|  |  |
| --- | --- |
|  | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI** |
|  | **---------------------------------------**  ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG  **THIẾT KẾ HỆ THỐNG GIÁM SÁT THIẾT BỊ TIÊU THỤ ĐIỆN TRONG HỘ GIA ĐÌNH** |
| **CBHD:**  **Sinh viên:**  **Mã số sinh viên:** |
| NGÀNH CNKT ĐIỆN TVIỄTHÔNG THÔNG | Hà Nội - 2024 |

# LỜI CẢM ƠN

Trong thời gian thực hiện đề tài đồ án tốt nghiệp, nhóm em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của các thầy cô bộ môn Điện Tử Viễn Thông cũng như thầy cô trong khoa Điện Tử trường Đại Học Công Nghiệp Hà Nội. Đồng thời chúng em đã được tiếp cận các trang thiết bị hiện đại của khoa để phục vụ vào mục đích nghiên cứu, học tập.

Nhóm em xin gửi lời cảm ơn chân thành đến TS……………, người đã tận tình hướng dẫn, chỉ dạy, giúp đỡ và cung cấp những tài liệu cũng như kinh nghiệm quý báu giúp em hoàn thành các nhiệm vụ được giao trong quá trình thực hiện.

Nhóm em cũng xin cảm ơn các thầy cô trong trường Đại học Công nghiệp Hà Nội nói chung, các thầy cô trong khoa Điện Tử nói riêng đã chỉ dạy những kiến thức quý báu, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên trong suốt quá trình học tập và hoàn thành đề tài đồ án tốt nghiệp.

Hà Nội, Ngày… tháng… năm 2024

Nhóm sinh viên thực hiện

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc164539416)

[DANH MỤC HÌNH VẼ iv](#_Toc164539417)

[DANH MỤC TỪ VIẾT TẮT vi](#_Toc164539418)

[LỜI MỞ ĐẦU 1](#_Toc164539419)

[Lý do chọn đề tài 1](#_Toc164539420)

[Mục đích, đối tượng và phạm vi nghiên cứu 2](#_Toc164539421)

[Ý nghĩa khoa học và thực tiễn của đề tài 2](#_Toc164539422)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN, ĐIỆN ÁP AC 3](#_Toc164539423)

[1.1 Đặt vấn đề 3](#_Toc164539424)

[1.2 Tổng quan về đề tài 3](#_Toc164539425)

[1.3 Một số dụng cụ đo dòng điện và điện áp phổ biến 4](#_Toc164539426)

[1.4 Kết luận chương 1 7](#_Toc164539427)

[CHƯƠNG 2: Các Công nghệ SỬ DỤNG TRONG HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN ĐIỆN ÁP AC 8](#_Toc164539428)

[2.1 Các chuẩn truyền thông được sử dụng 8](#_Toc164539429)

[2.1.1 Modbus 8](#_Toc164539430)

[2.1.2 Giao thức I2C 10](#_Toc164539431)

[2.1.3 Giao thức HTTP 16](#_Toc164539432)

[2.1.4 Giao thức MQTT 17](#_Toc164539433)

[2.2 Phần cứng của hệ thống 19](#_Toc164539434)

[2.2.1 Vi điều khiển ESP8266 19](#_Toc164539435)

[2.2.2 Cảm biến PZEM – 004T 21](#_Toc164539436)

[2.2.3 Màn hình LCD 16x2 24](#_Toc164539437)

[2.2.4 Module chuyển đổi giao tiếp I2C cho LCD 25](#_Toc164539438)

[2.3 Phần mềm thiết kế và lập trình 27](#_Toc164539439)

[2.3.1 Phần mềm thiết kế Altium Designer 27](#_Toc164539440)

[2.3.2 Phần mềm Visual Studio Code 28](#_Toc164539441)

[2.3.3 Platform IO 29](#_Toc164539442)

[2.3.4 Phần mềm Blynk 30](#_Toc164539443)

[2.4 Kết luận chương 2 32](#_Toc164539444)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN, ĐIỆN ÁP AC 33](#_Toc164539445)

[3.1 Sơ đồ khối của hệ thống 33](#_Toc164539446)

[3.2 Sơ đồ nguyên lý toàn mạch 34](#_Toc164539447)

[3.2.1 Khối nguồn và ổn áp nguồn 34](#_Toc164539448)

[3.2.2 Khối xử lý trung tâm 35](#_Toc164539449)

[3.2.3 Khối cảm biến 36](#_Toc164539450)

[3.2.4 Khối hiển thị 37](#_Toc164539451)

[3.3 Sơ đồ mạch in 37](#_Toc164539452)

[3.4 Thiết kế phần mềm 40](#_Toc164539453)

[3.5 Mô hình thực tế 41](#_Toc164539454)

[KẾT LUẬN 44](#_Toc164539455)

[TÀI LIỆU THAM KHẢO 45](#_Toc164539456)

[PHỤ LỤC 46](#_Toc164539457)

# DANH MỤC HÌNH VẼ

[Hình 1: Đồng hồ vạn năng 4](#_Toc164539490)

[Hình 2: Đồng hồ VAF36A 5](#_Toc164539491)

[Hình 3: Đồng hồ đo SELEC MA2301 6](#_Toc164539492)

[Hình 4: Giao thức Modbus 8](#_Toc164539493)

[Hình 5: Mô hình giao tiếp I2C 10](#_Toc164539494)

[Hình 6: Cách truyền dữ liệu I2C 11](#_Toc164539495)

[Hình 7: Mô hình một Master nhiều Slave 14](#_Toc164539496)

[Hình 8: Mô hình nhiều Master nhiều Slave 14](#_Toc164539497)

[Hình 9: Mô hình OSI và TCP/IP 16](#_Toc164539498)

[Hình 10: Mô hình Client – Server 16](#_Toc164539499)

[Hình 11: Mô hình khái quát của MQTT 18](#_Toc164539500)

[Hình 12: Các ngoại vi trên ESP8266 20](#_Toc164539501)

[Hình 13: Cảm biến PZEM-004T 21](#_Toc164539502)

[Hình 14: Sơ đồ đấu nối của cảm biến PZEM-004T 23](#_Toc164539503)

[Hình 15: Màn hình LCD16x2 24](#_Toc164539504)

[Hình 16: Module I2C cho LCD 26](#_Toc164539505)

[Hình 17: Một cửa sổ của Altium Designer 27](#_Toc164539506)

[Hình 18: Visual Studio Code 28](#_Toc164539507)

[Hình 19: Platform IO 29](#_Toc164539508)

[Hình 20: Phần mềm Blynk 30](#_Toc164539509)

[Hình 21: Sơ đồ khối của hệ thống 33](#_Toc164539510)

[Hình 22: Sơ đồ nguyên lý toàn mạch 34](#_Toc164539511)

[Hình 23: Khối nguồn và adapter 5V 34](#_Toc164539512)

[Hình 24: Khối xử lý trung tâm 35](#_Toc164539513)

[Hình 25: Khối cảm biến 36](#_Toc164539514)

[Hình 26: Khối hiển thị 37](#_Toc164539515)

[Hình 27: Sơ đồ mạch in 38](#_Toc164539516)

[Hình 28: Hình ảnh 3D của toàn mạch 39](#_Toc164539517)

[Hình 29: Lưu đồ thuật toán của hệ thống 40](#_Toc164539518)

[Hình 30: Mô hình thực tế của hệ thống 41](#_Toc164539519)

[Hình 31: Giao diện giám sát 42](#_Toc164539520)

[Hình 32: Tổng quan về hướng dẫn sử dụng 43](#_Toc164539521)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Tiếng anh | Tiếng việt |
| 1 | HTTP | Hyper Text Transfer Protocol | Giao thức truyền tải siêu văn bản |
| 2 | IC | Intergrated Circuit | Mạch tích hợp |
| 4 | DC | Direct Current | Dòng điện một chiều |
| 5 | LCD | Liquid Crystal Display | Màn hình tinh thể lỏng |
| 6 | IDE | Integrated Development Environment | Môi trường phát triển tích hợp |
| 7 | MCU | Multipoint Control Unit | Thiết bị điều khiển đa điểm |

# LỜI MỞ ĐẦU

## Lý do chọn đề tài

Lịch sử nhân loại đã trải qua nhiều cuộc cách mạng khoa học kĩ thuật, tinh thần tìm tòi sáng tạo giúp con người ngày càng có nhiều phát minh, sáng kiến, tìm ra những công cụ, phương tiện mới, con đường mới đế chinh phục tự nhiên. Ngày nay, chúng ta đang phấn đấu cho mục tiêu công nghiệp hóa - hiện đại hóa, rượt đuổi làn sóng các cuộc cách mạng khoa học, con người đã và đang tiếp cận với vi xử lý.

Trong những năm gần đây, công nghệ vi điện tử phát triển rất mạnh mẽ. Sự ra đời của các vi mạch cỡ lớn, cực lớn với giá thành giảm nhanh, khả năng lập trình ngày càng cao đã mang lại những thay đổi sâu sắc trong ngành kĩ thuật điện tử. Nền công nghiệp thế giới đã đạt được những thành tựu to lớn nhờ ứng dụng những tiến bộ của khoa học kĩ thuật và công nghệ, máy móc đã thay thế con người trong nhiều hoạt động lao động sản xuất.

Việc giám sát dòng điện và điện áp một chiều với vi điều khiển là phần việc không thể thiếu khi chế tạo những máy móc tự động. Trong một cỗ máy tự động, nếu phần cơ khí tạo nên hình dạng và một cơ cấu hoạt động linh hoạt thì phần lập trình và mạch điện tử như một bộ não điều khiển những hoạt động đó.

Để thực hiện công giám sát hiệu suất về dòng và áp của các máy hoạt động trong công nghiệp một cách khoa học và chính xác. Hệ thống đo và giám sát dòng điện và điện áp cho hệ thống là một trong các giải pháp nhà sản xuất đã và đang áp dụng vào dây truyền sản xuất nhằm đáp ứng được nhu cầu tự động hóa, giảm sức lao động cho người lao động và tiết kiệm được chi phí sản xuất.

Qua quá trình học tập và tìm hiểu một số môn học ở trên trường, chúng em đã quyết định chọn đề tài: **“Thiết kế hệ thống giám sát thiết bị tiêu thụ điện trong hộ gia đình”** là đề tài đồ án tốt nghiệp.

