Project 9 - Infrared Control WS2812B RGB Light

1. project description

Through this project, you can learn how to use an infrared remote control to control the WS2812B LED module to emit different colors . This function is: when you press "1" on the remote control, the WS2812 B displays a red light, when you press "2" on the remote control, the WS2812 B displays a green light, and when you press "3" on the remote control, the WS2812 B displays a blue light. When you press " 4 " on the remote control , the WS2812 displays white lights. When you press " 5 " on the remote control, all the lights of WS2812 B turn off . When you press " 6 " on the remote control , the WS2812 displays red, green and blue running water lights in sequence .

2. Introduction to modules

2.1 Infrared remote control kit

Infrared detectors are tiny chips with photocells that are tuned to monitor infrared light. They are almost always used for remote control detection - there is one on the front of every TV and DVD player that monitors the infrared signal from the remote control. There is a matching IR LED inside the remote that emits IR pulses to tell the TV to turn on, off, or change channels. Infrared light cannot be seen by the human eye, which means testing the setup requires more work.

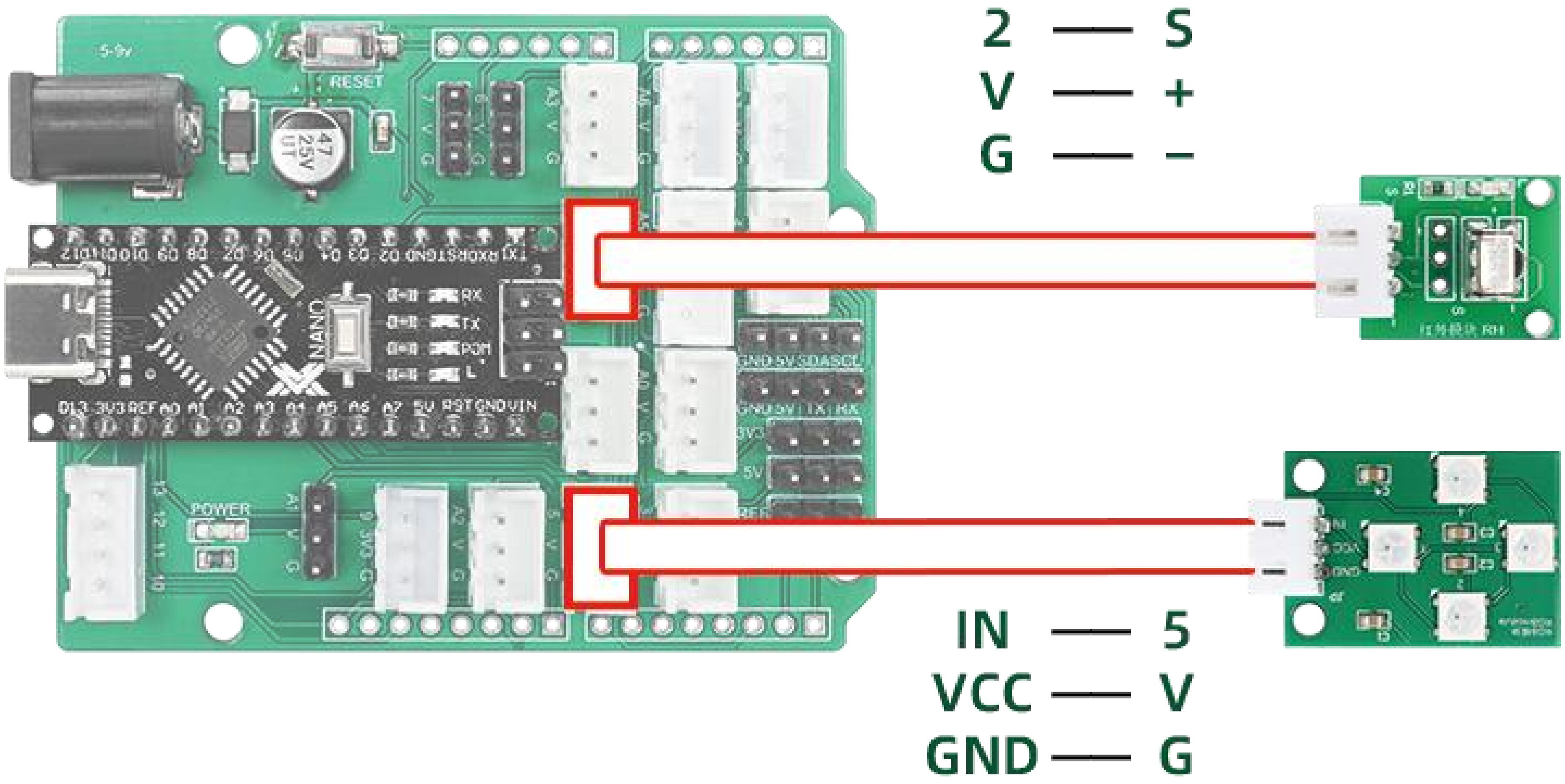
There are some differences between these and CdS phototubes:

IR detectors are specifically filtered for IR light, they are not good at detecting visible light. Photocells, on the other hand, are good at detecting yellow/green visible light, but not so good at detecting infrared light.

