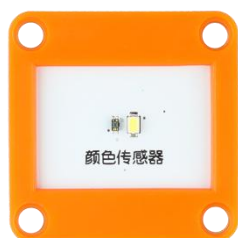


颜色传感器VEML6040用户手册



版本V1.0

目录

模块概述.....	3
简介.....	3
规格参数.....	3
芯片说明.....	3
外壳尺寸, 单位MM.....	4
引脚说明.....	5
应用示例-Arduino UNO.....	5
硬件及连接.....	5
程序及结果-Arduino IDE.....	6
程序及结果-Mind+.....	7
程序及结果-Mixly 暂未支持.....	8
应用示例-MicroBit.....	8
硬件及连接.....	8
程序及结果-Mind+.....	8
应用示例-掌控板.....	9

硬件及连接	9
程序及结果-Mind+	10
应用示例-ESPONE主板	10
硬件及连接	10
程序及结果-Mind+	11
附录	13
附录1-UNO接口说明	13
附录2-Arduino如何导入库?	13
附录3-Mind+如何导入扩展库?	14
附录4-Mixly如何导入扩展库?	14
版本说明	15
联系我们	15

模块概述

简介

颜色传感器，采用VEML6040颜色传感器芯片设计制作。通过I2C总线接口通信，可提供红光、绿光、蓝光和白光的检测与输出。

芯片采用 Filtron™ 技术实现了对红色、绿色和蓝色通道每个请求波段的中间的准确响应。此外，它还提供出色的温度补偿，在不断变化的温度下保持输出稳定。

颜色传感器具有统一的兼容乐高积木的外壳，可轻松完成乐高积木的拼接，实现创意设计。

重要提示：颜色的识别与环境的色温、亮度，被测物体的材质，检测距离、角度都有关联；如发现检测颜色结果与实际有差别属于正常情况，可以通过**更换检测材料、测量距离**等测试条件进行调节；或者通过**优化程序**来调节输出结果，可以参考：[为什么我的颜色传感器不准确？颜色传感器的正确使用方法？](#)

规格参数

工作电压：DC 3.3-5V

IC：VEML6040

通信方式：I2C，地址0x10

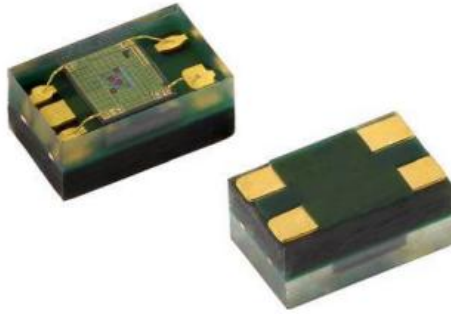
检测距离：1cm

每个通道（R、G、B、W）提供 16 位分辨率

最大检测范围（515.4、1031、2062、4124、8248 或 16496）勒克斯，

最高灵敏度为 0.007865 勒克斯/步

芯片说明



VEML6040是颜色传感器可感测红光、绿光、蓝光和白光，并将光电二极管、放大器和模拟/数字电路集成到单个CMOS芯片中。这种数字RGBW信息可用于反馈控制系统等，以监测和主动控制光源。例如，应用颜色微传感器后，可以根据环境光条件调整背光的亮度和色温，使面板看起来更舒适。

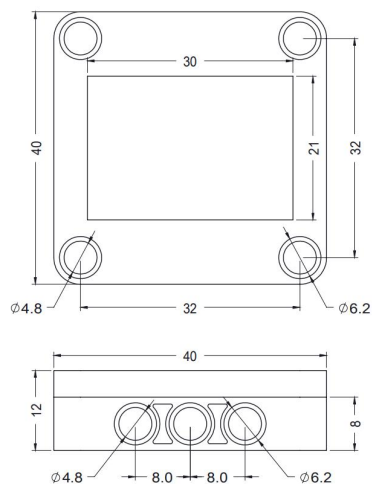
VEML6040采用 Filtron™ 技术实现了对红色、绿色和蓝色通道每个请求波段的中间的准确响应。此外，它还提供出色的温度补偿，在不断变化的温度下保持输出稳定。

特征

- 封装类型：表面贴装（封装：OPLGA）
- 尺寸（长 x 宽 x 高，毫米）：2.0 x 1.25 x 1.0
- 集成模块：颜色传感器（RGBW）和信号调理集成电路
- Filtron™ 技术提供与真实人眼反应相匹配的光谱
- 支持低透光率（深色）镜片设计
- 抗荧光灯闪烁
- 为每个通道（R、G、B、W）提供 16 位分辨率
- 可选择的最大检测范围（515.4、1031、2062、4124、8248 或 16496）勒克斯，最高灵敏度为 0.007865 勒克斯/步
- 温度补偿：-40 °C 至 +85 °C
- 输出类型：I2C总线，低功耗

