

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ
KHOA CÔNG NGHỆ THÔNG TIN



BÀI THI TIỂU LUẬN
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI

**GIÁM SÁT NĂNG LƯỢNG TIÊU THỤ
VỚI ESP32**

Thừa Thiên Huế, 2025

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC HUẾ
KHOA CÔNG NGHỆ THÔNG TIN

BÀI THI TIỂU LUẬN
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI

**GIÁM SÁT NĂNG LƯỢNG TIÊU THỤ
VỚI ESP32**

TÊN HỌC PHẦN – MÃ HỌC PHẦN
PHÁT TRIỂN ỨNG DỤNG IOT – 2024-2025.2.TIN4024.006

Giáo viên hướng dẫn: ThS. Võ Việt Dũng

Khóa : K45 – HỆ CHÍNH QUY

Thừa Thiên Huế, 2025

A. DANH MỤC CÁC TỪ VIẾT TẮT

- IoT (Internet of Things) : **Internet Vạn Vật**, hay cụ thể hơn là Mạng lưới vạn vật kết nối.
- ESP32: Vi điều khiển tích hợp Wifi và Bluetooth
- ACS712 -20A : là một **cảm biến dòng điện** (current sensor) được sử dụng phổ biến trong các dự án điện tử để **đo dòng điện một chiều (DC) hoặc xoay chiều (AC)**.
- MQTT (Message Queuing Telemetry Transport) : là một giao thức truyền tin **nhẹ (lightweight)**, **mở (open)** và **dễ triển khai**

B. MỤC LỤC

GIÁM SÁT NĂNG LƯỢNG TIÊU THỤ VỚI ESP32.....	2
A. DANH MỤC CÁC TỪ VIẾT TẮT	3
B. MỤC LỤC	3
C. MỤC LỤC HÌNH ẢNH.....	5
D. TÓM TẮT.....	7
E. NỘI DUNG	9
Chương 1. TỔNG QUAN	9
1.1 ĐẶT VẤN ĐỀ.....	9
1.2 MỤC TIÊU	9
1.3 NỘI DUNG NGHIÊN CỨU.....	10
1.4 GIỚI HẠN	10
Chương 2. CƠ SỞ LÝ THUYẾT	11
2.1 GIỚI THIỆU PHẦN CỨNG	11
2.1.1 Module cảm biến dòng điện ACS712 -20A.....	11
2.1.1.1 Giới thiệu module cảm biến dòng điện ACS712-20A	11
2.1.1.2 Các thông số kỹ thuật.....	11
2.1.2 Module vi điều khiển ESP32.....	13

2.1.2.1 Giới thiệu module vi điều khiển ESP32.....	13
2.1.2.2 Thông số kỹ thuật.....	14
2.1.2.3 Các ứng dụng của vi điều khiển ESP32	15
2.1.3 Thẻ SD – lưu trữ dữ liệu cục bộ	18
2.1.3.1 Giới thiệu thẻ SD	18
2.1.3.2 Các thông số kỹ thuật của thẻ SD	19
2.1.3.3 Ứng dụng thực tế của thẻ SD trong giám sát năng lượng	20
2.1.4 Kết nối Wi-Fi/MQTT - Gửi dữ liệu tới đám mây.....	20
2.1.4.1 Giới thiệu.....	20
2.1.4.2 Các thông số kỹ thuật của giao thức MQTT trên ESP32.....	21
2.1.4.3 Ứng dụng thực tế của MQTT trong giám sát năng lượng	22
2.1.5 Nguồn điện cho hệ thống.....	22
2.1.5.1 Giới thiệu.....	22
2.1.5.2 Thông số kỹ thuật nguồn cho các thành phần chính	24
2.1.5.3 Ứng dụng thực tế trong giám sát năng lượng.....	24
2.2 GIỚI THIỆU PHẦN MỀM.....	24
2.2.1 Thư viện cho ACS712.....	24
2.2.1.1 Giới thiệu.....	24
2.2.1.2 Các thông số kỹ thuật của thư viện ACS712:.....	26
2.2.1.3 Ứng dụng:	27
2.2.2 MQTT/HTTP	27
2.2.2.1 Giới thiệu MQTT/HTTP	27
2.2.2.2 Các thông số kỹ thuật của giao thức MQTT trên ESP32.....	29
2.2.2.3 Ứng dụng thực tế của MQTT trong giám sát năng lượng	30
2.2.3 BLYNK	30
2.2.3.1 Giới thiệu về ứng dụng Blynk – Giao diện giám sát và điều khiển từ xa	30
2.2.3.2 Ứng dụng thực tế về hệ thống giám sát và điều khiển từ xa	33
2.2.4 WOKWI	33
2.2.4.1 Giới thiệu Wokwi – Mô phỏng phần cứng trực tuyến.....	33
2.2.4.2 Ứng dụng thực tế của Wokwi trong dự án giám sát năng lượng	35

2.2.5	TELEGRAM	35
2.2.5.1	Giới thiệu Telegram - Cảnh báo và điều khiển thiết bị từ xa	35
2.2.5.2	Ứng dụng thực tế của Telegram	37
Chương 3.	MÔ PHỎNG	38
3.1	SƠ ĐỒ KHÔI LƯU THÔNG TIN TRÊN ĐÁM MÂY	38
3.1.1	SƠ ĐỒ KHÔI:	38
3.1.1.1	MÔ TẢ HOẠT ĐỘNG CỦA THIẾT BỊ:.....	39
3.1.1.2	SƠ ĐỒ KẾT NỐI:	39
3.1.2	SƠ ĐỒ TRÊN BLYNK:	41
3.1.2.1	MINH HOẠ TRÊN BLYNK:.....	41
3.1.2.2	TẠO DATASTREAMS TRÊN BLYNK:.....	41
3.1.3	NGUYÊN LÝ HOẠT ĐỘNG:	42
3.1.3.1	HOẠT ĐỘNG VỚI BLYNK:	42
3.1.3.2	HOẠT ĐỘNG VỚI TELEGRAM:.....	47
3.2	SƠ ĐỒ KHÔI LƯU THÔNG TIN TRÊN SD.....	48
3.2.1	SƠ ĐỒ KHÔI:.....	48
3.2.1.1	MÔ TẢ HOẠT ĐỘNG CỦA THIẾT BỊ:.....	49
3.2.1.2	SƠ ĐỒ KẾT NỐI:	50
3.2.2	NGUYÊN LÝ HOẠT ĐỘNG:	52
F.	KẾT LUẬN/NHẬN XÉT/ĐÁNH GIÁ/KIẾN NGHỊ:.....	53
Chương 1.	KẾT LUẬN:.....	53
Chương 2.	NHẬN XÉT:.....	54
Chương 3.	ĐÁNH GIÁ:	55
Chương 4.	KIẾN NGHỊ:.....	55
Chương 5.	TỔNG KẾT:	57
Tài liệu tham khảo	58

C. MỤC LỤC HÌNH ẢNH

Hình 1.1 Hình ảnh minh họa thông báo cảm biến và lưu lên đám mây

Hình 2.1 Hình ảnh của module và AC cảm biến dòng ASC 712

Hình 2.2 Hình ảnh minh họa kết nối với module ASC712

Hình 2.3 Hình ảnh module vi điều khiển ESP32

Hình 2.4 Hình ảnh minh họa sơ đồ chân của bo 30 chân ESP32

Hình 2.5 Cấu trúc của module ESP32

Hình 2.6 Hình ngôi nhà thông minh

Hình 2.7 Hình IOT trong công nghiệp

Hình 2.8 Hình Thiết bị đeo thông minh

Hình 2.9 Hình Robot

Hình 2.10 Hình Hệ thống đo lường và giám sát năng lượng

Hình 2.11 Hình ảnh thẻ SD

Hình 2.12 Hình ảnh minh họa MQTT

Hình 2.13 Hình ảnh minh họa sơ đồ cấp nguồn cho hệ thống ESP32

Hình 2.14 Hình ảnh minh họa MQTT

Hình 2.15 Hình ảnh minh họa Blynk trên mobile

Hình 2.16 Hình ảnh minh họa Blynk trên Web

Hình 2.17 Hình ảnh minh họa ESP32 trên wokwi

Hình 2.18 Hình ảnh minh họa Telegram

Hình 3.1: Hình ảnh minh họa sơ đồ khối trên wokwi

Hình 3.2: Hình ảnh minh họa trên Blynk Web

Hình 3.3: Hình ảnh tạo Datastreams trên Blynk Web

Hình 3.4: Hình ảnh minh họa Sơ đồ khối (Mới khởi động)

Hình 3.5: Hình ảnh Sơ đồ khối và Blynk khi chưa đổi giá trị

Hình 3.6: Hình ảnh Sơ đồ khối và Blynk sau khi đổi giá trị

Hình 3.7: Hình ảnh Sơ đồ khối và Blynk sau khi đã tắt thiết bị

Hình 3.8: Hình ảnh Sơ đồ khối khi chưa thay đổi giá trị

Hình 3.9: Hình ảnh Sơ đồ khối và telegram khi thay đổi giá trị

Hình 3.10: Hình ảnh minh họa sơ đồ khối có SD trên wokwi

Hình 3.11: Hình ảnh minh họa hoạt động của SD trên wokwi

D. TÓM TẮT

- Trong thời đại công nghệ phát triển mạnh mẽ, Internet of Things (IoT) đã trở thành một xu hướng quan trọng, mang lại nhiều tiện ích cho đời sống và sản xuất. Một trong những ứng dụng thiết thực của IoT là giám sát và quản lý năng lượng tiêu thụ, giúp người dùng theo dõi mức sử dụng điện năng của các thiết bị trong gia đình hoặc doanh nghiệp một cách chính xác và tiện lợi. Điều này không chỉ giúp tối ưu hóa việc sử dụng điện mà còn góp phần tiết kiệm chi phí và nâng cao ý thức sử dụng năng lượng hiệu quả.
- Hệ thống giám sát năng lượng tiêu thụ với ESP32 được xây dựng nhằm mục đích theo dõi lượng điện tiêu thụ theo thời gian thực, cung cấp thông tin chi tiết về công suất tiêu thụ của từng thiết bị. Hệ thống sử dụng cảm biến dòng điện (như ACS712) để đo dòng điện chạy qua thiết bị điện, từ đó tính toán công suất tiêu thụ và ghi nhận dữ liệu một cách liên tục. Toàn bộ dữ liệu này có thể được lưu trữ trên thẻ nhớ SD để phân tích sau này hoặc gửi lên nền tảng đám mây để người dùng theo dõi từ xa thông qua Internet.
- Với hệ thống này, người dùng có thể kiểm tra lịch sử sử dụng điện, phân tích xu hướng tiêu thụ theo từng khoảng thời gian nhất định và nhận thông báo khi có sự bất thường về mức tiêu thụ điện. Nhờ đó, người dùng có thể dễ dàng phát hiện các thiết bị tiêu tốn điện năng cao hoặc phát sinh sự cố rò rỉ điện, từ đó có biện pháp điều chỉnh và sử dụng điện hợp lý hơn. Ngoài ra, hệ thống còn có thể tích hợp với các ứng dụng di động hoặc giao diện web, giúp việc giám sát và điều khiển thiết bị điện trở nên đơn giản và thuận tiện hơn bao giờ hết.
- Trên thực tế, các thiết bị giám sát điện năng hiện nay thường có giá thành khá cao và khó tiếp cận với người dùng phổ thông. Chính vì vậy, nhóm đã nghiên cứu và triển khai đề tài **“Giám sát năng lượng tiêu**

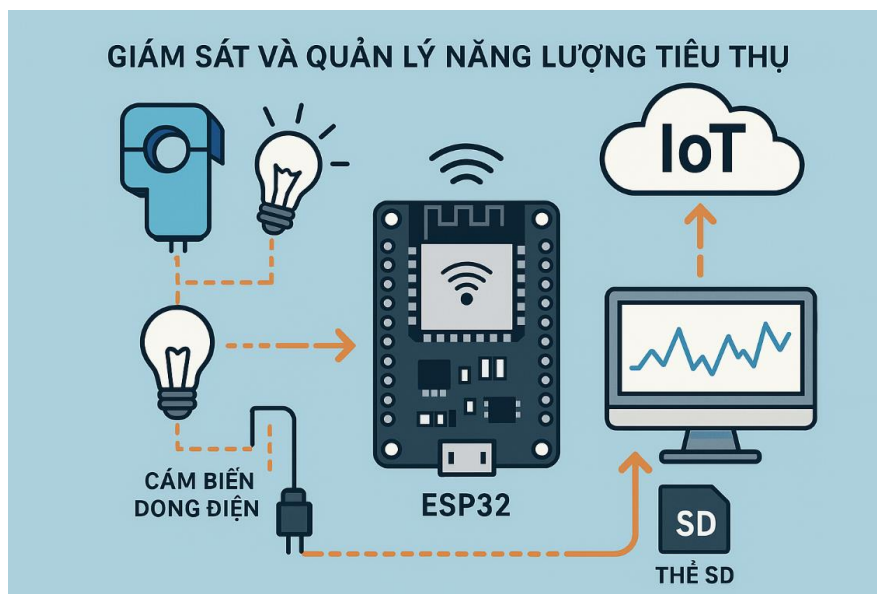
thụ với ESP32”, với mục tiêu tạo ra một giải pháp giám sát điện năng đơn giản, hiệu quả và chi phí thấp. Hệ thống không chỉ giúp người dùng cá nhân quản lý điện năng tiêu thụ mà còn có thể ứng dụng trong các mô hình lớn hơn như văn phòng, nhà máy hoặc các cơ sở sản xuất. Thông qua đề tài này, Tôi hy vọng sẽ mang đến một giải pháp hữu ích giúp người dùng nâng cao nhận thức về tiết kiệm năng lượng, đồng thời góp phần bảo vệ môi trường và hướng đến một hệ thống tiêu thụ điện thông minh hơn trong tương lai

E. NỘI DUNG

Chương 1. TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Giám sát và quản lý năng lượng tiêu thụ đang trở thành một yếu tố quan trọng trong các hệ thống điện, đặc biệt trong các ứng dụng IoT. Việc sử dụng cảm biến dòng điện, như ACS712, kết hợp với vi điều khiển ESP32 giúp đo lường công suất tiêu thụ và cung cấp thông tin chi tiết về tình trạng năng lượng sử dụng trong các hệ thống. Dữ liệu thu thập từ các cảm biến này có thể được lưu trữ cục bộ trên thẻ SD hoặc truyền tải lên đám mây để giám sát và phân tích từ xa.



Hình 1.1 Hình ảnh minh họa thông báo cảm biến và lưu lên đám mây

1.2 MỤC TIÊU

Mục tiêu của nghiên cứu này là thiết kế và triển khai một hệ thống giám sát năng lượng tiêu thụ dựa trên ESP32, sử dụng cảm biến ACS712 để đo dòng điện và công suất tiêu thụ. Hệ thống sẽ lưu trữ dữ liệu đo lường trên thẻ SD hoặc đám mây, giúp người dùng giám sát và quản lý năng lượng một cách hiệu quả. Cụ thể, các mục tiêu bao gồm:

- Đo và ghi nhận dòng điện, công suất tiêu thụ.
- Lưu trữ dữ liệu trên thẻ SD hoặc đám mây để truy xuất dễ dàng.

- Phát triển hệ thống giám sát từ xa qua kết nối Wi-Fi hoặc MQTT.
- Cung cấp giao diện người dùng để sử dụng để theo dõi dữ liệu năng lượng.

1.3 NỘI DUNG NGHIÊN CỨU

Nội dung nghiên cứu tập trung vào việc phát triển một hệ thống giám sát năng lượng tiêu thụ với các thành phần chính bao gồm:

- **ESP32:** Là vi điều khiển dùng để điều khiển hệ thống, kết nối Wi-Fi và xử lý dữ liệu.
- **Cảm biến ACS712:** Đo dòng điện trong mạch và tính toán công suất tiêu thụ.
- **Thẻ SD:** Lưu trữ dữ liệu đo lường cục bộ.
- **Đám mây hoặc MQTT:** Truyền dữ liệu lên đám mây để giám sát từ xa.
- **Phát triển phần mềm:** Cung cấp giao diện và các chức năng cho người dùng để giám sát và phân tích dữ liệu tiêu thụ

1.4 GIỚI HẠN

Mặc dù hệ thống giám sát năng lượng tiêu thụ sử dụng ESP32 có thể cung cấp nhiều tính năng hữu ích, nhưng nghiên cứu này sẽ có một số giới hạn, bao gồm:

- **Độ chính xác của cảm biến:** ACS712 có độ chính xác nhất định, có thể ảnh hưởng đến kết quả đo lường trong những ứng dụng yêu cầu độ chính xác cao.
- **Phạm vi đo lường:** Hệ thống chỉ có thể đo công suất tiêu thụ trong phạm vi dòng điện và điện áp mà cảm biến ACS712 hỗ trợ.
- **Kết nối mạng:** Dữ liệu được truyền tải lên đám mây hoặc qua MQTT phụ thuộc vào chất lượng kết nối mạng Wi-Fi, có thể gặp gián đoạn nếu kết nối mạng yếu.

