHA, RSA, AES, HMAC, Flash Encryption, Secure Boot, và Random Number Generator (RNG), đảm bảo an toàn dữ liệu và kết nối. Cuối cùng, hệ thống RTC (Real-Time Clock) và Power

Management Unit (PMU) hỗ trợ hoạt động ở chế độ tiêu thụ điện năng cực thấp, phù hợp cho các ứng dụng yêu cầu chạy liên tục trong thời gian dài.



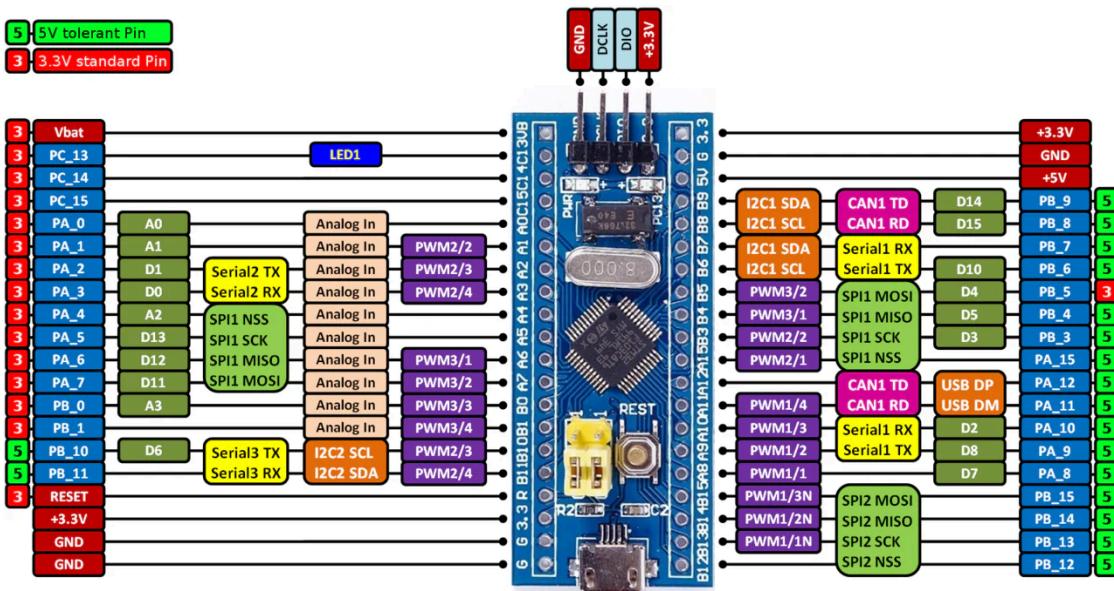
Hình 2.3: Sơ đồ khái niệm các chức năng của ESP32-C3

Ưu điểm nổi bật của ESP32-C3 nằm ở khả năng tiêu thụ điện năng thấp, bảo mật phản ứng mạnh mẽ và tính tương thích với các công cụ phát triển phổ biến như Arduino IDE, PlatformIO và ESP-IDF. Nhờ sự linh hoạt và hiệu suất cao, ESP32-C3 rất phù hợp cho các ứng dụng như điều khiển thiết bị thông minh, giám sát môi trường, nhà thông minh và các hệ thống điều khiển từ xa trong thời gian thực.

2.2 Giới thiệu về STM32F103:

STM32F103 là vi điều khiển 32-bit của STMicroelectronics, thuộc dòng STM32 sử dụng lõi ARM Cortex-M3. Nó có hiệu suất cao, tiêu thụ điện thấp và được ứng dụng

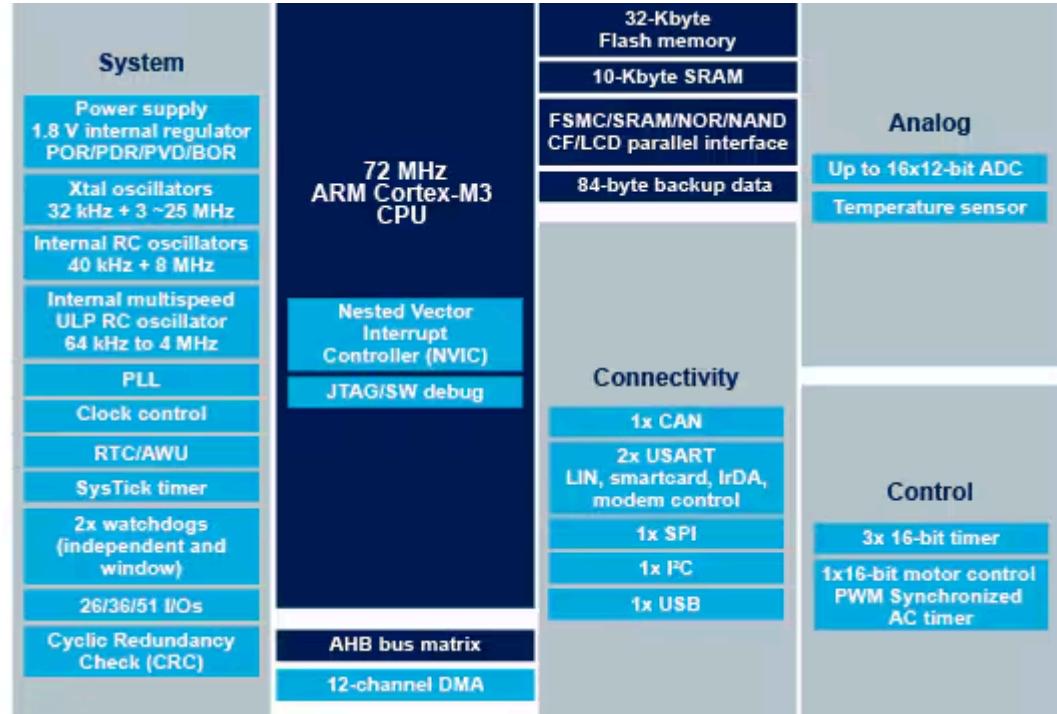
rộng rãi trong nhúng và IoT. STM32F103 nổi bật với hiệu suất ổn định, giá thành hợp lý và hỗ trợ thư viện FreeRTOS, STM32 HAL/CMSIS giúp phát triển dễ dàng.



Hình 2.4: Vị trí điều khiển và sơ đồ kết nối các chân của STM32F103

Type	Parameter
Core	ARM Cortex M3
Core Size	32-Bit Single-Core
Program Memory Size	32 kB
Data Bus Width	32 bit
ADC Resolution	12 bit
Maximum Clock Frequency	72 MHz
RAM Size	10K x 8
Supply Voltage - Min	1.8 V, 2 V
Supply Voltage - Max	3.6 V
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Connectivity	CANbus, I2C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	DMA, Motor Control PWM, PDR, POR, PVD, PWM, Temp Sensor, WDT
Number of I/Os	48 I/O
Operating Temperature	-40°C ~ 85°C (TA)
Package / Case	48-LQFP

Hình 2.5: Thông số kỹ thuật của STM32F103C6T6



Hình 2.6: Sơ đồ khái niệm các chức năng của STM32F103C6T6

STM32F103C6T6 sử dụng lõi xử lý ARM Cortex-M3 hoạt động ở tần số 72 MHz, được thiết kế cho các ứng dụng nhúng yêu cầu hiệu suất cao, độ tin cậy và tiêu thụ điện năng thấp. Vi điều khiển này tích hợp bộ nhớ Flash 32KB và SRAM 10KB, đáp ứng tốt các ứng dụng nhúng cỡ nhỏ đến trung bình.

Về hệ thống lõi, STM32F103C6T6 được trang bị bộ điều khiển ngắt vectơ (NVIC), JTAG/SWD debug, cùng với các bộ tạo dao động nội/ngoại đa dạng từ 32kHz đến 25MHz, hỗ trợ linh hoạt cho nhiều ứng dụng thời gian thực. Chip cũng có bộ cấp nguồn tích hợp 1.8V, bộ giám sát nguồn (POR/PDR/BOR) và PLL để điều chỉnh tần số hệ thống.