## Mục đích, đối tượng và phạm vi nghiên cứu

Mục đích nghiên cứu của đề tài nhằm giám sát dòng điện và điện áp xoay chiều (AC) của thiết bị để theo dõi, giám sát trạng thái của chúng. Từ các giá trị đó, người vận hành hay thao tác có thể đưa ra được các trường hợp, phân tích ra được các nguyên nhân lỗi

Đối tượng nghiên cứu của đề tài bao gồm: ESP8266, cảm biến dòng điện và điện áp AC PZEM-004T, màn hình LCD và lập trình ứng dụng sử dụng để theo dõi và giám sát.

Phạm vi nghiên cứu của đề tài là sử dụng cảm biến dòng điện điện áp PZEM-004T nhằm đo dòng điện và điện áp xoay chiều. Cảm biến giao tiếp với vi điều khiển ESP8266 thông qua chuẩn UART, dòng điện và điện áp hiển thị thông qua ứng dụng trên điện thoại.

## Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài tập trung vào giám sát và thống kê dòng điện và điện áp xoay chiều, dòng điện và điện áp xoay chiều là 2 thông số rất quan trọng trong hệ thống có sử dụng điện xoay chiều. Kết hợp giữa giám sát và hiển thị trên màn hình LCD. Hơn nữa đề tài còn có thể giám sát và đưa ra cảnh báo trong các hệ thống điện từ xa thông qua một giao diện web đơn giản và dễ dàng sử dụng. Vì vậy đề tài có ý nghĩa thực tiễn to lớn trong các nhà máy sản xuất cũng như các hệ thống sử dụng điện một chiều như: Hệ thống năng lượng mặt trời, hệ thống pin trên các loại xe …

# TỔNG QUAN VỀ HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN, ĐIỆN ÁP AC

## Đặt vấn đề

Trên thế giới hiện nay việc giám sát và thống kê dòng điện, điện áp của các thiết bị điện qua mạng internet không còn mới mẻ và xa lạ nữa. Xu hướng hiện nay là có thể giám sát và thống kê dòng điện, điện áp thiết bị thông qua phần mềm với giao diện được lập trình và được cài trên máy tính mà không cần đo đạc, thống kê một cách thủ công.

Ở phạm vi trong nước, vấn đề giám sát và thống kê dòng điện, điện áp các thiết bị điện từ xa luôn là tâm điểm của khoa học hiện nay. Với mong muốn ứng dụng khoa học thế giới vào cuộc sống, họ muốn cuộc sống trở nên công nghệ hơn, hiện đại hơn. Nhưng thực tế chưa có sản phẩm nào đáp ứng được nhiều vấn đề mà nhu cầu thực tế đặt ra và thường giá thành tới tay người tiêu dung để áp dụng và thực tế thường khá cao nên ít được mọi người chú ý tới.

Tóm lại, việc thiết kế sử dụng hệ thống giám sát và thống kê dòng điện, điện áp các thiết bị hiện nay ở Việt Nam đang còn mới mẻ và chưa đi vào thực tiễn ứng dụng nhiều. Điều này khiến nhóm em quyết định làm và đưa ra sản phẩm mang tính chất thực tế hơn và áp dụng được vào cuộc sống thực tiễn

## Tổng quan về đề tài

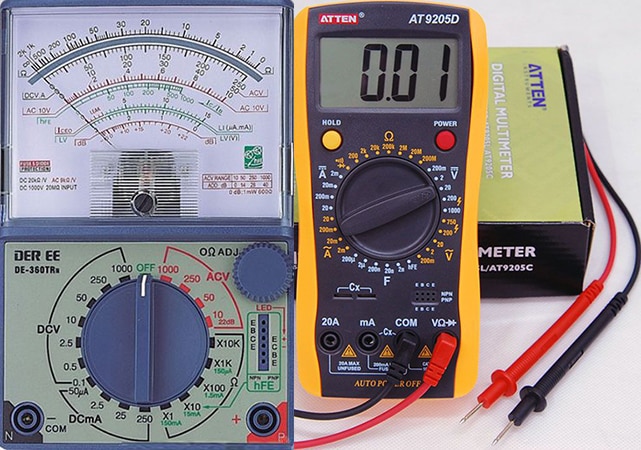
Trong đề tài này, nhóm em sẽ thực hiện thiết kế một mô hình sản phẩm hoàn thiện đảm bảo các chức năng giám sát và thống kê dòng điện, điện áp cho các thiết bị điện xoay chiều (AC). Hiện thị thông số các giá trị dòng điện và điện áp lên màn hình LCD và kiểm soát thông qua ứng dụng trên điện thoại.

Ngoài ra mô hình còn có các chức năng khác như: Báo quá áp hoặc quá dòng so với một ngưỡng đã đặt qua còi, có thể đặt ngưỡng để cắt hệ thống qua một relay với các ngưỡng về còi, về relay hoàn toàn có thể cài đặt từ xa thông qua webserver.

## Một số dụng cụ đo dòng điện và điện áp phổ biến

*Các loại đồng hồ vạn năng*

Đồng hồ vạn năng là một công cụ điện tử thường được sử dụng để đo điện áp và dòng điện trên các mạch. Ngoài ra tùy vào mục đích sử dụng, đồng hồ vạn năng còn có thể đo được các thông số khác như: Điện trở, điện dung, tần số …



Hình 1: Đồng hồ vạn năng

Đồng hồ vạn năng là một thiết bị đo với tốc độ nhanh và chính xác cao, vì vậy đồng hồ vạn năng thường được sử dụng như một dụng cụ không thể thiếu để hiệu chỉnh các thiết bị khác.

Trên thị trường hiện nay có 2 loại đồng hồ vạn năng chủ yếu là đồng hồ kim và đồng hồ dạng số điện tử.  Nếu nhu cầu sửa chữa mạch điện như kiểm tra diode, transistor, đo thông mạch, kiểm tra sự phóng nạp, dò, chập của tụ điện ... thì chỉ cần sử dụng loại đồng hồ vạn năng kim vì giá thành rẻ hơn. Nếu cần một chiếc đồng hồ đo điện đa năng thì nên chọn đồng hồ vạn năng số. Ưu điểm hiển thị nhiều kết quả đo đồng thời nhanh chóng và chính xác qua màn hình điện tử

*Đồng hồ đo VAF36A*



Hình 2: Đồng hồ VAF36A

Các thông số kỹ thuật của đồng hồ:

– Đồng hồ đo: Điện áp, dòng điện và tần số

– Hiển thị 3 hàng, 3 số, dạng LCD

– Cài đặt hệ số CT: 5A – 10.000A

– Mạng kết nối: 3 pha – 4 dây, 3 pha – 3 dây, 2 pha – 3 dây & 1 pha – 2 dây

– Cài đặt hệ số PT Sơ cấp: 100V – 500kV

– Cài đặt hệ số PT Thứ cấp: 100 – 500V AC (L-L)

– Đo được điện áp trung và cao thế

– Nguồn cấp: 230V AC ± 20%

*Đồng hồ đo SELEC MA2301*

**

Hình 3: Đồng hồ đo SELEC MA2301

*Các thông số kỹ thuật:*

* Kích thước 72x72mm.
* Đồng hồ đo Ampere 3 pha với công tắc chuyển mạch.
* Hiển thị 4 số, dạng LCD.
* Cài đặt hệ số CT: 5 – 5000A.
* Dải hiển thị: 0 – 6200A.
* Mạng kết nối: 3 pha 4 dây.
* Nguồn cấp: AC: 240V AC ±20%, (50 / 60Hz).
* Bảo vệ mặt trước: IP54.
* Chiều cao của LED hiển thị: 14.2 mm.
* Có biểu đồ hiển thị % tải.

## Kết luận chương 1

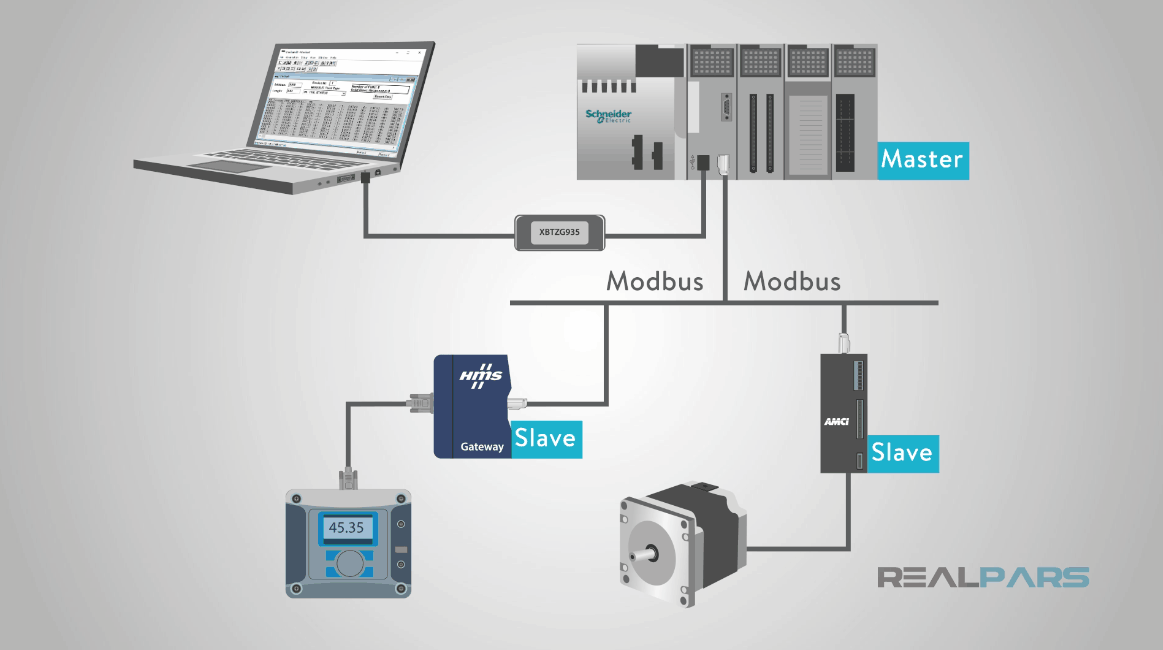
Chương 1 nói về việc sử dụng các thiết bị, các phần mềm và các công cụ đã dùng trong quá trình hoàn thành sản phẩm và các kiến trúc tổng quan phục vụ cho việc giải bài toán được đặt ra. Cùng với đó, chương 1 giới thiệu cũng như cách sử dụng các công cụ để lập trình và debug chương trình vi điều khiển ESP8266. Ngoài ra chương 2 cũng giới thiệu về các công cụ lập trình, giao diện hiển thị cho người dùng.