There is a demodulator inside the IR detector that looks for modulated IR at 38 KHz. The infrared LED that just glows will not be detected, it must flash at 38KHz. Photocells do not have any type of demodulator and can detect any frequency (including DC) within the photocell response speed (about 1KHz)

IR detectors are digital outputs - they either detect a 38KHz IR signal and output a low level (0V), or they detect no signal and output a high level (5V). Photocells are like resistors in that the resistance changes depending on how much light they are exposed to.

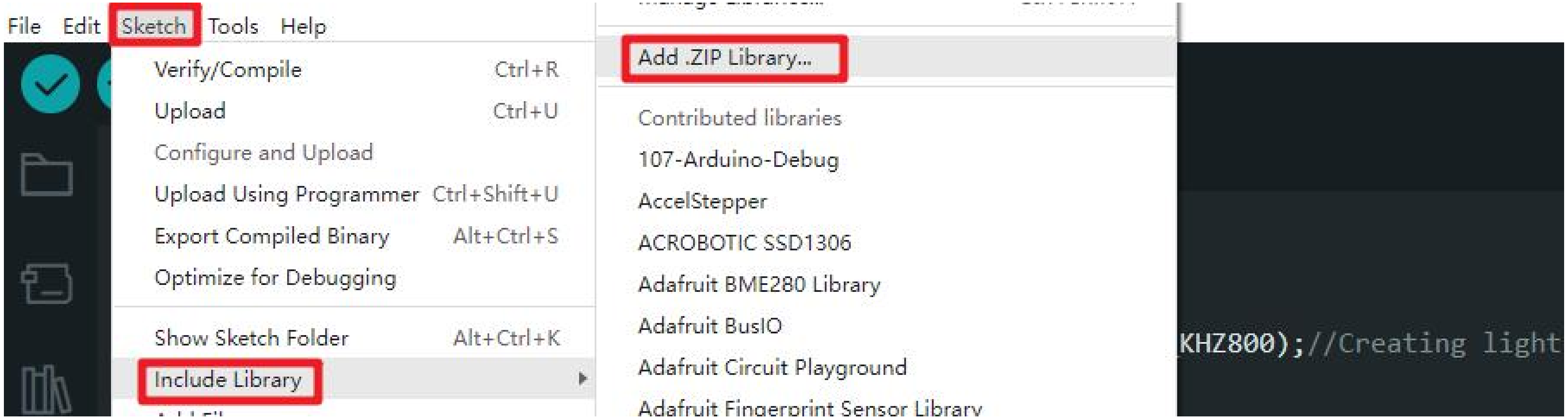
3. Project wiring diagram



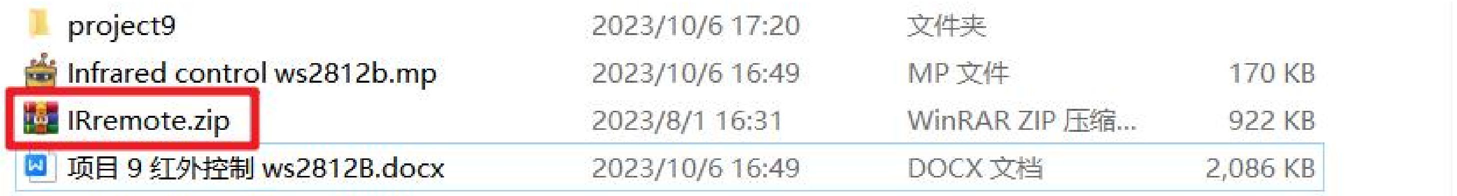
4. Import IRremote .zip library

Here, the infrared remote control uses *the IRremote library* .

In the Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library and at the top of the drop-down list, select the "Add .ZIP Library" option.