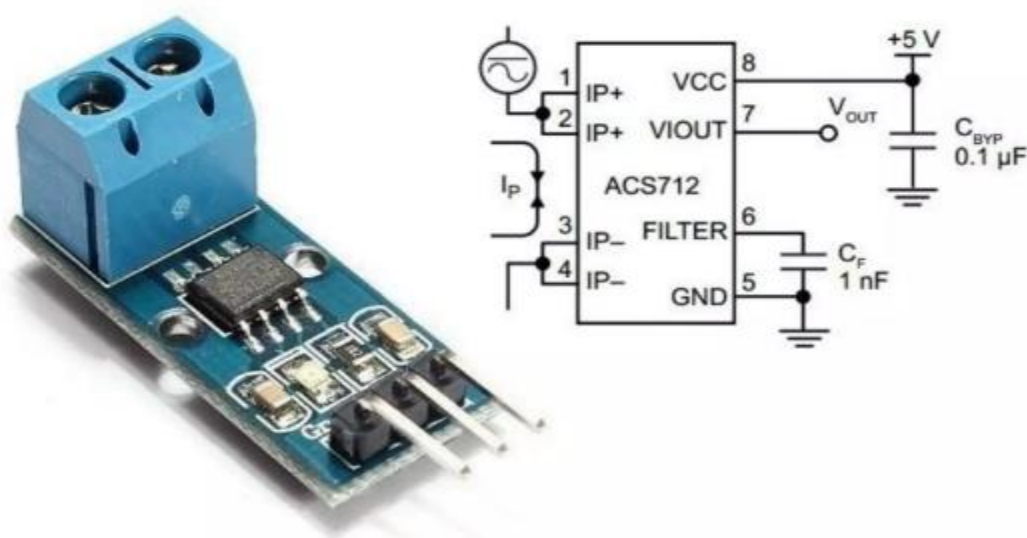
Chương 2. CƠ SỞ LÝ THUYẾT

2.1 GIỚI THIỆU PHẦN CỨNG

2.1.1 Module cảm biến dòng điện ACS712 -20A

2.1.1.1 Giới thiệu module cảm biến dòng điện ACS712-20A

- Để đo dòng điện AC, ta có thể dùng máy biến dòng CT, đây là một loại “công cụ đo lường dòng điện” được thiết kế nhằm tạo ra một dòng điện xoay chiều có cường độ tỷ lệ với cường độ dòng điện ban đầu. Tuy nhiên, giá thành của thiết bị này khá mắc. Ta có thể dùng cảm biến ACS712 được tích hợp sẵn vào module để thực hiện đo dòng điện với độ chính xác khá cao, kết nối đơn giản- thuận tiện và giá thành phải chăng.

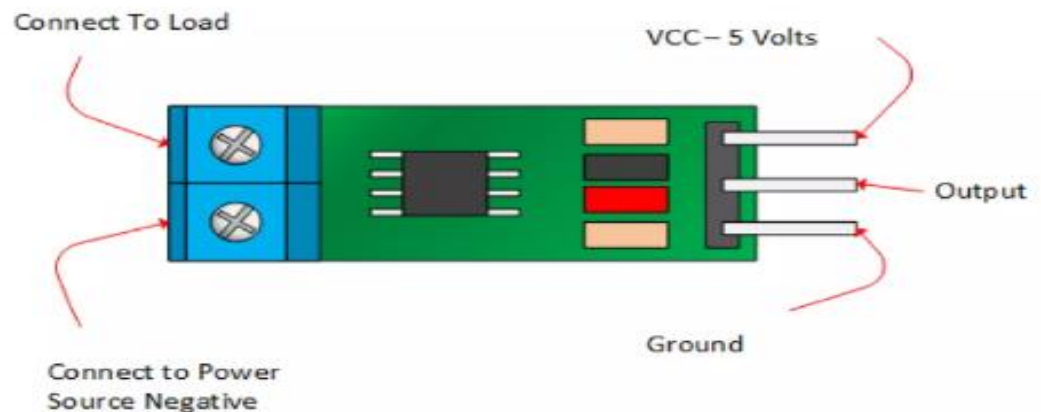


Hình 2.1 Hình ảnh của module và AC cảm biến dòng ASC 712

- Module ở hình trên sử dụng cảm biến dòng điện ACS712- đây là một cảm biến dòng tuyến tính trên hiệu ứng Hall, giúp chuyển dòng điện cần đo thành giá trị điện thế. Chân ASC 712 sẽ xuất ra một tín hiệu analog ở chân Vout biến đổi tuyến tính theo I_p (dòng điện cần đo) được lấy mẫu thứ cấp DC(hoặc AC) trong phạm vi cho phép. Tụ C_f dùng cho mục đích chống nhiễu.

2.1.1.2 Các thông số kỹ thuật

- Các thông số kỹ thuật của module ACS712 -20A:
 - Đường tín hiệu analog có độ nhiễu thấp
 - Thời gian tăng của đầu ra để đáp ứng với đầu vào là 5 μ s
 - Băng thông 80 Khz
 - Tổng lỗi ngõ ra tại $T_a = 25^{\circ}\text{C}$ là 1,5%
 - Điện trở dây dẫn trong là 1,2m Ω
 - Nguồn vận hành đơn : 5V
 - Dòng tiêu thụ (max): 13mA
 - Độ nhạy đầu ra từ 96 - 104mV/A
 - Điện áp ngõ ra tương ứng với dòng AC hoặc DC
 - Điện áp ngõ ra cực kỳ ổn định
 - I_p : từ -20A đến 20A
 - Nhiệt độ hoạt động từ -40 đến 85 $^{\circ}\text{C}$
- Ngoài ra còn có các loại cảm biến dòng khác như :
 - ACS712 - 5A: khoảng đo từ -5A đến 5A, độ nhạy điện áp 180 - 190mV/A
 - ACS712 - 30A: khoảng đo từ -30A đến 30A, độ nhạy điện áp 64 - 68mV/A



Hình 2.2 Hình ảnh minh họa kết nối với module ASC712

2.1.2 Module vi điều khiển ESP32

2.1.2.1 Giới thiệu module vi điều khiển ESP32

- Để điều khiển và quản lý các thiết bị điện tử, ta có thể sử dụng vi điều khiển truyền thống như Arduino hoặc STM32. Tuy nhiên, với nhu cầu kết nối Wi-Fi, Bluetooth và xử lý dữ liệu nhanh chóng, ESP32 là một lựa chọn tối ưu. ESP32 có giá thành phải chăng, tích hợp nhiều tính năng mạnh mẽ, giúp đơn giản hóa việc thiết kế và triển khai các hệ thống IoT.



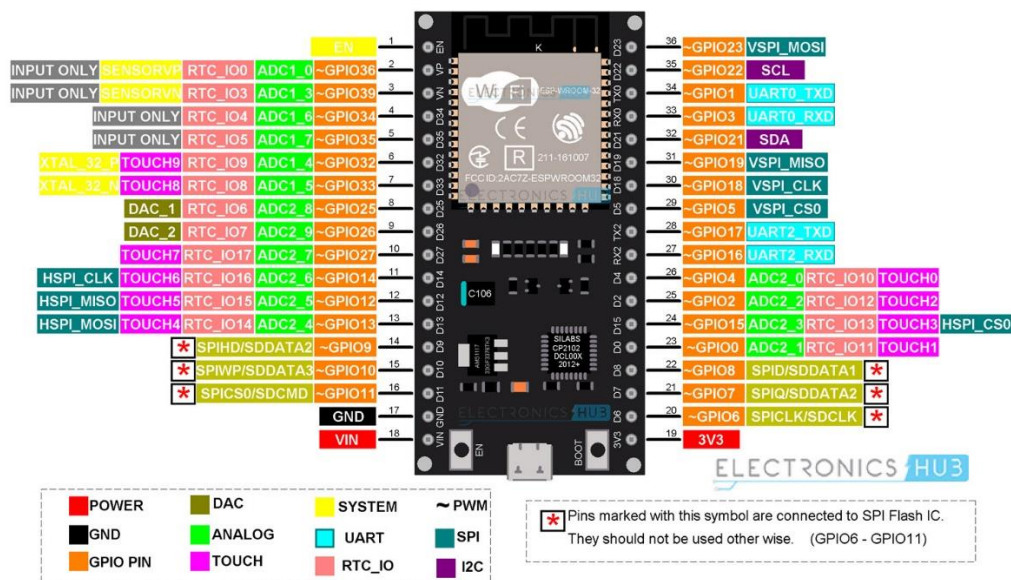
Hình 2.3 Hình ảnh module vi điều khiển ESP32

- **Đặc điểm nổi bật của ESP32:**
 - Tích hợp Wi-Fi và Bluetooth: ESP32 hỗ trợ Wi-Fi 802.11 b/g/n và Bluetooth 4.2 BLE, giúp truyền dữ liệu không dây nhanh chóng.
 - Hiệu suất cao: Với vi xử lý lõi kép Tensilica LX6 tốc độ 240MHz, ESP32 có thể xử lý nhiều tác vụ đồng thời.
 - Tiêu thụ điện năng thấp: Hỗ trợ nhiều chế độ tiết kiệm năng lượng như Deep Sleep, Light Sleep giúp kéo dài thời gian hoạt động của thiết bị.
 - Hỗ trợ nhiều giao thức giao tiếp: UART, SPI, I2C, PWM, ADC, DAC, I2S giúp kết nối linh hoạt với nhiều cảm biến và module khác.

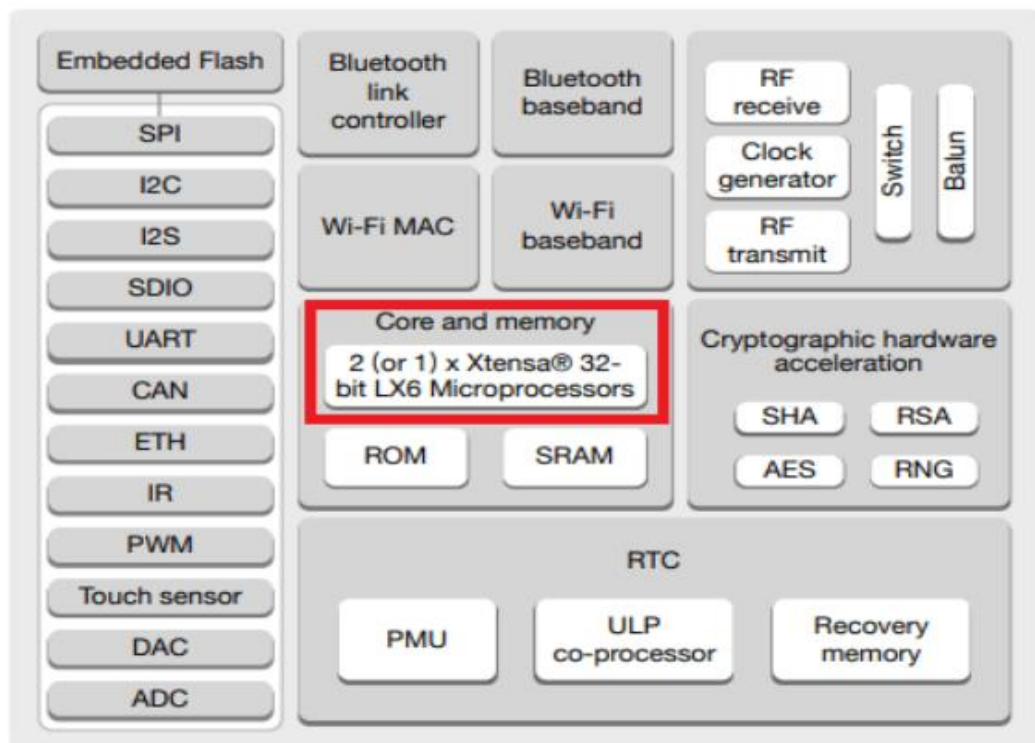
- Bảo mật cao: Hỗ trợ mã hóa AES, SHA, RSA và các tính năng bảo mật phần cứng khác.

2.1.2.2 Thông số kỹ thuật

- Các thông số kỹ thuật của ESP32
 - Vi xử lý: Dual-core Tensilica LX6 (240 MHz)
 - RAM: 520 KB SRAM
 - Bộ nhớ flash: 4MB hoặc hơn (tùy model)
 - Kết nối: Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BLE
 - GPIO: 34 chân I/O có thể lập trình
 - ADC: 18 kênh, độ phân giải 12-bit
 - DAC: 2 kênh, độ phân giải 8-bit
 - Giao tiếp: UART, SPI, I2C, PWM, I2S
 - Điện áp hoạt động: 3.3V
 - Dòng tiêu thụ ở chế độ chờ: 10µA
 - Hỗ trợ Deep Sleep: Giúp tiết kiệm năng lượng
 - Nhiệt độ hoạt động: -40°C đến 125°C



Hình 2.4 Hình ảnh minh họa sơ đồ chân của bo 30 chân ESP32



Hình 2.5 Cấu trúc của module ESP32

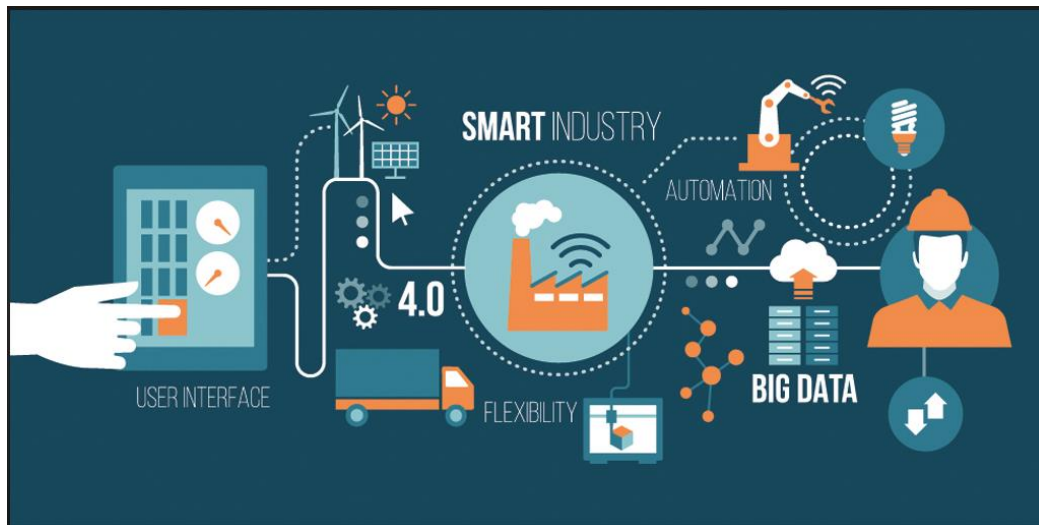
2.1.2.3 Các ứng dụng của vi điều khiển ESP32

- ESP32 được sử dụng rộng rãi trong nhiều lĩnh vực như:
 - Nhà thông minh (Smart Home): Điều khiển thiết bị từ xa, cảm biến nhiệt độ, camera an ninh.



Hình 2.6 Hình ngôi nhà thông minh

- IoT công nghiệp: Giám sát hệ thống, điều khiển động cơ, cảm biến môi trường.



Hình 2.7 Hình IOT trong công nghiệp

- Thiết bị đeo thông minh: Đồng hồ thông minh, thiết bị theo dõi sức khỏe.



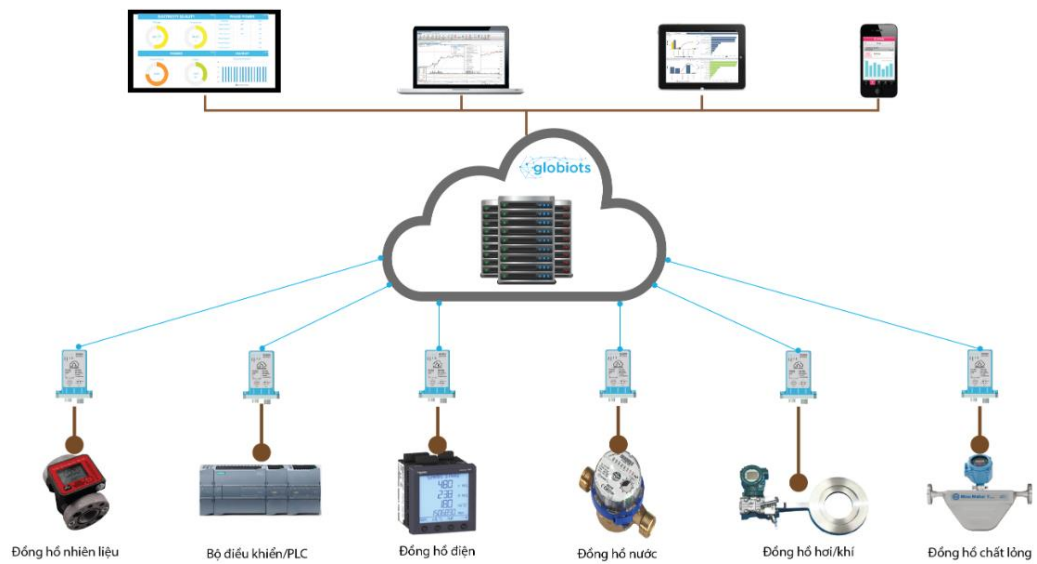
Hình 2.8 Hình Thiết bị đeo thông minh

- Robot và tự động hóa: Xe tự hành, máy in 3D, hệ thống điều khiển tự động.



Hình 2.9 Hình Robot

- Hệ thống đo lường và giám sát năng lượng: Theo dõi mức tiêu thụ điện, đo dòng điện, điện áp.



Hình 2.10 Hình Hệ thống đo lường và giám sát năng lượng

2.1.3 Thẻ SD – lưu trữ dữ liệu cục bộ

2.1.3.1 Giới thiệu thẻ SD

- Thẻ SD được sử dụng để lưu trữ dữ liệu đo lường từ cảm biến, giúp hệ thống có thể ghi lại lịch sử tiêu thụ năng lượng ngay cả khi không có kết nối internet.