# Các Công nghệ SỬ DỤNG TRONG HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN ĐIỆN ÁP AC

## Các chuẩn truyền thông được sử dụng

### Modbus

Modbus là một giao thức truyền thông dữ liệu ban đầu được phát triển bởi Modicon (nay là Schneider Electric) vào năm 1979 để sử dụng với PLC (Programable Logic Controller). Modbus đã trở thành một giao thức truyền thông tiêu chuẩn thực tế và hiện là một phương tiện thường có sẵn để kết nối các thiết bị điện tử công nghiệp.

Modbus là phổ biến trong môi trường công nghiệp vì nó được công khai và miễn phí bản quyền. Nó được phát triển cho các ứng dụng công nghiệp vì tương đối dễ triển khai và duy trì so với các tiêu chuẩn khác và đặt ra một số hạn chế đối với định dạng của dữ liệu sẽ được truyền đi.

Hình 4: Giao thức Modbus

Việc phát triển và cập nhật các giao thức Modbus đã được quản lý bởi Tổ chức Modbus. Tổ chức Modbus là một hiệp hội của người dùng và nhà cung cấp các thiết bị tuân thủ Modbus mà ủng hộ việc tiếp tục sử dụng công nghệ.

Giao thức Modbus sử dụng giao tiếp nối tiếp, Ethernet hoặc giao thức Internet làm lớp vận chuyển (Transport Layer). Modbus hỗ trợ giao tiếp đến và từ nhiều thiết bị được kết nối với cùng một mạng cáp hoặc mạng Ethernet như: Có 2 thiết bị, một thiết bị đo nhiệt độ và một thiết bị khác để đo độ ẩm được kết nối với cùng một đường truyền, cả hai đều truyền dữ liệu của các phép đo đến cùng một máy tính thông qua Modbus.

Modbus thường được sử dụng để kết nối máy tính giám sát nhà máy/hệ thống với thiết bị đầu cuối từ xa (RTU) trong các hệ thống giám sát và thu thập dữ liệu (SCADA) [1].

*Các phiên bản hay được sử dụng của Modbus:*

Modbus RTU

Được sử dụng trong giao tiếp nối tiếp và là phiên bản phổ biến nhất của Modbus. Modbus RTU sử dụng một biểu diễn nhị phân, nhỏ gọn của dữ liệu cho giao tiếp giao thức. Định dạng RTU tuân theo các lệnh/dữ liệu với kiểm tra dự phòng theo chu kỳ làm cơ chế kiểm tra lỗi để đảm bảo độ tin cậy của dữ liệu. Một thông điệp Modbus RTU phải được truyền liên tục mà không bị ngắt quãng giữa các ký tự.

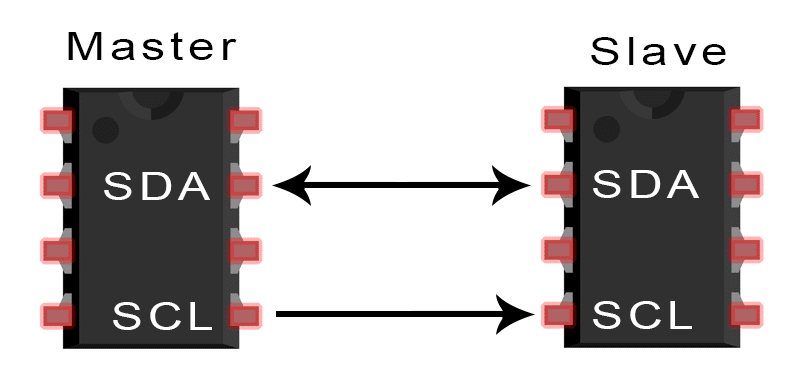
Modbus ASCII

Được sử dụng trong giao tiếp nối tiếp và sử dụng các ký tự ASCII để giao tiếp giao thức. Định dạng ASCII sử dụng kiểm tra kiểm tra dự phòng theo chiều dọc. Các tin nhắn Modbus ASCII được đóng khung bởi một ký tự (":") và theo dõi (CR/LF).

Modbus TCP/IP

Một biến thể Modbus được sử dụng để truyền tin qua mạng TCP/IP, kết nối qua cổng 502. Nó không yêu cầu tính toán tổng kiểm tra, vì các lớp thấp hơn đã cung cấp bảo vệ tổng kiểm tra.

### Giao thức I2C



Hình 5: Mô hình giao tiếp I2C

*Giới thiệu về giao tiếp I2C*

Giao tiếp I2C là sự kết hợp các tính năng tốt nhất của UART và SPI, trong giao tiếp I2C, một thiết bị có thể đóng vai trò là Master (chủ) kết nối tới nhiều Slave (tớ), hoặc có thể nhiều thiết bị đóng vai trò là Master kết nối tới một hoặc nhiều thiết bị là Slave.

I2C chỉ sử dụng 2 dây để kết nối các thiết bị với nhau:

* SDA (tín hiệu Data) - Master và Slave gửi và nhận dữ liệu.
* SCL (xung Clock) - Tín hiệu xung Clock

I2C là một giao thức giao tiếp nối tiếp, vì vậy dữ liệu được truyền từng bit theo một dây duy nhất (dây SDA)

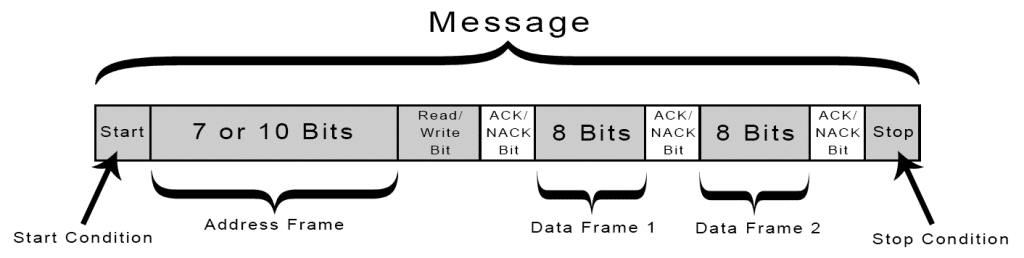
Giống như SPI, I2C sử dụng cơ chế đồng bộ do đó, đầu ra của bit được đồng bộ hóa với việc lấy mẫu các bit bằng tín hiệu Clock được chia sẻ giữa Master và Slave. Tín hiệu Clock luôn được kiểm soát bởi Master.

Thông số chính của giao tiếp I2C:

* Số lượng dây sử dụng: 2
* Tốc độ truyền tối đa:
* Chế độ tiêu chuẩn = 100 kbps
* Chế độ nhanh = 400 kbps
* Chế độ tốc độ cao = 3.4 Mbps
* Chế độ tốc độ rất cao = 5 Mbps
* Loại chế độ truyền: Đồng bộ
* Phương thức truyền: Nối tiếp
* Số lượng Master tối đa: Không giới hạn về số lượng
* Số lượng Slave tối đa: 1008

*Cách hoạt động của I2C*

Với I2C, dữ liệu được truyền trong các tin nhắn. Tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của địa chỉ slave và một hoặc nhiều khung dữ liệu chứa dữ liệu đang được truyền. Thông điệp cũng bao gồm điều kiện khởi động và điều kiện dừng, các bit đọc / ghi và các bit ACK / NACK giữa mỗi khung dữ liệu:



Hình 6: Cách truyền dữ liệu I2C

* Điều kiện khởi động: Đường SDA chuyển từ mức điện áp cao xuống mức điện áp thấp trước khi đường SCL chuyển từ mức cao xuống mức thấp.
* Điều kiện dừng: Đường SDA chuyển từ mức điện áp thấp sang mức điện áp cao sau khi đường SCL chuyển từ mức thấp lên mức cao.
* Khung địa chỉ: Một chuỗi 7 hoặc 10bit duy nhất cho mỗi slave để xác định slave khi master muốn giao tiếp với nó.
* Bit đọc / ghi: Một bit duy nhất chỉ định master đang gửi dữ liệu đến slave (mức điện áp thấp) hay yêu cầu dữ liệu từ nó (mức điện áp cao).
* Bit ACK / NACK:Mỗi khung trong một tin nhắn được theo sau bởi một bit xác nhận / không xác nhận. Nếu một khung địa chỉ hoặc khung dữ liệu được nhận thành công, một bit ACK sẽ được trả lại cho thiết bị gửi từ thiết bị nhận [2].

*Địa chỉ*

I2C không có các đường chọn Slave như SPI, vì vậy cần một cách khác để cho slave biết rằng dữ liệu đang được gửi đến slave này chứ không phải slave khác. Nó thực hiện điều này bằng cách định địa chỉ. Khung địa chỉ luôn là khung đầu tiên sau bit khởi động trong một tin nhắn mới.

Master gửi địa chỉ của slave mà nó muốn giao tiếp với mọi slave được kết nối với nó. Sau đó, mỗi slave sẽ so sánh địa chỉ được gửi từ master với địa chỉ của chính nó. Nếu địa chỉ phù hợp, nó sẽ gửi lại một bit ACK điện áp thấp cho master. Nếu địa chỉ không khớp, slave không làm gì cả và đường SDA vẫn ở mức cao.

*Bit đọc / ghi*

Khung địa chỉ bao gồm một bit duy nhất ở cuối tin nhắn cho slave biết master muốn ghi dữ liệu vào nó hay nhận dữ liệu từ nó. Nếu master muốn gửi dữ liệu đến slave, bit đọc / ghi ở mức điện áp thấp. Nếu master đang yêu cầu dữ liệu từ slave, thì bit ở mức điện áp cao.

