

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔN**G

----------

# BÁO CÁO

**Môn: Thiết kế ngoại vi và kỹ thuật ghép nối**

**Chuẩn giao tiếp nối tiếp RS485 giữa các ESP8266**

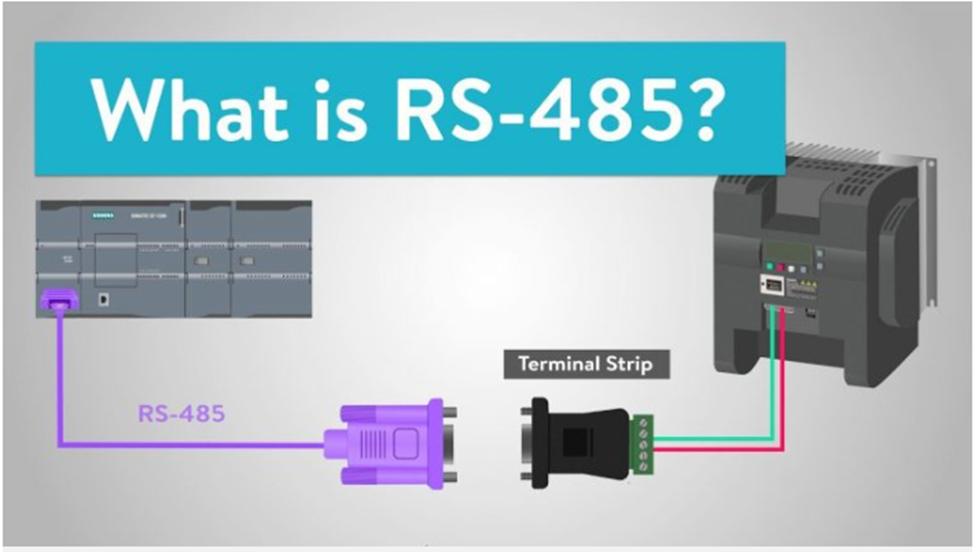
**(Điều khiển độ sáng đèn LED)**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **Dương Quang Duy** |
| **Sinh viên** | **Hồ Văn Thành Minh - B19DCDT142**  **Lê Công Năm - B19DCDT158**  **Nguyễn Đức Quân - B19DCDT178**  **Luyện Thị Ánh Nguyệt - B19DCDT162** |
| **Nhóm môn học** | **01** |
| **Nhóm BTL** | **05** |

**Hà Nội - 2023**

**MỤC LỤC**

**Mục Lục…………………………………………………………………………….2**

1. **Giới thiệu…………………………………………………………………….3**
2. **Tổng quan về chuẩn giao tiếp nối tiếp RS485……………………………..4**
3. **Định nghĩa và nguyên lí hoạt động RS485…………………………………….4**
4. **Các thông số kỹ thuật RS485…………………………………………………………4**
5. **Điểm mạnh và điểm yếu của RS485……………………………………………5**
6. **Thông số kỹ thuật các thiết bị phần cứng khác………………………………….5**
7. **ESP8266………………………………………………………………………………..5**
8. **Chiết áp………………………………………………………………………………..6**
9. **Oled SSD1306…………………………………………………………………………7**
10. **Led……………………………………………………………………………………..7**
11. **Thiết lập môi trường và kết nối phần cứng……………………………………..8**
12. **Sử dụng thư viện ModbusRTU để điều khiển LED………………………………...8**
13. **Thiết lập kết nối RS485 giữa các ESP8266………………………………………….9**
14. **Thực hiện giao tiếp nối tiếp RS485………………………………………………9**
15. **Thiết lập giao thức truyền thông ModbusRTU…………………………………….9**
16. **Gửi và nhận dữ liệu giữa các ESP8266……………………………………………..10**
17. **Xử lí dữ liệu để điều khiển độ sáng LED……………………………………………10**
18. **Kết luận và hướng phát triển ……………………………………………………11**
19. **Tóm tắt lại kết quả đạt được…………………………………………………………11**
20. **Những hạn chế của phương pháp này và hướng phát triển tiếp theo…………….11**
21. **Tài liệu tham khảo………………………………………………………………..12**
22. **Giới thiệu**

