

KEY Control RGB

KEY Control RGB

Device connection

Hardware connection

Software connection

Button control RGB

Control principle

Control pin

Code analysis

Experimental results

Switch the RGB light to display different colors by controlling the button (K1) on the Robduino expansion board.

Device connection

Hardware connection

Use Type-B data cable to connect Arduino Uno and computer.

Software connection

Open the "Arduino IDE" software and select the model and serial port number corresponding to the development board.

Button control RGB

Control principle

Associate the button value with the RGB light state. Different button values correspond to different RGB light effects.

Control pin

Peripheral module	Arduino Uno
RGB	6
K1	7

Code analysis

Here is only a brief introduction to the code content. For detailed code, it is recommended to refer to the corresponding code file, which is provided in the download area!

- Include `Adafruit_NeoPixel` library

```
#include <Adafruit_NeoPixel.h> // 包含Adafruit NeoPixel库
#include Adafruit_NeoPixel library
```

- Define RGB control pins and quantity

```
// 定义RGB控制引脚和数量 Define RGB control pins and quantity
#define RGB_PIN 6
#define RGB_NUM 1
// 定义按键引脚和控制状态 Define pin and key(button) states
#define KEY_PIN 7
#define Press_KEY 0
#define Release_KEY 1

unsigned int ikeyValue = 0;
```

- Enumerate common colors

```
// 枚举常见颜色 Enumerate common colors
enum ColorType {
    BLACK,
    RED,
    GREEN,
    BLUE,
    YELLOW,
    MAGENTA,
    CYAN,
    WHITE,
};
```

- Create an instance of the Adafruit_NeoPixel class

```
// 创建Adafruit_NeoPixel类的实例 Create an instance of the Adafruit_NeoPixel class
Adafruit_NeoPixel RGB = Adafruit_NeoPixel(RGB_NUM, RGB_PIN, NEO_GRB +
NEO_KHZ800);
```

- Set RGB display color

```
/**
 * @brief 设置RGB显示的颜色 Set RGB display color
 * @param color: 显示的颜色 Set the color
 * @retval 无 None
 */
void setRGBColor(ColorType color) {
    switch (color) {
        case RED:
            RGB.setPixelColor(0, RGB.Color(255, 0, 0));
            RGB.show();
            break;
        case GREEN:
            RGB.setPixelColor(0, RGB.Color(0, 255, 0));
            RGB.show();
            break;
        case BLUE:
            RGB.setPixelColor(0, RGB.Color(0, 0, 255));
            RGB.show();
            break;
```

```

    case YELLOW:
        RGB.setPixelColor(0, RGB.Color(255, 255, 0));
        RGB.show();
        break;
    case MAGENTA:
        RGB.setPixelColor(0, RGB.Color(255, 0, 255));
        RGB.show();
        break;
    case CYAN:
        RGB.setPixelColor(0, RGB.Color(0, 255, 255));
        RGB.show();
        break;
    case WHITE:
        RGB.setPixelColor(0, RGB.Color(255, 255, 255));
        RGB.show();
        break;
    default:
        RGB.setPixelColor(0, RGB.Color(0, 0, 0));
        RGB.show();
        break;
}
}

```

- Button detection

Note: This method will only respond to the button being pressed when the button is released, that is, it will only return to the pressed state once it is pressed and released!

```

/**
 * @brief 获取按键状态 Get key(button) status
 * @param pin: 按键控制引脚 Control key(button) pins
 * @retval 按键状态 Key(button) Status
 */
int getKeyState(uint8_t pin) {
    if (digitalRead(pin) == LOW) {
        delay(20);
        if (digitalRead(pin) == LOW) {
            while (digitalRead(pin) == LOW)
                ;
            return Press_KEY;
        }
        return Release_KEY;
    } else {
        return Release_KEY;
    }
}

```

- Initialization Code

```
void setup() {  
    pinMode(KEY_PIN, INPUT_PULLUP); // 设置按键KEY引脚上拉输入模式 Set the key(button)  
    pin to pull-up input mode  
    RGB.begin(); // 初始化RGB Initialize RGB  
    RGB.show(); // 刷新RGB显示 Refresh RGB display  
}
```

- Looping code

```
void loop() {  
    // 根据按键切换RGB灯显示 Switch the RGB light display according to the key  
    if (getKeyState(KEY_PIN) == Press_KEY) {  
        ikeyValue++;  
        ikeyValue = ikeyValue % 8;  
    }  
    setRGBColor(ikeyValue);  
}
```

Experimental results

After compiling the program successfully, upload the code to the Arduino Uno development board.

After the program starts, press and release the K1 button, and the RGB light will switch to a color!

The burning program cannot use other programs to occupy the serial port or an external serial communication module (for example: WiFi camera module), otherwise the program cannot be burned or an error message will be prompted!