*Khung dữ liệu*

Sau khi Master phát hiện bit ACK từ Slave, khung dữ liệu đầu tiên đã sẵn sàng được gửi.

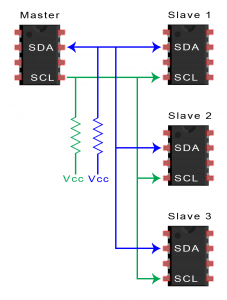
Khung dữ liệu luôn có độ dài 8bit và được gửi với bit quan trọng nhất trước. Mỗi khung dữ liệu ngay sau đó là một bit ACK / NACK để xác minh rằng khung đã được nhận thành công. Bit ACK phải được nhận bởi Master hoặc Slave (tùy thuộc vào cái nào đang gửi dữ liệu) trước khi khung dữ liệu tiếp theo có thể được gửi.

Sau khi tất cả các khung dữ liệu đã được gửi, Master có thể gửi một điều kiện dừng cho slave để tạm dừng quá trình truyền. Điều kiện dừng là sự chuyển đổi điện áp từ thấp lên cao trên đường SDA sau khi chuyển tiếp từ thấp lên cao trên đường SCL, với đường SCL vẫn ở mức cao.

*Các bước truyền dữ liệu I2C*

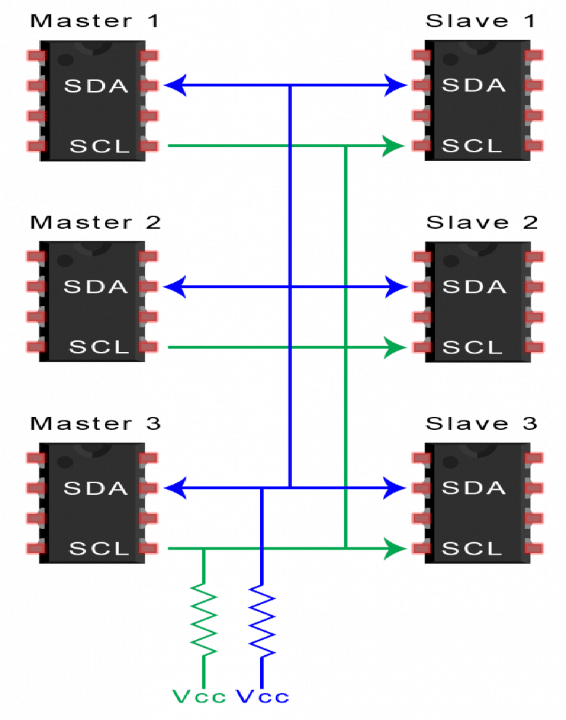
1. Master gửi điều kiện khởi động đến mọi slave được kết nối bằng cách chuyển đường SDA từ mức điện áp cao sang mức điện áp thấp trước khi chuyển đường SCL từ mức cao xuống mức thấp.
2. Master gửi cho mỗi slave địa chỉ 7 hoặc 10bit của slave mà nó muốn giao tiếp, cùng với bit đọc / ghi.
3. Mỗi slave sẽ so sánh địa chỉ được gửi từ master với địa chỉ của chính nó. Nếu địa chỉ trùng khớp, slave sẽ trả về một bit ACK bằng cách kéo dòng SDA xuống thấp cho một bit. Nếu địa chỉ từ master không khớp với địa chỉ của slave, slave rời khỏi đường SDA cao.
4. Master gửi hoặc nhận khung dữ liệu.
5. Sau khi mỗi khung dữ liệu được chuyển, thiết bị nhận trả về một bit ACK khác cho thiết bị gửi để xác nhận đã nhận thành công khung.
6. Để dừng truyền dữ liệu, master gửi điều kiện dừng đến slave bằng cách chuyển đổi mức cao SCL trước khi chuyển mức cao SDA.

**Một Master với nhiều Slave**

Vì I2C sử dụng định địa chỉ nên nhiều Slave có thể được điều khiển từ một master duy nhất. Với địa chỉ 7bit sẽ có 128 (2 mũ 7) địa chỉ duy nhất. Việc sử dụng địa chỉ 10bit không phổ biến, nhưng nó cung cấp 1.024 (2 mũ 10) địa chỉ duy nhất. Để kết nối nhiều slave đến một master duy nhất, bạn có thể đấu dây như thế này, với điện trở kéo lên 4,7K Ohm kết nối đường SDA và SCL với Vcc:

Hình 7: Mô hình một Master nhiều Slave

*Nhiều master với nhiều slave*

Nhiều master có thể được kết nối với một slave hoặc nhiều slave. Sự cố với nhiều master trong cùng một hệ thống xảy ra khi hai master cố gắng gửi hoặc nhận dữ liệu cùng một lúc qua đường SDA. Để giải quyết vấn đề này, mỗi master cần phải phát hiện xem đường SDA thấp hay cao trước khi truyền tin nhắn.

Hình 8: Mô hình nhiều Master nhiều Slave

Nếu đường SDA thấp, điều này có nghĩa là một master khác có quyền điều khiển bus và master đó phải đợi để gửi tin nhắn. Nếu đường SDA cao thì có thể truyền tin nhắn an toàn. Để kết nối nhiều master với nhiều slave, hãy sử dụng sơ đồ sau, với các điện trở kéo lên 4,7K Ohm kết nối các đường SDA và SCL với Vcc

*Ưu điểm*

* Chỉ sử dụng hai dây
* Hỗ trợ nhiều Master và nhiều Slave
* Bit ACK / NACK xác nhận mỗi khung được chuyển thành công
* Phần cứng ít phức tạp hơn so với UART
* Giao thức nổi tiếng và được sử dụng rộng rãi

*Nhược điểm*

* Tốc độ truyền dữ liệu chậm hơn SPI
* Kích thước của khung dữ liệu bị giới hạn ở 8 bit
* Cần phần cứng phức tạp hơn để triển khai so với SPI

### Giao thức HTTP

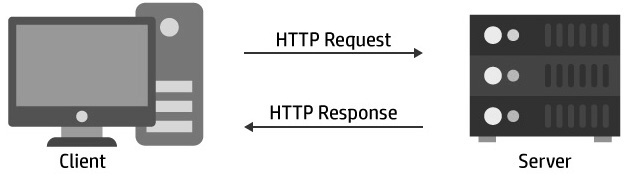
A white rectangular box with black text

Description automatically generated

Hình 9: Mô hình OSI và TCP/IP

HTTP (HyperText Transfer Protocol – Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web – WWW.

Giao thức HTTP hoạt động dựa trên mô hình Client – Server. Thông thường khi các bạn lướt web, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ.



Hình 10: Mô hình Client – Server

Ngoài ra, khi các hệ thống trao đổi dữ liệu với nhau, chúng cũng sử dụng giao thức này nhưng 2 bên đều là server [7].

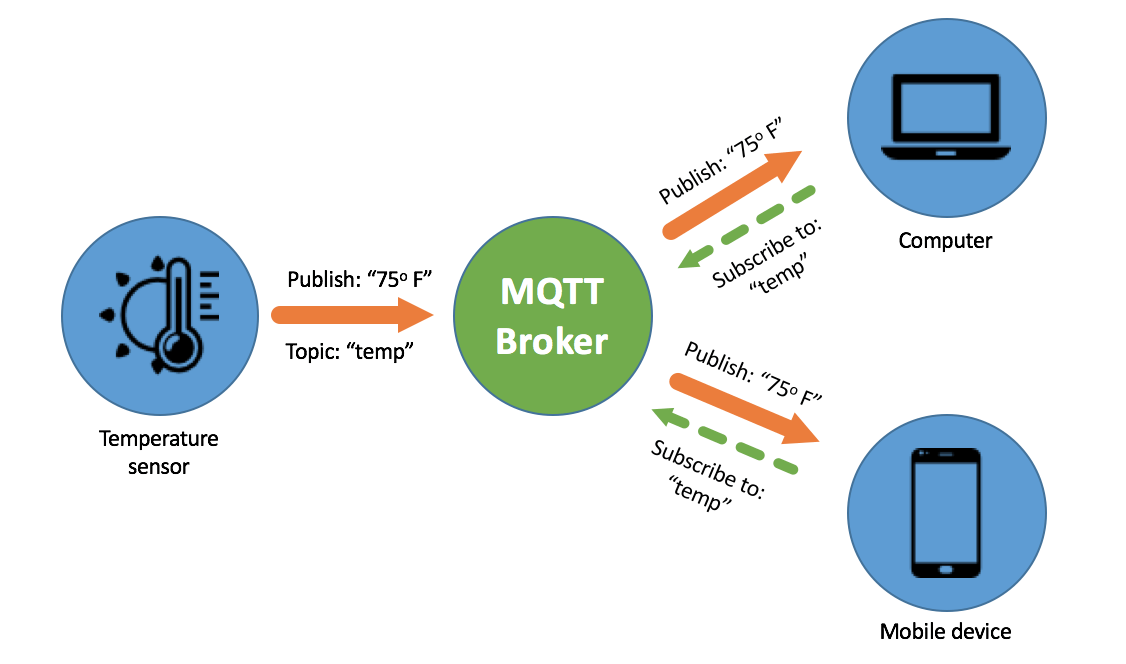
### Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp / thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.

MQTT là lựa chọn lý tưởng trong các môi trường như:

* Những nơi mà giá mạng viễn thông đắt đỏ hoặc băng thông thấp hay thiếu tin cậy.
* Khi chạy trên thiết bị nhúng bị giới hạn về tài nguyên tốc độ và bộ nhớ.
* Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M (Machine to Machine).
* MQTT cũng là giao thức được sử dụng trong Facebook Messenger

Một số ưu điểm nổi bật của MQTT như: băng thông thấp, độ tin cậy cao và có thể sử dụng ngay cả khi hệ thống mạng không ổn định, tốn rất ít byte cho việc kết nối với server và connection có thể giữ trạng thái open xuyên suốt, có thể kết nối nhiều thiết bị (MQTT client) thông qua một MQTT server (broker). Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng IoT [4].

* Dạng truyền thông điệp theo mô hình Pub/Sub cung cấp việc truyền tin phân tán một chiều, tách biệt với phần ứng dụng.
* Việc truyền thông điệp là ngay lập tức, không quan tâm đến nội dung được truyền.
* Sử dụng TCP/IP là giao thức nền.
* Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)
* Phần bao bọc dữ liệu truyền nhỏ và được giảm đến mức tối thiểu để giảm tải cho đường truyền.