更多VEML6040详情请参考数据手册。

外壳尺寸，单位MM



引脚说明



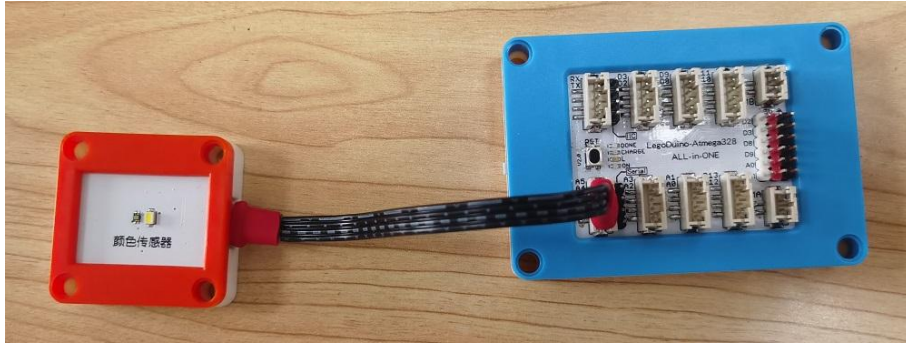
应用示例-Arduino UNO

硬件及连接

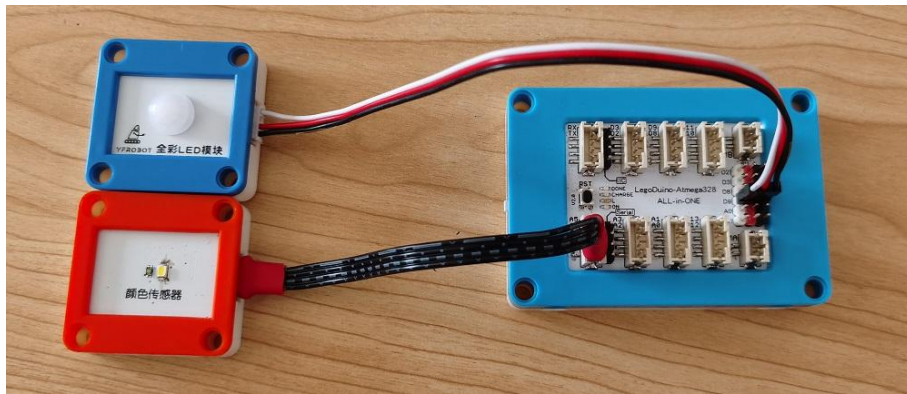
⚠ 注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！

Legoduino Pro主板、颜色传感器、RGB彩灯模块、线材。

颜色传感器的 G、V、SDA、SCL 分别连接 主板的 IIC 端口 GND、VCC、SDA(A4)、SCL(A5) 引脚。



彩灯模块的 G、V、NC(空引脚)、S 分别连接 主板的GND、VCC、D9引脚。

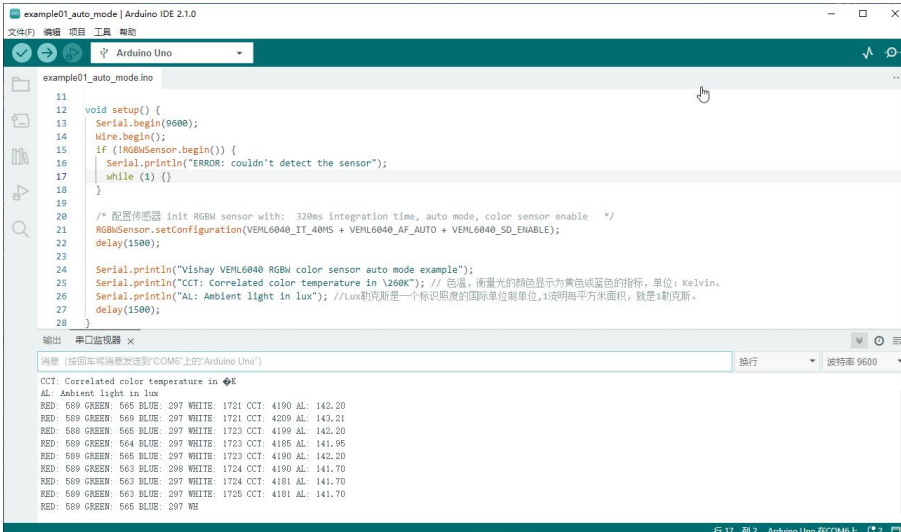


程序及结果-Arduino IDE

⚠ 注意：程序需要添加库文件“VEML6040-arduino-master.zip”，否则无法正常编译，添加方式见附录2。

例程一、识别颜色

打开例程“example01_auto_mode”编译上传。



```

example01_auto_mode.ino
11
12 void setup() {
13   Serial.begin(9600);
14   Wire.begin();
15   if (!RGBWSensor.begin()) {
16     Serial.println("ERROR: couldn't detect the sensor");
17     while (1) {}
18   }
19
20   /* 配置传感器 init RGBW sensor with: 320ms integration time, auto mode, color sensor enable */
21   RGBWSensor.setConfiguration(VEML6040_TI_60MS + VEML6040_AF_AUTO + VEML6040_SD_ENABLE);
22   delay(1500);
23
24   Serial.println("Vishay VEML6040 RGBW color sensor auto mode example");
25   Serial.println("CCT: Correlated color temperature in \260K"); // 色温, 衡量光的颜色显示为黄色或蓝色的指标, 单位: Kelvin.
26   Serial.println("AL: Ambient light in lux"); //Lux勒克斯是一个标识照度的国际单位制单位,1流明每平方米面积,就是1勒克斯.
27   delay(1500);
28 }