* Giao tiếp nối tiếp RS485 là một công nghệ truyền thông phổ biến được sử dụng trong các hệ thống điều khiển và giám sát. RS485 là một chuẩn giao tiếp truyền dữ liệu điểm đến điểm, cho phép truyền tín hiệu qua một đường truyền dẫn (bus) để kết nối các thiết bị điện tử với nhau. RS485 có thể hỗ trợ nhiều thiết bị trên cùng một đường truyền và có thể truyền dữ liệu trong khoảng cách lên tới 1,2 km.
* Trong báo cáo này, chúng ta sẽ trình bày về cách sử dụng giao tiếp nối tiếp RS485 để kết nối giữa các bảng ESP8266 và ESP8266 để điều khiển độ sáng đèn LED. ESP8266 và ESP8266 là hai bảng phát triển IoT (Internet of Things) được phát triển bởi Espressif Systems. Cả hai bảng đều hỗ trợ giao tiếp nối tiếp và có thể được sử dụng để điều khiển các thiết bị điện tử thông qua các giao thức truyền thông như Modbus RTU.
* Để điều khiển độ sáng đèn LED, chúng ta sử dụng giao thức truyền thông Modbus RTU. Modbus RTU là một giao thức truyền thông điều khiển được sử dụng rộng rãi trong các ứng dụng công nghiệp và điều khiển thiết bị. Giao thức này cho phép các thiết bị truyền thông với nhau thông qua giao diện nối tiếp nối tiếp RS485.
* Với việc sử dụng giao tiếp nối tiếp RS485, chúng ta có thể kết nối các bảng ESP8266 và ESP8266 để tạo thành một hệ thống điều khiển và giám sát các thiết bị điện tử như đèn LED. Các thông tin điều khiển sẽ được truyền tới từ bảng điều khiển chính thông qua giao thức Modbus RTU và gửi tín hiệu tới các bảng ESP8266 và ESP8266 để điều khiển độ sáng đèn LED.

Trong báo cáo này, chúng ta sẽ đưa ra một hướng dẫn chi tiết về cách thiết lập môi trường và kết nối phần cứng để thực hiện giao tiếp nối tiếp RS485 giữa các bảng ESP8266 và ESP8266

1. **Tổng quan về giao tiếp nối tiếp RS485**
2. **Định nghĩa và nguyên lý hoạt động của giao tiếp RS485**

* Giao tiếp nối tiếp RS485 là một chuẩn giao tiếp truyền dữ liệu điểm đến điểm giữa các thiết bị điện tử thông qua một đường truyền dẫn (bus). Nó được sử dụng rộng rãi trong các hệ thống điều khiển và giám sát vì khả năng truyền dữ liệu nhanh, tin cậy và có thể kết nối nhiều thiết bị trên cùng một đường truyền.
* Nguyên lý hoạt động của giao tiếp RS485 là truyền dữ liệu qua các tín hiệu điện. Mỗi thiết bị trên đường truyền đều được kết nối với hai dây dữ liệu: A và B. Khi truyền dữ liệu, tín hiệu sẽ được chuyển đổi thành các mức điện trên dây A và B. Tín hiệu truyền đi được mã hóa dưới dạng chuỗi các bit, được đồng bộ hóa với tần số xung clock được định sẵn.
* Một đặc điểm nổi bật của giao tiếp RS485 là khả năng truyền dữ liệu hai chiều trên cùng một đường truyền. Điều này được thực hiện bằng cách sử dụng các tín hiệu phản hồi (feedback signals) để đồng bộ hóa truyền và nhận dữ liệu giữa các thiết bị.

Để tránh xung đột dữ liệu, các thiết bị trên đường truyền sử dụng kỹ thuật truyền thông CSMA/CD (Carrier Sense Multiple Access with Collision Detection) để kiểm tra trạng thái của đường truyền trước khi truyền dữ liệu. Nếu đường truyền đang bận, thiết bị sẽ đợi cho đến khi đường truyền trống trước khi truyền dữ liệu của mình.