Hình 2.11 Hình ảnh thẻ SD

- Nguyên lý hoạt động của thẻ SD:
 - o Thẻ SD hoạt động bằng cách lưu trữ dữ liệu dưới dạng các tệp tin có thể truy xuất và đọc lại bất kỳ lúc nào. Khi ESP32 thu thập dữ liệu từ cảm biến, nó sẽ ghi dữ liệu này vào thẻ SD thông qua giao thức SPI. Hệ thống có thể mở, ghi, sửa đổi hoặc xóa các tệp tin trên thẻ, giúp quản lý dữ liệu linh hoạt. ESP32 sử dụng thư viện như SD.h hoặc SPIFFS để giao tiếp với thẻ SD, giúp đọc và ghi dữ liệu dễ dàng.

2.1.3.2 Các thông số kỹ thuật của thẻ SD

- Hỗ trợ thẻ nhớ: MicroSD, SDHC (hỗ trợ lên đến 32GB)
- Giao tiếp: SPI (Serial Peripheral Interface)
- Điện áp hoạt động: 3.3V - 5V
- Dòng tiêu thụ: Khoảng 100mA (tùy loại thẻ)
- Tốc độ đọc/ghi: Phụ thuộc vào tốc độ của thẻ SD (Class 4 - Class 10)
- Hệ thống file: FAT16, FAT32

- Ứng dụng: Lưu trữ dữ liệu cảm biến, nhật ký hệ thống, cập nhật firmware từ thẻ SD, ghi nhận dữ liệu ngoại tuyến khi không có kết nối mạng.
- Khả năng chịu nhiệt: -25°C đến 85°C (tùy loại thẻ)

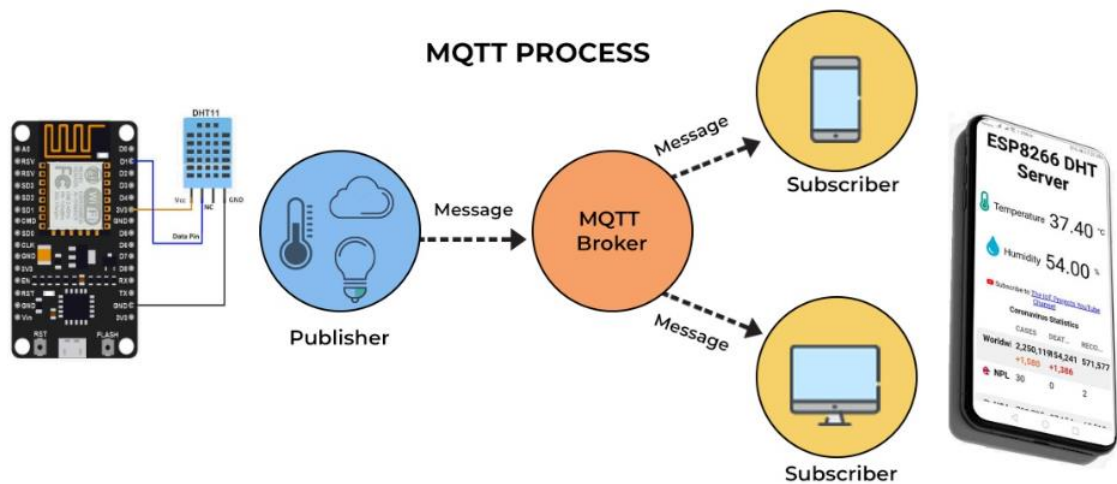
2.1.3.3 Ứng dụng thực tế của thẻ SD trong giám sát năng lượng

- Ghi nhận dữ liệu tiêu thụ điện: Mỗi lần ESP32 đo dòng điện và công suất, nó có thể lưu vào tệp nhật ký trên thẻ SD.
- Phân tích dữ liệu ngoại tuyến: Người dùng có thể lấy thẻ SD ra và phân tích dữ liệu trên máy tính để có cái nhìn tổng quan về mức tiêu thụ năng lượng.
- Lưu trữ dữ liệu dài hạn: Thẻ SD giúp hệ thống có thể hoạt động ngay cả khi mất kết nối internet, đảm bảo không mất dữ liệu.
- Cập nhật firmware qua thẻ SD: Một số hệ thống có thể sử dụng thẻ SD để cập nhật chương trình mới mà không cần kết nối với máy tính.

2.1.4 Kết nối Wi-Fi/MQTT - Gửi dữ liệu tới đám mây

2.1.4.1 Giới thiệu

- MQTT là một giao thức giao tiếp nhẹ dựa trên mô hình **publish-subscribe**. Giao thức này cho phép các thiết bị **IOT** gửi và nhận các tin nhắn nhỏ và đơn giản qua mạng Internet.
- ESP32 hỗ trợ Wi-Fi và MQTT giúp truyền dữ liệu đo lường từ cảm biến lên đám mây hoặc server, giúp giám sát dữ liệu từ xa một cách hiệu quả.



Hình 2.12 Hình ảnh minh họa MQTT

- Nguyên lý hoạt động của MQTT:
 - MQTT (Message Queuing Telemetry Transport) là một giao thức nhắn tin nhẹ sử dụng mô hình **publish/subscribe**, cho phép các thiết bị IoT gửi và nhận dữ liệu một cách hiệu quả thông qua broker MQTT.
 - **Publish:** ESP32 có thể gửi dữ liệu cảm biến lên broker MQTT.
 - **Subscribe:** Thiết bị khác có thể đăng ký (subscribe) để nhận dữ liệu từ ESP32.
 - **Broker:** Trung gian quản lý và chuyển tiếp dữ liệu giữa các thiết bị.

2.1.4.2 Các thông số kỹ thuật của giao thức MQTT trên ESP32

- **Giao thức:** MQTT 3.1.1
- **Tốc độ truyền tải:** 10ms - 100ms (tùy thuộc vào mạng)
- **Bảo mật:** Hỗ trợ TLS/SSL để mã hóa dữ liệu

- **Ứng dụng:** Giám sát dữ liệu từ xa, IoT, hệ thống tự động hóa công nghiệp
- **Thư viện hỗ trợ:** PubSubClient, AsyncMQTTClient

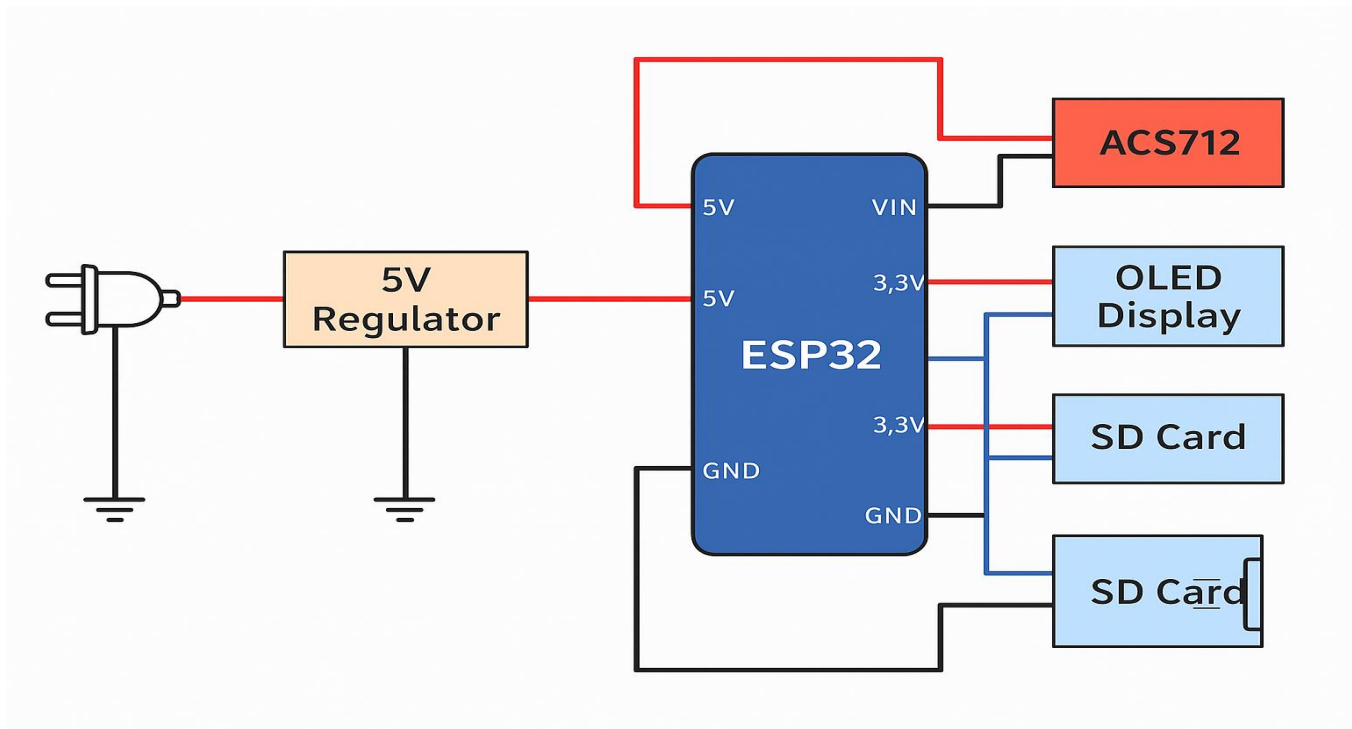
2.1.4.3 Ứng dụng thực tế của MQTT trong giám sát năng lượng

- **Giám sát dữ liệu theo thời gian thực:** Dữ liệu từ cảm biến có thể được gửi lên dashboard để người dùng theo dõi.
- **Tự động cảnh báo:** Khi phát hiện mức tiêu thụ điện bất thường, hệ thống có thể gửi cảnh báo đến điện thoại.
- **Điều khiển thiết bị từ xa:** Người dùng có thể điều khiển các thiết bị điện từ xa thông qua MQTT.

2.1.5 Nguồn điện cho hệ thống

2.1.5.1 Giới thiệu

- Nguồn điện là một thành phần thiết yếu trong bất kỳ hệ thống IoT nào, đặc biệt là với các hệ thống giám sát năng lượng sử dụng ESP32 và cảm biến dòng điện như ACS712. Một nguồn điện ổn định không chỉ đảm bảo thiết bị hoạt động chính xác mà còn tăng tuổi thọ linh kiện, giảm rủi ro sai số hoặc treo thiết bị.
- ESP32 có thể được cấp nguồn thông qua cổng USB (5V), chân Vin (5V) hoặc trực tiếp 3.3V nếu nguồn có khả năng ổn định tốt. Ngoài ra, các linh kiện đi kèm như cảm biến dòng, màn hình OLED, thẻ SD cũng cần được cung cấp đúng điện áp để hoạt động ổn định.



Hình 2.13 Hình ảnh minh họa sơ đồ cấp nguồn cho hệ thống ESP32

- Nguyên lý cấp nguồn trong hệ thống IoT
- Việc cấp nguồn trong một hệ thống giám sát năng lượng tiêu thụ cần tuân theo các nguyên tắc sau:
 - **Ổn định điện áp:** Các linh kiện như ESP32, ACS712, thẻ nhớ SD rất nhạy cảm với dao động điện áp. Nguồn cấp cần được lọc tốt và ổn định.
 - **Cung cấp đủ dòng điện:** Đảm bảo tổng dòng tiêu thụ của hệ thống không vượt quá khả năng cung cấp của nguồn.
 - **Tùy biến theo ứng dụng:**
 - **Hệ thống cố định:** Sử dụng adapter 5V-2A là giải pháp phổ biến.
 - **Hệ thống di động:** Dùng pin Li-ion 3.7V kết hợp với mạch sạc TP4056 và ổn áp AMS1117 (5V hoặc 3.3V).

- **Bảo vệ và cách ly:** Nên có các linh kiện bảo vệ như diode chống ngược cực, tụ lọc nhiễu và bảo vệ quá áp/quá dòng.

2.1.5.2 Thông số kỹ thuật nguồn cho các thành phần chính

Thiết bị	Điện áp hoạt động	Dòng tiêu thụ (xấp xỉ)
ESP32	3.3V	80mA – 250mA
Cảm biến ACS712	5V	~10mA
Màn hình OLED 0.96"	3.3V hoặc 5V	~20mA
Thẻ nhớ SD	3.3V	30mA – 100mA

2.1.5.3 Ứng dụng thực tế trong giám sát năng lượng

- **Sử dụng adapter 5V – 2A:** Phù hợp với hệ thống giám sát năng lượng cố định trong nhà, đảm bảo nguồn ổn định và đủ công suất.
- **Pin sạc Li-ion 3.7V + TP4056 + AMS1117:** Giải pháp lý tưởng cho hệ thống di động hoặc cần hoạt động khi mất điện.
- **Hệ thống ngoài trời (năng lượng mặt trời):** Có thể sử dụng tấm pin mặt trời nhỏ kết hợp pin sạc và mạch điều khiển để cung cấp điện liên tục.
- **Tích hợp mạch giám sát pin:** Giúp theo dõi mức năng lượng còn lại và gửi cảnh báo khi pin yếu.
- **Tự động chuyển nguồn:** Khi nguồn chính mất, hệ thống tự động chuyển sang nguồn phụ như pin dự phòng

2.2 GIỚI THIỆU PHẦN MỀM

2.2.1 Thư viện cho ACS712

2.2.1.1 Giới thiệu

- Thư viện ACS712 là một bộ công cụ lập trình mạnh mẽ và tiện dụng, được phát triển để hỗ trợ việc giao tiếp và xử lý tín hiệu từ cảm biến

ACS712, một loại cảm biến đo dòng điện sử dụng công nghệ hiệu ứng Hall. ACS712 được thiết kế để đo dòng điện xoay chiều (AC) và một chiều (DC) với độ chính xác cao, giúp chuyển đổi giá trị dòng điện thành tín hiệu điện áp tương ứng, mà cụ thể là tín hiệu analog tuyến tính. Đây là một trong những cảm biến phổ biến nhất trong các ứng dụng đo lường điện năng, nhờ vào sự đơn giản trong việc kết nối và khả năng cung cấp các giá trị đo đạc chính xác với chi phí rất hợp lý.

- Với việc sử dụng hiệu ứng Hall, cảm biến ACS712 không cần tiếp xúc trực tiếp với mạch điện, giúp đo dòng điện trong các mạch có điện áp cao mà không cần phải cắt hay can thiệp vào mạch, điều này rất quan trọng khi làm việc trong các hệ thống điện năng, nơi mà việc đo lường an toàn là yêu cầu hàng đầu. Tín hiệu đầu ra của cảm biến ACS712 là một tín hiệu analog tương ứng với dòng điện đo được, giúp dễ dàng tích hợp vào các vi điều khiển như Arduino, ESP32 hay các nền tảng khác để thu thập và xử lý dữ liệu.
- Thư viện ACS712 được thiết kế để đơn giản hóa việc sử dụng cảm biến này, với các hàm lập trình hỗ trợ người dùng trong việc đọc tín hiệu từ cảm biến, tính toán dòng điện, hiệu chuẩn và xử lý các tín hiệu này. Thư viện này không chỉ giúp đo dòng điện mà còn giúp tính toán các giá trị liên quan đến dòng điện, như dòng điện hiệu dụng (RMS) đối với dòng AC, hoặc giúp người dùng dễ dàng chuyển đổi giữa các đơn vị đo lường khác nhau, chẳng hạn như mA và A.
- Một trong những ưu điểm lớn của thư viện ACS712 là sự dễ dàng trong việc tích hợp với các dự án IoT, nơi mà việc theo dõi và phân tích dòng điện tiêu thụ là rất quan trọng. Dữ liệu đo được có thể được sử dụng trong nhiều ứng dụng, từ việc giám sát năng lượng tiêu thụ trong các hệ thống điện gia đình, công nghiệp, đến việc theo dõi các thiết bị tiêu thụ điện trong môi trường làm việc. Thư viện ACS712 hỗ trợ đọc liên tục giá trị dòng điện, giúp theo dõi tình trạng dòng điện theo thời gian thực, từ đó phát hiện các bất thường hoặc vấn đề trong mạch điện.

- Ngoài ra, thư viện ACS712 còn giúp người dùng dễ dàng hiệu chỉnh cảm biến, xử lý các tín hiệu nhiễu và đảm bảo độ chính xác cao trong suốt quá trình đo. Một số thư viện cũng hỗ trợ tính toán giá trị đỉnh hoặc giá trị RMS của dòng điện xoay chiều, giúp người dùng thu thập các chỉ số quan trọng về dòng điện thay vì chỉ đơn thuần là giá trị tức thời.
- Ứng dụng của cảm biến ACS712 và thư viện của nó rất rộng rãi trong nhiều lĩnh vực. Trong các hệ thống giám sát năng lượng tiêu thụ, cảm biến ACS712 và thư viện sẽ giúp đo lường dòng điện trong các thiết bị điện, qua đó giúp người dùng theo dõi mức độ tiêu thụ điện năng và thực hiện các biện pháp tiết kiệm năng lượng. Trong các dự án IoT, thư viện ACS712 có thể được sử dụng để phát triển các hệ thống giám sát năng lượng thông minh, giúp theo dõi và quản lý tiêu thụ điện năng từ xa, cung cấp thông tin chi tiết và báo cáo về mức độ sử dụng năng lượng của các thiết bị hoặc toàn bộ hệ thống.
- Thư viện ACS712 là công cụ không thể thiếu trong việc phát triển các ứng dụng đo lường và giám sát dòng điện, giúp tối ưu hóa việc sử dụng điện năng và đảm bảo hiệu quả trong các hệ thống điện.