Hình 11: Mô hình khái quát của MQTT

Các thành phần của MQTT bao gồm:

Client: Publisher - Nơi gửi thông điệp và Subscriber - Nơi nhận thông điệp

Broker - Máy chủ môi giới:

Trong đó Broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ Client (Publisher/Subscriber). Nhiệm vụ chính của Broker là nhận thông điệp (message) từ Publisher, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể. Nhiệm vụ phụ của Broker là nó có thể đảm nhận thêm một vài tính năng liên quan tới quá trình truyền thông như: bảo mật message, lưu trữ message, logs, ....

Client thì được chia thành hai nhóm là Publisher và Subscriber. Client chỉ làm ít nhất một trong 2 việc là publish các thông điệp (message) lên một/nhiều topic cụ thể hoặc subscribe một/nhiều topic nào đó để nhận message từ topic này.

## Phần cứng của hệ thống

### Vi điều khiển ESP8266

Bảng phát triển NodeMCU ESP8266 đi kèm với mô-đun ESP-12E chứa chip ESP8266 có bộ vi xử lý Tensilica Xtensa 32-bit LX106 RISC. Bộ vi xử lý này hỗ trợ RTOS và hoạt động ở tần số xung nhịp có thể điều chỉnh từ 80MHz đến 160 MHz.

NodeMCU có 128 KB RAM và 4MB bộ nhớ Flash để lưu trữ dữ liệu và chương trình. Sức mạnh xử lý cao của nó với Wi-Fi / Bluetooth và các tính năng Điều hành Ngủ sâu tích hợp khiến nó trở nên lý tưởng cho các dự án IoT.

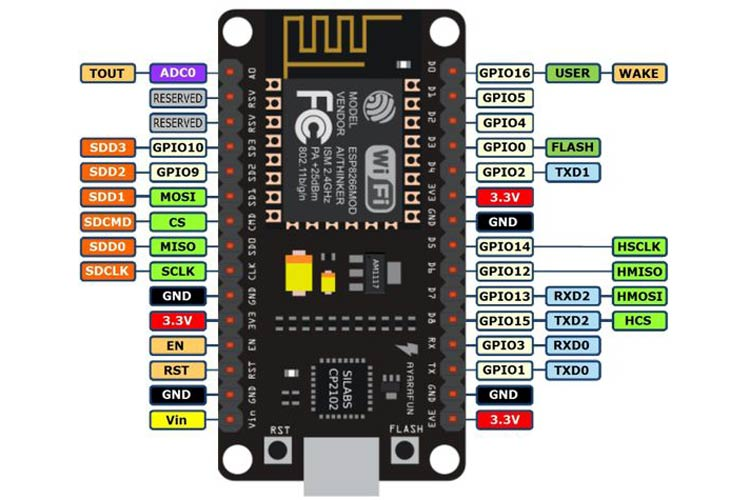
NodeMCU có thể được cấp nguồn bằng giắc cắm Micro USB và chân VIN (Chân nguồn cung cấp bên ngoài). Nó hỗ trợ giao diện UART, SPI và I2C.

Thông số cơ bản của ESP8266 [4]:

* Điện áp nguồn (USB): 5V DC
* Đầu vào/Đầu ra điện áp: 3.3V DC
* Dòng điện: 15mA CPU và 20uA ở chế độ chờ
* In/Out: 18
* Bộ nhớ ROM: Max 16MByte
* Bộ nhớ SRAM: Max 50KByte
* Mô hình: 30 chân
* Thạch anh: 24Mhz – 52MHz
* Các chuẩn giao tiếp: I²C, SPI, UART / USART, USB, CAN
* Nhiệt độ hoạt động: -40°C ~ 125°C
* Cổng ADC: 10bit – 1 Kênh
* Chip USB-Serial: CP2102

Thông số wifi:

* Hỗ trợ 802.11 b/g/n
* Hỗ trợ 802.11n (2,4 GHz), lên tới 72,2 Mbps
* Chống phân mảnh
* 2 x giao diện Wi-Fi ảo
* Giám sát đèn hiệu tự động (TSF phần cứng)
* Hỗ trợ cơ sở hạ tầng Chế độ trạm BSS/Chế độ SoftAP



Hình 12: Các ngoại vi trên ESP8266

Chức năng các chân:

* 17 chân GPIO: Được sử dụng để đọc dữ liệu từ các cảm biến, điều khiển các thiết bị đầu ra, hoặc giao tiếp với các thiết bị khác như LED, động cơ, nút nhấn, v.v.
* 1 kênh ADC: 1 kênh ADC có độ chính xác 10bit theo công nghệ SAR ADC.
* 2 giao tiếp UART: 2 giao tiếp UART hỗ trợ điều khiển dòng dữ liệu.
* 4 đầu ra PWM: 4 chân PWM để điều khiển tốc độ động cơ hoặc độ sáng của đèn LED.
* 2 giao tiếp SPI và 1 giao tiếp I2C:2 giao tiếp SPI và một giao tiếp I2C để kết nối các cảm biến và thiết bị ngoại vi khác.
* Giao tiếp I2S: Một giao tiếp I2S để thêm âm thanh vào dự án của bạn.

### Cảm biến PZEM – 004T

Module đo điện AC PZEM-004T được sử dụng để đo và theo dõi gần như các thông số về điện năng tiêu thụ của mạch điện như: Điện áp hoạt động, dòng tiêu thụ, công suất và năng lượng tiêu thụ, giao tiếp UART dễ dàng kết nối với vi điều khiển hoặc máy tính.

Module đo điện AC PZEM-004T gồm 2 phân loại: cổng PZCT-02 100A và cổng CT tròn 100A được nhập khẩu chính hãng Peacefair nhỏ gọn dễ lắp đặt, máy được sử dụng để đo dòng cách ly lên tới 100A, mạch có chất lượng gia công linh kiện tốt, độ bền cao.



Hình 13: Cảm biến PZEM-004T

Thông số kỹ thuật [5]:

* Model: PZEM-004T
* Thương hiệu: Peacefair
* Phân loại: các thông số giống nhau, chỉ khác cấu tạo CT
  + Cổng CT tròn 100A: cố định không đóng mở
  + Cổng PZCT-02 100A: có thể đóng mở CT
* Điện áp
* Dải đo: 80 ~ 260VAC
* Tần số: 45 – 65Hz
* Độ phân giải: 0.1V
* Độ chính xác của phép đo: 0.5%
* Dòng điện
* Dải đo: 0 ~ 100A
* Phép đo bắt đầu: 0.02A
* Độ phân giải: 0.001A
* Độ chính xác của phép đo: 0.5%
* Điện năng
* Dải đo: 0 ~ 23kW
* Phép đo bắt đầu: 0.4W
* Độ phân giải: 0.1W
* Độ chính xác của phép đo: 0.5%
* Kích thước board mạch chủ: 78 x 38 x 20mm
* Kích thước CT: 31 x 29 x 47mm

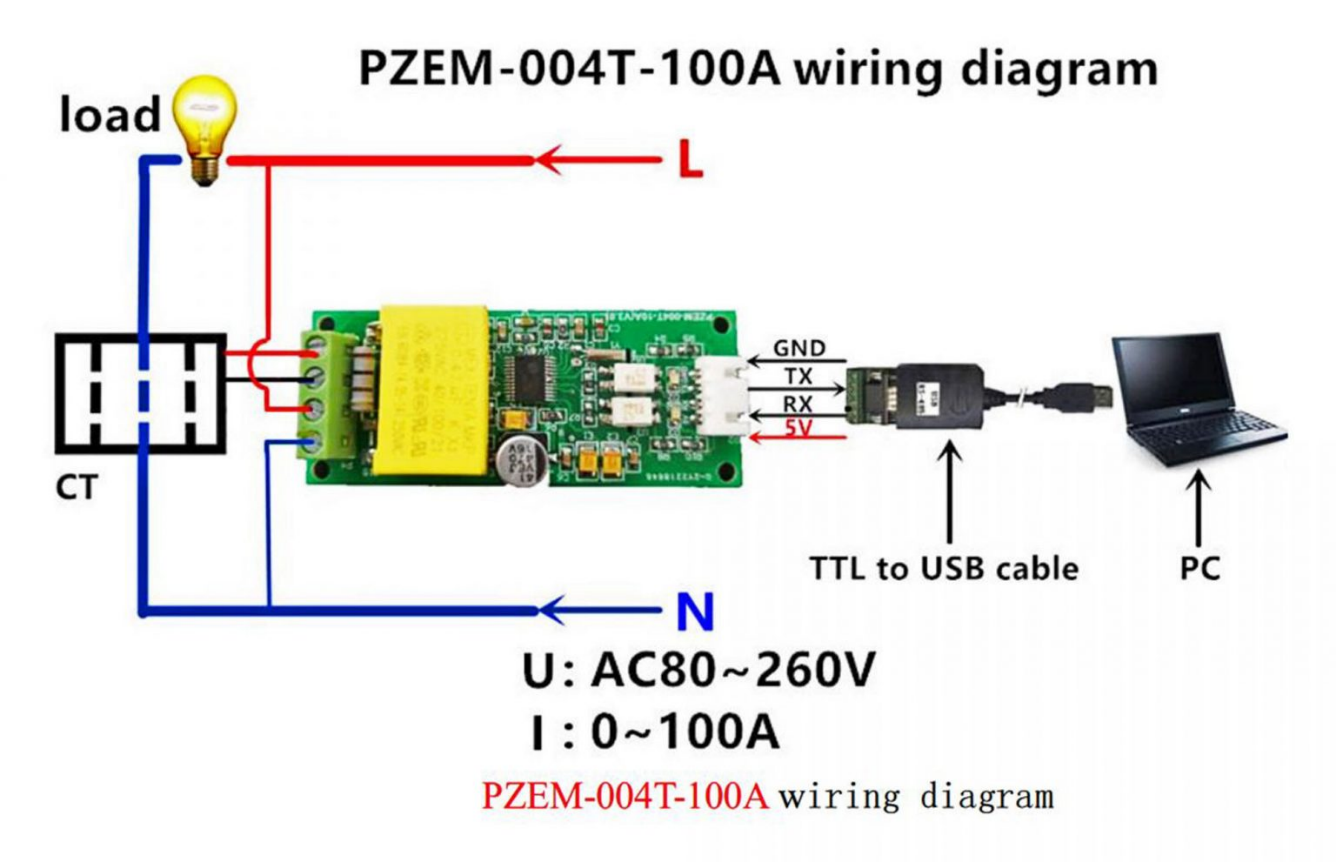
Module đo điện AC 100 A PZEM-004T được dùng để đo và theo dõi các thông số điện AC như: dòng tải tiêu thụ, điện áp, công suất, công suất tiêu thụ...