1. **Các thông số kỹ thuật của giao tiếp RS485**

* Tốc độ truyền dữ liệu: Tốc độ truyền dữ liệu của giao tiếp RS485 có thể được thiết lập từ 300 bps đến 10 Mbps. Tốc độ truyền dữ liệu thường được thiết lập bằng cách chọn một trong các tốc độ chuẩn, chẳng hạn như 9600 bps, 19200 bps, 38400 bps, v.v.
* Khoảng cách truyền: Giao tiếp RS485 cho phép truyền dữ liệu trên khoảng cách lên đến 1200 mét với tốc độ truyền dữ liệu tối đa 100 kbps. Khi tốc độ truyền dữ liệu giảm xuống, khoảng cách truyền cũng tăng lên.
* Điện áp và dòng tiêu thụ: Điện áp hoạt động của giao tiếp RS485 là 5VDC hoặc 3.3VDC. Điện áp tối đa trên đường truyền là ± 5V, và dòng tiêu thụ của các thiết bị trên đường truyền là khoảng 1 mA.
* Độ dài thông điệp: Độ dài thông điệp của giao tiếp RS485 tối đa là 256 byte. Tuy nhiên, thông điệp thường được chia thành các gói nhỏ hơn để tăng tính ổn định và độ tin cậy của giao tiếp.
* Độ lỗi: Giao tiếp RS485 cho phép phát hiện và sửa chữa các lỗi truyền dữ liệu, chẳng hạn như lỗi kiểm soát chẵn lẻ, lỗi kiểm soát tổng quan, v.v. Điều này giúp tăng tính tin cậy và độ ổn định của giao tiếp.
* Số lượng thiết bị: Giao tiếp RS485 có khả năng kết nối đến 32 thiết bị trên cùng một đường truyền. Tuy nhiên, số lượng thiết bị tối đa có thể kết nối phụ thuộc vào độ dài của đường truyền và tốc độ truyền dữ liệu.
* Khả năng chống nhiễu: Giao tiếp RS485 được thiết kế để chống nhiễu và có khả năng hoạt động tốt trong môi trường có nhiều tín hiệu nhiễu. Các tín hiệu nhiễu có thể được giảm thiểu bằng cách sử dụng cáp đường truyền chất lượng tốt và cài đặt thêm các bộ lọc

1. **Điểm mạnh và điểm yếu của giao tiếp RS485**

+) Điểm mạnh của giao tiếp RS485 bao gồm:

* Khoảng cách truyền dữ liệu lớn: RS485 có thể truyền dữ liệu trên khoảng cách lên đến 1200 mét, cho phép kết nối các thiết bị ở khoảng cách xa nhau mà không cần sử dụng các bộ khuếch đại tín hiệu.
* Khả năng kết nối nhiều thiết bị: Giao tiếp RS485 có khả năng kết nối đến 32 thiết bị trên cùng một đường truyền. Tuy nhiên, số lượng thiết bị tối đa có thể kết nối phụ thuộc vào độ dài của đường truyền và tốc độ truyền dữ liệu.
* Khả năng chống nhiễu: Giao tiếp RS485 được thiết kế để chống nhiễu và có khả năng hoạt động tốt trong môi trường có nhiều tín hiệu nhiễu. Các tín hiệu nhiễu có thể được giảm thiểu bằng cách sử dụng cáp đường truyền chất lượng tốt và cài đặt thêm các bộ lọc.
* Độ tin cậy cao: Giao tiếp RS485 có khả năng phát hiện và sửa chữa các lỗi truyền dữ liệu, giúp tăng tính tin cậy và độ ổn định của giao tiếp.

+) Tuy nhiên, giao tiếp RS485 cũng có một số điểm yếu:

* Chi phí đắt đỏ hơn so với một số giao tiếp khác: Chi phí cài đặt và sử dụng giao tiếp RS485 thường cao hơn so với một số giao tiếp khác, chẳng hạn như giao tiếp UART hay I2C.
* Thời gian đáp ứng chậm hơn: Do quá trình phát hiện và sửa chữa lỗi, thời gian đáp ứng của giao tiếp RS485 có thể chậm hơn so với một số giao tiếp khác.
* Cáp đường truyền phải có chất lượng tốt: Để đảm bảo tính ổn định và độ tin cậy của giao tiếp RS485, cáp đường truyền phải có chất lượng tốt. Việc sử dụng cáp đường truyền kém chất lượng có thể dẫn đến các lỗi truyền dữ liệu.