2.2.1.2 Các thông số kỹ thuật của thư viện ACS712:

- Hỗ trợ cảm biến ACS712 với các dải đo: $\pm 5A$, $\pm 20A$ và $\pm 30A$.
- Cho phép cấu hình hệ số nhạy (Sensitivity) theo từng phiên bản cảm biến.
- Tích hợp tính năng hiệu chuẩn điểm 0 (zero offset calibration).
- Hỗ trợ lấy mẫu trung bình để tăng độ chính xác.
- Hỗ trợ đo RMS cho dòng AC.
- Đầu ra có thể quy đổi thành mA hoặc A.
- **Nguyên lý hoạt động:** Khi dòng điện chạy qua ACS712, cảm biến sử dụng hiệu ứng Hall để tạo ra một điện áp đầu ra tuyến tính với dòng điện. Thư viện sẽ đọc tín hiệu analog từ chân đầu ra Vout của cảm biến

qua ADC của ESP32. Từ giá trị điện áp đo được, thư viện sử dụng công thức tuyến tính hoặc RMS (với dòng AC) để tính toán dòng điện thực tế.

- Một số thư viện còn cho phép lấy mẫu liên tục và trung bình nhiều giá trị trong khoảng thời gian ngắn để giảm nhiễu, tăng độ tin cậy của kết quả đo.

2.2.1.3 Ứng dụng:

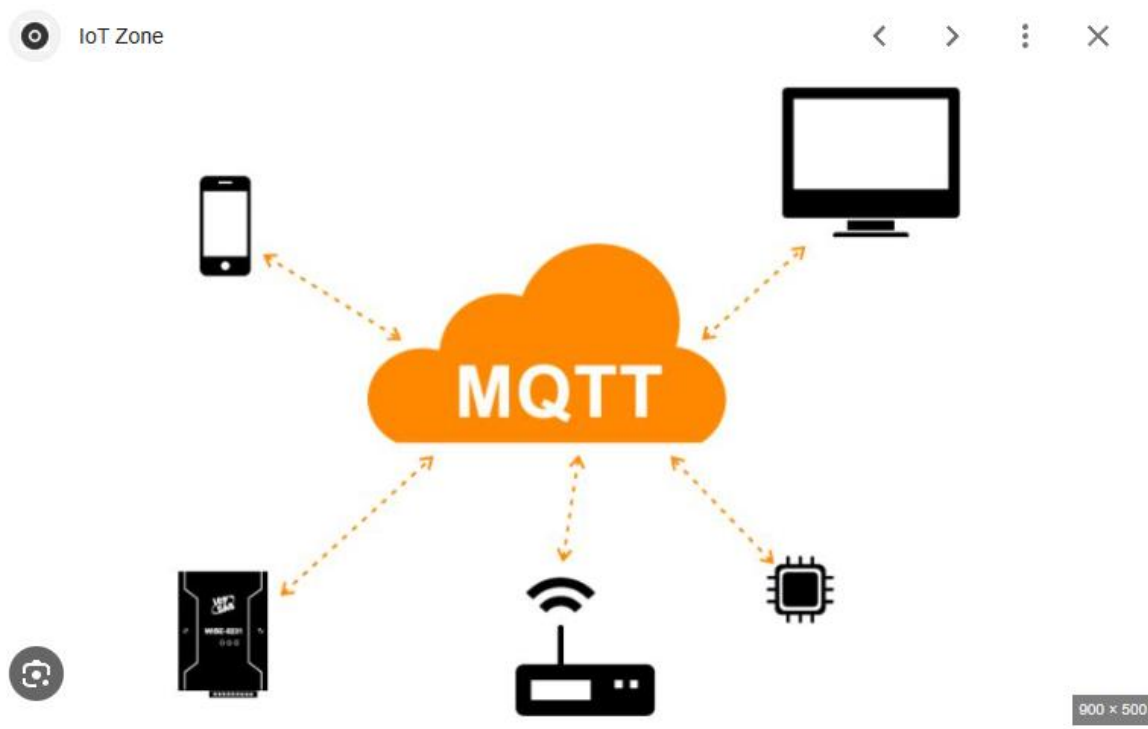
- Giám sát dòng điện tiêu thụ trong hộ gia đình, tòa nhà hoặc nhà máy.
- Bảo vệ mạch điện bằng cách phát hiện dòng điện vượt ngưỡng.
- Thu thập dữ liệu dòng điện cho hệ thống báo cáo, phân tích.
- Tích hợp vào các ứng dụng IoT để truyền dữ liệu lên cloud thông qua MQTT hoặc HTTP.
- Được sử dụng rộng rãi trong các hệ thống quản lý năng lượng thông minh.
- Một số thư viện nổi bật hiện nay là:
 - ACS712.h của Rob Tillaart.
 - EmonLib – thư viện phổ biến cho các dự án đo năng lượng.
 - Tự viết hàm đọc analog kết hợp lọc và hiệu chỉnh nếu không dùng thư viện có sẵn.
- Việc sử dụng thư viện giúp rút ngắn thời gian phát triển và tăng độ chính xác khi làm việc với cảm biến ACS712 trong các hệ thống giám sát năng lượng.

2.2.2 MQTT/HTTP

2.2.2.1 Giới thiệu MQTT/HTTP

- Hỗ trợ cảm biến ACS712 với các dải đo: $\pm 5A$, $\pm 20A$ và $\pm 30A$.

- Cho phép cấu hình hệ số nhạy (Sensitivity) theo từng phiên bản cảm biến.
- Tích hợp tính năng hiệu chuẩn điểm 0 (zero offset calibration).
- Hỗ trợ lấy mẫu trung bình để tăng độ chính xác.
- Hỗ trợ đo RMS cho dòng AC.
- Đầu ra có thể quy đổi thành mA hoặc A.
- **Nguyên lý hoạt động:** Khi dòng điện chạy qua ACS712, cảm biến sử dụng hiệu ứng Hall để tạo ra một điện áp đầu ra tuyến tính với dòng điện. Thư viện sẽ đọc tín hiệu analog từ chân đầu ra Vout của cảm biến qua ADC của ESP32. Từ giá trị điện áp đo được, thư viện sử dụng công thức tuyến tính hoặc RMS (với dòng AC) để tính toán dòng điện thực tế.
- Một số thư viện còn cho phép lấy mẫu liên tục và trung bình nhiều giá trị trong khoảng thời gian ngắn để giảm nhiễu, tăng độ tin cậy của kết quả đo.
- **MQTT** là một giao thức nhắn tin nhẹ dựa trên mô hình publish/subscribe, thường được sử dụng trong các hệ thống IoT để truyền dữ liệu giữa các thiết bị một cách nhanh chóng và tiết kiệm băng thông. Nhờ tính linh hoạt, tốc độ cao và khả năng mở rộng, MQTT trở thành giải pháp lý tưởng để kết nối các thiết bị nhúng như ESP32 với các dịch vụ đám mây.
- ESP32 hỗ trợ kết nối Wi-Fi tích hợp và sử dụng giao thức MQTT để truyền dữ liệu đo lường từ cảm biến lên đám mây hoặc server một cách hiệu quả. Thông qua đó, người dùng có thể theo dõi trạng thái tiêu thụ điện năng của hệ thống từ xa, bất kỳ lúc nào, ở bất cứ đâu chỉ với một thiết bị có kết nối internet. Điều này đặc biệt hữu ích trong các ứng dụng giám sát năng lượng thời gian thực, tiết kiệm điện năng hoặc phát hiện sự cố điện.



Hình 2.14 Hình ảnh minh họa MQTT

Nguyên lý hoạt động của MQTT

MQTT là một giao thức nhắn tin nhẹ sử dụng mô hình **publish/subscribe**, cho phép các thiết bị IoT gửi và nhận dữ liệu một cách hiệu quả thông qua broker MQTT.

- **Publish:** ESP32 có thể gửi dữ liệu cảm biến lên broker MQTT.
- **Subscribe:** Thiết bị khác có thể đăng ký (subscribe) để nhận dữ liệu từ ESP32.
- **Broker:** Trung gian quản lý và chuyển tiếp dữ liệu giữa các thiết bị.

2.2.2.2 Các thông số kỹ thuật của giao thức MQTT trên ESP32

- **Giao thức:** MQTT 3.1.1
- **Tốc độ truyền tải:** 10ms - 100ms (tùy thuộc vào mạng)
- **Bảo mật:** Hỗ trợ TLS/SSL để mã hóa dữ liệu
- **Ứng dụng:** Giám sát dữ liệu từ xa, IoT, hệ thống tự động hóa công nghiệp
- **Thư viện hỗ trợ:** PubSubClient, AsyncMQTTClient

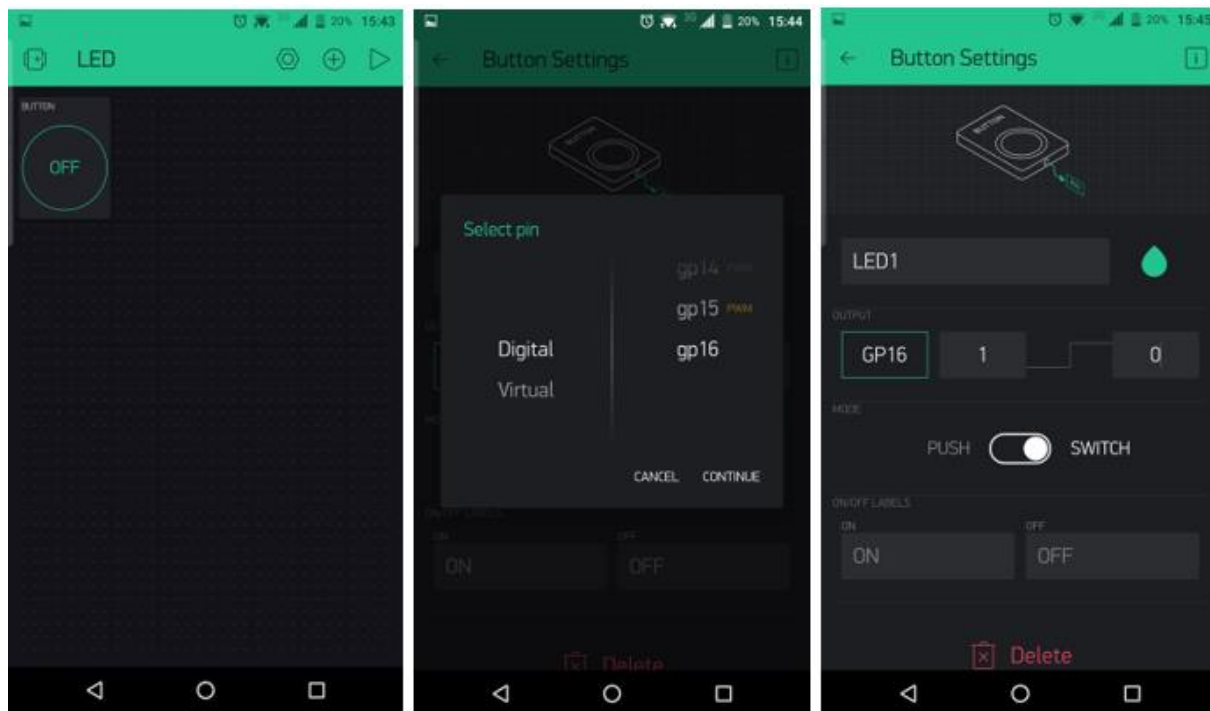
2.2.2.3 Ứng dụng thực tế của MQTT trong giám sát năng lượng

- Giám sát dữ liệu theo thời gian thực: Dữ liệu từ cảm biến có thể được gửi lên dashboard để người dùng theo dõi.
- Tự động cảnh báo: Khi phát hiện mức tiêu thụ điện bất thường, hệ thống có thể gửi cảnh báo đến điện thoại.
- Điều khiển thiết bị từ xa: Người dùng có thể điều khiển các thiết bị điện từ xa thông qua MQTT.
- Kết nối với dịch vụ đám mây: Dễ dàng tích hợp với các nền tảng như Firebase, Blynk, Node-RED, Home Assistant...
- Tính linh hoạt cao: Hệ thống có thể mở rộng để kết nối nhiều thiết bị cùng lúc thông qua broker trung gian.

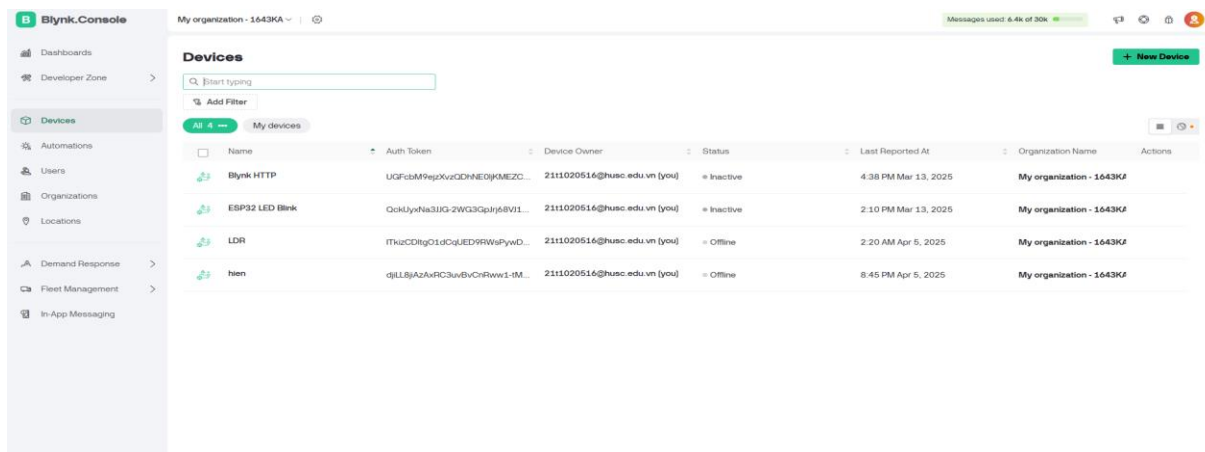
2.2.3 BLYNK

2.2.3.1 Giới thiệu về ứng dụng Blynk – Giao diện giám sát và điều khiển từ xa

Blynk là một nền tảng IoT mạnh mẽ và linh hoạt, cho phép người dùng dễ dàng xây dựng các ứng dụng di động để điều khiển, giám sát thiết bị từ xa. Thay vì phải lập trình giao diện thủ công, Blynk cung cấp môi trường trực quan, cho phép kéo - thả các widget như nút nhấn, đồng hồ đo, biểu đồ... để tạo dashboard điều khiển mà không cần nhiều kiến thức về lập trình giao diện.



Hình 2.15 Hình ảnh minh họa Blynk trên mobile



Hình 2.16 Hình ảnh minh họa Blynk trên Web

Nguyên lý hoạt động của Blynk

Blynk hoạt động theo mô hình client-server:

- ESP32 (thiết bị phần cứng): gửi và nhận dữ liệu từ cảm biến hoặc các thiết bị đầu cuối.

- Blynk Cloud hoặc Server riêng: trung gian tiếp nhận, xử lý và chuyển tiếp dữ liệu.
- Ứng dụng Blynk trên điện thoại: giao diện người dùng, hiển thị và điều khiển dữ liệu theo thời gian thực.

ESP32 giao tiếp với server Blynk thông qua giao thức TCP hoặc MQTT thông qua thư viện Blynk hỗ trợ sẵn. Giao tiếp này đảm bảo việc truyền nhận dữ liệu ổn định và nhanh chóng giữa thiết bị và điện thoại.

Các tính năng nổi bật của Blynk

- Hiển thị dữ liệu cảm biến thời gian thực: Bằng các widget như Gauge (đồng hồ đo), Value Display (giá trị), hoặc biểu đồ (Graph).
- Điều khiển thiết bị từ xa: Dùng nút nhấn (Button), công tắc (Switch), thanh trượt (Slider)...
- Cảnh báo tự động: Gửi thông báo (push notification) khi có sự kiện xảy ra (ví dụ vượt ngưỡng điện năng).
- Lập lịch hoạt động: Tự động bật/tắt thiết bị dựa trên thời gian định sẵn (Widget Timer).
- Tương thích đa thiết bị: Một tài khoản có thể giám sát và điều khiển nhiều thiết bị ESP32 khác nhau cùng lúc.
- Đồng bộ dữ liệu trên đám mây: Không bị giới hạn địa điểm truy cập, chỉ cần có kết nối internet.

Ưu điểm khi sử dụng Blynk

- Không cần xây dựng server riêng: Giảm chi phí và công sức bảo trì hệ thống backend.
- Giao diện dễ dùng, trực quan: Không cần kỹ năng lập trình giao diện.
- Hỗ trợ đa nền tảng: iOS, Android và cả trình duyệt web (với Blynk.Console).

- Tính linh hoạt và mở rộng: Hỗ trợ nhiều loại vi điều khiển và dễ dàng tích hợp với các giao thức khác như MQTT, HTTP API.