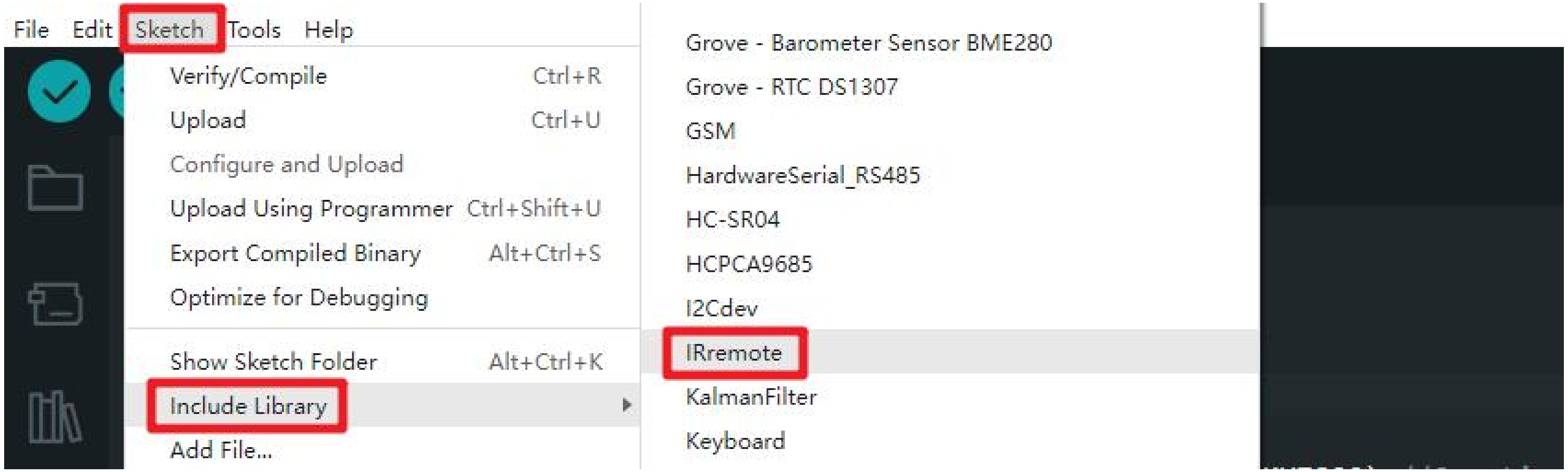


The system will prompt you to select the library to be added , as shown below, navigate to the path location of the saved *IRremote* .zip file on your computer (*Item 9 Infrared Control ws2812B\IRremote.zip*) and open it .



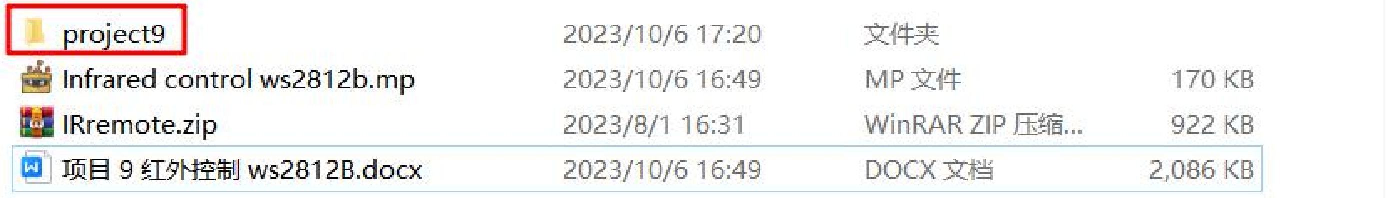
project9	2023/10/6 17:20	文件夹	
Infrared control ws2812b.mp	2023/10/6 16:49	MP 文件	170 KB
IRremote.zip	2023/8/1 16:31	WinRAR ZIP 压缩...	922 KB
项目 9 红外控制 ws2812B.docx	2023/10/6 16:49	DOCX 文档	2,086 KB

Open the Sketch > Include Library menu. You should now see Libraries at the bottom of the drop-down menu. It's ready to use in your sketches .



5. Download Arduino code

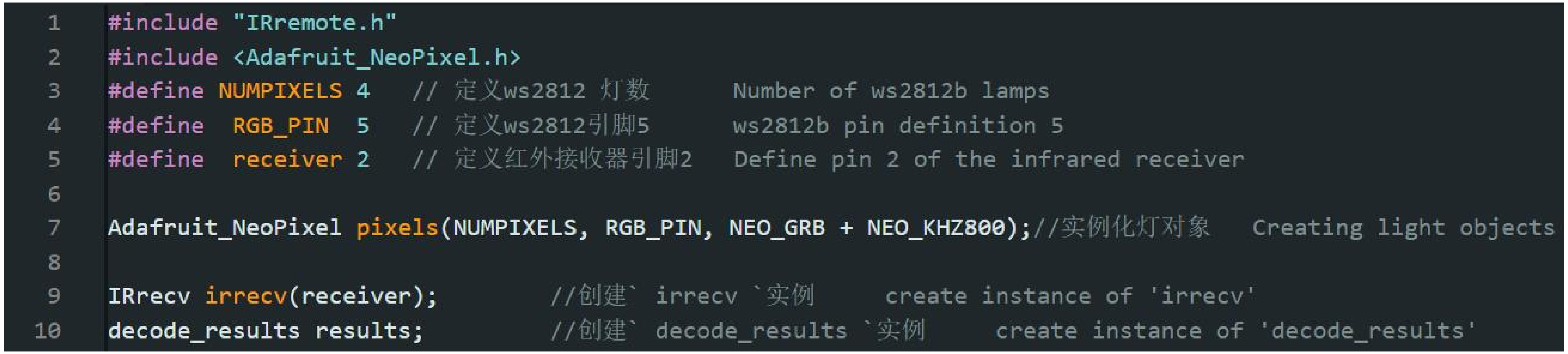
Open the project Arduino code file (path: project 9 infrared control ws2812B\project9\project9.ino)