1. **Thông số kỹ thuật các thiết bị phần cứng khác**
2. **ESP8266**

* ESP8266 là một module Wi-Fi được phát triển bởi Espressif Systems, được sử dụng để kết nối với internet thông qua Wi-Fi. Module này là một trong những sản phẩm phổ biến và được sử dụng rộng rãi trong các ứng dụng IoT (Internet of Things) như đo lường, điều khiển và giám sát thông qua internet.
* Thông số kỹ thuật của ESP8266:

+) Chip: ESP8266EX

+) Bộ nhớ flash tích hợp: 1MB hoặc 2MB (có thể được nâng cấp lên đến 16MB bằng cách sử dụng bộ nhớ ngoài)

+) RAM tích hợp: 80KB

+) Tần số hoạt động: 80MHz

+) Giao diện: UART, GPIO, I2C, SPI, ADC

+) Chuẩn Wi-Fi: 802.11 b/g/n

+) Dải tần số: 2.4GHz

+) Tốc độ truyền dữ liệu tối đa: 150Mbps

+) Tiêu thụ điện năng thấp: 10µA khi trong chế độ ngủ

+) Điện áp hoạt động: 3.0V đến 3.6V

* ESP8266 có thể được lập trình bằng nhiều ngôn ngữ khác nhau như C/C++, Lua, Python và Arduino IDE. Nó cũng hỗ trợ nhiều giao thức truyền thông như MQTT, HTTP, TCP/UDP.

1. **Chiết áp 3 pha**

* Chiết áp (hay còn gọi là transformer) là một thiết bị điện tử được sử dụng để chuyển đổi điện áp từ một mức điện áp sang một mức điện áp khác. Nó được sử dụng rộng rãi trong các ứng dụng điện tử và điện lực để cung cấp điện áp cho các thiết bị điện tử khác nhau.
* Thông số kỹ thuật của chiết áp bao gồm:
* Điện áp định mức (Rated voltage): Là điện áp tối đa mà chiết áp có thể hoạt động ổn định.
* Công suất định mức (Rated power): Là công suất tối đa mà chiết áp có thể chuyển đổi một cách ổn định. Điều này phụ thuộc vào kích thước của chiết áp và số lượt cuộn.
* Tần số hoạt động (Operating frequency): Là tần số của nguồn cung cấp cho chiết áp. Tần số thường được chọn là 50 Hz hoặc 60 Hz.
* Hiệu suất (Efficiency): Là tỉ lệ giữa công suất đầu ra và công suất đầu vào của chiết áp.
* Số lượng lượt cuộn (Number of turns): Là số lượng lượt cuộn trên mỗi vòng dây của chiết áp. Số lượt cuộn thường được chọn để đảm bảo điện áp đầu ra đạt được.
* Dải điện áp đầu vào (Input voltage range): Là dải điện áp tối đa và tối thiểu mà chiết áp có thể chuyển đổi.
* Dải điện áp đầu ra (Output voltage range): Là dải điện áp tối đa và tối thiểu mà chiết áp có thể cung cấp.

Các loại chiết áp được chia thành nhiều loại khác nhau, bao gồm chiết áp đơn (single-phase transformer), chiết áp ba pha (three-phase transformer), chiết áp cách ly (isolation transformer) và chiết áp tự ngẫu (autotransformer). Các loại chiết áp này được sử dụng trong các ứng dụng khác nhau và có các thông số kỹ thuật khác nhau

1. **Oled SSD1306**

* Màn hình OLED SSD1306 là một loại màn hình hiển thị có độ phân giải cao và tiêu thụ năng lượng thấp. Nó được sử dụng rộng rãi trong các ứng dụng nhúng và các thiết bị di động để hiển thị các thông tin như văn bản, hình ảnh và biểu đồ.
* Thông số kỹ thuật của màn hình OLED SSD1306 bao gồm:
* Độ phân giải: 128x64 hoặc 128x32 điểm ảnh.
* Kích thước màn hình: Thường có kích thước từ 0,96 inch đến 1,3 inch.
* Điện áp hoạt động: Thường là 3,3 V hoặc 5 V.
* Giao diện: I2C hoặc SPI.
* Độ sáng: Có thể được điều chỉnh từ rất sáng đến rất tối.
* Góc nhìn: Góc nhìn của màn hình là rộng, từ khoảng 160 độ.
* Độ dày: Thường là từ 1mm đến 2mm.
* Điều khiển đèn nền: Màn hình OLED SSD1306 có thể được điều khiển bằng cách sử dụng đèn nền LED.
* Tốc độ làm mới: Thường là từ 60 Hz đến 100 Hz.

