RGB Control

Control the RGB lights on the Robduino expansion board to display different colors.

Device connection

Hardware connection

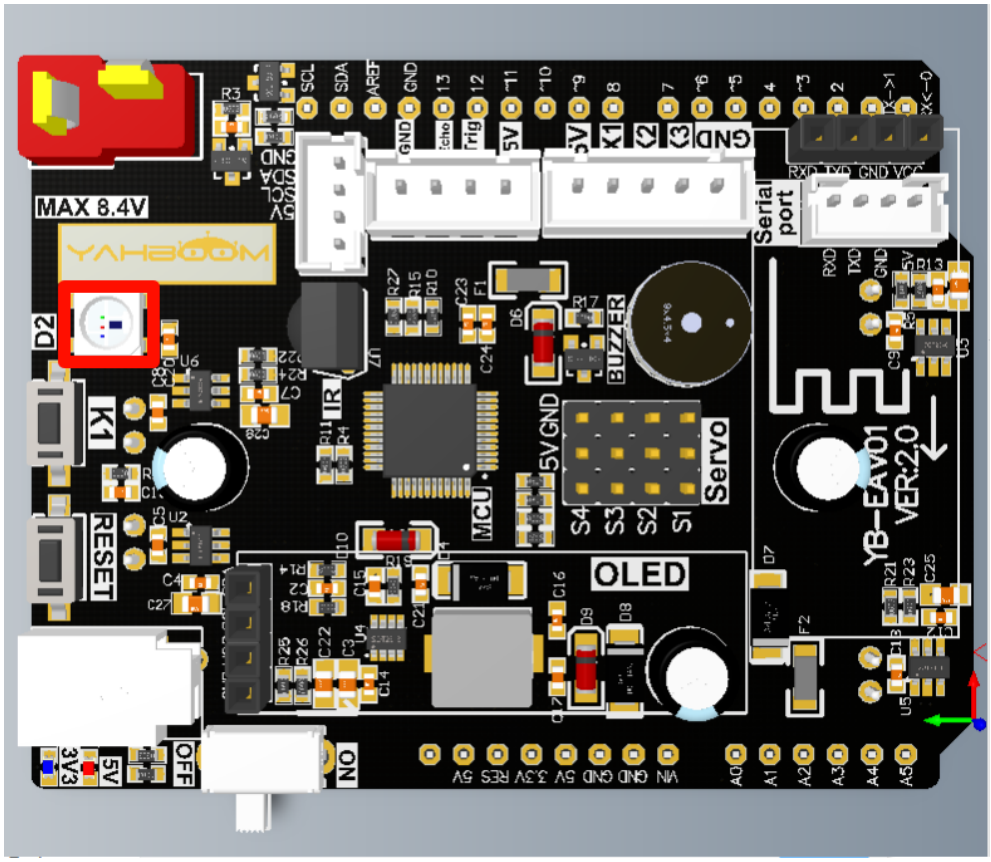
Use a Type-B data cable to connect Arduino Uno and the computer.

Software connection

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

Control RGB

The location selected by the red box is the location of the RGB light module:



Control Principle

RGB lamp is composed of three colors of LED lamps: red, green and blue. By combining different color LEDs and controlling the brightness, the RGB lamp can display various colors.

Control Pin

Peripheral Module	Arduino Uno
RGB	6

Code Analysis

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

- Included `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 the color of the RGB display

```
/**
 * @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;
  }
}
```

- 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() {
  // 切换RGB显示颜色 Switch the color of RGB display
  for (int iRgbState = 0; iRgbState < 8; iRgbState++) {
    setRGBColor(iRgbState);
    delay(1000);
  }
}
```

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 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!
```