```

输出 串口监视器 ×

消息 (按回车将消息发送到“COM6”上的“Arduino Uno”)

```

CCT: Correlated color temperature in 405
AL: Ambient light in lux
RED: 589 GREEN: 565 BLUE: 297 WHITE: 1721 CCT: 4190 AL: 142.20
RED: 589 GREEN: 569 BLUE: 297 WHITE: 1721 CCT: 4209 AL: 143.21
RED: 588 GREEN: 565 BLUE: 297 WHITE: 1723 CCT: 4199 AL: 142.20
RED: 589 GREEN: 564 BLUE: 297 WHITE: 1723 CCT: 4185 AL: 141.95
RED: 588 GREEN: 565 BLUE: 297 WHITE: 1723 CCT: 4190 AL: 142.20
RED: 589 GREEN: 563 BLUE: 298 WHITE: 1724 CCT: 4190 AL: 141.70
RED: 589 GREEN: 563 BLUE: 297 WHITE: 1724 CCT: 4181 AL: 141.70
RED: 589 GREEN: 563 BLUE: 297 WHITE: 1725 CCT: 4181 AL: 141.70
RED: 589 GREEN: 565 BLUE: 297 WH

```

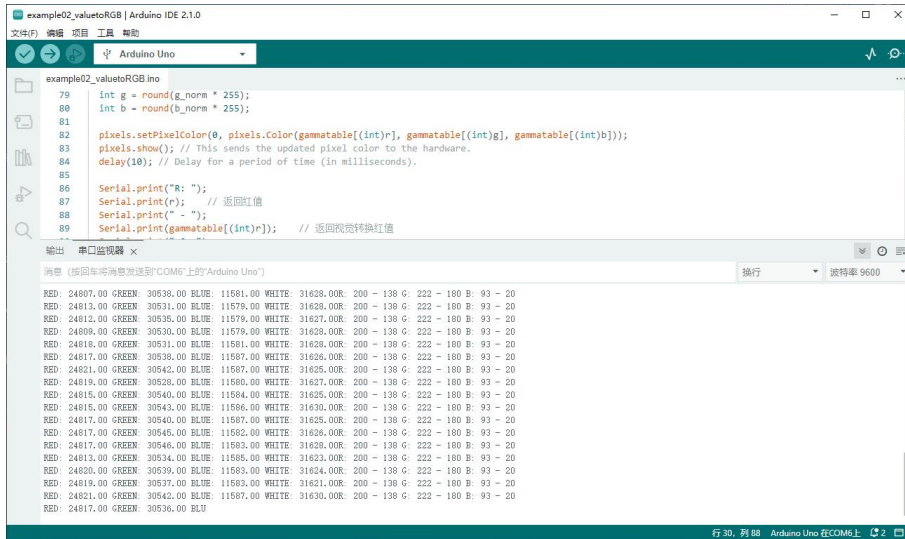
行 17, 列 2. Arduino Uno: E2COM6.L

运行结果

观察上图中，串口打印识别到的颜色原始值：RED红色，GREEN绿色，BLUE蓝色，WHITE白色，CCT色温，AL环境光亮度（勒克斯）。

例程二、识别颜色并点亮RGB LED模块

打开例程“example02_valuetoRGB”编译上传。



```

example02_valuetoRGB.ino
79 int g = round(g_norm * 255);
80 int b = round(b_norm * 255);
81
82 pixels.setPixelColor(0, pixels.Color(gammatable[(int)r], gammatable[(int)g], gammatable[(int)b]));
83 pixels.show(); // This sends the updated pixel color to the hardware.
84 delay(10); // Delay for a period of time (in milliseconds).
85
86 Serial.print("R: ");
87 Serial.print(r); // 返回红值
88 Serial.print(" - ");
89 Serial.print(gammatable[(int)r]); // 返回视觉转换红值