Các chức năng ngoại vi (peripherals) của vi điều khiển này rất phong phú, bao gồm:

- Analog: tích hợp bộ ADC 12-bit với 16 kênh, cảm biến nhiệt độ nội.
- Kết nối: hỗ trợ nhiều giao tiếp như 2x USART, SPI, I2C, CAN, USB, đáp ứng đa dạng nhu cầu truyền thông.

- Điều khiển (Control): có 3 bộ đếm 16-bit thông thường, cùng với 1 bộ PWM/AC Timer đồng bộ 16-bit, thích hợp cho các ứng dụng điều khiển động cơ hoặc tạo xung chính xác.
- Bộ định thời và giám sát: bao gồm SysTick timer, RTC/AWU, và 2 watchdogs độc lập, giúp tăng độ ổn định và an toàn của hệ thống.
- DMA và I/O: trang bị 12 kênh DMA, hỗ trợ 26/36/51 chân I/O tùy theo dạng đóng gói.

Với khả năng xử lý mạnh mẽ, giao tiếp phong phú và hoạt động ổn định, STM32F103C6T6 là lựa chọn phổ biến trong các hệ thống nhúng như thiết bị công nghiệp, điều khiển tự động, và các ứng dụng IoT.

2.3 Giới thiệu về FreeRTOS:



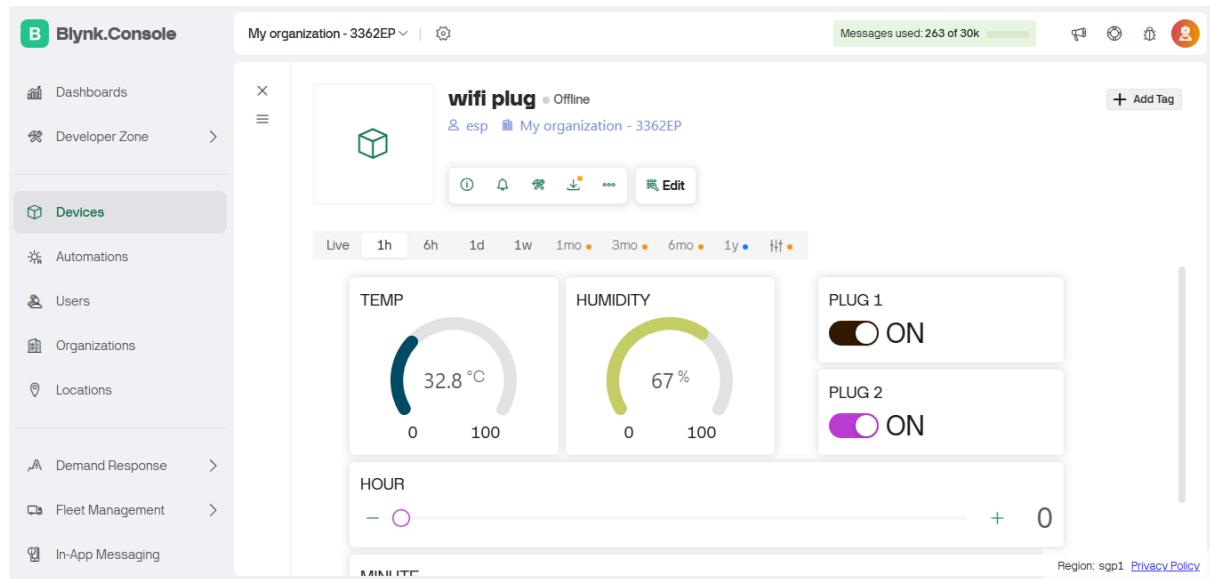
Hình 2.7: Logo của FreeRTOS

FreeRTOS (Free Real-Time Operating System) là một hệ điều hành thời gian thực mã nguồn mở, được thiết kế dành cho các hệ thống nhúng có tài nguyên hạn chế. Với khả năng hoạt động nhẹ, hiệu quả và dễ tích hợp, FreeRTOS cho phép quản lý đa tác vụ (multitasking), đồng thời hỗ trợ các cơ chế đồng bộ như semaphore, mutex, hàng đợi (queue) và timer mềm.

FreeRTOS cung cấp khả năng lập lịch ưu tiên (priority-based scheduling), giúp các tác vụ quan trọng được thực thi kịp thời, đảm bảo tính đáp ứng thời gian thực trong hệ thống. Hệ điều hành này hỗ trợ nhiều kiến trúc vi điều khiển khác nhau như ARM Cortex-M, RISC-V, AVR, MSP430, v.v., và có thể dễ dàng mở rộng hoặc tích hợp với các thư viện trung gian như TCP/IP, FAT, hoặc MQTT.

Với mã nguồn nhỏ gọn, tài liệu phong phú và cộng đồng phát triển mạnh mẽ, FreeRTOS là lựa chọn lý tưởng cho các ứng dụng nhúng như thiết bị IoT, hệ thống điều khiển công nghiệp, robot và các thiết bị gia dụng thông minh.

2.4 Giới thiệu về Blynk:

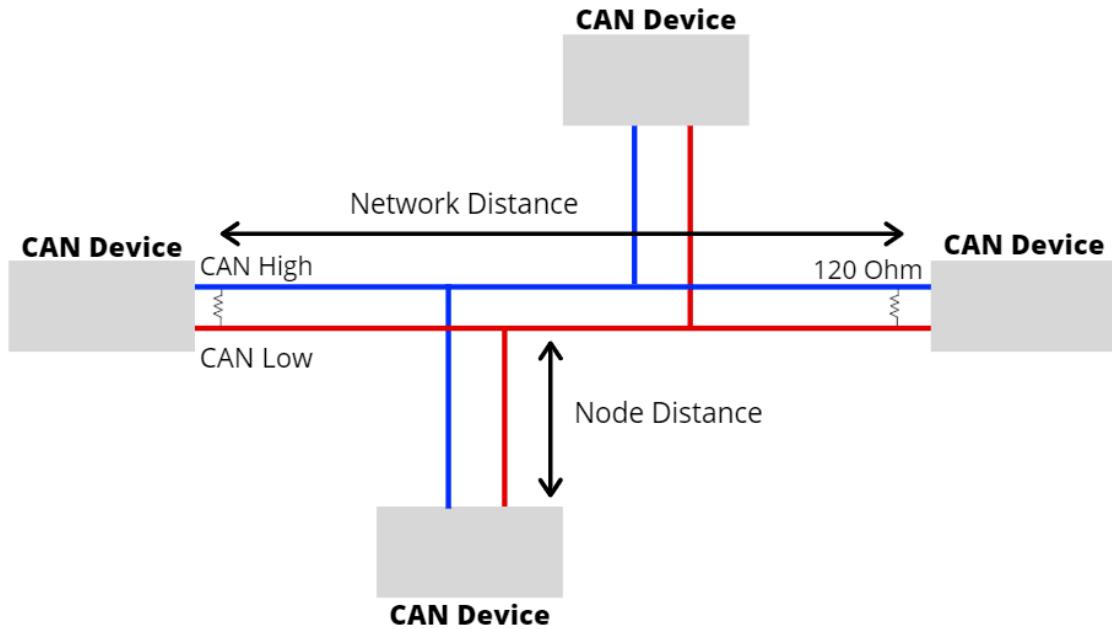


Hình 2.8: Giao diện quản lý của người dùng trên web Blynk

Blynk là một nền tảng phát triển ứng dụng IoT (Internet of Things) mạnh mẽ, cho phép người dùng dễ dàng điều khiển và giám sát các thiết bị phần cứng từ xa thông qua smartphone. Với Blynk, người dùng có thể tạo giao diện điều khiển trực quan trên điện thoại (Android/iOS) bằng cách kéo thả các widget như nút nhấn, công tắc, biểu đồ, màn hình LCD ảo, v.v., mà không cần viết code giao diện.