Module sử dụng giao tiếp UART dễ dàng kết nối truyền dữ liệu tới Vi điều kiển hoặc máy tính, thích hợp cho các ứng dụng theo dõi năng lượng, nhà thông minh, IoT, ...

Module đo điện AC 100 A PZEM-004T với thiết kế nhỏ gọn, dễ lắp đặt, dễ sử dụng, độ bền cao....

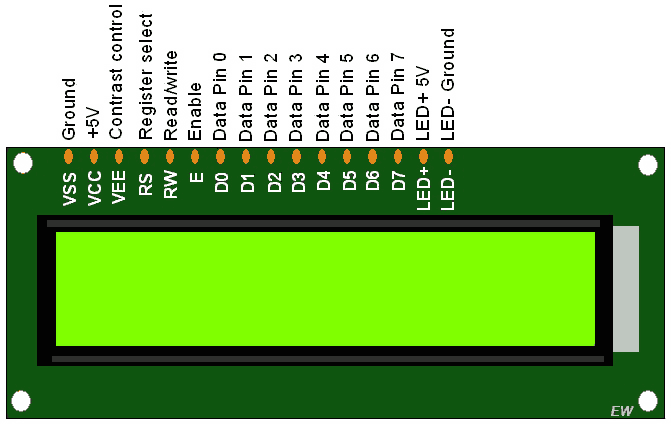
Phương pháp đo dòng cách li an toàn, khả năng đo dòng điện tới 100A.

Sơ đồ đấu nối của cảm biến:



Hình 14: Sơ đồ đấu nối của cảm biến PZEM-004T

### Màn hình LCD 16x2

*Giới thiệu*

Hình 15: Màn hình LCD16x2

LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị kí tự đa dạng, trực quan dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn ít tài nguyên hệ thống và giá thành rẻ.[5]

*Cấu tạo*

LCD 16x2 có 2 hàng, mỗi hàng 16 ký tự, trong 16 chân của LCD được chia làm 3 dạng tín hiệu như sau:

* Các chân cấp nguồn: Chân số 1 nối mass (0V), chân số 2 là VDD nối với nguồn 5V, chân số 3 dùng để chỉnh contrast thường nối với biến trở.
* Các chân điều khiển: Chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
* Các chân dữ liệu DB0 - DB7: Là chân từ số 7 đến 14 dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.
* Chân 15 nối nguồn +5V hoặc 4.2V nối với led, chân 16 nối GND.

*Ứng dụng*

LCD thường được sử dụng trong các mạch điện tử, hiển thị thời gian thực, giá trị, kết quả, hiệu ứng.

*Module chuyển I2C LCD 16x2*

Module I2C LCD ra đời và giải quyết vấn đề LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16x2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối

Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16x2, LCD 20x4, ...) và tương thích với hầu hết các vi điều khiển hiện nay.

Ưu điểm của nó là tiết kiệm chân cho vi điều khiển, dễ dàng kết nối với LCD.

Một số thông số kỹ thuật:

* Điện áp hoạt động: 2.5-6V DC.
* Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
* Giao tiếp: I2C.
* Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
* Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
* Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

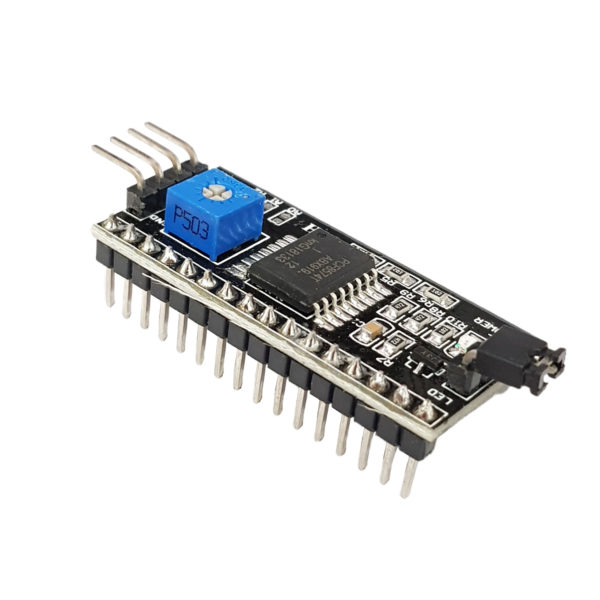
### Module chuyển đổi giao tiếp I2C cho LCD

Để sử dụng các loại LCD có driver là HD44780 (LCD 1602, LCD 2004, …) cần có ít nhất 6 chân của MCU kết nối với các chân RS, EN, D7, D6, D5 và D4 để có thể giao tiếp với LCD.

Nhưng với mạch chuyển đổi giao tiếp I2C cho LCD, các bạn chỉ cần 2 chân (SDA và SCL) của MCU kết nối với 2 chân (SDA và SCL) của module là đã có thể hiển thị thông tin lên LCD. Ngoài ra có thể điều chỉnh được độ tương phản bởi biến trở gắn trên module.

Thông số kỹ thuật:

* Kích thước: 41.5mm(L)X19mm(W)X15.3MM(H)
* Trọng lượng: 5g
* Điện áp hoạt động: 2.5v-6v
* Jump chốt: Cung cấp đèn cho LCD hoặc ngắt.
* Biến trở xoay độ tương phản cho LCD



Hình 16: Module I2C cho LCD

## Phần mềm thiết kế và lập trình

### Phần mềm thiết kế Altium Designer

Altium Designer cung cấp một môi trường thiết kế hợp nhất, giúp cho các kỹ sư có một cái nhìn duy nhất về mọi khía cạnh của quy trình thiết kế PCB từ sơ đồ, đến bố cục PCB, để thiết kế. Bằng cách truy cập tất cả các công cụ thiết kế ở một nơi, các kỹ sư có thể hoàn thành toàn bộ quy trình thiết kế của họ trong cùng một môi trường trực quan và cung cấp các sản phẩm chất lượng cao một cách nhanh chóng.

Altium Designer dễ dàng tạo ra các thiết kế PCB với giao diện trực quan kết nối hoàn toàn bạn với mọi chi tiết của quy trình thiết kế điện tử từ ý tưởng đến sản xuất.

Biên dịch các dự án để tạo ra một mô hình duy nhất, gắn kết có thể đặt trung tâm cho quy trình thiết kế. Dễ dàng truy cập và thao tác dữ liệu chi tiết trong mô hình (sơ đồ, bố cục, mô phỏng) từ mô hình trung tâm này thay vì lưu trữ dữ liệu riêng cho từng phần tử thiết kế.

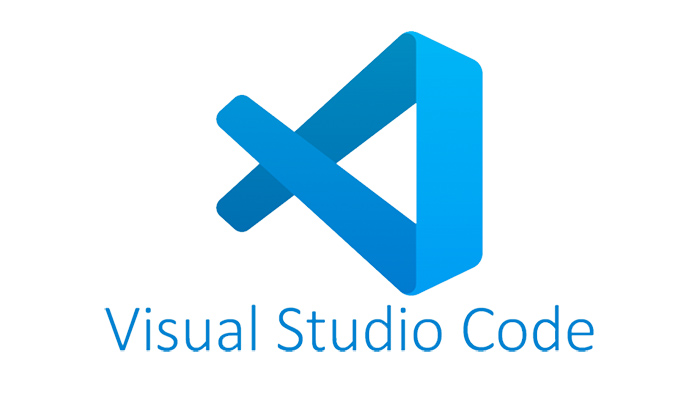
Hình 17: Một cửa sổ của Altium Designer

Ngoài ra, các phiên bản mới hiện nay hỗ trợ Altium 365 - là nền tảng thiết kế sản phẩm điện tử dựa trên đám mây, hợp nhất thiết kế PCB, MCAD, quản lý dữ liệu và làm việc cùng nhau.

### Phần mềm Visual Studio Code

Visual Studio Code là một trình soạn thảo, biên tập code hoàn toàn miễn phí dành được Microsoft phát triển cho các lập trình viên và có mặt trên hầu hết các hệ điều hành phổ biến như: Windows, Linux và macOS. Có thể nói rằng, Visual Studio Code là một sự kết hợp độc đáo – đỉnh cao giữa IDE và Code Editor.

Không chỉ là soạn thảo, chỉnh sửa code, Visual Studio Code còn có thể hỗ trợ làm được rất nhiều việc như: đổi theme, hàng loạt phím tắt tiện dụng, có chức năng debug đi kèm, hỗ trợ Git, syntax highlighting hỗ trợ quá trình gõ code, phần gợi ý code thông minh, …



Hình 18: Visual Studio Code

Visual Studio Code tích hợp rất nhiều tính năng giúp người dùng có thể lập trình, thao tác với các thư mục của bạn một cách chuyên nghiệp nhất mà không cần phải thay đổi giữa các màn hình hay phải trở về thư mục gốc.

Visual Studio Code cũng có thể mở cùng lúc nhiều tệp tin, thư mục mà không cần thiết phải liên quan đến nhau.

Visual Studio Code là một phần mềm miễn phí và được sử dụng hầu hết với các lập trình viên.

### Platform IO

PlatformIO là một hệ sinh thái mã nguồn mở được viết trên Python để phát triển IoT và là một IDE đa nền tảng với trình gỡ lỗi hợp nhất chạy trên Windows, Mac và Linux.

Hình 19: Platform IO

PlatformIO đi kèm với trình quản lý thư viện cho các nền tảng như Arduino hay MBED cùng với kiểm thử phần mềm và cập nhật firmware.

