RGB Breath

Control the RGB light on the Robduino expansion board to display the breathing effect of 7 common colors.

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.

RGB breathing effect

Control principle

RGB lights are composed of red, green and blue LED lights:

Through the combination and brightness control of different color LEDs, the effect of RGB lights displaying various colors can be achieved;

On this basis, we control the brightness of different color LEDs to achieve the effect of breathing lights.

Control pins

Peripheral module	Arduino Uno
RGB	6

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
```

- 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;
  }
}
```

- Set the RGB breath effect

```
/**
 * @brief 设置RGB灯呼吸效果 Set the RGB breath effect
 * @param red: 红色亮度 Red brightness
 * @param green: 绿色亮度 Green brightness
 * @param blue: 蓝色亮度 Blue brightness
 * @retval 无 None
```

```
 */
void setColorBreathShow(uint8_t red, uint8_t green, uint8_t blue) {
  for (int i = 0; i < 256; i++) {
    RGB.setPixelColor(0, RGB.Color((red * i) / 255, (green * i) / 255, (blue * i) / 255));
    RGB.show();
    delay(10);
  }

  for (int i = 255; i >= 0; i--) {
    RGB.setPixelColor(0, RGB.Color((red * i) / 255, (green * i) / 255, (blue * i) / 255));
    RGB.show();
    delay(10);
  }
}
```

- Initialization Code

```
void setup() {
  RGB.begin();  // 初始化RGB Initialize RGB
  RGB.show();   // 刷新RGB显示 Refresh RGB display
}
```

- Looping code

```
void loop() {
  setColorBreathShow(0, 0, 0);         // 黑色 BLACK
  setColorBreathShow(255, 0, 0);       // 红色 RED
  setColorBreathShow(0, 255, 0);       // 绿色 GREEN
  setColorBreathShow(0, 0, 255);       // 蓝色 BLUE
  setColorBreathShow(255, 255, 0);     // 黄色 YELLOW
  setColorBreathShow(255, 0, 255);     // 品红色 MAGENTA
  setColorBreathShow(0, 255, 255);     // 青色 CYAN
  setColorBreathShow(255, 255, 255);  // 白色 WHITE
}
```

Experimental results

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

After the program is started, the RGB light will display 7 common colors of breathing effects in turn!

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