Blynk hỗ trợ nhiều dòng vi điều khiển và bo mạch phổ biến như ESP8266, ESP32, Arduino, Raspberry Pi,... đồng thời giao tiếp qua các giao thức như WiFi, Ethernet, hoặc mạng di động. Hệ thống hoạt động dựa trên ba thành phần chính: ứng dụng di động, máy chủ Blynk (có thể dùng server riêng hoặc cloud), và thư viện Blynk được cài vào thiết bị phần cứng.

2.5 Giới thiệu về giao tiếp CAN bus:



Hình 2.9: Minh họa sơ đồ kết nối giao tiếp CAN bus

CAN (Controller Area Network) là một chuẩn giao tiếp được phát triển bởi hãng Bosch, chuyên dùng để truyền thông giữa các thiết bị nhúng trong môi trường yêu cầu độ tin cậy cao như ô tô, công nghiệp và tự động hóa. CAN Bus cho phép nhiều thiết bị (node) kết nối trên cùng một đường truyền, trao đổi dữ liệu theo cơ chế phân quyền ưu tiên dựa trên định danh (ID), giúp giảm xung đột và tăng hiệu quả truyền tải.

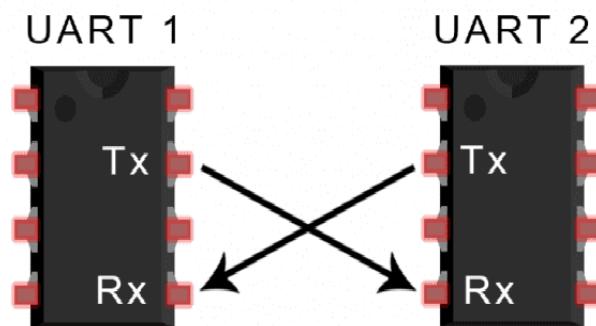
CAN Bus (Controller Area Network) hoạt động dựa trên nguyên lý truyền thông vi sai (differential signaling) giữa hai dây dẫn: **CAN High (CANH)** và **CAN Low (CANL)**. Khi truyền dữ liệu, mức điện áp trên hai dây này sẽ thay đổi ngược nhau — một dây tăng trong khi dây kia giảm — giúp hệ thống chống nhiễu hiệu quả và đảm bảo độ tin cậy cao trong môi trường công nghiệp hay ô tô.

Mạng CAN có cấu trúc dạng đường trực (bus topology), nơi tất cả các thiết bị (node) được kết nối song song vào hai dây chính. Ở hai đầu của đường truyền, cần gắn điện trở 120 Ohm để triệt tiêu tín hiệu phản xạ, giúp giữ ổn định tín hiệu và đảm bảo truyền

thông chính xác. Mỗi thiết bị có thể gửi và nhận dữ liệu mà không cần thiết bị chủ điều khiển trung tâm.

CAN sử dụng cơ chế định danh khung tin (identifier) để phân quyền truyền dữ liệu, trong đó khung có độ ưu tiên cao hơn sẽ được truyền trước mà không gây xung đột trên bus. Nhờ đặc điểm này, CAN Bus phù hợp với hệ thống yêu cầu giao tiếp nhanh, đáng tin cậy, và có nhiều thiết bị hoạt động đồng thời.

2.6 Giới thiệu về giao tiếp UART:



Hình 2.10: Sơ đồ giao tiếp UART

UART (Universal Asynchronous Receiver/Transmitter) là một chuẩn giao tiếp nối tiếp không đồng bộ, được sử dụng phổ biến để truyền và nhận dữ liệu giữa các vi điều khiển, module cảm biến, máy tính và thiết bị ngoại vi. Giao tiếp UART đơn giản, chi phí thấp, dễ triển khai nên rất thích hợp cho các ứng dụng nhúng và hệ thống nhúng thời gian thực.

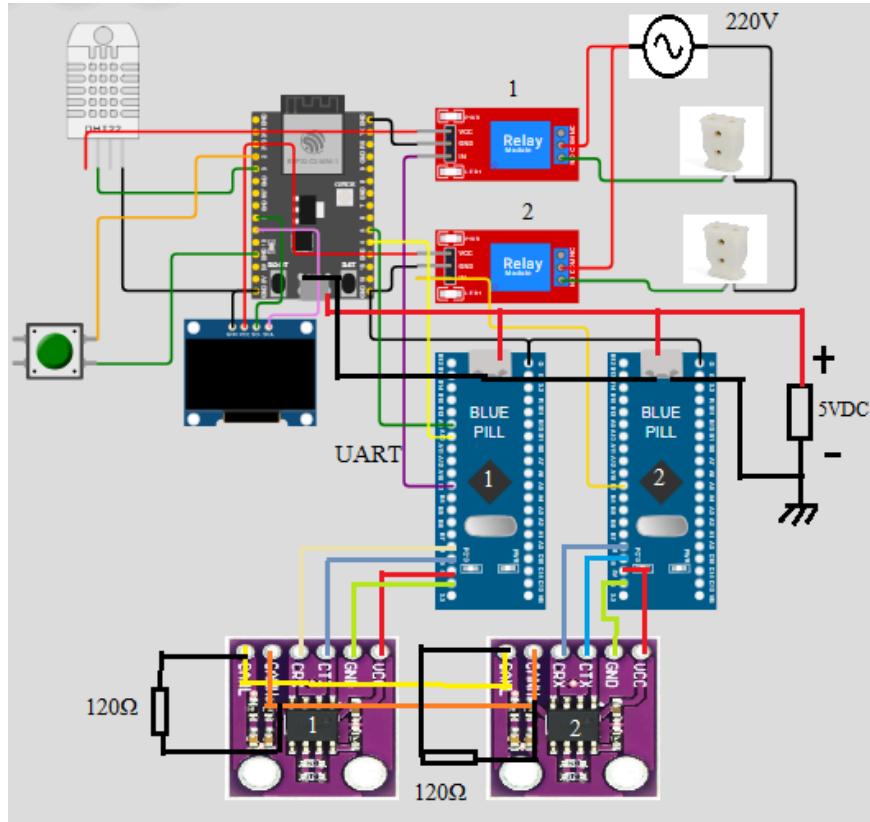
UART hoạt động dựa trên nguyên lý truyền dữ liệu tuần tự từng bit một qua hai đường dây chính: **TX (Transmit)** và **RX (Receive)**. Do là giao tiếp không đồng bộ, UART không dùng xung clock chung giữa các thiết bị. Thay vào đó, hai bên truyền và nhận cần thống nhất trước các tham số như: tốc độ baud rate (tốc độ truyền), số bit dữ liệu, bit chẵn lẻ (parity), và số bit stop (stop bit).

Dữ liệu sẽ được truyền theo từng frame gồm: 1 bit start (bắt đầu), từ 5–9 bit dữ liệu, 0 hoặc 1 bit kiểm tra chẵn lẻ, và 1 hoặc 2 bit stop. Bit start giúp thiết bị nhận đồng bộ

tín hiệu bắt đầu, còn bit stop đánh dấu kết thúc một khung dữ liệu. Quá trình truyền diễn ra theo từng byte, đơn giản nhưng hiệu quả trong những ứng dụng không yêu cầu tốc độ quá cao.

CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN

3.1 Sơ đồ kết nối thiết bị:



Hình 3.1: Sơ đồ kết nối thiết bị

3.2 Các linh kiện sử dụng trong dự án:

STT	Tên linh kiện	Số lượng	Chức năng
1	ESP32-C3	1	Bộ xử lý các chức năng chính của hệ thống, hỗ trợ WiFi cho hệ thống
2	DHT11	1	Đọc giá trị nhiệt độ, độ ẩm
3	Nút nhấn	1	Reset hệ thống, thiết lập lại kết nối WiFi
4	OLED 0.96 inch	1	Hiển thị giá trị nhiệt độ, độ ẩm, ngày giờ
5	Module relay 5V	2	Cách ly điện áp, điều khiển các ổ cắm

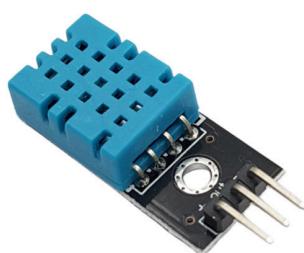
6	STM32F103 C6T6	2	Bộ xử lý các chức năng điều khiển phích cắm, mỗi STM32 sẽ điều khiển 1 phích cắm
7	Phích cắm âm	2	Cung cấp nguồn cho các thiết bị khác (đóng vai trò là ổ cắm)
8	MCP2551	2	Bộ thu phát (transceiver) CAN giúp chuyển đổi tín hiệu giữa vi điều khiển và bus CAN
9	Điện trở 120Ω	2	Được dùng làm terminating resistor trong mạng CAN. Nó giúp triệt tiêu phản xạ tín hiệu, đảm bảo truyền dữ liệu ổn định và đúng chuẩn.
10	Tải tiêu thụ (ví dụ bóng đèn 1W)	2	Kiểm tra chức năng hoạt động của thiết bị (ổ cắm)

Bảng 1: Tóm tắt các linh kiện sử dụng trong dự án

3.2.1 Vi điều khiển ESP32 và STM32:

Vi điều khiển ESP32-C3 và STM32 (trong đề tài này nhóm sử dụng loại STM32F103C6T6) đóng vai trò như “bộ não” của hệ thống. Có chức năng xử lý, truyền tải thông tin, hỗ trợ kết nối với máy chủ bên ngoài (Blynk) thông qua modun Wifi của ESP. Cấu tạo chi tiết và chức năng đã được trình bày cụ thể ở Chương 2: Cơ sở lý thuyết.

3.2.2 Cảm biến DHT11:



Hình 3.2: Cảm biến DHT11

DHT11 là một loại cảm biến kỹ thuật số dùng để đo nhiệt độ và độ ẩm môi trường, được sử dụng rộng rãi trong các hệ thống điều khiển, giám sát và tự động hóa. Cảm biến này có kích thước nhỏ gọn, giá thành thấp, dễ sử dụng và giao tiếp trực tiếp với các vi điều khiển như Arduino, ESP32 hay STM32 thông qua một chân dữ liệu duy nhất.

DHT11 hoạt động dựa trên nguyên lý của điện trở nhiệt (theristor) để đo nhiệt độ và một cảm biến điện dung để đo độ ẩm. Giá trị đo được sẽ được số hóa và truyền về vi điều khiển theo giao thức một dây (single-wire), trong đó dữ liệu được gửi dưới dạng các xung cao – thấp để thể hiện các bit nhị phân. Mỗi chu kỳ truyền gồm 5 byte: 2 byte độ ẩm, 2 byte nhiệt độ và 1 byte kiểm tra CRC để đảm bảo tính chính xác.

Thông số kỹ thuật của cảm biến:

- Điện áp hoạt động: 3,3-5VDC
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số $\pm 5\%$ RH
- Khoảng đo nhiệt độ: 0-50°C sai số $\pm 2^\circ\text{C}$
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
- Kích thước: 28mm x 12mm x10m

3.2.3 Màn hình OLED:



Hình 3.3: Màn hình OLED 0.96 inch (SSD1306)

Màn hình OLED 0.96 inch SSD1306 là một loại màn hình hiển thị đơn sắc, sử dụng công nghệ OLED (Organic Light Emitting Diode) với độ phân giải 128x64 pixel, rất phổ biến trong các ứng dụng nhúng và IoT nhờ vào kích thước nhỏ gọn, tiêu thụ điện năng thấp và khả năng hiển thị rõ nét kể cả trong điều kiện thiếu sáng.

Màn hình sử dụng chip điều khiển SSD1306 do hãng Solomon Systech phát triển, hỗ trợ giao tiếp qua I2C, giúp dễ dàng kết nối với các vi điều khiển như Arduino, STM32, ESP8266 hay ESP32. Dữ liệu hình ảnh được gửi đến màn hình dưới dạng ma trận pixel, cho phép hiển thị các ký tự văn bản, biểu tượng, đồ họa hoặc cảm biến dữ liệu thời gian thực.

Thông số kỹ thuật của màn hình:

- Điện áp sử dụng: 2.2~5.5VDC.
- Công suất tiêu thụ: 0.04w
- Góc hiển thị: lớn hơn 160 độ
- Số điểm hiển thị: 128×64 điểm.
- Độ rộng màn hình: 0.96 inch
- Màu hiển thị: Trắng
- Giao tiếp: I2C
- Driver: SSD1306

3.2.4 Relay:



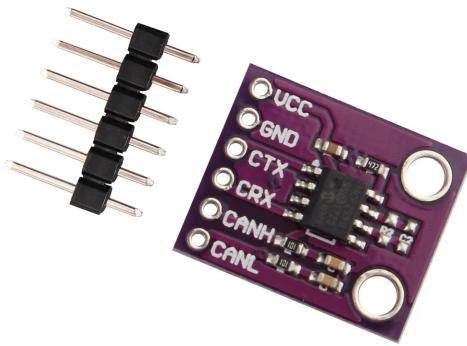
Hình 3.4: Module Relay 2 kênh 5V

Module relay là một linh kiện điện tử dùng để điều khiển các thiết bị điện công suất cao (như đèn, quạt, máy bơm...) thông qua tín hiệu điều khiển điện áp thấp từ vi điều khiển như Arduino, ESP32, hoặc STM32. Relay hoạt động như một công tắc điện tử, cho phép đóng/ngắt mạch điện xoay chiều (AC) hoặc một chiều (DC). Module thường tích hợp mạch cách ly quang (optocoupler) giúp tăng độ an toàn và bảo vệ mạch điều khiển khỏi nhiễu hoặc hư hại do dòng điện cao.

Thông số kỹ thuật của Relay 2 kênh:

- Sử dụng điện áp nuôi: 3,3-5VDC
- 2 Relay đóng ngắt ở điện thế kích bằng 0V nên có thể sử dụng cho cả tín hiệu 5V hay 3V3 (cần cấp nguồn ngoài), mỗi Relay tiêu thụ dòng khoảng 80mA.
- Điện thế đóng ngắt tối đa: AC250V – 10A hoặc DC30V – 10A.
- Có đèn báo đóng ngắt trên mỗi Relay.

3.2.5 Module MCP2551:



Hình 3.5: Module MCP2551

Module MCP2551 là một bộ thu phát (transceiver) CAN Bus được sử dụng để chuyển đổi tín hiệu logic từ vi điều khiển sang tín hiệu vi sai tiêu chuẩn của giao tiếp CAN. MCP2551 cho phép thiết bị nhúng như STM32 hoặc Arduino giao tiếp trong mạng CAN, đảm bảo tốc độ truyền cao (lên đến 1 Mbps) và khả năng chống nhiễu tốt. Nó chịu trách nhiệm truyền và nhận dữ liệu qua hai đường dây CANH và CANL, đồng thời bảo vệ chống ngắn mạch và nhiễu điện áp.

Để đảm bảo tín hiệu ổn định và giảm phản xạ trên đường truyền CAN, cần sử dụng điện trở 120Ω đặt ở hai đầu của đường bus. Điện trở này đóng vai trò trở kháng kết thúc (termination resistor), giúp hấp thụ năng lượng dư trong tín hiệu và duy trì độ chính xác của giao tiếp, đặc biệt trong các mạng CAN có khoảng cách dây dài.

Thông số kỹ thuật:

- Tốc độ dữ liệu: 1 Mb/s
- Điện áp hoạt động: 5V
- Nhiệt độ hoạt động tối đa: 85 độ C
- Nhiệt độ hoạt động tối thiểu: -40 độ C
- Số lượng chân (pin): 6 - gồm có VCC, GND, CTX, CRX, CANH và CANL

- Màu sắc: tím.
- Trọng lượng: 1,4g.

3.2.6 Các thiết bị hỗ trợ khác:



Hình 3.6: Nút nhấn

Nút nhấn trong dự án có vai trò đặt thiết bị về trạng thái ban đầu (Reset), được sử dụng khi muốn kết nối thiết bị với mạng Wifi khác so với lần kết nối trước đó.



Hình 3.7: Phích cắm âm

Phích cắm âm có vai trò là nguồn cung cấp điện cho các vật dụng khác mà người dùng muốn kết nối vào, chức năng tương tự như các ổ cắm điện bình thường khác.