PlatformIO hỗ trợ một số nền tảng như Arduino, ESP32, ESP8266 và đi kèm với một số ví dụ và thư viện. Nó độc lập với nền tảng mà nó đang chạy và chỉ yêu cầu Python được cài đặt trên máy tính.

PlatformIO giúp người dùng viết chương trình nhanh chóng với các tính năng như hoàn tất code C/C ++ và công cụ lập trình thông minh để phát triển dự án chuyên nghiệp và nhanh chóng. Các tính năng này vốn không có trong Arduino IDE. Nó cũng đi kèm với gợi ý thông minh và định dạng khi lập trình. Các tính năng cốt lõi bao gồm hệ thống xây dựng đa nền tảng, trình quản lý thư viện, theo dõi cổng nối tiếp …

### Phần mềm Blynk

Blynk là một nền tảng với các ứng dụng iOS và Android để điều khiển Arduino, Raspberry Pi và các ứng dụng tương tự qua Internet.

Nó là một bảng điều khiển kỹ thuật số nhờ đó có thể xây dựng giao diện đồ họa cho dự án bằng cách kéo và thả các widget.

Blynk không bị ràng buộc với một số bo hoặc shield cụ thể. Thay vào đó, nó hỗ trợ các phần cứng như Arduino hoặc Raspberry Pi được liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk và sẵn sàng cho IoT [3].



Hình 20: Phần mềm Blynk

**Cách hoạt động của Blynk**

Blynk được thiết kế cho IoT. Nó có thể điều khiển phần cứng từ xa, nó có thể hiển thị dữ liệu cảm biến, nó có thể lưu trữ dữ liệu, trực quan hóa và làm nhiều thứ hay ho khác.

Có ba thành phần chính trong nền tảng:

* Ứng dụng Blynk - cho phép tạo giao diện cho các dự án của mình bằng cách sử dụng các widget khác nhau.
* Blynk Server - chịu trách nhiệm về tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Có thể sử dụng Blynk Cloud hoặc chạy cục bộ máy chủ Blynk riêng của mình. Nó là mã nguồn mở, có thể dễ dàng xử lý hàng nghìn thiết bị và thậm chí có thể được khởi chạy trên Raspberry Pi.
* Thư viện Blynk - dành cho tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và lệnh đi.

**Đặc tính**

* API và giao diện người dùng tương tự cho tất cả phần cứng và thiết bị được hỗ trợ
* Kết nối với đám mây bằng cách sử dụng:
  + Wifi
  + Bluetooth và BLE
  + Ethernet
  + USB (Nối tiếp)
  + GSM
  + …
* Bộ Widget dễ sử dụng
* Thao tác ghim trực tiếp mà không cần viết mã
* Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng ghim ảo
* Theo dõi dữ liệu lịch sử qua tiện ích SuperChart
* Giao tiếp giữa thiết bị với thiết bị sử dụng Bridge Widget
* Gửi email, tweet, push notification...
* Các tính năng mới liên tục được bổ sung.

## Kết luận chương 2

Chương 2 nói về việc sử dụng các thiết bị, các phần mềm và các công cụ đã dùng trong quá trình hoàn thành sản phẩm và các kiến trúc tổng quan phục vụ cho việc giải bài toán được đặt ra. Cùng với đó, chương 2 giới thiệu cũng như cách sử dụng các công cụ để lập trình và debug chương trình vi điều khiển ESP8266.

# THIẾT KẾ HỆ THỐNG HỆ THỐNG GIÁM SÁT VÀ THỐNG KÊ DÒNG ĐIỆN, ĐIỆN ÁP AC

## Sơ đồ khối của hệ thống

Sơ đồ khối của hệ thống:

A diagram of a company

Description automatically generated

Hình 21: Sơ đồ khối của hệ thống

Mô hình bao gồm các khối chính như sau:

* Khối xử lý trung tâm: Bao gồm module ESP8266, khối này có nhiệm vụ đọc dữ liệu từ cảm biến dòng áp.
* Khối cảm biến: Bao gồm mạch cảm biến dòng áp (PZEM – 004T) có nhiệm vụ xử lý tín hiệu dòng áp và gửi dữ liệu đến khối xử lý trung tâm.
* Webserver: Có nhiệm vụ tiếp nhận thông tin từ ESP8266 và hiển thị thông tin lên webserver
* Khối hiển thị: Hiển thị dòng và áp đo được qua LCD 16x2.
* Khối nguồn: Cấp nguồn cho toàn bộ hệ thống.

## Sơ đồ nguyên lý toàn mạch

A diagram of a device

Description automatically generated

Hình 22: Sơ đồ nguyên lý toàn mạch

### Khối nguồn và ổn áp nguồn

*Khối nguồn*

A close-up of a power cord

Description automatically generated

Hình 23: Khối nguồn và adapter 5V

Mạch sử dụng 5V-2A Delta chuyên dụng.

Thông số kỹ thuật [6]:

* Điện áp đầu vào: 100-240V-0.5A
* Tần số hoạt động: 50-60 Hz
* Điện áp đầu ra: 5V-2A
* Model: ADP-24ZB
* Kích thước jack đầu ra: 5.5 x 2.1mm
* Chiều dài dây cắm từ ổ điện tới Adapter: 44 cm
* Chiều dài dây từ Adapter tới jack 12V: 48 cm
* Chiều dài sợi dây nối dài: 1.5m

Độ bảo vệ:

* Bảo vệ quá tải.
* Bảo vệ quá áp.
* Bảo vệ ngắn mạch.

### Khối xử lý trung tâm

A diagram of a computer

Description automatically generated

Hình 24: Khối xử lý trung tâm

* Khối xử lý trung tâm với vi điều khiển ESP8266 xử lý tất cả các tác vụ bên trong hệ thống, bao gồm:
* Giao tiếp với cảm biến: Khối xử lý trung tâm giao tiếp với cảm biến thông qua 2 chân TX và RX của chuẩn UART.
* Gửi các dữ liệu dòng điện và điện và điện áp đọc được lên webserver để thống kê và giám sát.

### Khối cảm biến

A close-up of a number

Description automatically generated

Hình 25: Khối cảm biến

Khối cảm biến sử dụng cảm biến PZEM004T đo dòng, đo áp của dòng điện xoay chiều. Sau đó các tín hiệu điện áp, dòng điện sẽ được gửi đến bộ xử lý trung tâm qua chuẩn UART bằng hai chân Tx, Rx.

* Chân +5v dùng để cấp nguồn cho module
* Chân GND được nối với mass.
* Chân Tx được sử dụng để gửi dữ liệu từ cảm biến đến bộ xử lý trung tâm, Tx của cảm biến được nối với Rx của bộ xử lý trung tâm.
* Chân Rx được sử dụng để nhận dữ liệu từ bộ xử lý trung tâm, Rx của cảm biến được nối với Tx của bộ xử lý trung tâm.

### Khối hiển thị

A diagram of a circuit

Description automatically generated

Hình 26: Khối hiển thị

Kết nối tới màn hình LCD16x2 thông qua module I2C:

* Chân GND: nối GND
* Chân VCC: nối với +5V
* Chân SDA, SCL: 2 chân dữ liệu của giao tiếp I2C

## Sơ đồ mạch in

*Đặt luật cho mạch in:*

* Khoảng cách giữa các thành phần (Clearance): 1mm
* Chiều rộng dây tín hiệu: 0.8mm
* Chiều rộng dây GND: 1mm
* Chiều rộng dây VCC: 1mm

*Sơ đồ mạch in:*

A blue circuit board with yellow and red lines

Description automatically generated

Hình 27: Sơ đồ mạch in

A close-up of a circuit board

Description automatically generated

Hình 28: Hình ảnh 3D của toàn mạch

## Thiết kế phần mềm

A diagram of a computer flowchart

Description automatically generated

Hình 29: Lưu đồ thuật toán của hệ thống

Sau khi hệ thống được cấp nguồn, quá trình khởi động được bắt đầu, sau đó vi điều khiển sẽ đọc dữ liệu từ cảm biến và hiển thị lên màn hình LCD cũng như hiển thị lên trên App, Web.

## Mô hình thực tế

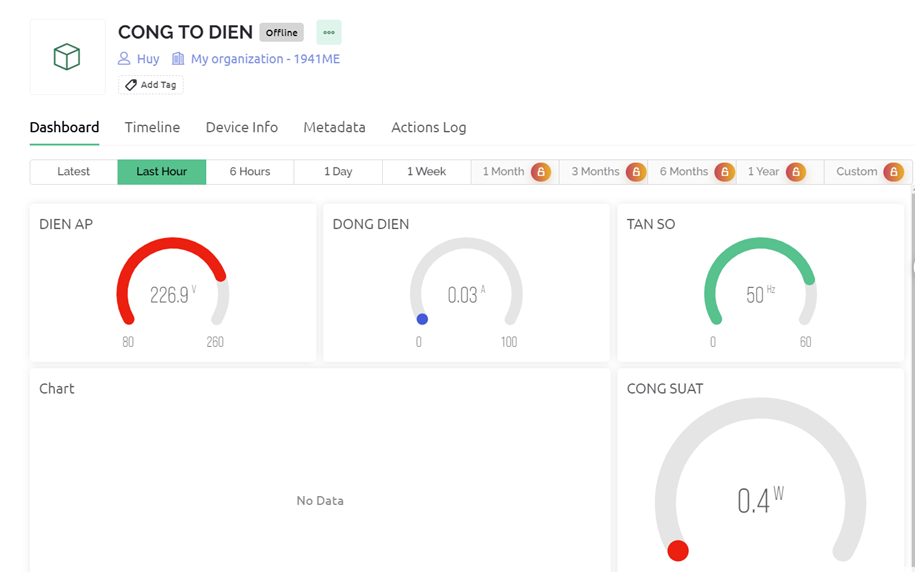
**A circuit board with wires and a display

Description automatically generated**

Hình 30: Mô hình thực tế của hệ thống

Mô hình thực tế của hệ thống bao gồm những thành phần sau:

* Vi điều khiển ESP8266 thực hiện nhiệm vụ đọc dữ liệu đo dòng, áp từ cảm biến PZEM-004T và hiển thị giá trị của chúng lên LCD. Bên cạnh đó, vi điều khiển ESP8266 còn thực hiện chức năng đóng cắt relay 5V.
* Khối hiển thị sử dụng LCD1602 thực hiện chức năng hiển thị thông tin về dòng điện, điện áp, công suất…được nhận từ ESP8266.
* Khối cảm biến sử dụng một cảm biến đo dòng, áp xoay chiều AC PZEM-004t để thực hiện chức năng đo dòng, đo áp và gửi dữ liệu đo được đến ESP8266 thông qua chuẩn kết nối UART.
* Thiết bị điện xoay chiều được bật tắt thông qua relay 5V được điều khiển bởi ESP8266.