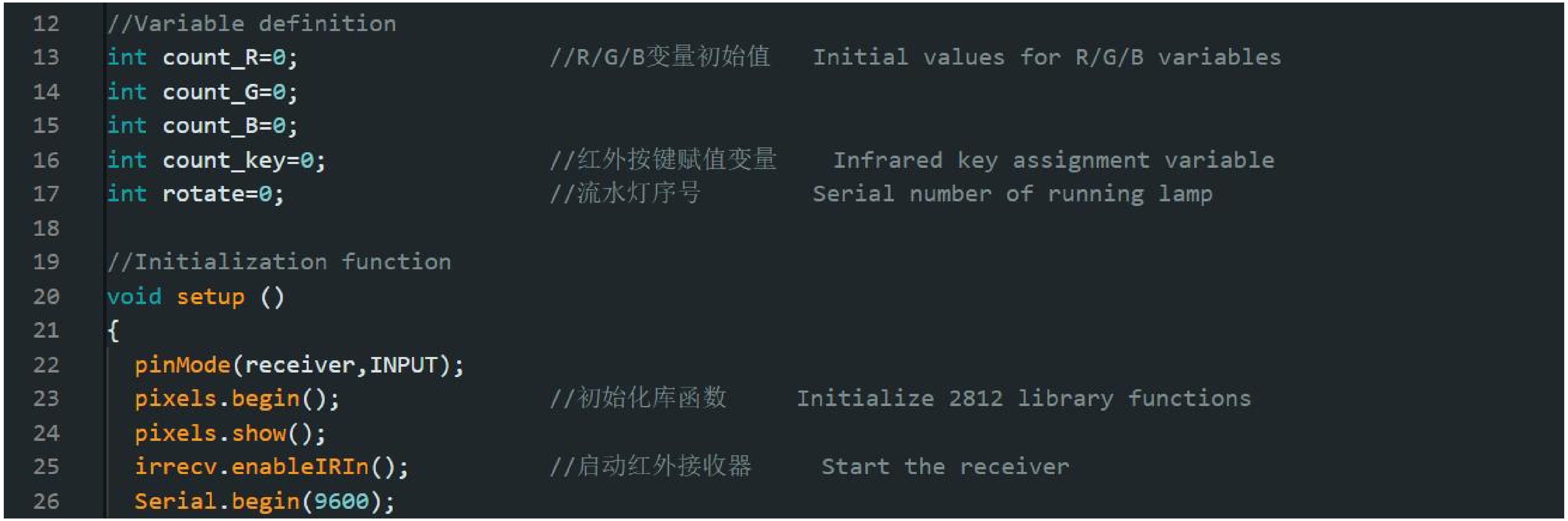
project9	2023/10/6 17:20	文件夹	
Infrared control ws2812b.mp	2023/10/6 16:49	MP 文件	170 KB
IRremote.zip	2023/8/1 16:31	WinRAR ZIP 压缩...	922 KB
项目 9 红外控制 ws2812B.docx	2023/10/6 16:49	DOCX 文档	2,086 KB

Connect the main control board to the computer using USB, select the board type as Nano, select the newly displayed COM number, click "Download" to start compiling and downloading the program to the main control board.

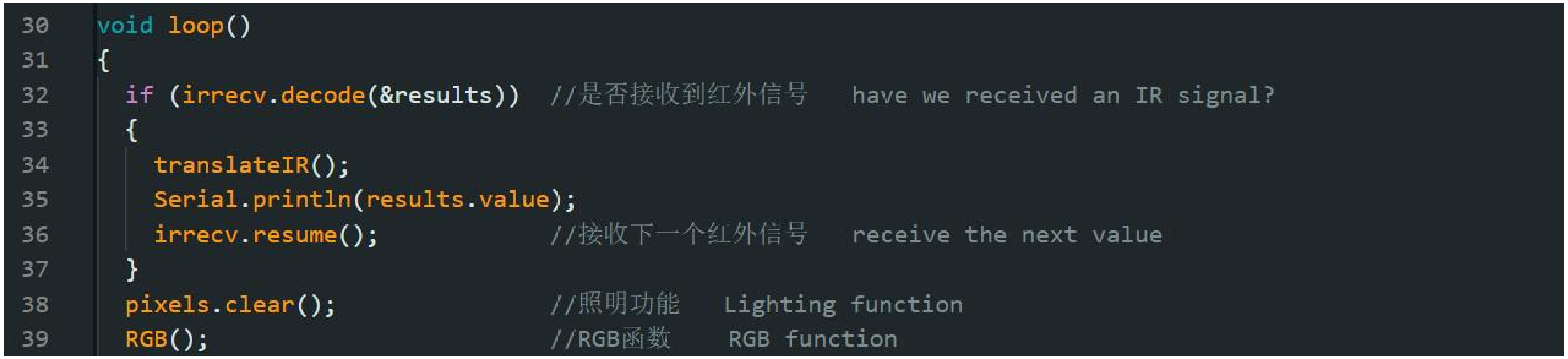
Code analysis:



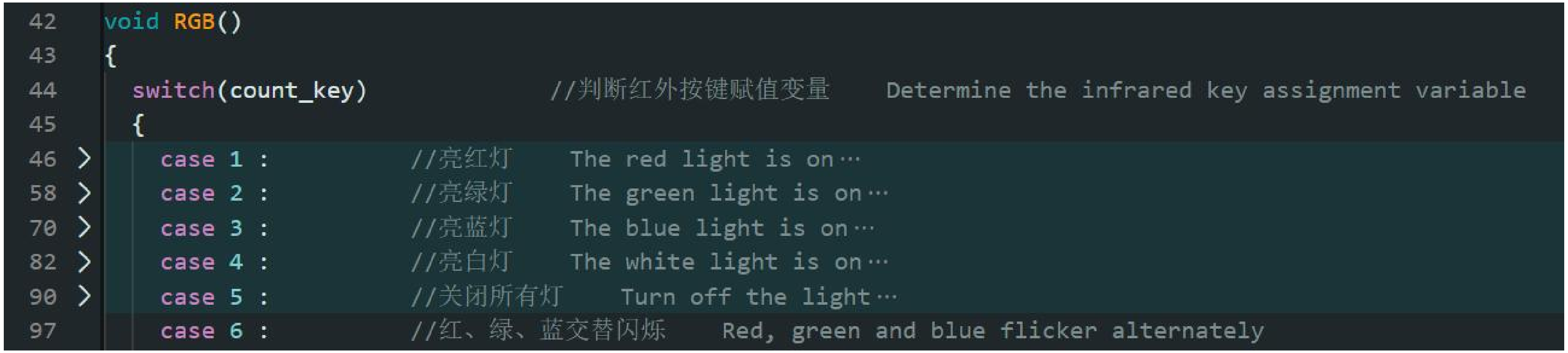
```
#include "IRremote.h"
#include <Adafruit_NeoPixel.h>
#define NUMPIXELS 4   // 定义ws2812 灯数      Number of ws2812b lamps
#define  RGB_PIN   5  // 定义ws2812引脚5      ws2812b pin definition 5
#define  receiver 2   // 定义红外接收器引脚2   Define pin 2 of the infrared receiver

Adafruit_NeoPixel pixels(NUMPIXELS, RGB_PIN, NEO_GRB + NEO_KHZ800);//实例化灯对象   Creating light objects

IRrecv irrecv(receiver);        //创建` irrecv `实例     create instance of 'irrecv'
decode_results results;         //创建` decode_results `实例     create instance of 'decode_results'
```



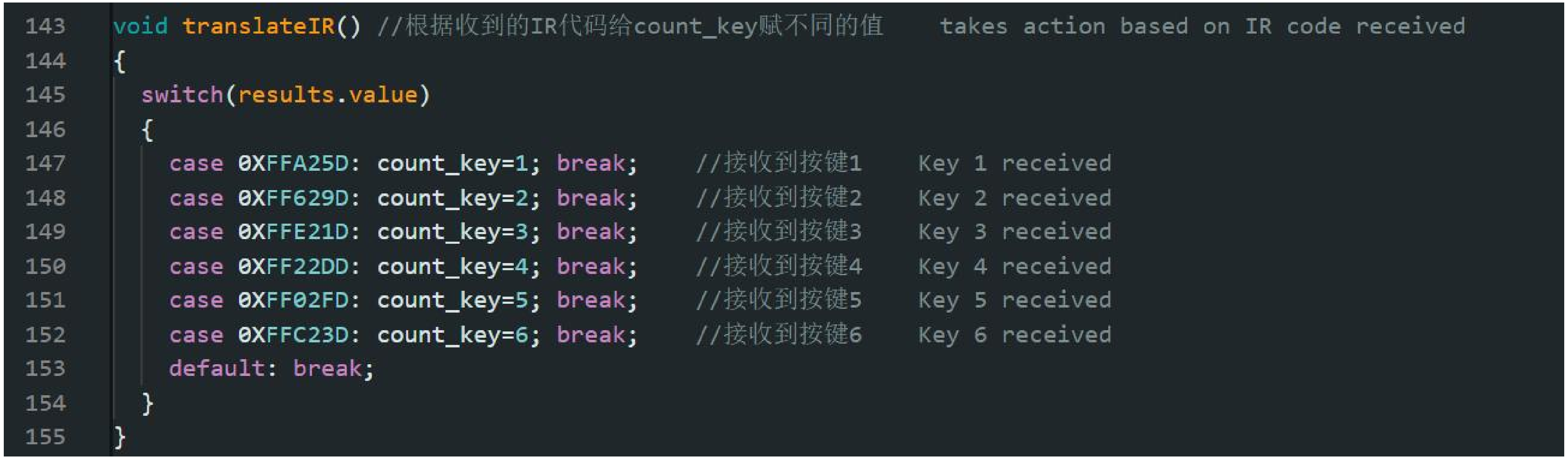
```
//Variable definition
int count_R=0;                  //R/G/B变量初始值   Initial values for R/G/B variables
int count_G=0;
int count_B=0;
int count_key=0;                //红外按键赋值变量    Infrared key assignment variable
int rotate=0;                   //流水灯序号         Serial number of running lamp

//Initialization function
void setup ()
{
  pinMode(receiver,INPUT);
  pixels.begin();               //初始化库函数      Initialize 2812 library functions
  pixels.show();
  irrecv.enableIRIn();          //启动红外接收器     Start the receiver
  Serial.begin(9600);
```



```
void loop()
{
  if (irrecv.decode(&results))  //是否接收到红外信号   have we received an IR signal?
  {
    translateIR();
    Serial.println(results.value);
    irrecv.resume();            //接收下一个红外信号   receive the next value
  }
  pixels.clear();               //照明功能   Lighting function
  RGB();                        //RGB函数    RGB function
```