Màn hình OLED SSD1306 được tích hợp sẵn bộ điều khiển đồ họa, cho phép hiển thị đa dạng các ký tự, hình ảnh, biểu đồ, đồ thị và các đối tượng khác. Màn hình này có thể được sử dụng với nhiều loại vi điều khiển nhúng và các thiết bị di động để hiển thị các thông tin cần thiết.

1. **LED**

* LED (Light Emitting Diode) là một loại bóng đèn đặc biệt có khả năng tỏa sáng khi được điện truyền qua. LED được sử dụng rộng rãi trong các thiết bị điện tử, đèn chiếu sáng, đèn trang trí và các ứng dụng khác.
* Thông số kỹ thuật của LED bao gồm:
* Điện áp hoạt động: Thường là 2V đến 3,5V.
* Dòng điện hoạt động: Thường là từ vài mA đến vài chục mA.
* Công suất hoạt động: Thường là từ vài mW đến vài W.
* Góc chiếu: LED có thể được thiết kế với góc chiếu từ 5 đến 180 độ.
* Độ sáng: Độ sáng của LED được đo bằng đơn vị lumens (lm).
* Nhiệt độ màu: Nhiệt độ màu của LED được đo bằng đơn vị Kelvin (K).
* Tuổi thọ: LED có tuổi thọ cao, thường từ 25.000 đến 100.000 giờ.

Có nhiều loại LED khác nhau như LED đơn, LED 7 đoạn, LED đa màu, LED SMD, LED COB và nhiều loại khác nữa. Mỗi loại LED có tính năng và ứng dụng khác nhau. LED là một phần quan trọng của công nghệ hiện đại và được sử dụng rộng rãi trong nhiều lĩnh vực, từ công nghiệp đến gia đình.

1. **Thiết lập môi trường và kết nối phần cứng**
2. **Sử dụng thư viện ModbusRTU để điều khiển đèn LED**

* Để sử dụng giao tiếp RS485 để điều khiển đèn LED, chúng ta có thể sử dụng thư viện ModbusRTU. Thư viện này cung cấp các hàm để thực hiện các lệnh đọc và ghi trên đường truyền RS485 theo chuẩn Modbus RTU.
* Các bước để sử dụng thư viện ModbusRTU để điều khiển đèn LED như sau:

Bước 1: Chuẩn bị phần cứng

* Kết nối module RS485 với các bảng điều khiển đèn LED (ESP8266 hoặc ESP8266) thông qua cổng UART.
* Kết nối module RS485 với PC hoặc thiết bị điều khiển khác thông qua cổng UART và chuyển đổi tín hiệu RS232/RS485.

Bước 2: Cài đặt thư viện ModbusRTU trên ESP8266 hoặc ESP8266

* Tải thư viện ModbusRTU về máy tính.
* Giải nén và chép thư mục ModbusRTU vào thư mục libraries trong thư mục cài đặt Arduino IDE.
* Khởi động lại Arduino IDE.

Bước 3: Viết code điều khiển đèn LED bằng thư viện ModbusRTU trên ESP8266 hoặc ESP8266

* Bắt đầu bằng khai báo các biến và hằng số, cài đặt các chân kết nối, tốc độ truyền dữ liệu và các thông số khác cho giao tiếp RS485.
* Sử dụng hàm ModbusRTU để giao tiếp với module đèn LED. Các lệnh giao tiếp bao gồm đọc và ghi trạng thái đèn LED, cài đặt độ sáng, thời gian chớp nháy, v.v.

Bước 4: Upload code và kiểm tra hoạt động

* Kết nối module ESP8266 hoặc ESP8266 với PC qua cổng USB.
* Chọn board và cổng COM tương ứng trên Arduino IDE.
* Upload code vào module ESP8266 hoặc ESP8266.
* Kiểm tra hoạt động của đèn LED thông qua các lệnh điều khiển.