```

输出 串口监视器 x

消息 (按回车将消息发送到“COM6”上的“Arduino Uno”)

```

RED: 24897.00 GREEN: 30538.00 BLUE: 11581.00 WHITE: 31628.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24813.00 GREEN: 30531.00 BLUE: 11579.00 WHITE: 31628.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24812.00 GREEN: 30535.00 BLUE: 11579.00 WHITE: 31627.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24809.00 GREEN: 30530.00 BLUE: 11579.00 WHITE: 31628.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24818.00 GREEN: 30531.00 BLUE: 11581.00 WHITE: 31628.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24817.00 GREEN: 30538.00 BLUE: 11587.00 WHITE: 31626.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24821.00 GREEN: 30542.00 BLUE: 11587.00 WHITE: 31625.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24819.00 GREEN: 30528.00 BLUE: 11580.00 WHITE: 31627.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24815.00 GREEN: 30540.00 BLUE: 11584.00 WHITE: 31625.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24815.00 GREEN: 30543.00 BLUE: 11586.00 WHITE: 31630.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24817.00 GREEN: 30540.00 BLUE: 11587.00 WHITE: 31625.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24817.00 GREEN: 30545.00 BLUE: 11582.00 WHITE: 31626.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24817.00 GREEN: 30546.00 BLUE: 11583.00 WHITE: 31628.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24813.00 GREEN: 30534.00 BLUE: 11585.00 WHITE: 31623.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24820.00 GREEN: 30539.00 BLUE: 11583.00 WHITE: 31624.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24819.00 GREEN: 30537.00 BLUE: 11583.00 WHITE: 31621.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24821.00 GREEN: 30542.00 BLUE: 11587.00 WHITE: 31630.00R: 200 - 138 G: 222 - 180 B: 93 - 20
RED: 24817.00 GREEN: 30536.00 BLU

```

行 30, 列 88 Arduino Uno (COM6) 行 30, 列 88

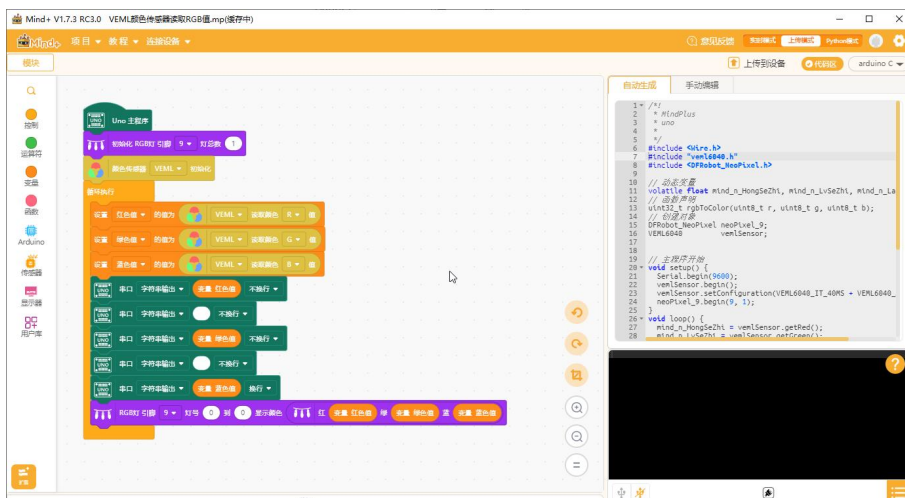
运行结果

观察上图中，串口打印识别到的颜色原始值：RED红色，GREEN绿色，BLUE蓝色，WHITE白色，R(255)颜色值，G(255)颜色值，B(255)颜色值；通过识别转换后的颜色值，反映点亮RGB彩灯模块。

程序及结果-Mind+

例程一、识别颜色并点亮RGB LED模块

打开例程“VEML颜色传感器读取RGB值.mp”编译上传。



```

1 //
2 #include <Arduino.h>
3 #include <Adafruit_NeoPixel.h>
4 #include <Adafruit_VEM_L6040.h>
5 #include <Adafruit_NeoPixel.h>
6
7 // 定义引脚
8 volatile float mind_n_hongse2h1, mind_n_luse2h1, mind_n_la
9 // 定义引脚
10 uint32_t rgbToColor(uint8_t r, uint8_t g, uint8_t b);
11 // 定义引脚
12 #define R_PIN 13
13 #define G_PIN 12
14 #define B_PIN 11
15 #define VEML6040_A0 VEML6040_A0
16 #define VEML6040_A1 VEML6040_A1
17
18 // 主程序开始
19 void setup() {
20   Serial.begin(9600);
21   pinMode(R_PIN, OUTPUT);
22   pinMode(G_PIN, OUTPUT);
23   pinMode(B_PIN, OUTPUT);
24   pinMode(VEML6040_A0, INPUT);
25   pinMode(VEML6040_A1, INPUT);
26 }
27 void loop() {
28   mind_n_hongse2h1 = veML6040.getRed();
29   mind_n_luse2h1 = veML6040.getGreen();

```


运行结果

观察串口打印识别到的颜色原始值: R(255)颜色值, G(255)颜色值, B(255)颜色值; 通过识别转换后的颜色值, 反映点亮RGB彩灯模块。

更多YFRobot扩展库参考: [YFRobot库 For Mind+ - YFRobotwiki](#)

程序及结果-Mixly 暂未支持

更多YFRobot扩展库参考: [YFRobot三方库ForMixly1.0/2.0 - YFRobotwiki](#)