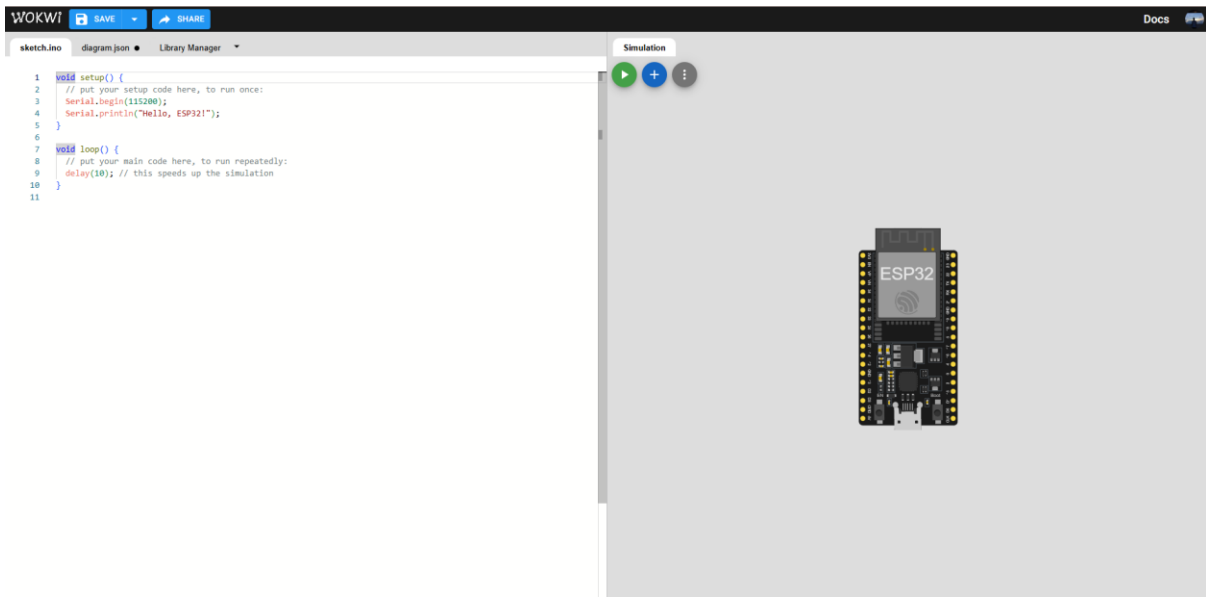
2.2.3.2 Ứng dụng thực tế về hệ thống giám sát và điều khiển từ xa

- Theo dõi dòng điện, điện áp, công suất theo thời gian thực trên điện thoại.
- Nhận thông báo cảnh báo khi hệ thống tiêu thụ vượt ngưỡng thiết lập.
- Điều khiển từ xa các thiết bị tiêu thụ điện như quạt, đèn, máy lạnh... để tối ưu năng lượng.
- Lập lịch tự động ngắt thiết bị trong giờ cao điểm nhằm giảm hóa đơn tiền điện.
- Giao diện đẹp mắt, dễ sử dụng cho cả người không chuyên.

2.2.4 WOKWI

2.2.4.1 Giới thiệu Wokwi – Mô phỏng phần cứng trực tuyến

- Wokwi là một nền tảng mô phỏng phần cứng trực tuyến, cho phép người dùng xây dựng, kiểm tra và mô phỏng các mạch điện tử với các vi điều khiển như Arduino, ESP32, ESP8266,... ngay trên trình duyệt web mà không cần phần cứng thực tế. Đây là công cụ lý tưởng cho sinh viên, nhà phát triển hoặc những người mới học về IoT và lập trình vi điều khiển.



Hình 2.17 Hình ảnh minh họa ESP32 trên wokwi

Nguyên lý hoạt động của Wokwi

- Wokwi hoạt động như một trình mô phỏng vi điều khiển và mạch điện tử. Người dùng có thể:
 - Kéo thả các thành phần như ESP32, cảm biến, LED, nút nhấn, màn hình OLED...
 - Kết nối dây theo sơ đồ mạch thực tế
 - Lập trình với ngôn ngữ Arduino C/C++
 - Nhấn "Start Simulation" để chạy chương trình và xem kết quả mô phỏng.
 - Wokwi mô phỏng hoạt động của phần cứng một cách chính xác, từ đọc tín hiệu cảm biến đến xuất dữ liệu lên màn hình hoặc giao tiếp với các module khác.
- Các tính năng chính của Wokwi
 - Mô phỏng ESP32, Arduino, và nhiều vi điều khiển khác
 - Hỗ trợ cảm biến phổ biến: DHT22, ACS712, PIR, LM35, v.v...
 - Hỗ trợ hiển thị: OLED SSD1306, LCD 16x2, 7-segment...
 - Lập trình bằng Arduino IDE tích hợp
 - Tự động biên dịch và hiển thị lỗi lập trình

- Chia sẻ dự án dễ dàng qua đường link
- Hỗ trợ các thư viện Arduino phổ biến
- Ưu điểm khi sử dụng Wokwi
 - Không cần phần cứng thật: Tiết kiệm chi phí thiết bị
 - Không cần cài đặt phần mềm: Hoạt động hoàn toàn trên trình duyệt
 - Dễ sử dụng: Giao diện kéo - thả thân thiện, mô phỏng mạch nhanh
 - Hỗ trợ học tập và giảng dạy: Giáo viên có thể tạo bài thực hành mô phỏng
 - Phản hồi tức thì: Có thể sửa lỗi và kiểm tra kết quả ngay lập tức

2.2.4.2 Ứng dụng thực tế của Wokwi trong dự án giám sát năng lượng

- Mô phỏng ESP32 và cảm biến ACS712 để kiểm tra chương trình đo dòng điện
- Xem trực tiếp giá trị dòng điện hoặc công suất qua Serial Monitor mô phỏng
- Thử nghiệm kết nối với module SD, MQTT, Blynk mà không cần phần cứng
- Tối ưu chương trình và sơ đồ mạch trước khi triển khai thực tế
- Giảm rủi ro khi xây dựng mạch thật: Do đã thử nghiệm kỹ trên mô phỏng

Wokwi giúp quá trình phát triển và kiểm thử hệ thống giám sát năng lượng trở nên dễ dàng, tiết kiệm thời gian và chi phí, đồng thời giúp người học nắm vững kiến thức nền tảng về lập trình và phần cứng.

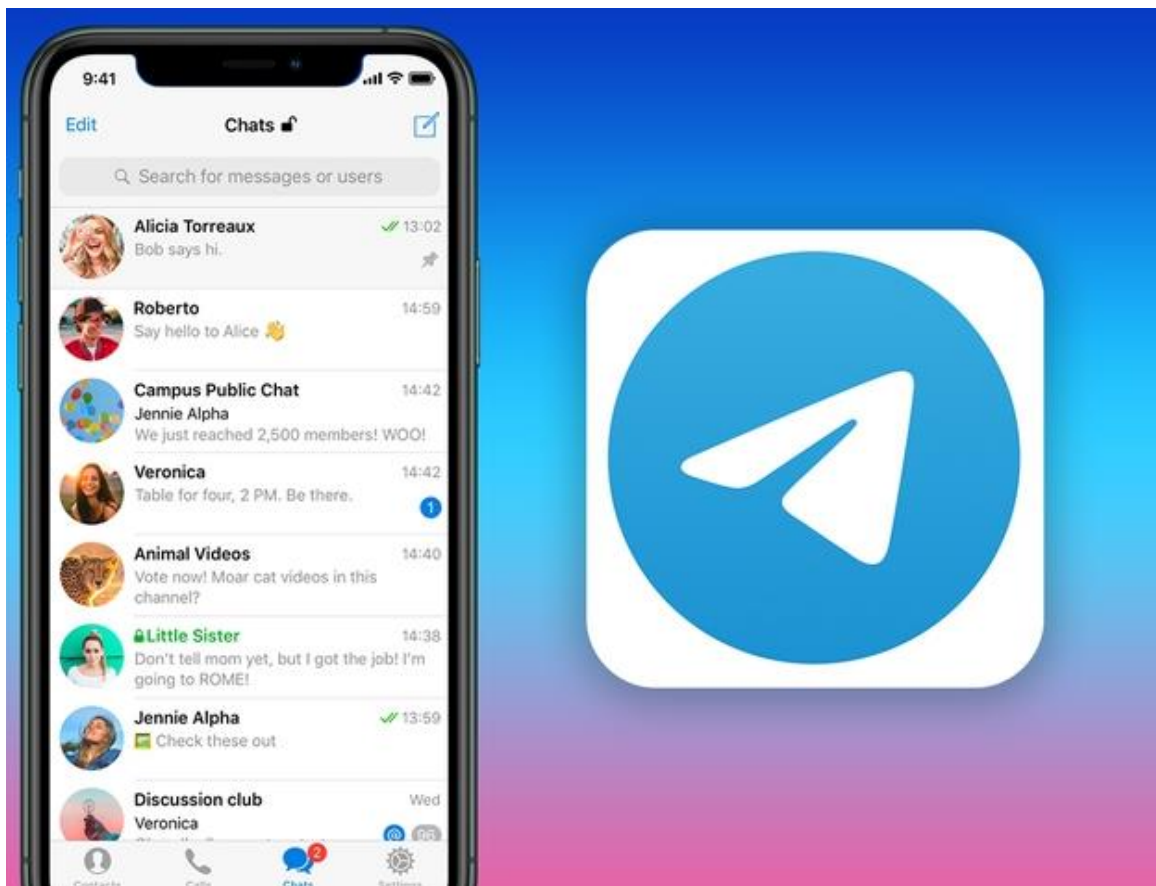
2.2.5 TELEGRAM

2.2.5.1 Giới thiệu Telegram - Cảnh báo và điều khiển thiết bị từ xa

- Telegram là một ứng dụng nhắn tin đa nền tảng nổi tiếng với khả năng bảo mật, tốc độ cao và hỗ trợ bot tự động. Không chỉ dừng lại ở vai trò ứng dụng nhắn tin thông thường, Telegram còn trở thành một công cụ mạnh mẽ trong các hệ thống IoT, đặc biệt là với các thiết bị như ESP32. Bằng cách tích hợp Telegram Bot, người dùng có thể dễ dàng nhận thông báo cảnh báo từ hệ thống, giám sát trạng

thái tiêu thụ điện, và thậm chí gửi lệnh điều khiển từ xa cho các thiết bị điện qua tin nhắn.

- Telegram sử dụng kiến trúc client-server và cho phép thiết bị IoT giao tiếp với người dùng thông qua các tin nhắn văn bản hoặc nút lệnh. Nhờ vào sự hỗ trợ của API mở rộng và giao tiếp bảo mật HTTPS, Telegram có thể tích hợp một cách linh hoạt và an toàn với ESP32. Điều này giúp cho việc thiết lập hệ thống cảnh báo hoặc điều khiển từ xa trở nên đơn giản hơn bao giờ hết.



Hình 2.18 Hình ảnh minh họa Telegram

Nguyên lý hoạt động của Telegram trong hệ thống IoT

- Telegram hoạt động như một trung gian giao tiếp giữa người dùng và thiết bị IoT thông qua Bot. Quy trình bao gồm:
 - ESP32 kết nối Wi-Fi và sử dụng HTTPS hoặc thư viện Telegram Bot.

- Gửi dữ liệu hoặc cảnh báo đến người dùng thông qua một bot Telegram.
 - Nhận phản hồi từ người dùng (chẳng hạn như lệnh bật/tắt) gửi về thiết bị.
- Người dùng chỉ cần có ứng dụng Telegram, tạo bot và lấy Token từ BotFather để tích hợp vào chương trình ESP32.
 - Các tính năng chính của Telegram Bot trong IoT
 - Gửi cảnh báo tức thì: Cảnh báo khi dòng điện vượt ngưỡng hoặc sự cố điện xảy ra.
 - Điều khiển từ xa: Gửi lệnh bật/tắt thiết bị bằng tin nhắn.
 - Xem dữ liệu đo lường: Nhận báo cáo định kỳ về công suất, dòng điện...
 - Giao diện thân thiện: Người dùng không cần lập trình, chỉ thao tác với ứng dụng Telegram quen thuộc.
 - Đa nền tảng: Hoạt động trên điện thoại, máy tính, trình duyệt.
 - Ưu điểm khi sử dụng Telegram trong hệ thống giám sát năng lượng
 - Phản hồi thời gian thực: Gửi/nhận tin nhắn tức thì.
 - An toàn và bảo mật: Giao tiếp mã hóa HTTPS.
 - Không cần ứng dụng riêng: Tận dụng nền tảng sẵn có.
 - Miễn phí và không quảng cáo.
 - Tích hợp dễ dàng với ESP32 thông qua thư viện Telegram Bot Arduino.

2.2.5.2 Ứng dụng thực tế của Telegram

- Gửi cảnh báo khi tiêu thụ vượt mức quy định, giúp người dùng phát hiện sớm tình trạng bất thường về điện năng.
- Nhận lệnh bật/tắt thiết bị tiêu thụ điện từ người dùng thông qua tin nhắn, hỗ trợ điều khiển từ xa trong trường hợp khẩn cấp hoặc để tiết kiệm điện.
- Theo dõi dữ liệu đo lường trực tiếp từ điện thoại di động, hỗ trợ giám sát liên tục mọi lúc, mọi nơi.

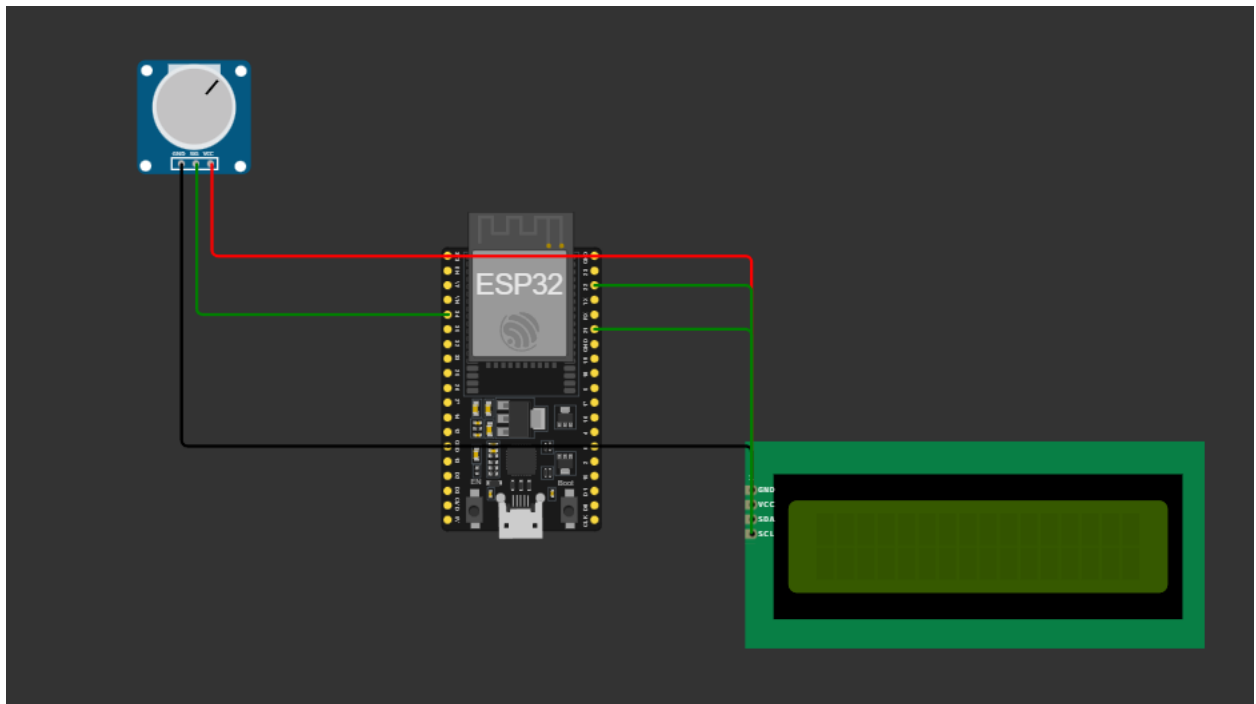
- Quản lý nhiều thiết bị từ xa trên cùng một nền tảng, phù hợp cho các hệ thống nhiều điểm giám sát hoặc hộ gia đình thông minh.
- Gửi báo cáo định kỳ theo thời gian cấu hình, giúp lưu trữ dữ liệu hoạt động và hỗ trợ phân tích xu hướng sử dụng điện.
- Cảnh báo mất kết nối hoặc sự cố hệ thống, đảm bảo hệ thống luôn trong trạng thái giám sát chủ động.
- Tương tác đa chiều giữa người và thiết bị, không chỉ nhận cảnh báo mà còn có thể gửi lệnh điều chỉnh linh hoạt.

Chương 3. MÔ PHỎNG

3.1 SƠ ĐỒ KHỐI LƯU THÔNG TIN TRÊN Đám MÂY

3.1.1 SƠ ĐỒ KHỐI:

- Sơ đồ minh họa được thực hiện trên wokwi



Hình 3.1: Hình ảnh minh họa sơ đồ khối trên wokwi

- Sơ đồ bao gồm:
 - **ESP32** làm bộ điều khiển chính.
 - **Potentiometer** (giả lập cảm biến dòng điện, như ACS712 trong thực tế).
 - **LCD1602 (I2C)** để hiển thị dữ liệu.

3.1.1.1 MÔ TẢ HOẠT ĐỘNG CỦA THIẾT BỊ:

- **Potentiometer:**
 - Giả lập cảm biến dòng điện (ACS712 trong thực tế).
 - Tín hiệu analog từ chân SIG được gửi đến GPIO34 của ESP32.
- **ESP32:**
 - Đọc tín hiệu analog từ GPIO34, chuyển đổi thành giá trị dòng điện.
 - Tính công suất tiêu thụ dựa trên điện áp cố định (220V).
 - Gửi dữ liệu đến LCD qua giao thức I2C (SDA: GPIO21, SCL: GPIO22).
 - In dữ liệu ra Serial Monitor để giám sát.
- **LCD1602:**
 - Hiển thị giá trị dòng điện (A) và công suất (W) theo thời gian thực.

3.1.1.2 SƠ ĐỒ KẾT NỐI:

3.1.1.2.1 Potentiometer (pot1) ↔ ESP32 (esp)

- **VCC (pot1) → 3V3 (esp):**
 - Cung cấp nguồn 3.3V từ ESP32 để cấp điện cho potentiometer.
 - Dây màu: **Đỏ**.
- **GND (pot1) → GND.1 (esp):**
 - Nối đất chung giữa potentiometer và ESP32.