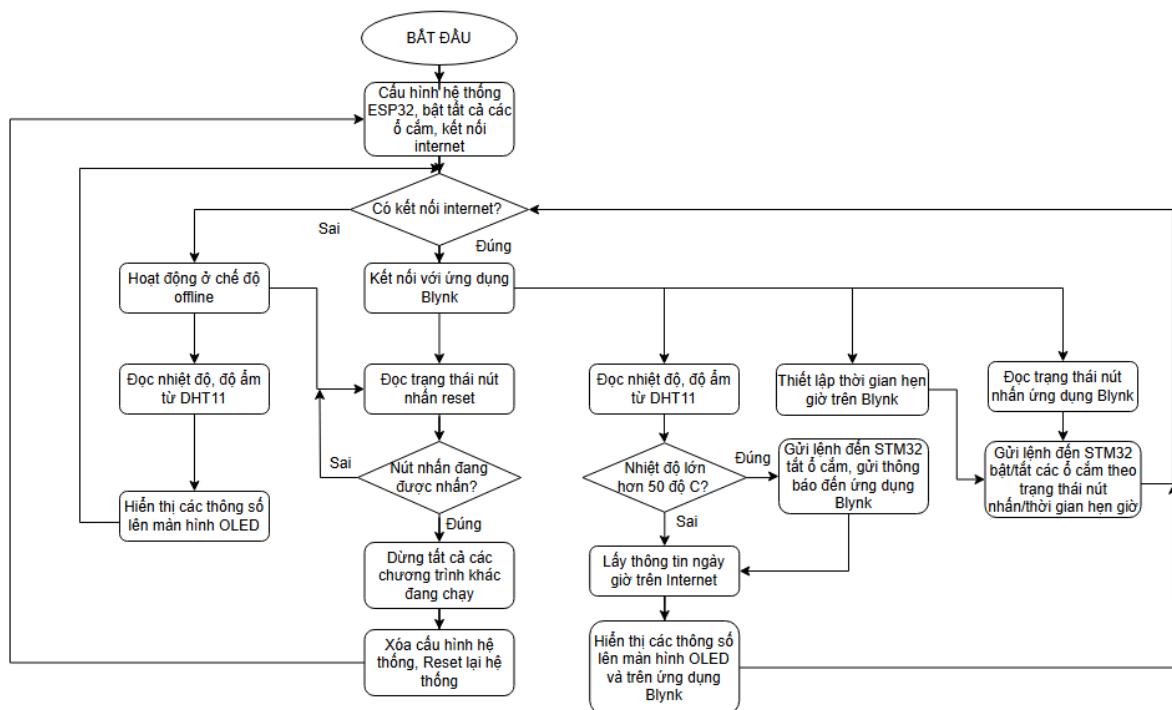


Hình 3.8: Bóng đèn 1W

Để kiểm tra ổ cảm điện thông minh có hoạt động tốt hay không, nhóm sử dụng tải tiêu thụ là 2 bóng đèn 1W. Mặc dù có thể chịu tải dòng điện lên đến 10A, để đảm bảo an toàn và quá trình kiểm tra trực quan dễ nhìn hơn, nhóm chỉ sử dụng tải có công suất rất nhỏ.

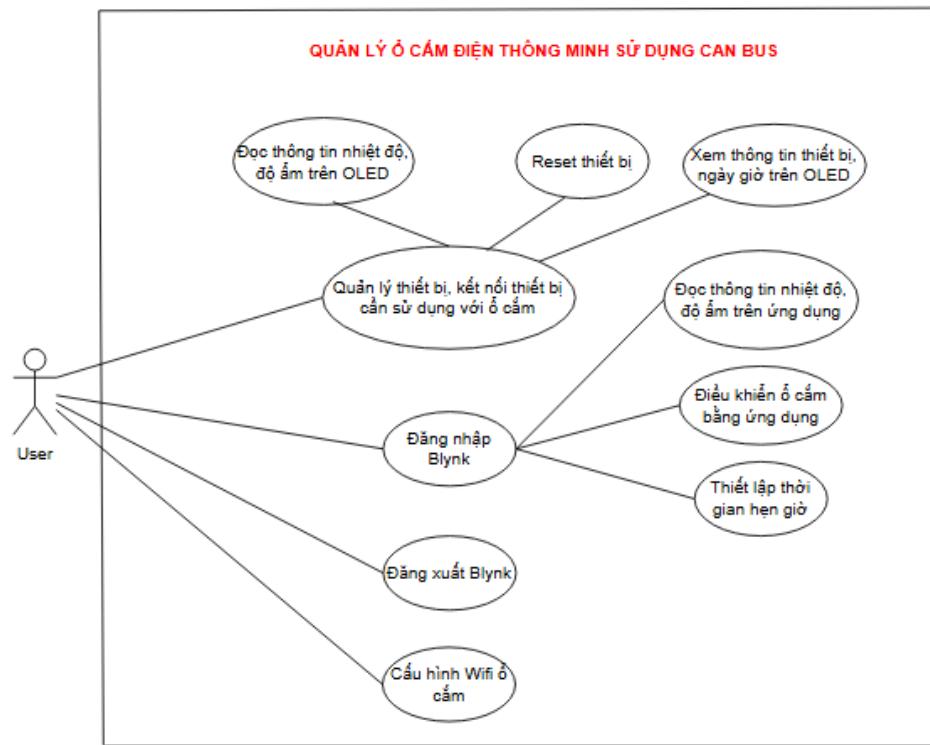
3.3 Sơ đồ mô tả hệ thống:

3.3.1 Lưu đồ thuật toán:



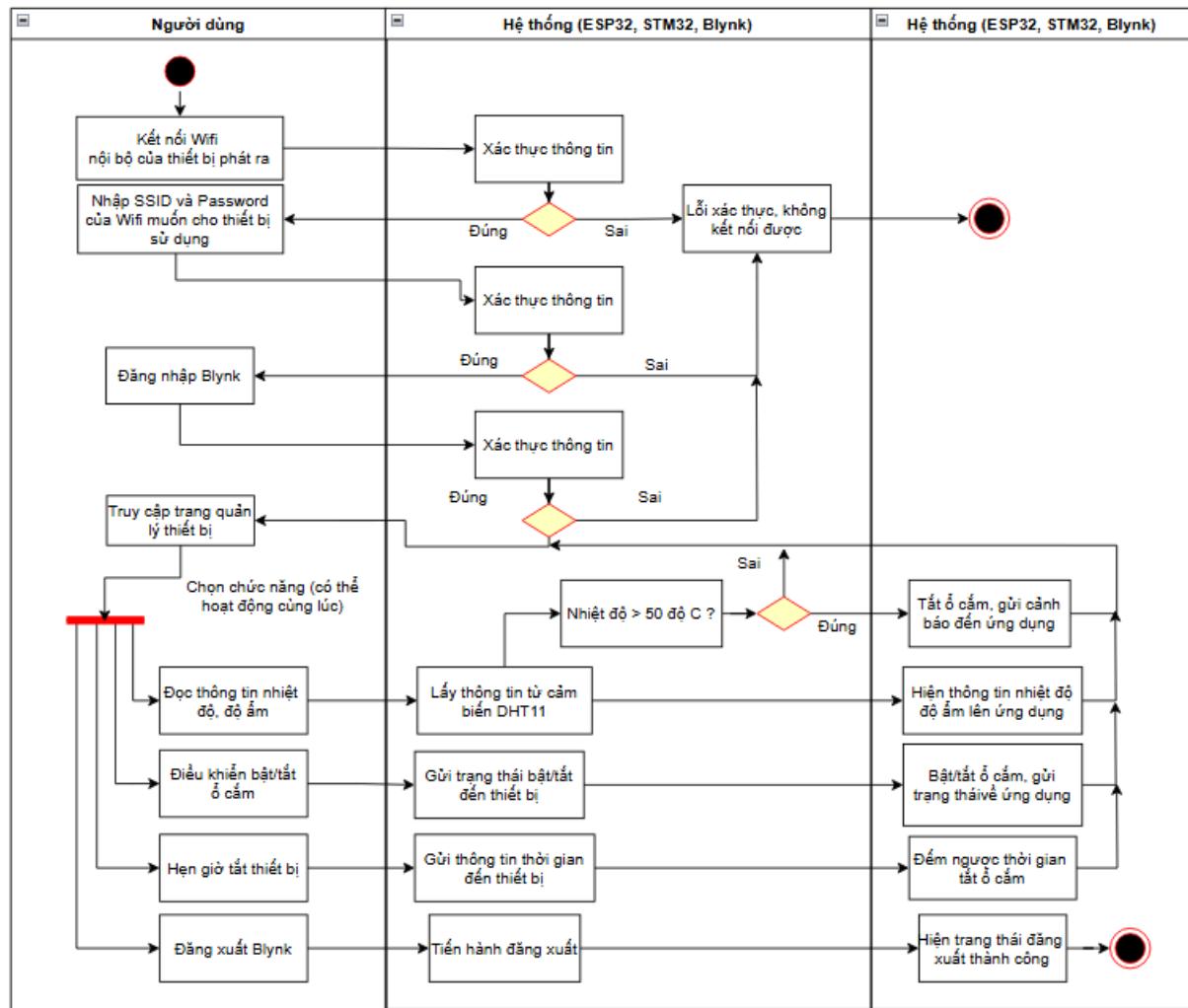
Hình 3.9: Lưu đồ thuật toán của dự án

3.3.2 Sơ đồ Use Case:



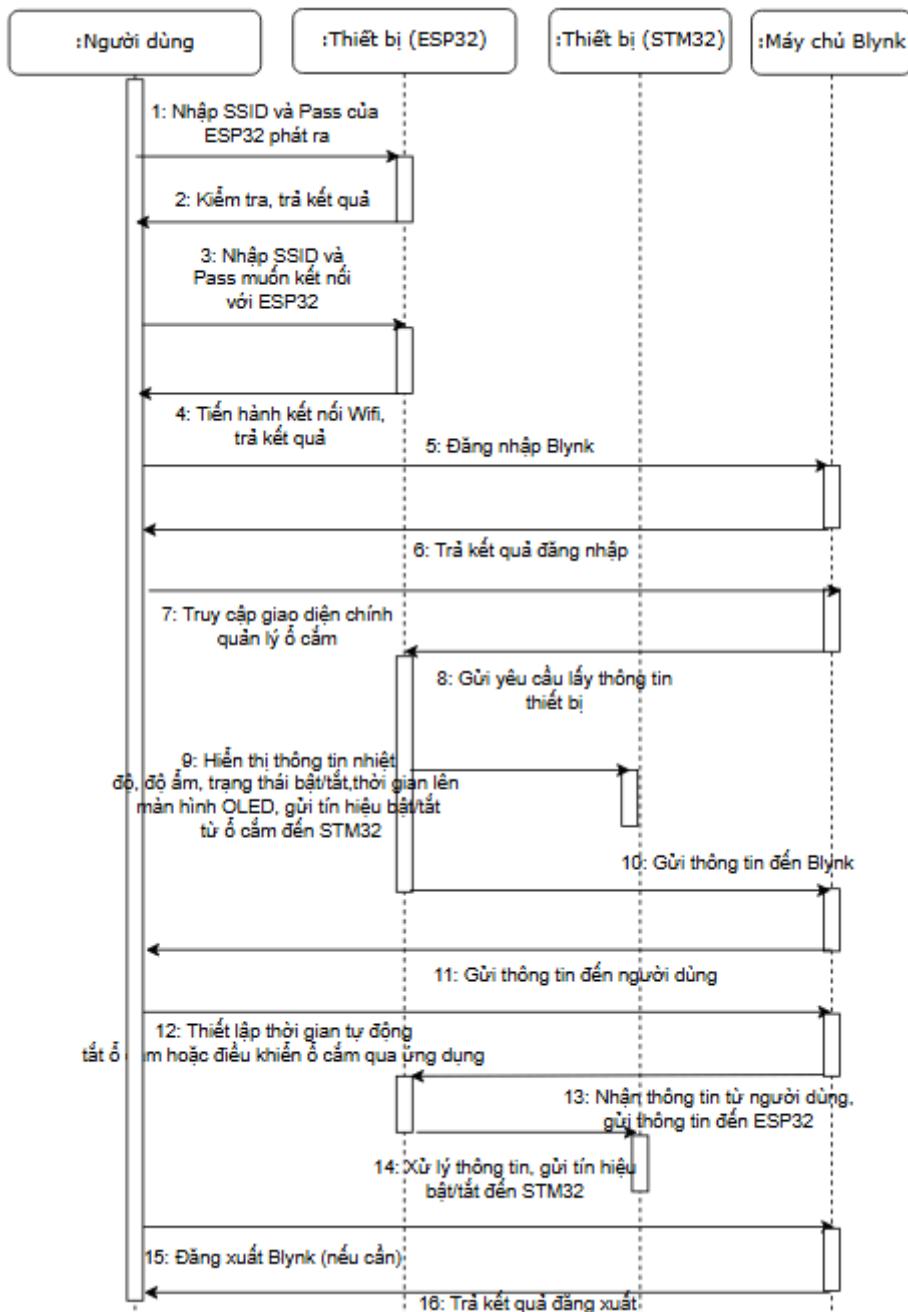
Hình 3.10: Sơ đồ Use Case của dự án

3.3.3 Sơ đồ Activity:



Hình 3.11: Sơ đồ Activity của dự án

3.3.4 Sơ đồ Sequence:



Hình 3.12: Sơ đồ Sequence của dự án

3.4 Nguyên lý làm việc của thiết bị:

Đầu tiên, nguồn điện xoay chiều 220V sẽ cung cấp điện áp cho tải (thiết bị cảm vào phích cảm âm). Nguồn điện 5VDC sẽ đóng vai trò cấp nguồn cho các thiết bị còn lại

(1 ESP32 và 2 STM32) thông qua cổng Micro-USB, nguồn này có thể lấy từ laptop hoặc sạc dự phòng hoặc qua biến áp. Lưu ý là cả 3 vi điều khiển khi cấp nguồn cần nối chung GND.

ESP32-C3 được kết nối với Blynk nhờ có hỗ trợ module Wifi (ban đầu người dùng sẽ tự thiết lập cho ESP32 bằng cách kết nối mạng Wifi nội bộ do ESP phát ra, chọn SSID Wifi và mật khẩu để ESP kết nối). Nếu mạng Wifi thay đổi, người dùng có thể thiết lập lại bằng cách nhấn nút RESET trên ố cắm. Sau khi thiết lập lại xong, chỉ cần rút nguồn thiết bị và cắm lại là tiếp tục sử dụng được.

Người dùng cần đăng nhập tài khoản Blynk đã được cấu hình liên kết sẵn với thiết bị. Sau khi đăng nhập thành công, trên ứng dụng Blynk, người dùng có thể xem các thông tin như nhiệt độ, độ ẩm, đồng thời có thể điều khiển bật/tắt các ố cắm từ xa (nếu ở chế độ online) theo cách thủ công hoặc hẹn giờ tắt các ố cắm.

ESP32-C3 sẽ đọc thông tin nhiệt độ, độ ẩm từ DHT11; đọc thông tin bật/tắt thủ công, thông tin hẹn giờ trên Blynk và thông tin ngày giờ lấy trên Internet. Sau đó ESP32 sẽ hiển thị thông tin nhiệt độ/độ ẩm, ngày giờ, trạng thái bật/tắt ố cắm lên màn hình OLED và Blynk. Đồng thời sẽ gửi trạng thái bật/tắt ố cắm qua STM32 thứ 1 qua giao thức UART

STM32 thứ 1 nhận dữ liệu từ ESP32 qua UART, rồi điều khiển bật/tắt Relay thứ 1 dựa vào dữ liệu nhận được. STM32 thứ 1 sẽ tiếp tục gửi dữ liệu trạng thái bật/tắt ố cắm còn lại qua giao thức CAN. Sau đó STM32 thứ 2 nhận dữ liệu từ STM32 thứ 1, rồi điều khiển bật/tắt Relay thứ 2 dựa vào dữ liệu nhận được

Ngoài ra, ố cắm điện thông minh được tích hợp thêm chức năng phát hiện nhiệt độ cao: khi cảm biến DHT11 của thiết bị đọc nhiệt độ lớn hơn 50 độ thì sẽ gửi thông tin đến ESP32, ESP tiếp tục gửi tín hiệu đến STM32 để tiến hành đặt tín hiệu mức 0 của Relay, điều này sẽ tắt ố cắm, giúp thiết bị an toàn không bị quá nhiệt. Đồng thời ESP32 gửi cảnh báo đến người dùng qua ứng dụng Blynk để dễ dàng phát hiện, xử lý kịp thời.

