

Tài liệu thư viện **RGBLed**

Thư viện **RGBLed** được thiết kế để đơn giản hóa việc điều khiển dải đèn RGB NeoPixel (WS2812), phù hợp với các dự án giáo dục cho học sinh THPT. Thư viện cung cấp các hàm dễ sử dụng để hiển thị màu sắc, hiệu ứng và điều chỉnh độ sáng một cách trực quan.

Cài đặt

1. Sao chép file **rgb_library.h** và **rgb_library.cpp** vào thư mục trong dự án của bạn.
2. Thêm vào file **main.cpp**:

```
#include <rgb_library.h>
```

Cách sử dụng cơ bản

```
RGBLed led(15, 15); // Dùng chân D15, 15 bóng LED
```

```
void setup() {  
    led.begin();  
    led.showColor(COLOR_RED, 1000);  
}
```

```
void loop() {  
    led.rainbowCycle();  
    delay(1000);  
}
```

Class RGBLed

1, Khởi tạo đối tượng

```
RGBLed(uint8_t pin, uint16_t count);
```

Tham số:

- **pin**: chân tín hiệu kết nối với dải LED
- **count**: số lượng LED trong dải

2, begin()

```
void begin(uint8_t brightness = 50);
```

Khởi động dải LED và thiết lập độ sáng mặc định.

Tham số:

- **brightness**: độ sáng tổng thể (0–255)

3, showColor()

```
void showColor(RGBColor color, uint32_t duration_ms = 1500);
```

Hiển thị một màu lên toàn bộ dải LED trong khoảng thời gian nhất định.

Tham số:

- **color**: màu sắc cần hiển thị (sử dụng **RGBColor**)
- **duration_ms**: thời gian hiển thị (tính bằng mili giây)

4, rainbowCycle()

```
void rainbowCycle(uint8_t cycles = 5, uint16_t wait_ms = 10);
```

Tạo hiệu ứng cầu vồng động trên dải LED.

Tham số:

- **cycles**: số lần hiệu ứng cầu vồng chạy toàn bộ dải LED (mỗi vòng là một chu kỳ đầy đủ chuyển màu từ đầu đến cuối)
- **wait_ms**: độ trễ giữa mỗi khung hiệu ứng

5, **showAllColor()**

```
void showAllColor(uint16_t wait_ms = 500);
```

Lần lượt hiển thị toàn bộ 20 màu có sẵn trong thư viện.

Tham số:

- **wait_ms**: thời gian hiển thị mỗi màu (tính bằng mili giây)

6, **setPixelColor()**

```
void setPixelColor(uint16_t index, RGBColor color, BLEVEL brightness);
```

Thay đổi màu cho từng LED cụ thể với độ sáng chỉ định.

Tham số:

- **index**: vị trí LED cần điều khiển (bắt đầu từ 0)
- **color**: màu sắc cần hiển thị
- **brightness**: độ sáng riêng cho LED đó (**BLEVEL**)

7, **blink()**

```
void blink(RGBColor color, uint8_t times, uint16_t on_time, uint16_t off_time);
```

Nhấp nháy dải LED với màu đã chọn.

Tham số:

- **color**: màu cần nhấp nháy
- **times**: số lần nhấp nháy
- **on_time**: thời gian sáng (ms)
- **off_time**: thời gian tắt (ms)

8, **fadeInOut()**

```
void fadeInOut(RGBColor color, uint8_t steps = 20, uint16_t delay_ms = 30);
```

Tạo hiệu ứng sáng dần và tắt dần một màu.

Tham số:

- **color**: màu cần hiệu ứng
- **steps**: số bước tăng/giảm độ sáng
- **delay_ms**: độ trễ giữa các bước

9, **printColorListToSerial()**

```
void printColorListToSerial();
```

In danh sách các màu có sẵn ra Serial Monitor.

10, **turnOff()**

```
void turnOff();
```

Tắt toàn bộ LED.

11, **setBrightness()**

```
void setBrightness(uint8_t level);
```

Điều chỉnh độ sáng toàn bộ dải LED.

Tham số:

- `level`: độ sáng từ 0 (tối) đến 255 (sáng nhất)
-

Enum: RGBColor

Liệt kê sẵn 20 màu phổ biến như `COLOR_RED`, `COLOR_GREEN`, `COLOR_BLUE`, v.v.

Enum: BLEVEL

Định nghĩa các mức sáng đơn giản:

- `BRIGHTNESS_NORMAL`
 - `BRIGHTNESS_HIGH`
-