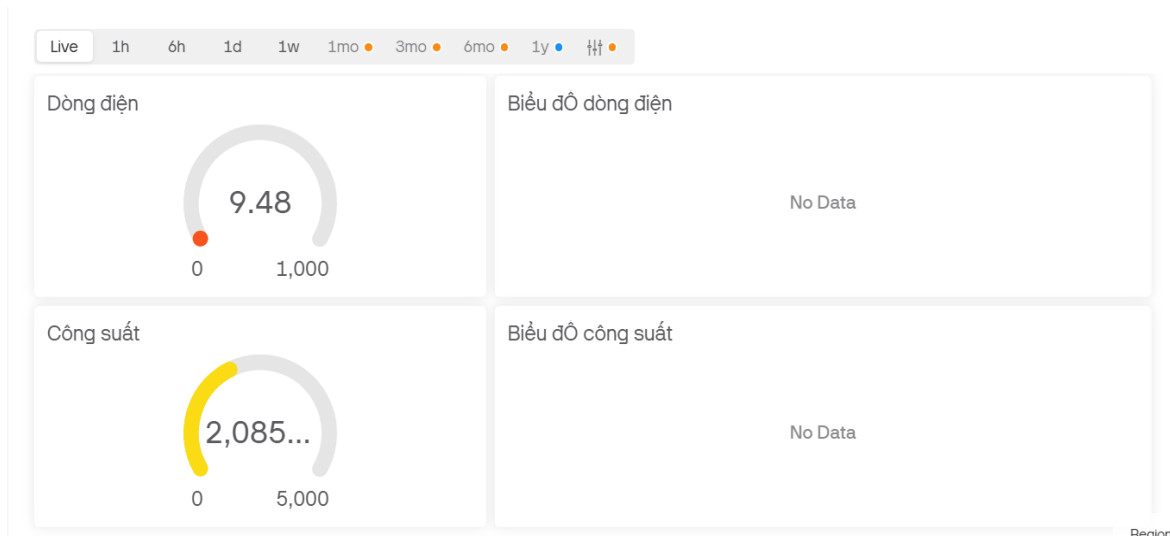
- Dây màu: **Đen**.
- **SIG (pot1) → GPIO34 (esp):**
 - Tín hiệu analog từ potentiometer được gửi đến chân GPIO34 (ADC1_CH6) của ESP32 để đo giá trị dòng điện.
 - Dây màu: **Xanh lá**.

3.1.1.2.2 LCD1602 (lcd1) ↔ ESP32 (esp):

- **VCC (lcd1) → 3V3 (esp):**
 - Cung cấp nguồn 3.3V từ ESP32 để cấp điện cho LCD.
 - Dây màu: **Đỏ**.
- **GND (lcd1) → GND.1 (esp):**
 - Nối đất chung giữa LCD và ESP32.
 - Dây màu: **Đen**.
- **SDA (lcd1) → GPIO21 (esp):**
 - Dữ liệu I2C từ ESP32 đến LCD qua chân SDA (GPIO21 mặc định cho I2C trên ESP32).
 - Dây màu: **Xanh lá**.
- **SCL (lcd1) → GPIO22 (esp):**
 - Tín hiệu đồng hồ I2C từ ESP32 đến LCD qua chân SCL (GPIO22 mặc định cho I2C trên ESP32).
 - Dây màu: **Xanh lá**.
- **Tổng kết:**
 - **Potentiometer:** Được nối với ESP32 qua 3 chân:
 - VCC (nguồn), GND (đất), và SIG (tín hiệu) vào GPIO34 để đo giá trị analog.
 - **LCD1602:** Giao tiếp với ESP32 qua giao thức I2C:
 - SDA (dữ liệu) và SCL (đồng hồ) lần lượt nối vào GPIO21 và GPIO22.
 - Nguồn và đất được cấp từ 3V3 và GND của ESP32.

3.1.2 SƠ ĐỒ TRÊN BLYNK:

3.1.2.1 MINH HOẠ TRÊN BLYNK:



Hình 3.2: Hình ảnh minh họa trên Blynk Web

- Bao gồm 4 khung:
 - Khung biểu thị dòng điện (màu đỏ): để hiển thị giá trị dòng điện hiện tại (Min: 0, Max: 1000).
 - Khung hiển thị giá trị công suất (Màu vàng): để hiển thị giá trị công suất hiện tại (Min: 0, Max: 5000).
 - Khung thống kê giá trị dòng điện theo thời gian thực (Biểu đồ dòng điện).
 - Khung thống kê giá trị công suất theo thời gian thực (Biểu đồ công suất).

3.1.2.2 TẠO DATASTREAMS TRÊN BLYNK:

The screenshot shows the 'Datastreams' management interface in Blynk Web. On the left is a sidebar with navigation links: Home, Datastreams (selected), Web Dashboard, Automation Templates, Metadata, and Connection Lifecycle. The main content area has a header with a 'hien' widget icon and an 'Edit' button. Below the header is a search bar labeled 'Search datastream'. A table lists the configured datastreams.

Id	Name	Pin	Color	Data Type	Units	Is Raw	Min	Max
1	V0	V0	Red	Double		false	0	1000
2	V1	V1	Yellow	Double		false	0	5000

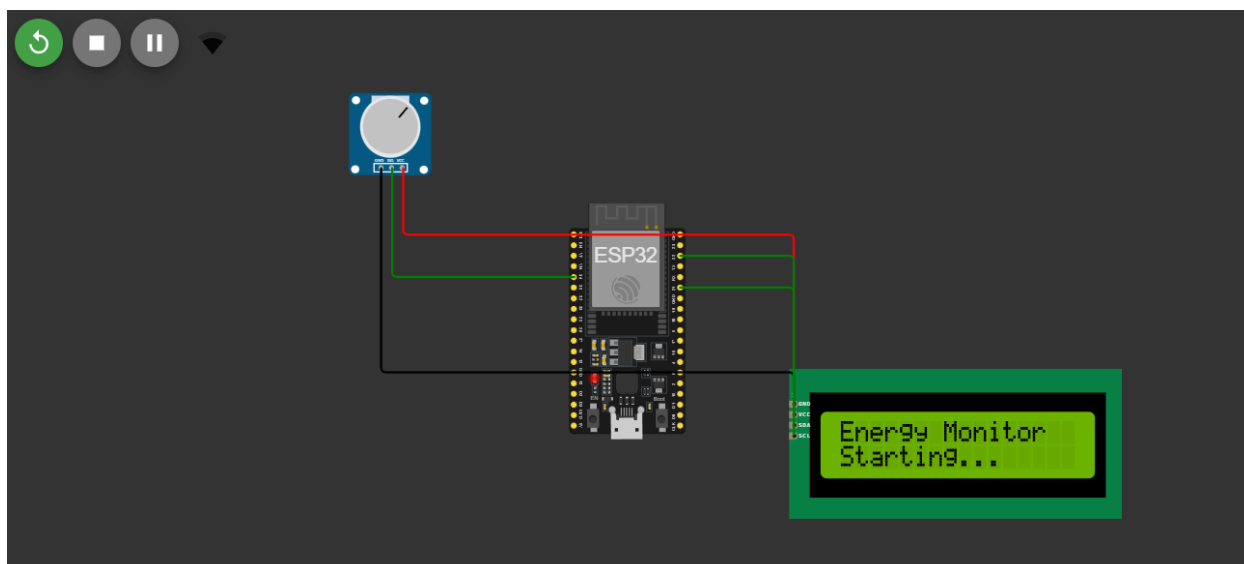
Hình 3.3: Hình ảnh tạo Datastreams trên Blynk Web

- Data gồm 2 đối tượng là V0 và V1:
 - V0: Dùng để hiển thị giá trị dòng điện
 - Type: Double
 - Min: 0
 - Max: 1000
 - V1: Dùng để hiển thị giá trị công suất
 - Type: Double
 - Min: 0
 - Max: 5000

3.1.3 NGUYÊN LÝ HOẠT ĐỘNG:

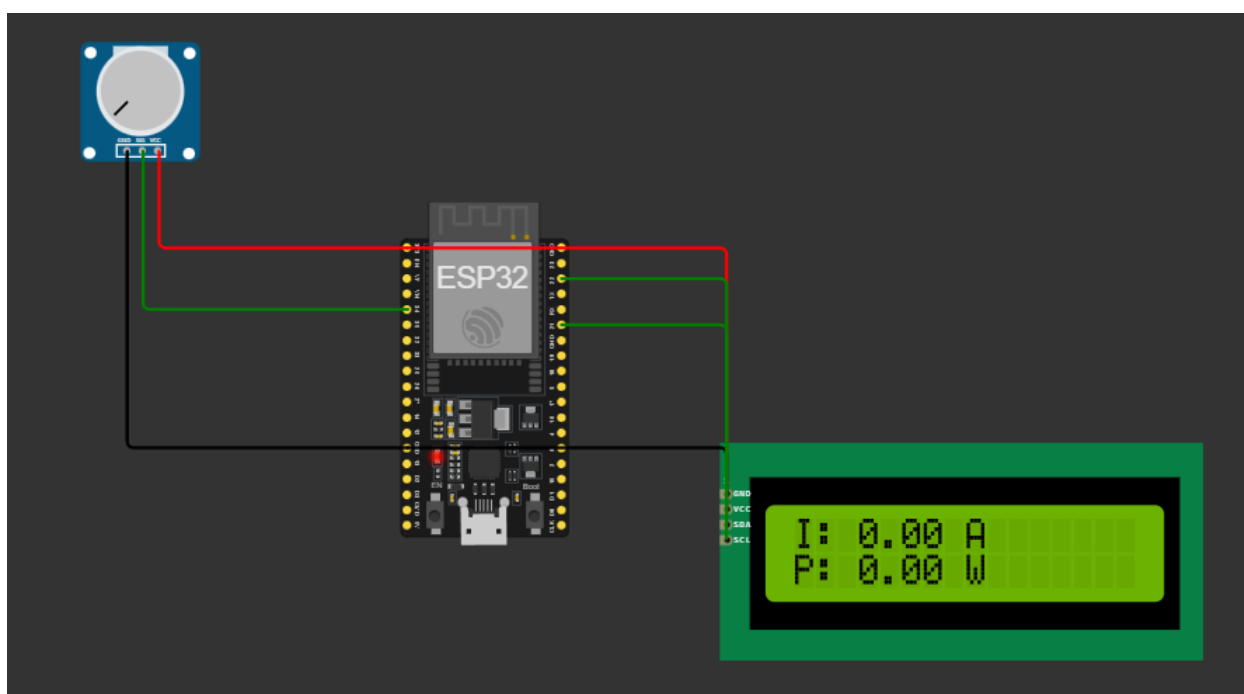
3.1.3.1 HOẠT ĐỘNG VỚI BLYNK:

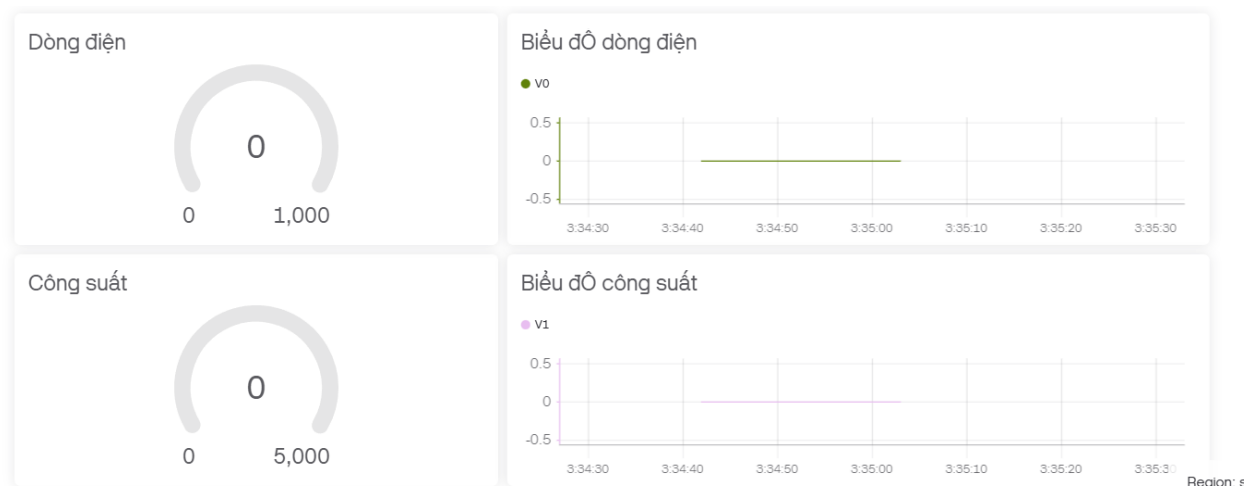
- Khi mới hoạt động màn hình sẽ hiển thị như **hình 3.4**



Hình 3.4: Hình ảnh minh họa Sơ đồ khối (Mới khởi động)

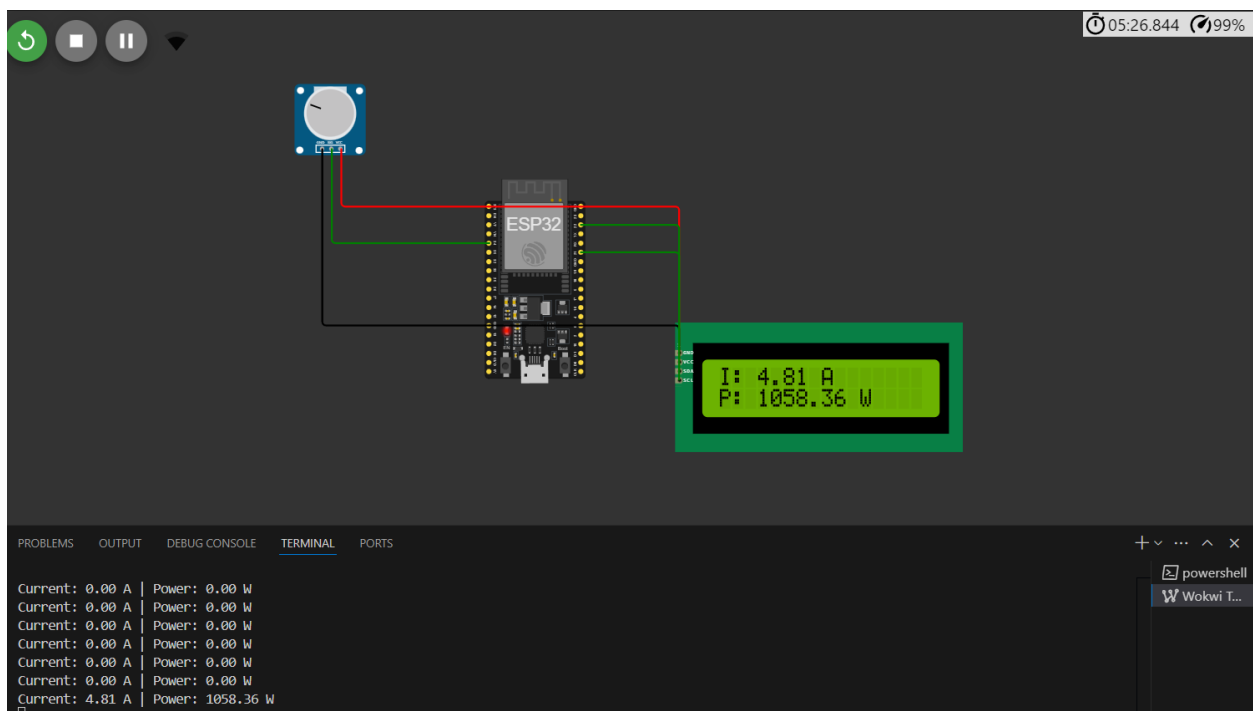
- Giá trị ban đầu khi không điều khiển **Potentiometer** trong đó I là giá trị dòng điện và p là giá trị công suất:

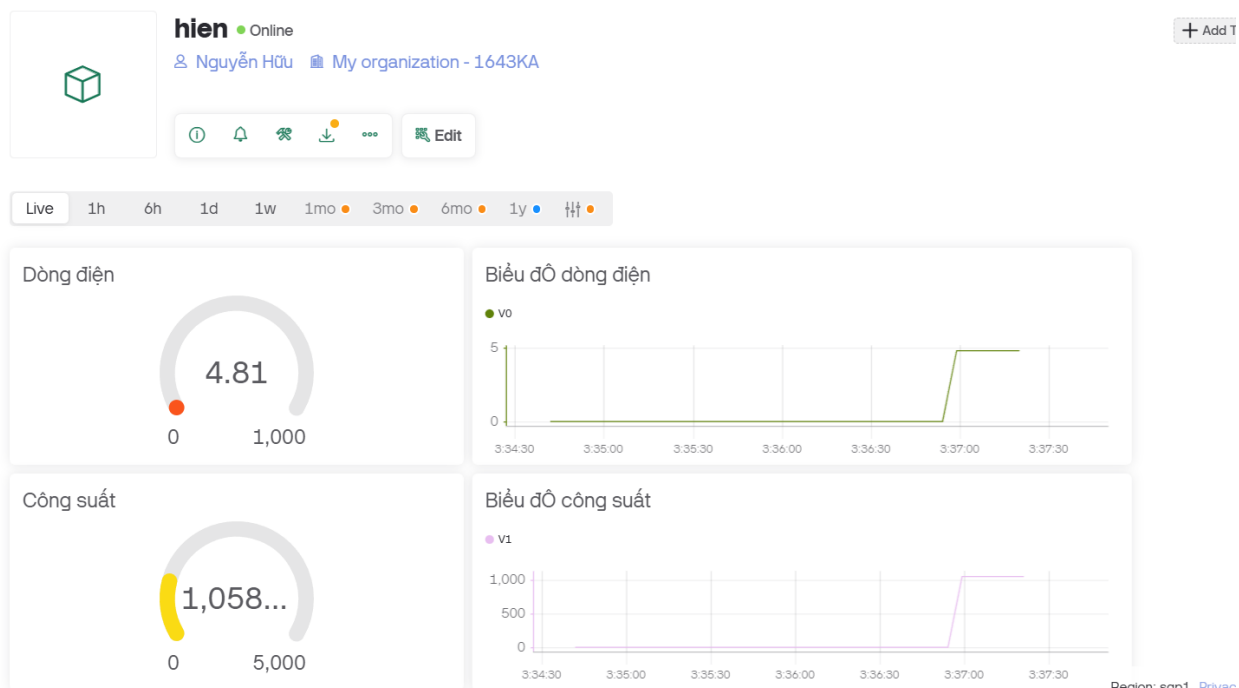




Hình 3.5: Hình ảnh Sơ đồ khối và Blynk khi chưa đổi giá trị

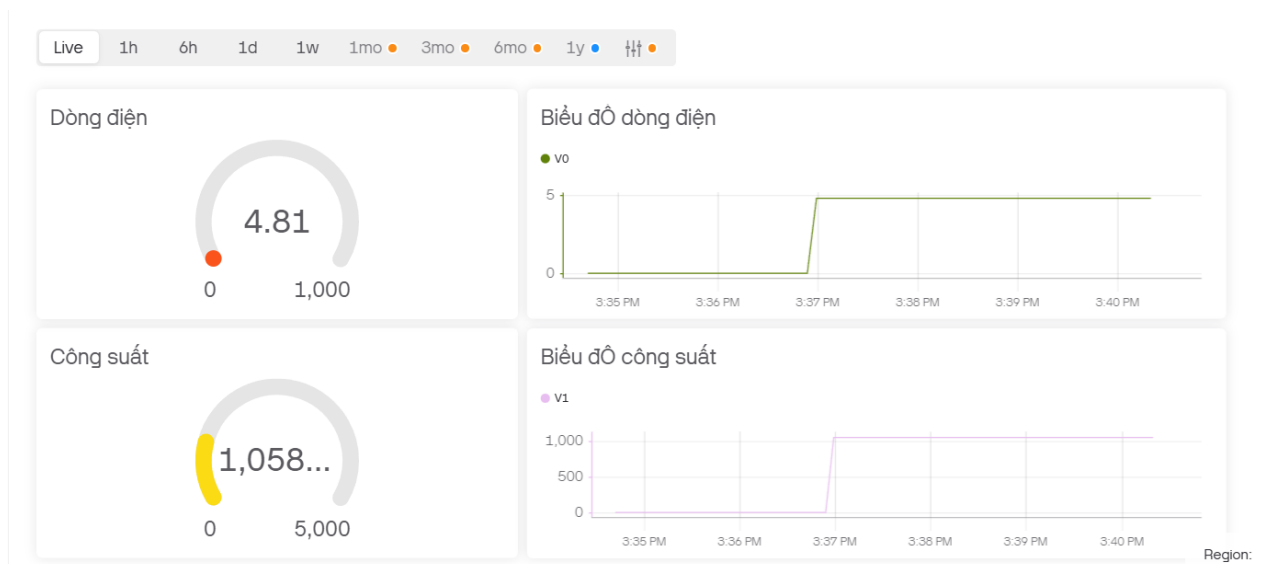
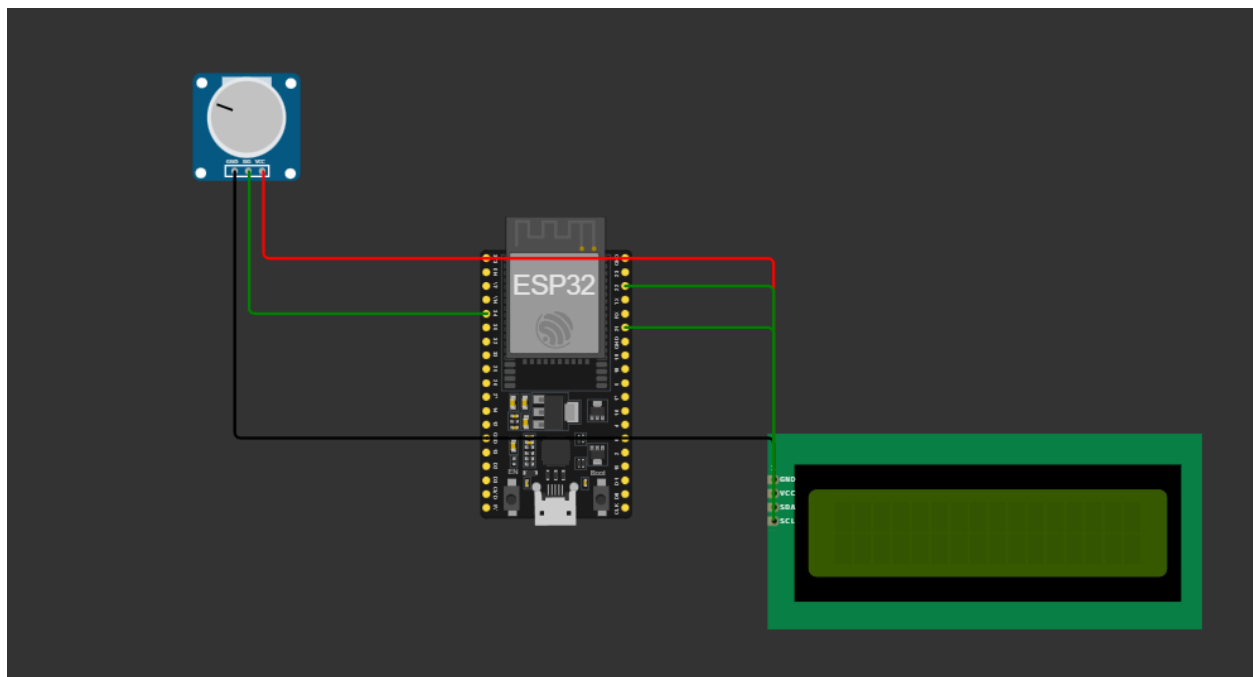
- Lúc này giá trị bên sơ đồ đang là I: 0 và P: 0 nên bên Blynk cũng hiển thị như vậy và biểu đồ cũng thống kê theo thời gian thực





Hình 3.6: Hình ảnh Sơ đồ khối và Blynk sau khi đổi giá trị

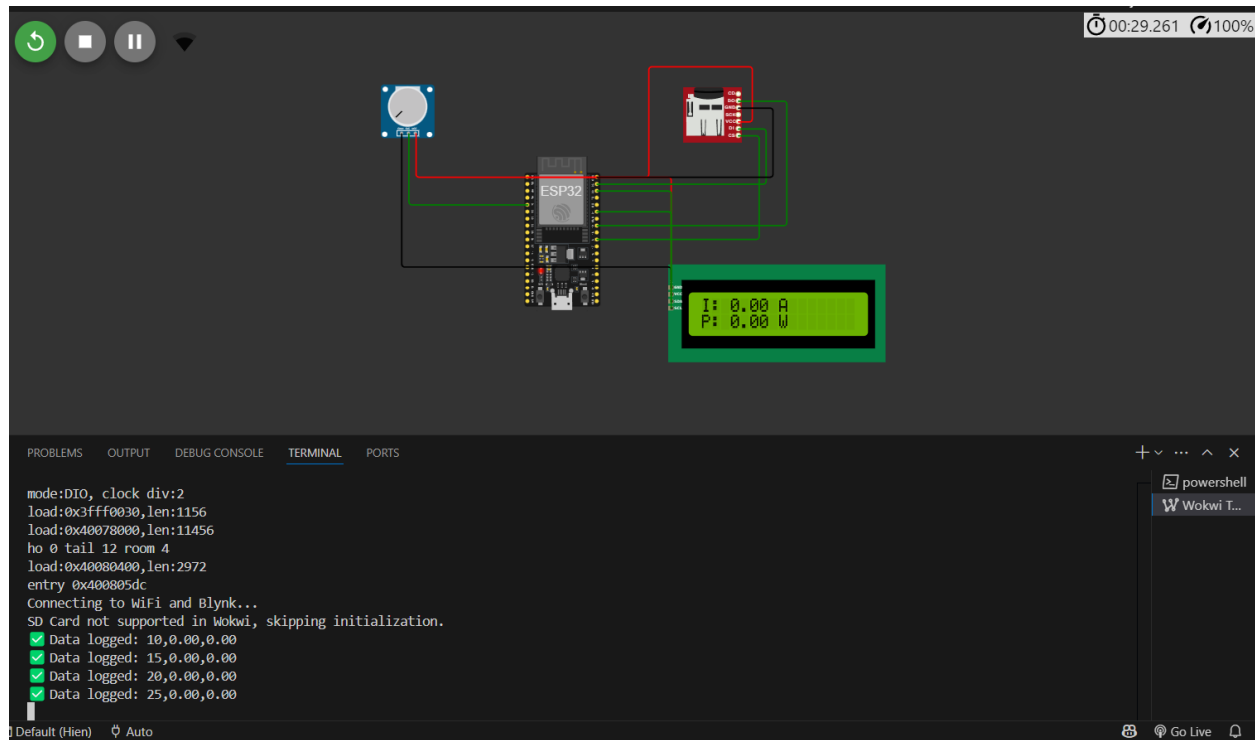
- Khi giá trị I và P được thay đổi như trong hình 3.6 thì lập tức giá trị dòng điện và công suất bên blynk cũng sẽ thay đổi theo và biểu đồ hoạt động cũng sẽ thay đổi theo thời gian thực, điều đó chứng tỏ giá trị đã được lưu vào Đám mây trên Blynk và sẽ thống kê giá trị liên tục



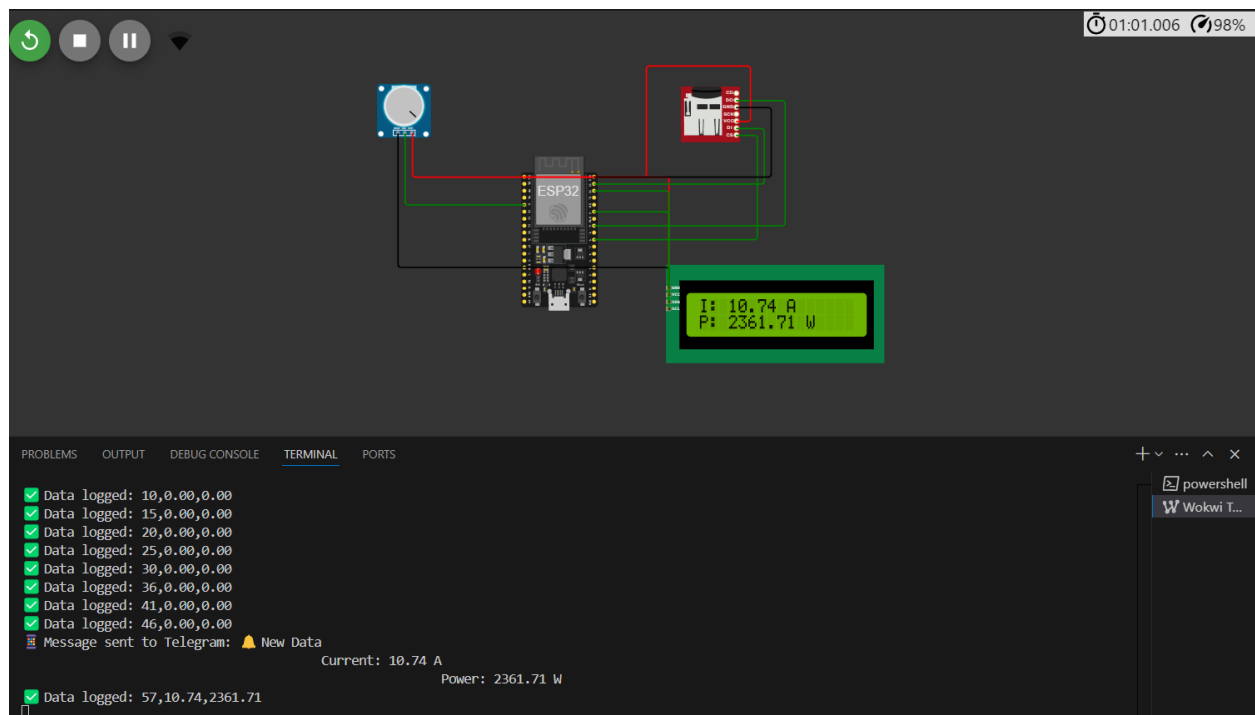
Hình 3.7: Hình ảnh Sơ đồ khối và Blynk sau khi đã tắt thiết bị

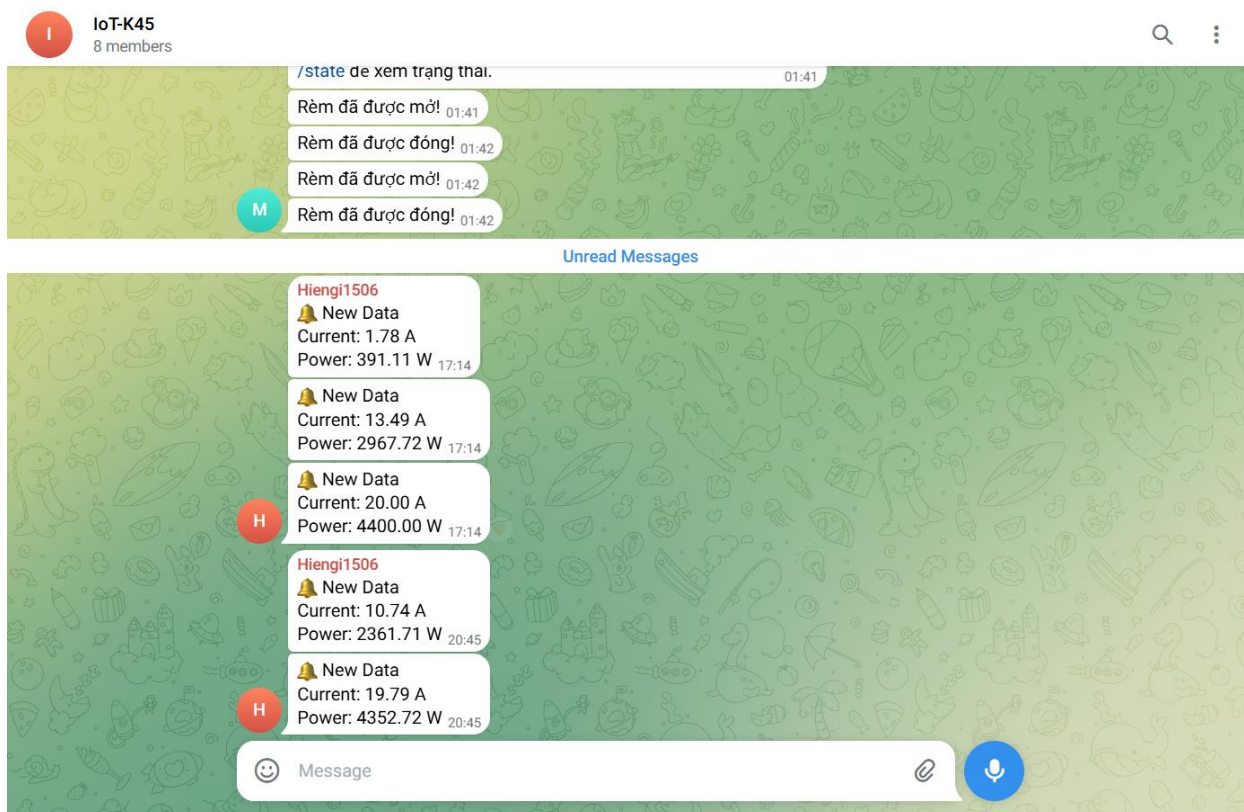
- Khi thiết bị đã tắt thì trên blynk vẫn lưu kết quả cũ vì các dữ liệu đã được lưu vào bộ toán đám mây trên Blynk

3.1.3.2 HOẠT ĐỘNG VỚI TELEGRAM:



Hình 3.8: Hình ảnh Sơ đồ khối khi chưa thay đổi giá trị



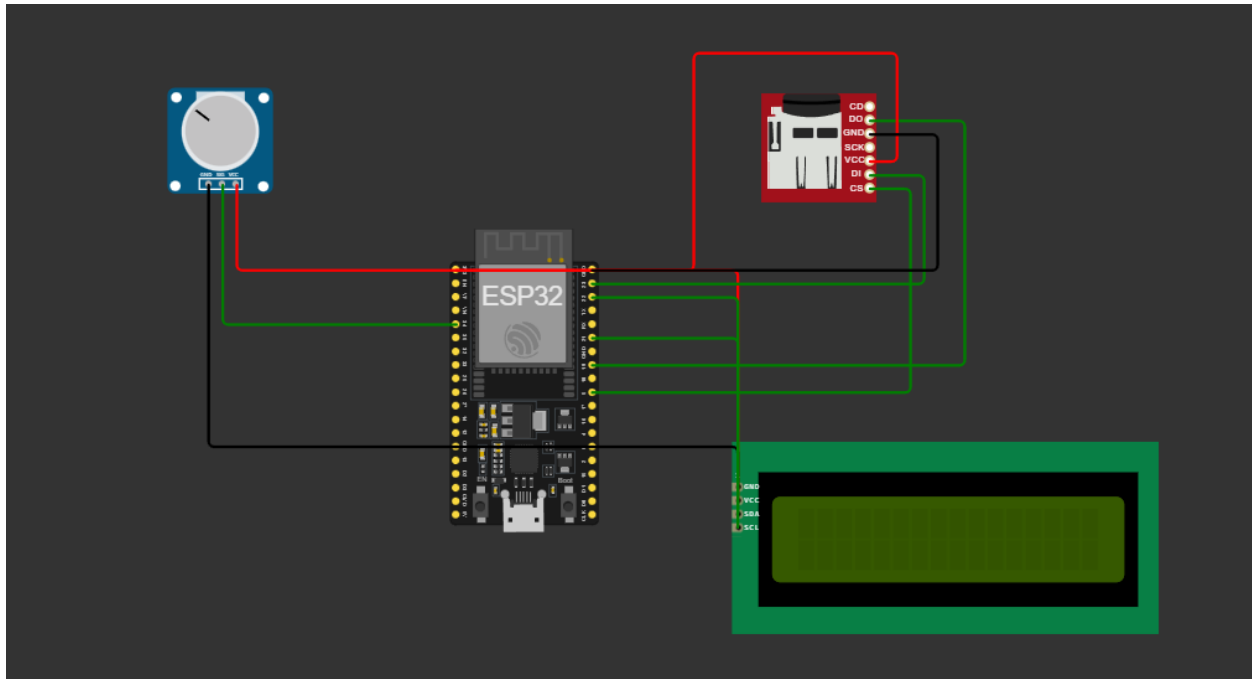


Hình 3.9: Hình ảnh Sơ đồ khối và telegram khi thay đổi giá trị

- Khi thiết bị thay đổi giá trị dòng điện và công suất thì lập tức ở telegram bot sẽ thông báo cho chúng ta giá trị dòng điện và công suất mới như hình 3.8.
- Còn nếu như giá trị không có gì thay đổi thì telegram bot sẽ không thông báo cho chúng ta.