3.5 Phần mềm sử dụng:



Hình 3.13: Logo của ArduinoIDE và STM32CubeIDE

Arduino IDE là môi trường phát triển mã nguồn mở dành cho lập trình vi điều khiển Arduino và các bo mạch tương thích (ESP32-C3). Nó hỗ trợ viết code, biên dịch và nạp chương trình dễ dàng với giao diện đơn giản.

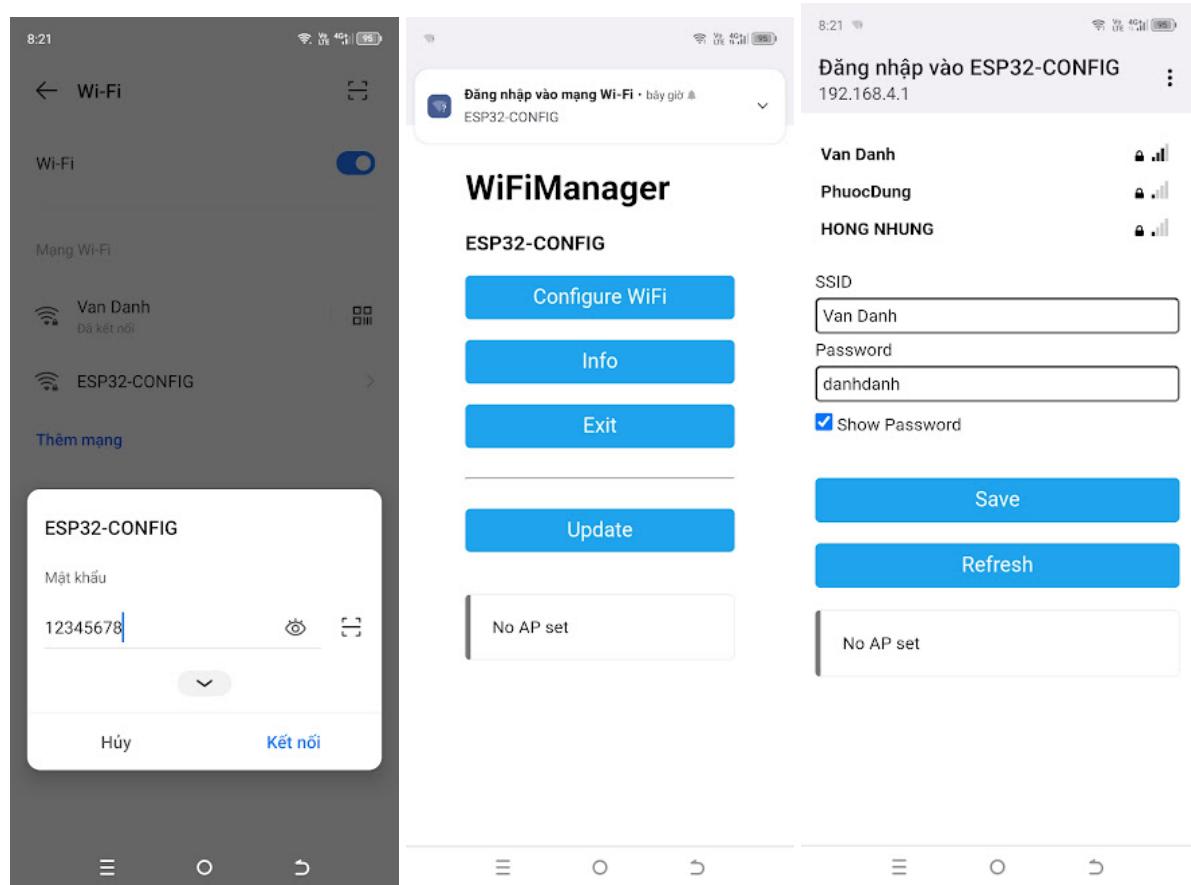
STM32CubeIDE là môi trường phát triển tích hợp (IDE) của STMicroelectronics, hỗ trợ lập trình, debug và cấu hình code tự động cho vi điều khiển STM32. Nó tích hợp CubeMX giúp thiết lập ngoại vi trực quan, phù hợp cho các dự án nhúng phức tạp.

CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ, HƯỚNG PHÁT TRIỂN

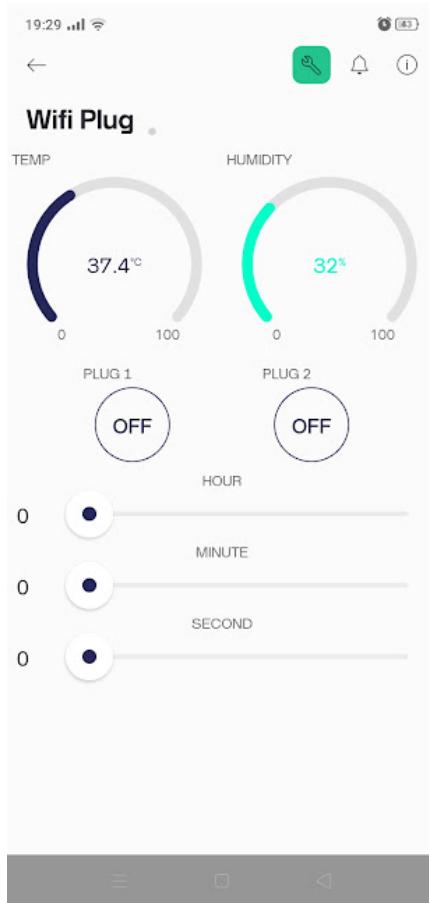
4.1 Kết quả đạt được:



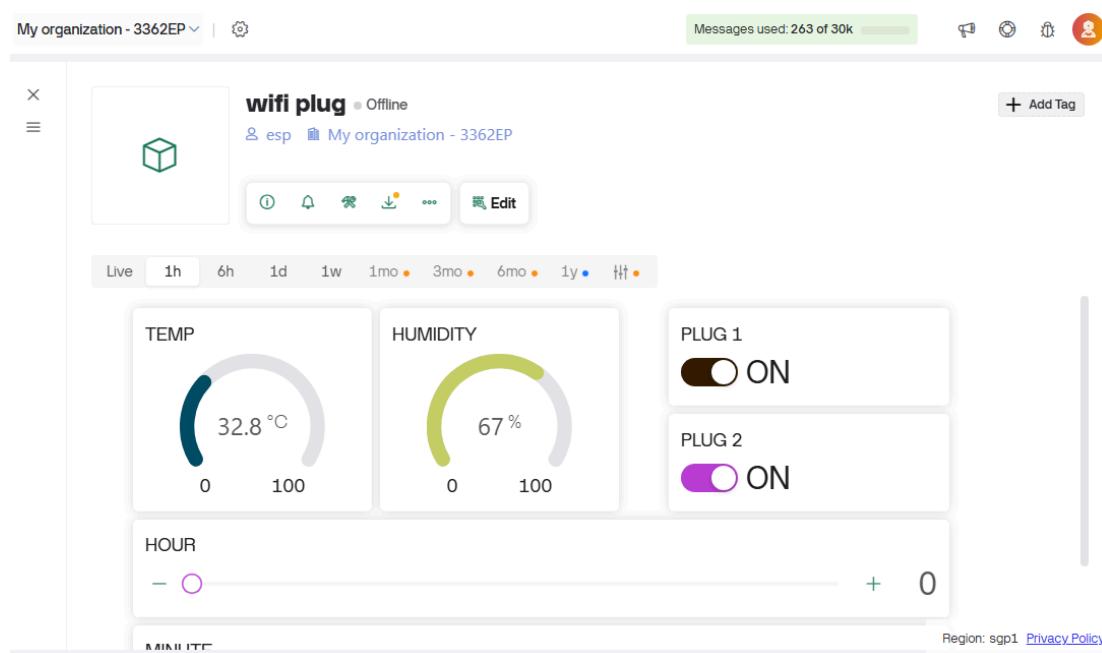
Hình 4.1: Sản phẩm thực tế của dự án



Hình 4.2: Giao diện cấu hình WiFi thiết bị



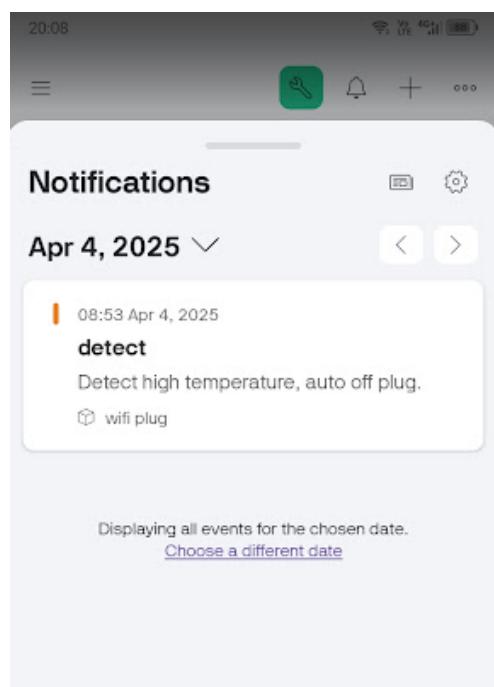
Hình 4.3: Giao diện quản lý ổ cắm trên điện thoại



Hình 4.4: Giao diện quản lý ổ cắm trên máy tính

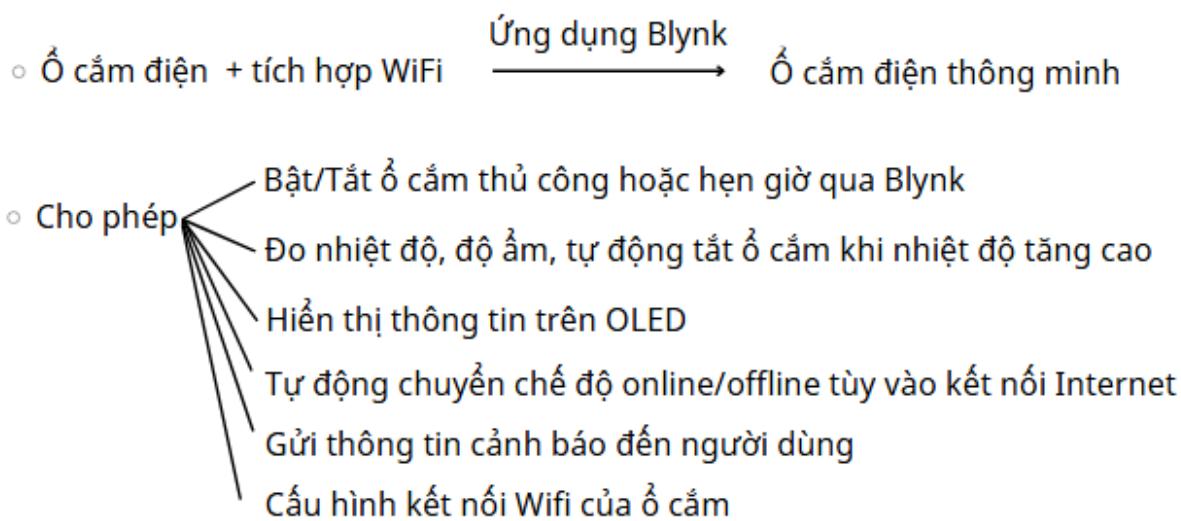


Hình 4.5: Giao diện màn hình OLED trên ổ cắm



Hình 4.6: Thông báo phát hiện nhiệt cao trên Blynk

Tóm tắt kết quả đạt được: Thiết bị hoạt động ổn định, phản hồi nhanh, hoạt động đúng chức năng đã giới thiệu như hình dưới. (Chi tiết ở video demo ở phụ lục code)



Hình 4.6: Tóm tắt các chức năng của ô cắm điện thông minh

4.2 Hướng phát triển của đề tài:

Đề tài: ô cắm điện thông minh sử dụng CAN BUS có nhiều tiềm năng phát triển trong tương lai. Các tính năng có thể phát triển thêm kể đến như: chức năng điều khiển bật/tắt ô cắm bằng giọng nói, chức năng hiển thị dòng điện, điện áp của các thiết bị kết nối với ô cắm điện thông minh, chức năng hẹn giờ bật thiết bị. Thông qua học phần Hệ thống thời gian thực, nhóm tích lũy được nhiều kiến thức quý báu và mong muốn đề tài này sẽ được ứng dụng trong thực tiễn nhiều hơn nữa.

TÀI LIỆU THAM KHẢO

- [1]“ESP32-C3-DevKitM-1 - ESP32-C3 - — ESP-IDF Programming Guide v5.2 documentation,” *Espressif.com*, 2016. <https://docs.espressif.com/projects/esp-idf/en/v5.2/esp32c3/hw-reference/esp32c3/user-guide-devkitm-1.html> (accessed Jun. 06, 2025).
- [2]O. www.ovaga.com, “STM32F103C6T6 Datasheet, Pinout, and Specifications - Ovaga Technologies,” *Ovaga.com*, 2024. <https://www.ovaga.com/blog/motor-control/stm32f103c6t6-datasheet-pinout-and-specifications> (accessed Jun. 06, 2025).
- [3]“STM32F103C6 - STMicroelectronics,” *STMicroelectronics*, 2025. https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f1-series/stm32f103/stm32f103c6.html (accessed Jun. 06, 2025).
- [4]“Introduction to CAN BUS and How to use it with Arduino,” *Latest open tech from seeed studio*, Nov. 27, 2019. <https://www.seeedstudio.com/blog/2019/11/27/introduction-to-can-bus-and-how-to-use-it-with-arduino/>
- [5]“Giao tiếp UART là gì,” *Dientutuonglai.com*, 2022. <https://dientutuonglai.com/giao-tiep-uart-la-gi.html> (accessed Jun. 06, 2025).
- [6]Module Cảm Biến Độ Âm, Nhiệt Độ DHT11 (UIJ3)#Shorts, “Module Cảm Biến Độ Âm, Nhiệt Độ DHT11,” *Nshopvn.com*, 2019. <https://nshopvn.com/product/module-cam-bien-do-am-nhiet-do-dht11/>
- [7]Inch, “Màn Hình LCD Oled 0.96 Inch Giao Tiếp I2C (White),” *Nshopvn.com*, 2019. <https://nshopvn.com/product/man-hinh-lcd-oled-0-96-inch-giao-tiep-i2c-white/> (accessed Jun. 07, 2025).

[8]C. Ly, “Module 2 Relay Vói Opto Cách Ly (5VDC),” *Nshopvn.com*, 2019. <https://nshopvn.com/product/module-2-relay-voi-opto-cach-ly-5vdc/> (accessed Jun. 07, 2025).

[9]M. High, “HALJIA CJMCU-2551 MCP2551 High Speed CAN Communication Protocol Control,” *HALJIA*, 2025. <https://www.haljia.com/products/haljia-cjmcu-2551-mcp2551-high-speed-can-communication-protocol-controller-bus-interface-module> (accessed Jun. 07, 2025).

PHỤ LỤC CODE

[1] Source code của dự án:
[an2101/SMART-POWER-OUTLET-BASED-ON-CAN-BUS](#)



[2] Video demo dự án: [Demo_project.mp4 - Google Drive](#)