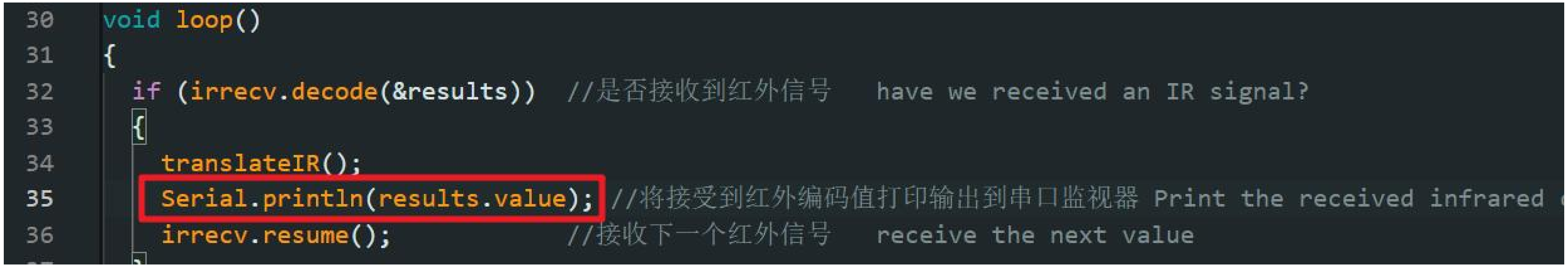
```
void RGB()
{
  switch(count_key)                //判断红外按键赋值变量    Determine the infrared key assignment variable
  {
    case 1 :           //亮红灯     The red light is on…
    case 2 :           //亮绿灯     The green light is on…
    case 3 :           //亮蓝灯     The blue light is on…
    case 4 :           //亮白灯     The white light is on…
    case 5 :           //关闭所有灯     Turn off the light…
    case 6 :           //红、绿、蓝交替闪烁     Red, green and blue flicker alternately
```



```
void translateIR() //根据收到的IR代码给count_key赋不同的值    takes action based on IR code received
{
  switch(results.value)
  {
    case 0XFFA25D: count_key=1; break;    //接收到按键1    Key 1 received
    case 0XFF629D: count_key=2; break;    //接收到按键2    Key 2 received
    case 0XFFE21D: count_key=3; break;    //接收到按键3    Key 3 received
    case 0XFF22DD: count_key=4; break;    //接收到按键4    Key 4 received
    case 0XFF02FD: count_key=5; break;    //接收到按键5    Key 5 received
    case 0XFFC23D: count_key=6; break;    //接收到按键6    Key 6 received
    default: break;
  }
}
```

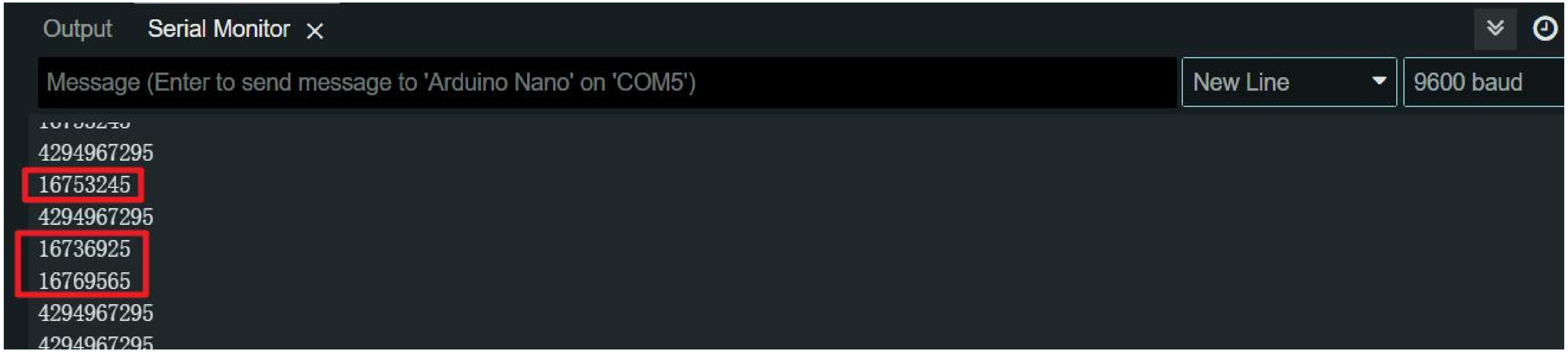
Get the code value sent by the infrared remote control:

The following statement code prints out the received infrared encoding value to the serial monitor.



```
void loop()
{
  if (irrecv.decode(&results))  //是否接收到红外信号   have we received an IR signal?
  {
    translateIR();
    Serial.println(results.value); //将接受到红外编码值打印输出到串口监视器 Print the received infrared
    irrecv.resume();            //接收下一个红外信号   receive the next value
```

Open the serial port monitor and you will see that the "8-digit" value printed out is the infrared encoding value corresponding to the button.