应用示例-MicroBit

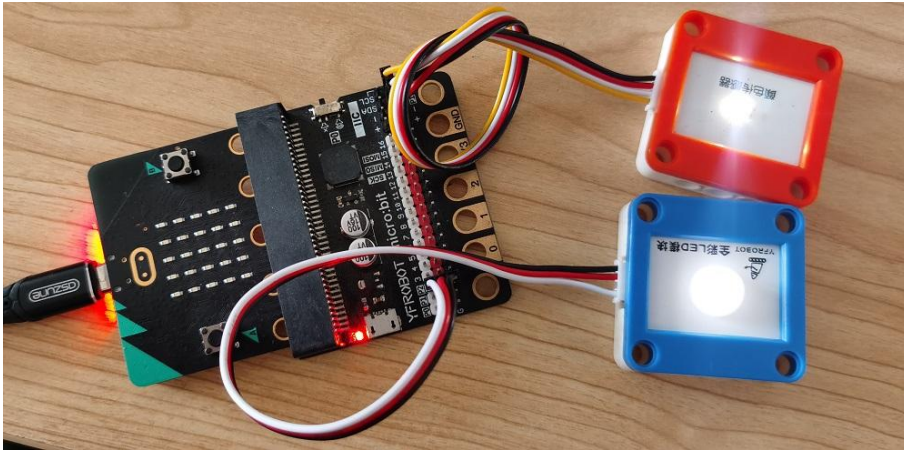
硬件及连接

⚠ 注意: 硬件应用于示例演示, 可能需要另购; 如有不明请咨询本司客服!

Micro:Bit主板、颜色传感器、彩灯模块、线材。

颜色传感器的 G、V、SDA、SCL分别连接 Micro:Bit的IIC端口GND、VCC、SDA(P20)、SCL(P19)引脚。

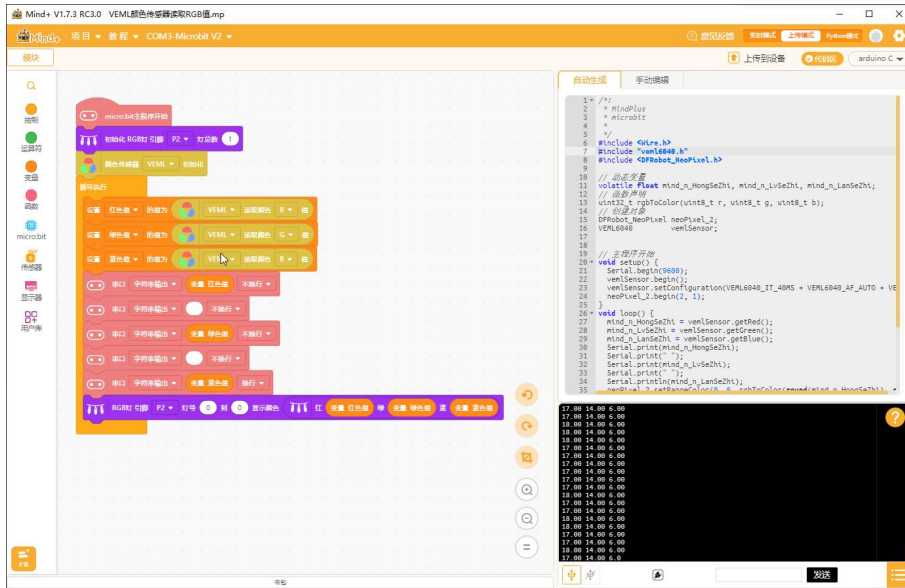
彩灯模块的 G、V、NC(空引脚)、S分别连接 主板的GND、VCC、P1引脚。



程序及结果-Mind+

例程一、识别颜色串口打印输出, 点亮RGB

打开例程 “VEML颜色传感器读取RGB值.mp” 上传。



运行结果

观察串口打印识别到的颜色值：R(255)颜色值，G(255)颜色值，B(255)颜色值；通过颜色值，反映点亮RGB彩灯模块。

更多YFRobot扩展库参考：[YFRobot库 For Mind+ - YFRobotwiki](#)