****

Hình 31: Giao diện giám sát

Giao diện giám sát có thể được nhìn thấy và hiển thị thông qua app hoặc web Blynk. Giao diện bao gồm một số thông tin như sau:

* Hiển thị trạng thái online/offline của thiết bị
* Hiển thị thông tin điện áp xoay chiều
* Hiển thị thông tin dòng điện xoay chiều
* Hiển thị tần số.
* Hiển thị công suất của thiết bị tiêu thụ điện
* Hiển thị biểu đồ dữ liệu theo thời gian.

Hướng dẫn sử dụng mô hình

*Tổng quan về hướng dẫn sử dụng*

A close up of a circuit board

Description automatically generated

Hình 32: Tổng quan về hướng dẫn sử dụng

*Các bước thực hiện:*

Bước 1: Dùng máy tính hoặc điện thoại phát mạng wifi 2.4G với ssid và password như sau:

* SSID: demo
* Passwords: 11111111

Bước 2: Cấp nguồn cho hệ thống

Bước 2: Truy cập vào App Blynk để giám sát và điều khiển bật tắt relay

Bước 3: Màn hình LCD hiển thị dòng điện và điện áp cần đo

Bước 4: Giao diện giám sát dòng điện, điện áp và một số thông số khác trên giao diện app.

# KẾT LUẬN

Qua quá trình thực hiện làm đồ án, nhóm em đã trình bày các cơ sở lý thuyết liên quan và chạy thành công **“Thiết kế hệ thống giám thiết bị tiêu thụ điện trong hộ gia đình”**. Tuy thời gian làm đồ án thực sự không quá dài nhưng được sự giúp đỡ tận tình của TS……………. cùng với sự nỗ lực và cố gắng của bản thân, sự chỉ bảo của các Thầy Cô trong khoa Điện Tử, nhóm em đã hoàn thành đề tài theo yêu cầu và đúng thời gian quy định với những nội dung sau:

- Nghiên cứu và tìm hiểu về các loại cảm biến đo dòng áp trên thị trường, đặc biệt là cảm biến đo dòng áp PZEM-004T

- Nghiên cứu và tìm hiểu về vi điều khiển ESP8266, các ngoại vi và lập trình ESP8266 dựa trên Arduino Framework

- Tìm hiểu về các chuẩn truyền thông có dây trong công nghiệp như Modbus, I2C, UART, tìm hiểu về chuẩn truyền thông không dây như WiFi

**Hướng phát triển đề tài:**

- Tạo giao diện phong phú và đa dạng hơn

- Giảm độ trễ khi đo và giám sát

- Phát triển đa nền tảng (Android, IOS…)

- Điều khiển và giám sát thêm các thiết bị khác

Thông qua quá trình làm đồ án, chúng em đã được vận dụng những kiến thức chuyên ngành trong 4 năm học. Qua đó đã giúp cho chúng em rèn luyện được kỹ năng, cách tiếp cận với các vấn đề, các bài toán thực tế phức tạp tại các doanh nghiệp, nhà máy khi ra trường làm việc.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "https://mesidas.com," [Online]. Available: https://mesidas.com/modbus/. |
| [2] | "https://dientutuonglai.com," [Online]. Available: https://dientutuonglai.com/chuan-giao-tiep-i2c-la-gi.html. |
| [3] | "https://blynk.io/," [Online]. Available: https://blynk.io/. |
| [4] | "https://hshop.vn," [Online]. Available: https://hshop.vn/products/kit-rf-thu-phat-wifi-esp8266-nodemcu. |
| [5] | "https://nshopvn.com," [Online]. Available: https://nshopvn.com/product/module-do-dien-ac-pzem-004t/. |
| [6] | "https://chotroihn.vn," [Online]. Available: https://chotroihn.vn/nguon-adapter-5v-2a-5-5-2-1mm. |
| [7] | "https://vi.wikipedia.org," [Online]. Available: https://vi.wikipedia.org/wiki/Hypertext\_Transfer\_Protocol. |
|  |  |

# PHỤ LỤC

Code ESP8266

#define BLYNK\_TEMPLATE\_ID "TMPL6gFicPlha"

#define BLYNK\_TEMPLATE\_NAME "CONG TO DIEN 3"

#define BLYNK\_AUTH\_TOKEN "NXnS7ZW3mVTTGY5pwOINpCOrUFdENHJa"

#define BLYNK\_PRINT Serial

#include <BlynkSimpleEsp8266.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

#include <ESP8266WiFi.h>

#include <WiFiManager.h>

#include <SPI.h>

#include <Wire.h>

#include <PZEM004Tv30.h>

#include <SoftwareSerial.h>

#define SW\_PIN 0

unsigned long TimeCount=0;

unsigned long timeReadPzem = millis();

#define DIENAP V0

#define DONGDIEN V1

#define CONGSUAT V2

#define DIENNANG V3

#define CONTROL V4

#define RELAY 13

char blynk\_token[] = BLYNK\_AUTH\_TOKEN;

BlynkTimer timer;

byte RLStatus=0;

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

#if defined(ESP32)

#error "Software Serial is not supported on the ESP32"

#endif

/\*

\* Pin 12 Rx (Connects to the Tx pin on the PZEM)

\* Pin 13 Tx (Connects to the Rx pin on the PZEM)

\*/

#if !defined(PZEM\_RX\_PIN) && !defined(PZEM\_TX\_PIN)

#define PZEM\_RX\_PIN 12

#define PZEM\_TX\_PIN 14

#endif

SoftwareSerial pzemSWSerial(PZEM\_RX\_PIN, PZEM\_TX\_PIN);

PZEM004Tv30 pzem(pzemSWSerial);

void controlON()

{

digitalWrite(RELAY,LOW);

RLStatus=1;

}

void controlOFF()

{

digitalWrite(RELAY,HIGH);

RLStatus=1;

}

void setup() {

Serial.begin(115200);

pinMode(RELAY,OUTPUT);

controlOFF();

connect();

lcd.init();

lcd.backlight();

lcd.setCursor(3,0);

lcd.print("WELCOME!");

lcd.setCursor(1,1);

lcd.print("INITIALIZATION");

lcd.init();

}

void loop() {

Blynk.run();

Serial.print("Custom Address:");

Serial.println(pzem.readAddress(), HEX);

// Doc gia tri cam bien

float voltage = pzem.voltage();

float current = pzem.current();

float power = pzem.power();

float energy = pzem.energy();

float frequency = pzem.frequency();

float pf = pzem.pf();

// kiem tra cam bien

if(isnan(voltage)){

Serial.println("Error reading voltage");

} else if (isnan(current)) {

Serial.println("Error reading current");

} else if (isnan(power)) {

Serial.println("Error reading power");

} else if (isnan(energy)) {

Serial.println("Error reading energy");

} else if (isnan(frequency)) {

Serial.println("Error reading frequency");

} else if (isnan(pf)) {

Serial.println("Error reading power factor");

} else {

// in ra serial monitor

Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");

Serial.print("Current: "); Serial.print(current); Serial.println("A");

Serial.print("Power: "); Serial.print(power); Serial.println("W");

Serial.print("Energy: "); Serial.print(energy,3); Serial.println("kWh");

Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");

Serial.print("PF: "); Serial.println(pf);

Blynk.virtualWrite(DIENAP,voltage);

Blynk.virtualWrite(DONGDIEN,current);

Blynk.virtualWrite(CONGSUAT,power);

Blynk.virtualWrite(DIENNANG,energy);

}

lcd.clear();

lcd.setCursor(0,0);

lcd.print("U:");

lcd.print(voltage,1);

lcd.setCursor(9,0);

lcd.print("I:");

lcd.print(current,2);

lcd.setCursor(0,1);

lcd.print("P:");

lcd.print(power,2);

lcd.setCursor(9,1);

lcd.print("E:");

lcd.print(energy,2);

timeReadPzem = millis();

Serial.println();

delay(2000);

}

void connect()

{

WiFiManager wifiManager;// Khởi tạo đối tượng cho WiFiManager

Serial.println("Delete old wifi? Press Flash button within 3 second");

for(int i=3;i>0;i--)

{

Serial.print(String(i)+" ");

delay(1000);

}

if(digitalRead(SW\_PIN)==LOW)// press button

{

Serial.println();

Serial.println("Reset wifi config!");

wifiManager.resetSettings();

}

WiFiManagerParameter custom\_blynk\_token("Blynk", "blynk token", blynk\_token, 34);

wifiManager.addParameter(&custom\_blynk\_token);

wifiManager.autoConnect("ESP8266\_PZEM004T","12345678");// tai khoan, mat khau wifi esp8266 phat ra

//wifiManager.autoConnect();// use this to display host as ESP name + CHIPID

// if can go next mean already connected wifi

Serial.println("YOU ARE CONNECTED TO WIFI");

Serial.println(custom\_blynk\_token.getValue());

Blynk.config(custom\_blynk\_token.getValue());

if (WiFi.status() == WL\_CONNECTED)

{

Serial.println("WIFI CONNECTED SUCCESSFULLY! NOW TRY TO CONNECT BLYNK...");

delay(1000);

Blynk.connect(3333); // try to connect to Blynk with time out 10 second

if(Blynk.connected())

{

Serial.println("BLYNK CONNECTED! System ready");

delay(1000);

}

else

{

Serial.println(" BLYNK Not connect. warning!");

delay(1000);

}

}

else

{

Serial.println("WIFI Not connect. warning!");

delay(1000);

}

}

BLYNK\_CONNECTED() {

Blynk.syncAll();

}

BLYNK\_WRITE(V4)

{

int pinValue = param.asInt();

if(pinValue==1)

{

controlON();

}

else

{

controlOFF();

}

TimeCount=millis();

}