Message (Enter to send message to 'Arduino Nano' on 'COM5')

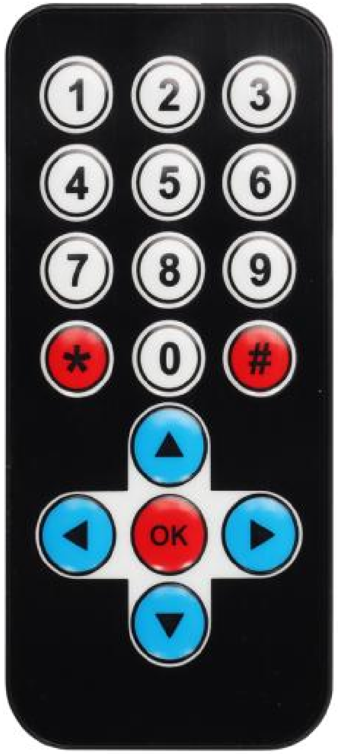
New Line

9600 baud

```
16753245
4294967295
16753245
4294967295
16736925
16769565
4294967295
4294967295
```

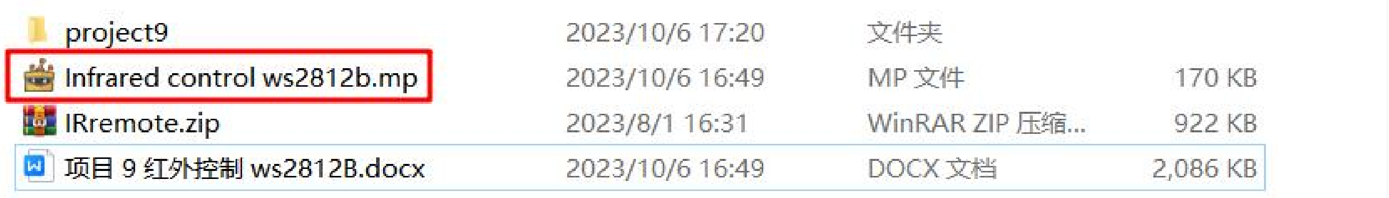
Through testing, you can get the coding values corresponding to all keys and then convert them into hexadecimal.

Key	Key value 十六进制	Key value 十进制
▲	FF18E7	16718055
▼	FF4AB5	16730805
◀	FF10EF	16716015
▶	FF5AA5	16734885
OK	FF38C7	16726215
①	FFA25D	16753245
②	FF629D	16736925
③	FFE21D	16769565
④	FF22DD	16720605
⑤	FF02FD	16712445
⑥	FFC23D	16761405
⑦	FFE01F	16769055
⑧	FFA857	16754775
⑨	FF906F	16748655



6. Download Mind+ graphical code

Open the project Mind+code file (path: project 9 infrared control ws2812B\Infrared control ws2812b.mp)