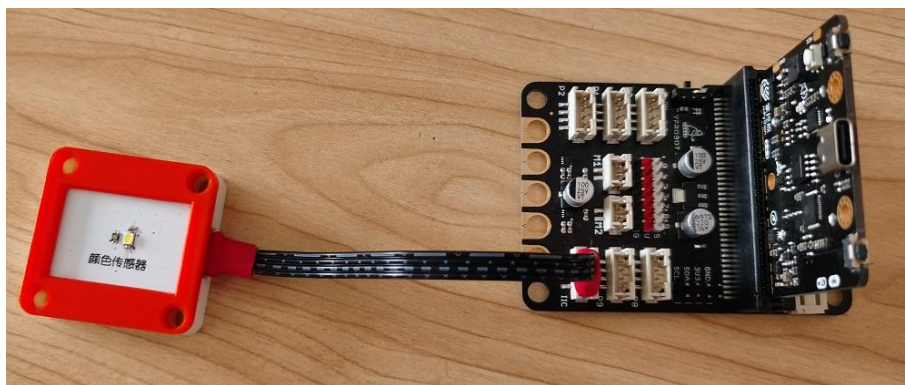
应用示例-掌控板

硬件及连接

⚠ 注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！

掌控板、颜色传感器、线材。

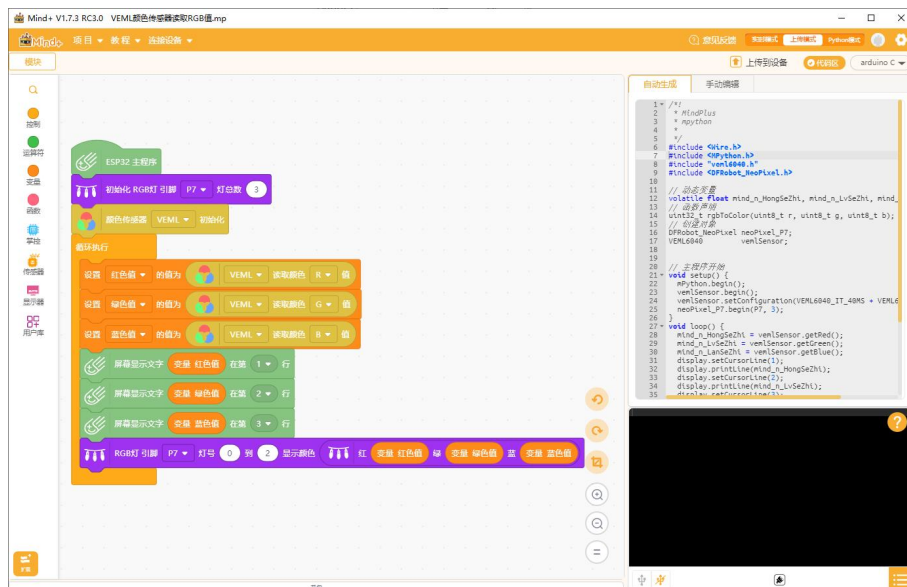
颜色传感器的 G、V、SDA、SCL 分别连接掌控板的 IIC 端口 GND、VCC、SDA(P20)、SCL(P19) 引脚。



程序及结果-Mind+

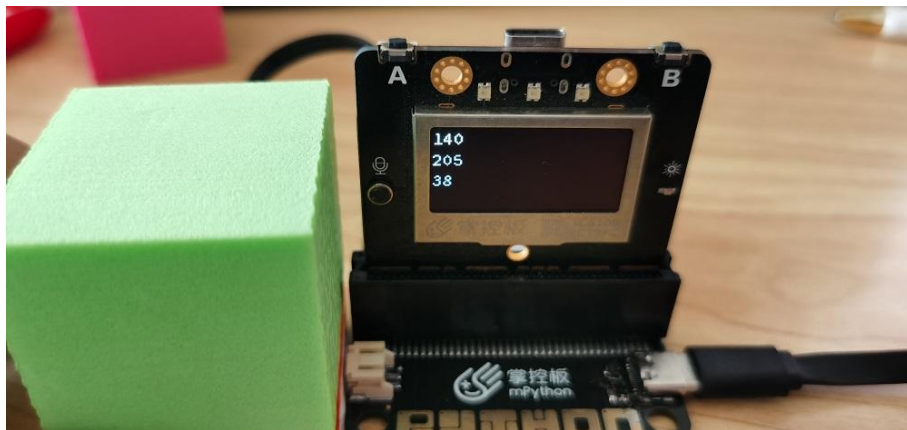
例程一、识别颜色串口打印输出，点亮RGB

打开例程“VEML颜色传感器读取RGB值.mp”上传。



运行结果

观察屏幕显示识别到的颜色值：R(255)颜色值，G(255)颜色值，B(255)颜色值；通过颜色值，反映点亮板载RGB彩灯模块。



更多YFRobot扩展库参考：[YFRobot库 For Mind+ - YFRobotwiki](#)

应用示例-ESPONE主板

硬件及连接

⚠ 注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本公司客服！

ESPONE主板、颜色传感器、彩灯模块、线材。

颜色传感器的 G、V、SDA、SCL分别连接掌控板的IIC端口GND、VCC、SDA(P20)、SCL(P19)引脚。

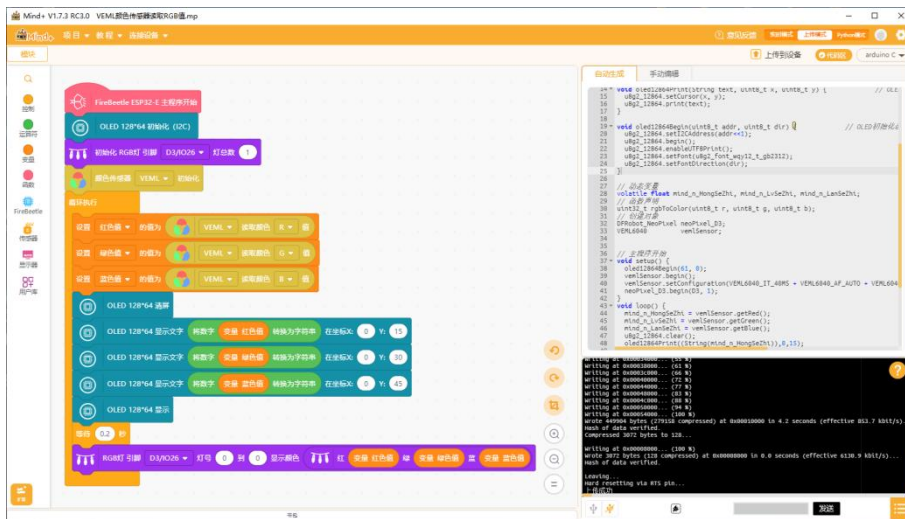
彩灯模块的 G、V、NC(空引脚)、S分别连接 主板的GND、VCC、P1引脚。



