

Lesson 18 Alarm Light

18.1 Overview

In this course, we will delve into how to use the Adafruit NeoPixel library on an Arduino board to achieve precise control of WS2812 LED lights and create alarm light effects.

18.2 Principle Introduction

The warning light system utilizes the Adafruit_NeoPixel library and carefully optimized code logic to achieve efficient control and diverse lighting effects presentation of WS2812 LED lights. The specific working principle is as follows:

- 1. Communication connection and initialization:** Build a communication bridge between the Arduino board and the WS2812 LED light through the Adafruit_NeoPixel library. Firstly, specify the Arduino pins for connecting the WS2812 LED lights and accurately set the number of LED lights. In the `WS2812_Setup()` function, call the `pixels.begin()` function to complete the initialization of the Adafruit_NeoPixel library. This operation is the key foundation for subsequent stable communication and precise control. At the same time, setting the initial brightness of LED lights reasonably lays a good foundation for the subsequent display of lighting effects.
- 2. Color processing:** Users can flexibly customize the color and brightness of LED lights by calling different functional functions. In the `WS2812Color()` function, users only need to specify the target LED light number and the required RGB color value. The function will use `pixels.setPixelColor()` and `pixels.show()` functions internally to transfer accurate color data to the corresponding LED light, thereby achieving personalized color settings for individual LED lights. The `WS2812ClorAll()` function is designed specifically for batch control. It loops through all LED lights, assigns them the same RGB color values, and then calls the `pixels.show()` function to achieve synchronized color updates for all LED lights. In addition, the `WS2811_Brightness()` function provides users with a convenient interface for dynamically adjusting the brightness of LED lights, which can flexibly change the brightness of the lights according to actual needs.

18.3 Main Code

```
01 void WS2812Blink() {  
02     pinMode(A3, OUTPUT);  
03     WS2812ColorAll(255, 0, 0);
```

```
04    delay(200);  
05    WS2812ColorAll(0, 0, 0);  
06    delay(200);  
07    WS2812ColorAll(0, 0, 255);  
08    delay(200);  
09    WS2812ColorAll(0, 0, 0);  
10    delay(200);  
11 }
```

For the complete code, please refer to the "IR_Control.ino" and "Adeept_Car_For_Arduino.cpp" files.