1. **Thiết lập kết nối nối tiếp RS485 giữa các ESP8266**

* Để thiết lập kết nối nối tiếp RS485 giữa các bảng ESP8266 và ESP8266, chúng ta cần kết nối module RS485 với các bảng điều khiển đèn LED thông qua cổng UART và cấu hình các thông số kết nối như sau:

Bước 1: Kết nối module RS485 với bảng điều khiển đèn LED

* Kết nối chân A (hoặc +) của module RS485 với chân A (hoặc +) của bảng điều khiển đèn LED.
* Kết nối chân B (hoặc -) của module RS485 với chân B (hoặc -) của bảng điều khiển đèn LED.

Bước 2: Kết nối module RS485 với bảng ESP8266 hoặc ESP8266

* Kết nối chân TX của module RS485 với chân RX của bảng ESP8266 hoặc ESP8266.
* Kết nối chân RX của module RS485 với chân TX của bảng ESP8266 hoặc ESP8266.
* Kết nối chân GND của module RS485 với chân GND của bảng ESP8266 hoặc ESP8266.

Bước 3: Cấu hình thông số kết nối RS485

* Thông số tốc độ truyền dữ liệu: Chọn tốc độ truyền dữ liệu phù hợp với module RS485 và bảng điều khiển đèn LED, thông thường là 9600 bps hoặc 115200 bps.
* Địa chỉ thiết bị: Các bảng đèn LED thường được đánh địa chỉ mặc định là 1, 2, 3, v.v. Chúng ta cần cấu hình địa chỉ thiết bị cho module RS485 tương ứng với địa chỉ của bảng đèn LED mà chúng ta muốn điều khiển.

Sau khi cấu hình các thông số kết nối RS485, chúng ta có thể sử dụng thư viện ModbusRTU để điều khiển đèn LED.

1. **Thực hiện giao tiếp nối tiếp RS485**
2. **Thiết lập giao thức truyền thông ModbusRTU**

* Để sử dụng thư viện ModbusRTU để điều khiển đèn LED thông qua giao tiếp RS485, chúng ta cần thiết lập giao thức truyền thông ModbusRTU trên các bảng ESP8266 và ESP8266. Các bước thực hiện như sau:

Bước 1: Cài đặt thư viện ModbusRTU trên Arduino IDE

* Truy cập mục Sketch -> Include Library -> Manage Libraries trên Arduino IDE.
* Tìm kiếm thư viện ModbusRTU và cài đặt thư viện.

Bước 2: Cấu hình ModbusRTU trên ESP8266 và ESP8266

* Khai báo thư viện ModbusRTU trong code của ESP8266 hoặc ESP8266.
* Cấu hình thông số kết nối RS485 như tốc độ truyền dữ liệu và địa chỉ thiết bị.
* Khởi tạo đối tượng ModbusRTU và thiết lập các tham số truyền thông như tốc độ truyền dữ liệu, địa chỉ thiết bị, timeout, v.v.

Bước 3: Thực hiện các lệnh điều khiển đèn LED thông qua ModbusRTU

* Sử dụng các hàm của thư viện ModbusRTU để gửi các lệnh điều khiển đèn LED, như đặt giá trị độ sáng, bật/tắt đèn LED, v.v.
* Chúng ta cần truyền vào địa chỉ của bảng đèn LED mà chúng ta muốn điều khiển và các giá trị điều khiển tương ứng.

Sau khi thiết lập giao thức truyền thông ModbusRTU trên các bảng ESP8266 và ESP8266, chúng ta có thể điều khiển đèn LED thông qua giao tiếp RS485 bằng các lệnh của thư viện ModbusRTU.

1. **Gửi và nhận dữ liệu giữa các ESP8266**

* Sau khi thiết lập kết nối nối tiếp RS485 giữa các bảng ESP8266 và ESP8266 và cấu hình giao thức truyền thông ModbusRTU, chúng ta có thể gửi và nhận dữ liệu giữa các bảng ESP8266 và ESP8266 thông qua giao tiếp RS485. Các bước thực hiện như sau:

Bước 1: Chuẩn bị các lệnh điều khiển dữ liệu

* Để gửi và nhận dữ liệu giữa các bảng ESP8266 và ESP8266, chúng ta cần chuẩn bị các lệnh điều khiển dữ liệu trong code của mỗi bảng.
* Ví dụ, chúng ta có thể sử dụng các lệnh như đọc hoặc ghi giá trị độ sáng của đèn LED, đặt địa chỉ của bảng đèn LED, v.v.