程序及结果-Mind+

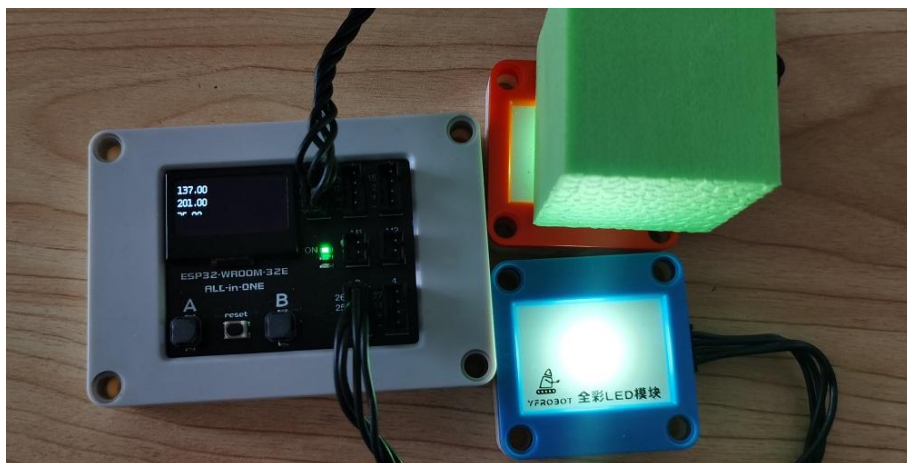
例程一、识别颜色串口打印输出，点亮RGB

打开例程“VEML颜色传感器读取RGB值.mp”上传。



运行结果

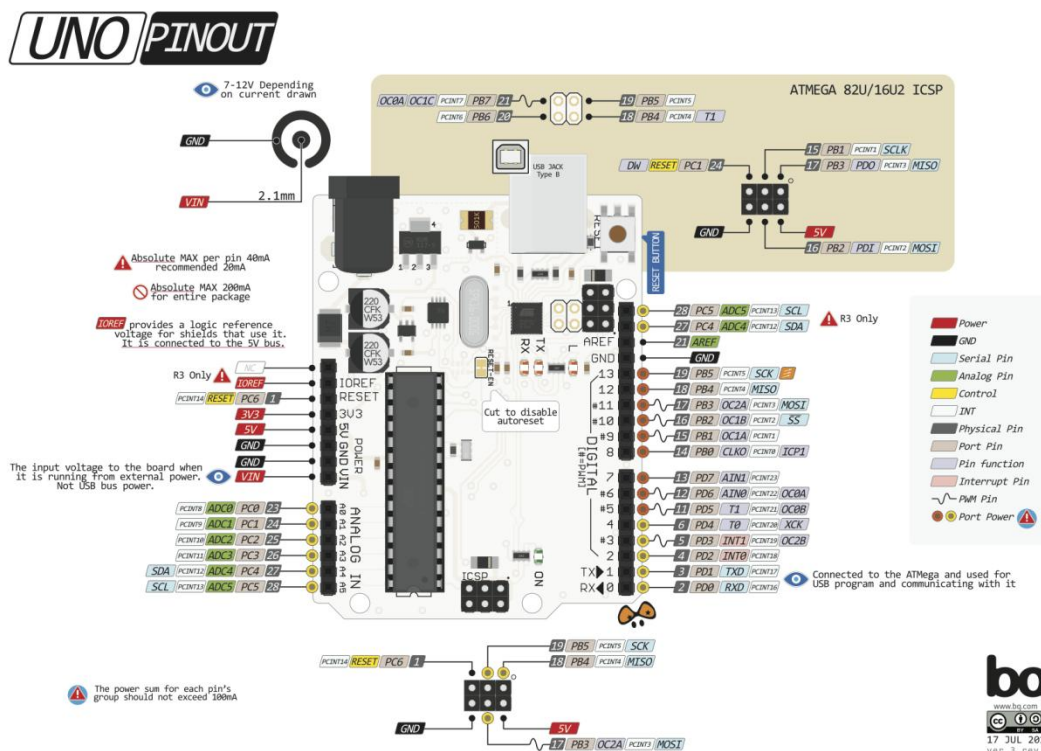
观察屏幕显示识别到的颜色值：R(255)颜色值，G(255)颜色值，B(255)颜色值；通过颜色值，反映点亮板载RGB彩灯模块。



更多YFRobot扩展库参考: [YFRobot库 For Mind+ - YFRobotwiki](#)