3.2 SƠ ĐỒ KHỐI LƯU THÔNG TIN TRÊN SD

3.2.1 SƠ ĐỒ KHỐI:



Hình 3.10: Hình ảnh minh họa sơ đồ khối có SD trên wokwi

- Sơ đồ bao gồm:
 - **ESP32** làm bộ điều khiển chính.
 - **Potentiometer** (giả lập cảm biến dòng điện, như ACS712 trong thực tế).
 - **LCD1602 (I2C)** để hiển thị dữ liệu.
 - **Thẻ SD** để lưu trữ thông tin

3.2.1.1 MÔ TẢ HOẠT ĐỘNG CỦA THIẾT BỊ:

- **Potentiometer:**
 - Giả lập cảm biến dòng điện (ACS712 trong thực tế).
 - Tín hiệu analog từ chân SIG được gửi đến GPIO34 của ESP32.
- **ESP32:**
 - Đọc tín hiệu analog từ GPIO34, chuyển đổi thành giá trị dòng điện.

- Tính công suất tiêu thụ dựa trên điện áp cố định (220V).
- Gửi dữ liệu đến LCD qua giao thức I2C (SDA: GPIO21, SCL: GPIO22).
- In dữ liệu ra Serial Monitor để giám sát.
- **LCD1602:**
 - Hiển thị giá trị dòng điện (A) và công suất (W) theo thời gian thực.
- **Thẻ SD:**
 - Lưu trữ thông tin khi mà biến áp thay đổi chỉ số với form như sau: Time | Current(A) | Power(W)

3.2.1.2 SƠ ĐỒ KẾT NỐI:

3.1.1.2.1 Potentiometer (pot1) ↔ ESP32 (esp)

- **VCC (pot1) → 3V3 (esp):**
 - Cung cấp nguồn 3.3V từ ESP32 để cấp điện cho potentiometer.
 - Dây màu: **Đỏ**.
- **GND (pot1) → GND.1 (esp):**
 - Nối đất chung giữa potentiometer và ESP32.
 - Dây màu: **Đen**.
- **SIG (pot1) → GPIO34 (esp):**
 - Tín hiệu analog từ potentiometer được gửi đến chân GPIO34 (ADC1_CH6) của ESP32 để đo giá trị dòng điện.
 - Dây màu: **Xanh lá**.

3.1.1.2.2 LCD1602 (lcd1) ↔ ESP32 (esp):

- **VCC (lcd1) → 3V3 (esp):**
 - Cung cấp nguồn 3.3V từ ESP32 để cấp điện cho LCD.
 - Dây màu: **Đỏ**.
- **GND (lcd1) → GND.1 (esp):**
 - Nối đất chung giữa LCD và ESP32.

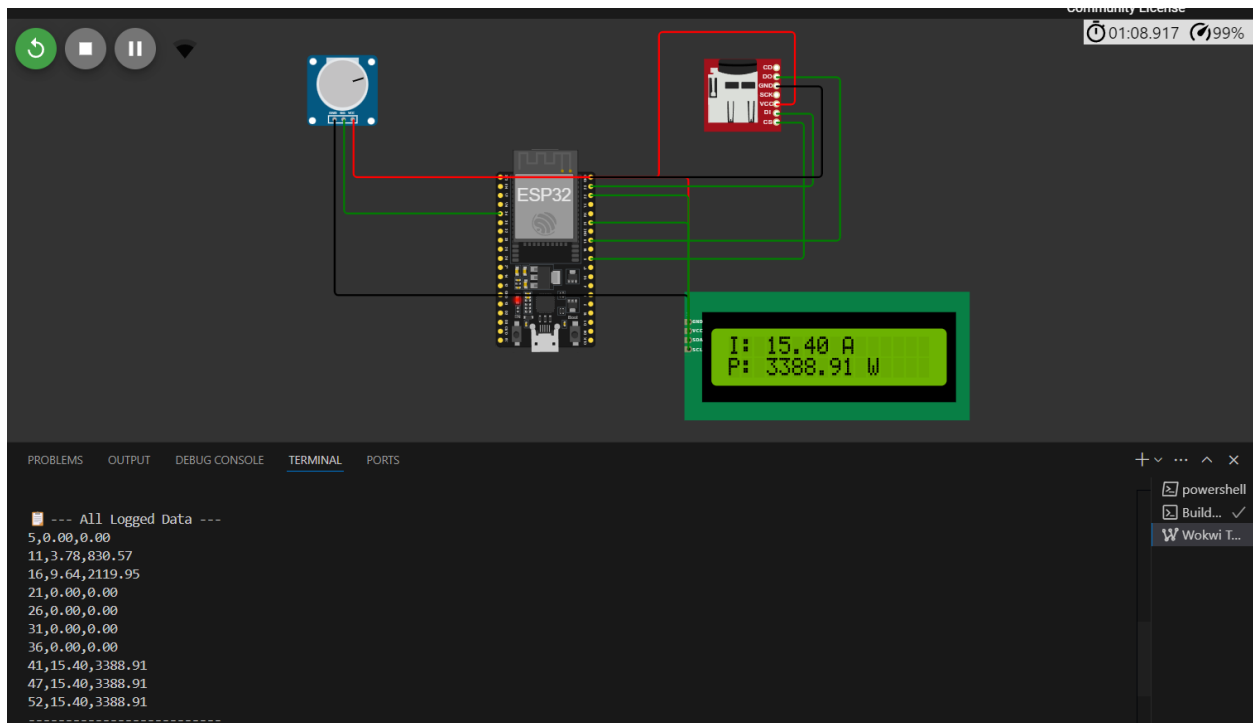
- Dây màu: **Đen**.
- **SDA (lcd1) → GPIO21 (esp):**
 - Dữ liệu I2C từ ESP32 đến LCD qua chân SDA (GPIO21 mặc định cho I2C trên ESP32).
 - Dây màu: **Xanh lá**.
- **SCL (lcd1) → GPIO22 (esp):**
 - Tín hiệu đồng hồ I2C từ ESP32 đến LCD qua chân SCL (GPIO22 mặc định cho I2C trên ESP32).
 - Dây màu: **Xanh lá**.

3.1.1.2.3 MicroSD Card (sd2) ↔ ESP32 (esp):

- **CS (sd2) → GPIO5 (esp):**
 - Dây CS (chip select) của thẻ SD được kết nối với GPIO5 của ESP32.
 - Dây màu: **Xanh lá**.
- **DI (sd2) → GPIO23 (esp):**
 - Dây DI (data input) của thẻ SD kết nối với GPIO23 của ESP32.
 - Dây màu: **Xanh lá**.
- **DO (sd2) → GPIO19 (esp):**
 - Dây DO (data output) của thẻ SD kết nối với GPIO19 của ESP32.
 - Dây màu: **Xanh lá**.
- **CLK (sd2) → GPIO18 (esp):**
 - Dây CLK (clock) của thẻ SD kết nối với GPIO18 của ESP32.
 - Dây màu: **Xanh lá**.
- **VCC (sd2) → 3V3 (esp):**
 - Cung cấp nguồn 3.3V từ ESP32 cho thẻ SD.
 - Dây màu: **Đỏ**.
- **GND (sd2) → GND.2 (esp):**

- Nối đất chung giữa thẻ SD và ESP32.
- Dây màu: **Đen**.
- **Tổng kết:**
 - **Potentiometer:** Được nối với ESP32 qua 3 chân:
 - VCC (nguồn), GND (đất), và SIG (tín hiệu) vào GPIO34 để đo giá trị analog.
 - **LCD1602:** Giao tiếp với ESP32 qua giao thức I2C:
 - SDA (dữ liệu) và SCL (đồng hồ) lần lượt nối vào GPIO21 và GPIO22.
 - Nguồn và đất được cấp từ 3V3 và GND của ESP32.
 - **MicroSD Card:** Nối với ESP32 qua các chân:
 - CS (Chip Select) → GPIO5 (ESP32).
 - DI (Data Input) → GPIO23 (ESP32).
 - DO (Data Output) → GPIO19 (ESP32).
 - CLK (Clock) → GPIO18 (ESP32).
 - VCC (Nguồn) → 3V3 (ESP32).
 - GND (Đất) → GND.2 (ESP32).

3.2.2 NGUYÊN LÝ HOẠT ĐỘNG:



Hình 3.11: Hình ảnh minh họa hoạt động của SD trên wokwi

- Khi hoạt động, với mỗi giá trị thay đổi sẽ lưu vào thẻ nhớ SD theo form sau : TimeHoatDong | A | W, và từ thẻ nhớ SD sẽ lưu vào file .txt trong máy tính.
- Cứ mỗi 10 dữ liệu được lưu vào thì sẽ hiển thị ra Serial Monitor danh sách của 10 giá trị thay đổi như trên hình 3.8

F. KẾT LUẬN/NHẬN XÉT/ĐÁNH GIÁ/KIẾN NGHỊ:

Chương 1. KẾT LUẬN:

- Qua quá trình nghiên cứu, tìm hiểu lý thuyết và triển khai thực tiễn đề tài “Giám sát năng lượng tiêu thụ với ESP32”, tôi đã đạt được những kết quả tích cực và thiết thực. Với mục tiêu chính là xây dựng một hệ thống giám sát điện năng tiêu thụ bằng cách sử

dụng vi điều khiển ESP32 kết hợp với cảm biến dòng điện ACS712, tôi đã thành công trong việc đo đặc công suất tiêu thụ của thiết bị điện, ghi nhận dữ liệu theo thời gian thực và lưu trữ chúng lên thẻ nhớ SD hoặc đẩy lên các nền tảng đám mây Blynk và bên cạnh đó tôi còn sử dụng được telegram để hiển thị dữ liệu mỗi khi giá trị thay đổi.

- Hệ thống hoạt động ổn định, cho phép người dùng theo dõi biến động năng lượng một cách chính xác và liên tục trên đa nền tảng nói chung và blynk, telegram nói riêng. Đây là một bước tiến quan trọng trong việc hướng đến các giải pháp tiết kiệm năng lượng và xây dựng các mô hình nhà hiện đại ngày nay. Trong bối cảnh nhu cầu sử dụng điện ngày càng tăng, đặc biệt trong các đô thị lớn, việc có một hệ thống giám sát đơn giản, dễ triển khai nhưng hiệu quả như vậy là điều vô cùng cần thiết và có tính ứng dụng cao.

Chương 2. NHẬN XÉT:

- Thông qua đề tài này, tôi nhận thấy rằng việc ứng dụng các công nghệ vi điều khiển và cảm biến trong lĩnh vực IoT ngày nay đã trở nên dễ tiếp cận hơn rất nhiều so với trước kia. ESP32 là một nền tảng mã nguồn mở mạnh mẽ, tích hợp sẵn WiFi và Bluetooth, rất phù hợp cho các dự án IoT có yêu cầu kết nối không dây. Cảm biến dòng ACS712 tuy có độ chính xác chưa cao như các dòng cảm biến chuyên dụng, nhưng hoàn toàn đáp ứng được nhu cầu giám sát cơ bản trong các ứng dụng dân dụng hoặc bán chuyên nghiệp.

- Ngoài ra, việc lưu trữ dữ liệu lên thẻ nhớ SD giúp đảm bảo an toàn thông tin trong những trường hợp mất kết nối Internet, trong khi việc tích hợp kết nối với các nền tảng đám mây (như Firebase, Google Sheets, hoặc MQTT broker) mở ra cơ hội cho các mô hình phân tích dữ liệu từ xa, theo thời gian thực.

Chương 3. ĐÁNH GIÁ:

- Từ kết quả đạt được, tôi đánh giá rằng hệ thống giám sát năng lượng tiêu thụ sử dụng ESP32 là một giải pháp có tính khả thi cao, vừa tiết kiệm chi phí, vừa có tính linh hoạt trong việc mở rộng và tích hợp với các hệ thống khác. Với chi phí thấp, linh kiện phổ biến, dễ mua và dễ lắp đặt, hệ thống hoàn toàn có thể được áp dụng trong thực tế – đặc biệt là trong các hộ gia đình, văn phòng nhỏ hoặc các xưởng sản xuất quy mô nhỏ.
- Tuy nhiên, tôi cũng nhận thấy một số hạn chế nhất định như: sai số khi đo dòng điện ở mức thấp, độ nhiễu tín hiệu trong môi trường điện từ cao, và khả năng xử lý dữ liệu chưa thực sự tối ưu nếu lượng thiết bị giám sát tăng lên đáng kể. Những hạn chế này cần được tiếp tục nghiên cứu và cải tiến trong các giai đoạn phát triển tiếp theo.

Chương 4. KIẾN NGHỊ:

- Từ những trải nghiệm mô phỏng trong quá trình thực hiện đề tài thông qua wokwi cùng một số ứng dụng khác, tôi xin đưa ra một

số kiến nghị nhằm hoàn thiện hơn nữa hệ thống và mở rộng hướng phát triển trong tương lai:

- **Nâng cao độ chính xác của hệ thống đo lường:** Nên thay thế hoặc kết hợp ACS712 với các cảm biến có độ nhạy và độ chính xác cao hơn như SCT-013 hoặc các cảm biến dòng kiểu Hall để phục vụ cho các ứng dụng chuyên sâu hơn.
- **Tăng cường bảo mật và đồng bộ hóa dữ liệu:** Khi triển khai lưu trữ dữ liệu lên đám mây, cần có cơ chế mã hóa và xác thực để đảm bảo an toàn thông tin, đồng thời lựa chọn nền tảng đám mây có độ tin cậy và tính ổn định cao.
- **Phát triển giao diện hiển thị trực quan:** Việc xây dựng một ứng dụng web hoặc mobile có giao diện thân thiện sẽ giúp người dùng dễ dàng theo dõi và phân tích dữ liệu tiêu thụ điện của từng thiết bị theo thời gian.
- **Tích hợp chức năng cảnh báo sớm:** Có thể bổ sung chức năng cảnh báo (qua email, SMS hoặc app notification) khi mức tiêu thụ vượt ngưỡng cho phép, giúp người dùng kiểm soát và điều chỉnh kịp thời nhằm tiết kiệm điện năng và đảm bảo an toàn.
- **Ứng dụng AI vào dự báo tiêu thụ năng lượng:** Trong tương lai, tôi kỳ vọng có thể tích hợp thêm các mô hình học máy (machine learning) để dự đoán xu hướng tiêu thụ điện dựa trên hành vi sử dụng thực tế,

từ đó tối ưu hóa chi phí và nâng cao hiệu suất năng lượng.

Chương 5. TỔNG KẾT:

- **Tổng kết lại**, đề tài không chỉ giúp tôi hiểu rõ hơn về cách ứng dụng vi điều khiển ESP32 và cảm biến dòng điện trong thực tiễn, mà còn rèn luyện kỹ năng triển khai một hệ thống IoT hoàn chỉnh – từ mô phỏng, từ việc xử lý dữ liệu đến tư duy mở rộng chức năng. Đây là tiền đề quý báu cho những nghiên cứu và phát triển sâu hơn trong tương lai về lĩnh vực giám sát, tự động hóa và tiết kiệm năng lượng trong thời đại công nghệ số.

Tài liệu tham khảo

- [1] **Ứng dụng IoT giám sát mực tiêu tiêu thụ điện nước.** Truy cập từ: <https://fr.slideshare.net/slideshow/luan-van-ung-dung-iot-giam-sat-muc-tieu-thu-dien-nuoc-hay/206575047>.
- [2] **Giới thiệu về IoT trong giám sát năng lượng.** Truy cập từ: <https://cps.hau.edu.vn/media/29/uffile-upload-no-title29800.pdf>.
- [3] **Sở Khoa học và Công nghệ TP. Cần Thơ.** Truy cập từ: <https://sokhcn.cantho.gov.vn/default.aspx?pid=57&nid=26614>.
- [4] **Blynk IoT và ESP32.** Truy cập từ: <https://dienthongminhesmart.com/lap-trinh-esp32/blynk-iot-va-esp32/>.