Bước 2: Gửi dữ liệu từ bảng ESP8266

* Sử dụng hàm của thư viện ModbusRTU để gửi các lệnh điều khiển dữ liệu từ bảng ESP8266 đến bảng ESP8266.
* Chúng ta cần truyền vào địa chỉ của bảng ESP8266, địa chỉ của thanh ghi hoặc biến cần ghi hoặc đọc, v.v.

Bước 3: Nhận dữ liệu từ bảng ESP8266

* Sử dụng hàm của thư viện ModbusRTU để nhận dữ liệu từ bảng ESP8266 và lưu trữ vào các biến trên bảng ESP8266.
* Chúng ta cần truyền vào địa chỉ của bảng ESP8266, địa chỉ của thanh ghi hoặc biến cần đọc, v.v.

1Sau khi thực hiện các bước trên, chúng ta có thể gửi và nhận dữ liệu giữa các bảng ESP8266 và ESP8266 thông qua giao tiếp RS485. Việc này cho phép chúng ta thực hiện các tác vụ phức tạp hơn và tăng tính linh hoạt trong việc điều khiển các thiết bị.1

1. **Xử lý dữ liệu để điều khiển độ sáng đèn LED**

* Sau khi đã gửi và nhận dữ liệu giữa các bảng ESP8266 và ESP8266 thông qua giao tiếp RS485 và thực hiện được các lệnh điều khiển dữ liệu, chúng ta cần xử lý dữ liệu để điều khiển độ sáng đèn LED. Các bước thực hiện như sau:

Bước 1: Lưu trữ dữ liệu nhận được

* Khi nhận được dữ liệu từ bảng ESP8266, chúng ta cần lưu trữ vào các biến trên bảng ESP8266.
* Các biến này sẽ lưu trữ các giá trị của các thanh ghi hoặc các biến trên bảng ESP8266, chẳng hạn như độ sáng đèn LED.

Bước 2: Xử lý dữ liệu

* Sau khi đã lưu trữ dữ liệu, chúng ta có thể xử lý dữ liệu để điều khiển độ sáng đèn LED.
* Ví dụ, nếu giá trị của thanh ghi độ sáng LED là 0-255, chúng ta có thể chuyển đổi giá trị này sang dạng tín hiệu PWM để điều khiển độ sáng đèn LED.
* Chúng ta cũng có thể kiểm tra các giá trị đầu vào, kiểm tra tính hợp lệ của các giá trị, v.v.

Bước 3: Điều khiển đèn LED

* Sau khi đã xử lý dữ liệu, chúng ta có thể điều khiển độ sáng của đèn LED.
* Sử dụng các hàm của thư viện ESP8266 để điều khiển độ sáng của đèn LED, chẳng hạn như hàm analogWrite() để điều khiển tín hiệu PWM.

Sau khi thực hiện các bước trên, chúng ta có thể điều khiển độ sáng của đèn LED thông qua giao tiếp RS485 giữa các bảng ESP8266 và ESP8266. Việc này cho phép chúng ta thực hiện các tác vụ phức tạp hơn và tăng tính linh hoạt trong việc điều khiển các thiết bị.

1. **Kết luận và hướng phát triển**
2. **Tóm tắt lại các kết quả đạt được**
   * Thông qua việc kết nối và giao tiếp giữa các bảng ESP8266 và ESP8266 thông qua giao tiếp nối tiếp RS485, chúng ta đã có thể điều khiển độ sáng của đèn LED thông qua các lệnh được gửi qua mạng. Cụ thể, chúng ta đã thực hiện được các công việc sau:
   * Thiết lập kết nối nối tiếp RS485 giữa các bảng ESP8266 và ESP8266.
   * Thiết lập giao thức truyền thông ModbusRTU để điều khiển đèn LED.
   * Gửi và nhận dữ liệu giữa các bảng ESP8266 và ESP8266 thông qua giao tiếp RS485.
   * Xử lý dữ liệu để điều khiển độ sáng đèn LED thông qua các hàm của thư viện ESP8266.
   * Điều khiển độ sáng của đèn LED thông qua giao tiếp RS485 giữa các bảng ESP8266 và ESP8266.