附录

附录1-UNO接口说明



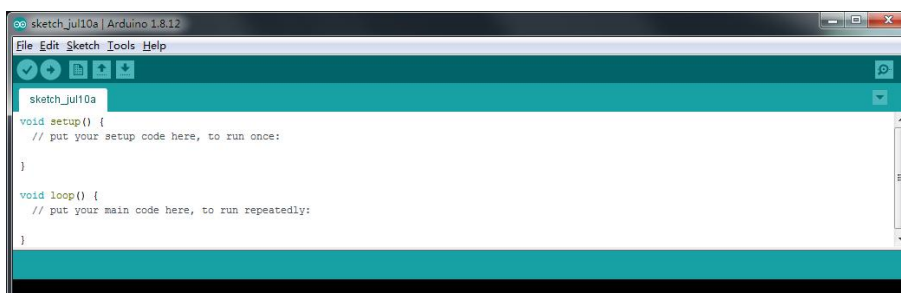
注：UNO官方版本和兼容版本大部分功能都相同

附录2-Arduino如何导入库？

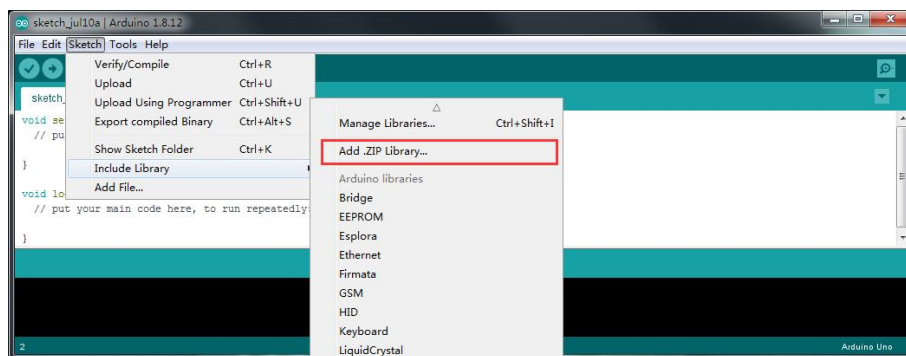
教程中有些需要使用库；如何将其导入到自己的Arduino IDE编译器中？

Arduino IDE导入库视频教程：点击[视频链接](#)查看。

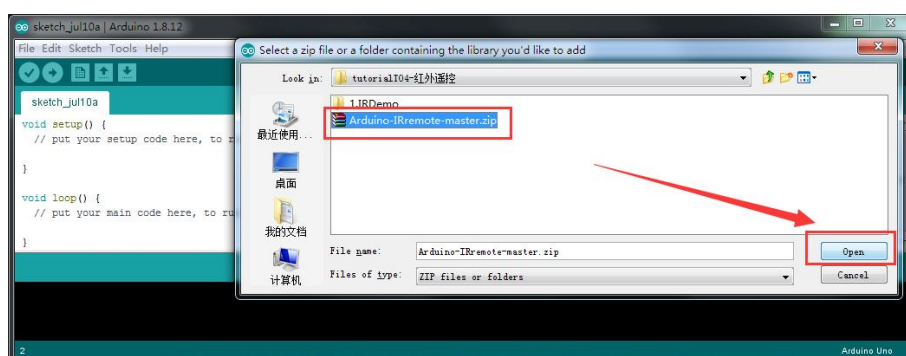
➤ 第一步：打开ARDUINO IDE



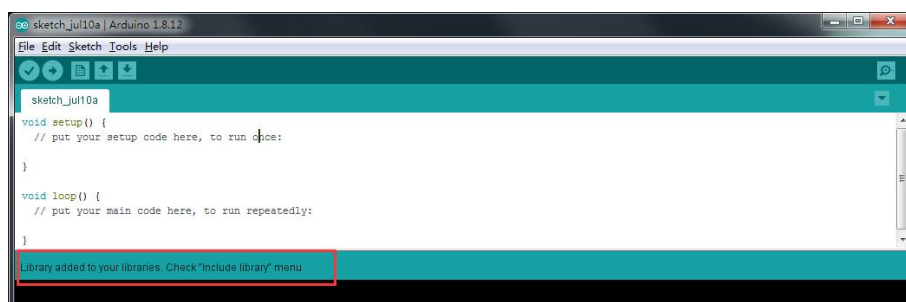
➤ 第二步：Sketch > Include Library > Add ZIP Library...



- 第三步：找到需要添加的zip格式库文件，选择并点击打开（Open）



- 第四步：导入成功后提示



附录3-Mind+如何导入扩展库？

Mind+图形化软件导入扩展库视频教程：点击[链接](#)。

附录4-Mixly如何导入扩展库？

Mixly图形化软件导入扩展库视频教程：点击[链接](#)。

版本说明

版本	日期	内容
V1.0	20230510	初版

联系我们

YFROBOT网站	www.yfrobot.com / www.yfrobot.com.cn		
手机	17696701116 (微信/QQ同号)		
QQ群	243067479		
邮件	yfrobot@qq.com		
技术微信二维码		微信公众号	YFRobotStudio 

免责声明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各所所有者的财产，特此声明。

版权归 © 2023 YFROBOT 所有。保留所有权利