Các kết quả này cho phép chúng ta có thể điều khiển độ sáng của đèn LED từ xa thông qua một mạng có dây, giúp tăng tính linh hoạt trong việc điều khiển các thiết bị và giảm chi phí cho việc lắp đặt và bảo trì.

1. **Những hạn chế của phương pháp này và hướng phát triển tiếp theo**
   * Mặc dù phương pháp điều khiển độ sáng đèn LED thông qua giao tiếp nối tiếp RS485 giữa các bảng ESP8266 và ESP8266 có nhiều ưu điểm, nhưng cũng còn một số hạn chế và giới hạn. Sau đây là một số hạn chế và hướng phát triển tiếp theo của phương pháp này:
   * Giới hạn về tốc độ truyền thông: Giao tiếp nối tiếp RS485 có tốc độ truyền thông khá chậm, giới hạn ở mức khoảng 115.2 kbps. Điều này có thể khiến cho việc truyền dữ liệu trở nên chậm chạp và hạn chế trong việc điều khiển các thiết bị với tần số cao.
   * Độ dài cáp giới hạn: Tín hiệu truyền qua cáp RS485 chỉ có thể truyền được trong khoảng cách giới hạn, thường là từ 1200 đến 12000 feet, phụ thuộc vào tốc độ truyền dữ liệu. Điều này có thể khiến cho việc triển khai mạng RS485 khó khăn và giới hạn trong việc điều khiển các thiết bị ở khoảng cách xa.
   * Khó khăn trong việc bảo trì và sửa chữa: Khi sử dụng giao tiếp nối tiếp RS485, việc bảo trì và sửa chữa có thể trở nên khó khăn và tốn nhiều thời gian, do yêu cầu phải kiểm tra và sửa chữa trên từng mô-đun điều khiển độc lập.
   * Hướng phát triển tiếp theo của phương pháp này có thể là sử dụng các giao tiếp mạng không dây như Wi-Fi hoặc Bluetooth để thay thế cho giao tiếp nối tiếp RS485. Điều này sẽ giúp tăng tốc độ truyền dữ liệu và giảm độ dài cáp giới hạn, giúp cho việc triển khai mạng điều khiển trở nên linh hoạt và tiện lợi hơn. Ngoài ra, cũng có thể phát triển các phần mềm và ứng dụng điều khiển đèn LED trên nền tảng web để có thể điều khiển từ xa bằng các thiết bị di động hoặc máy tính, giúp tăng tính tiện lợi và linh hoạt trong việc điều khiển các thiết bị.
2. **Tài liệu tham khảo**
   * Đặng Trần Tùng, Nguyễn Thanh Sơn. (2020). Tìm hiểu giao tiếp nối tiếp RS485 và ứng dụng điều khiển tốc độ động cơ bằng Arduino. Hội nghị khoa học và công nghệ trẻ toàn quốc lần thứ 7, Hanoi University of Industry.
   * Vũ Thanh Lai, Lê Thanh Hải, Phan Thanh Sơn. (2019). Ứng dụng giao thức Modbus trong điều khiển tốc độ động cơ bằng PLC. Kỷ yếu hội thảo quốc gia lần thứ XXVII - 2019, Trường Đại học Bách khoa Hà Nội.
   * Modbus-IDA. (2012). Modbus over serial line specification and implementation guide V1.02.
   * PyModbus Development Team. (2021). PyModbus: A Python Modbus Stack. <https://pymodbus.readthedocs.io/en/stable/>.
   * Espressif Systems. (2021). ESP8266EX Datasheet. <https://www.espressif.com/sites/default/files/documentation/esp8266ex_datasheet_en.pdf>.
   * Espressif Systems. (2021). ESP8266 Datasheet. <https://www.espressif.com/sites/default/files/documentation/ESP8266_datasheet_en.pdf>.
   * Link slide : https://www.canva.com/design/DAFfyDAskwk/bSZsbvwBelo43YbZfn56ww/edit?utm\_content=DAFfyDAskwk&utm\_campaign=designshare&utm\_medium=link2&utm\_source=sharebutton&fbclid=IwAR3Byvw0Wht4ifFiEww3Ti96EeUNFJBHnLGk8ruy0YX5dkbERm61Tlqr4QE