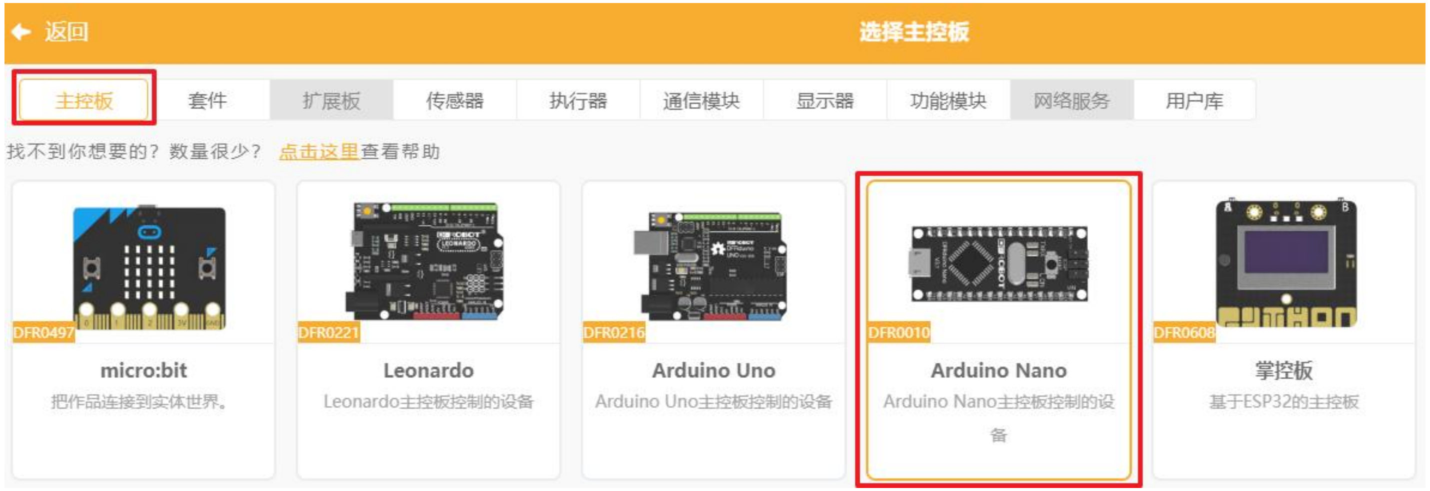


project9	2023/10/6 17:20	文件夹	
Infrared control ws2812b.mp	2023/10/6 16:49	MP 文件	170 KB
IRremote.zip	2023/8/1 16:31	WinRAR ZIP 压缩...	922 KB
项目 9 红外控制 ws2812B.docx	2023/10/6 16:49	DOCX 文档	2,086 KB

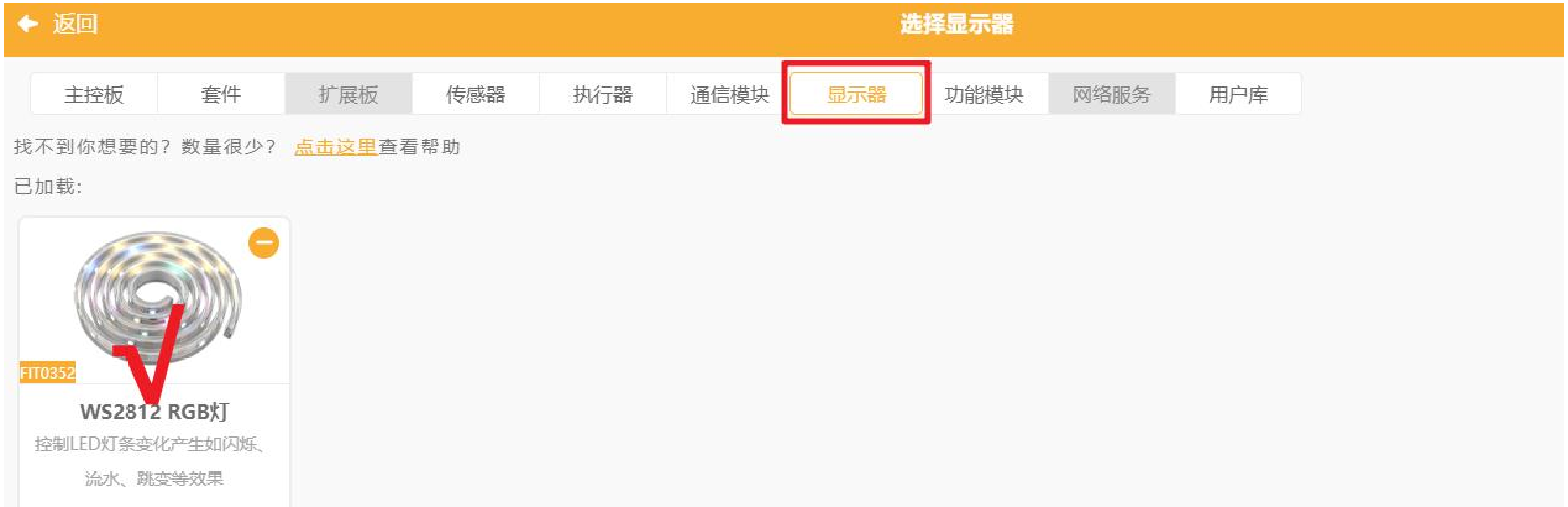
Connect the main control board to the computer with a USB cable and select the newly appeared CH340 serial port COM number. Click "Upload to Device" to complete the code upload.

Programming analysis:

Click "Extension" in the lower left corner, and then select the main control board type as Nano.



Add the WS2812 RGB light library file : click the "Display" type and select the WS2812 RGB light



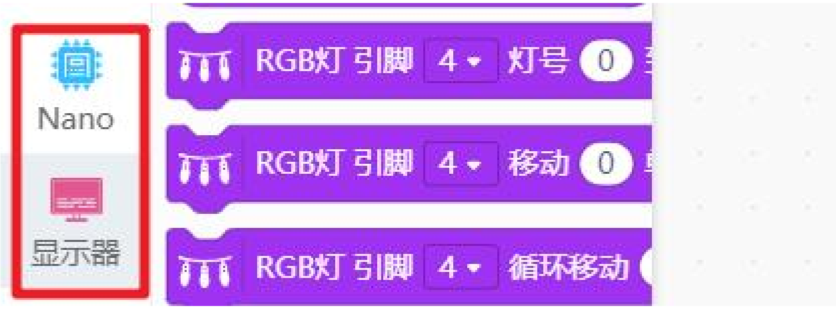
找不到你想要的？ 数量很少？ 点击这里查看帮助

已加载：

WS2812 RGB灯

控制LED灯条变化产生如闪烁、流水、跳变等效果

After the addition is successful, you can see that there are two more categories in the programming block column on the left: Nano and "Display"

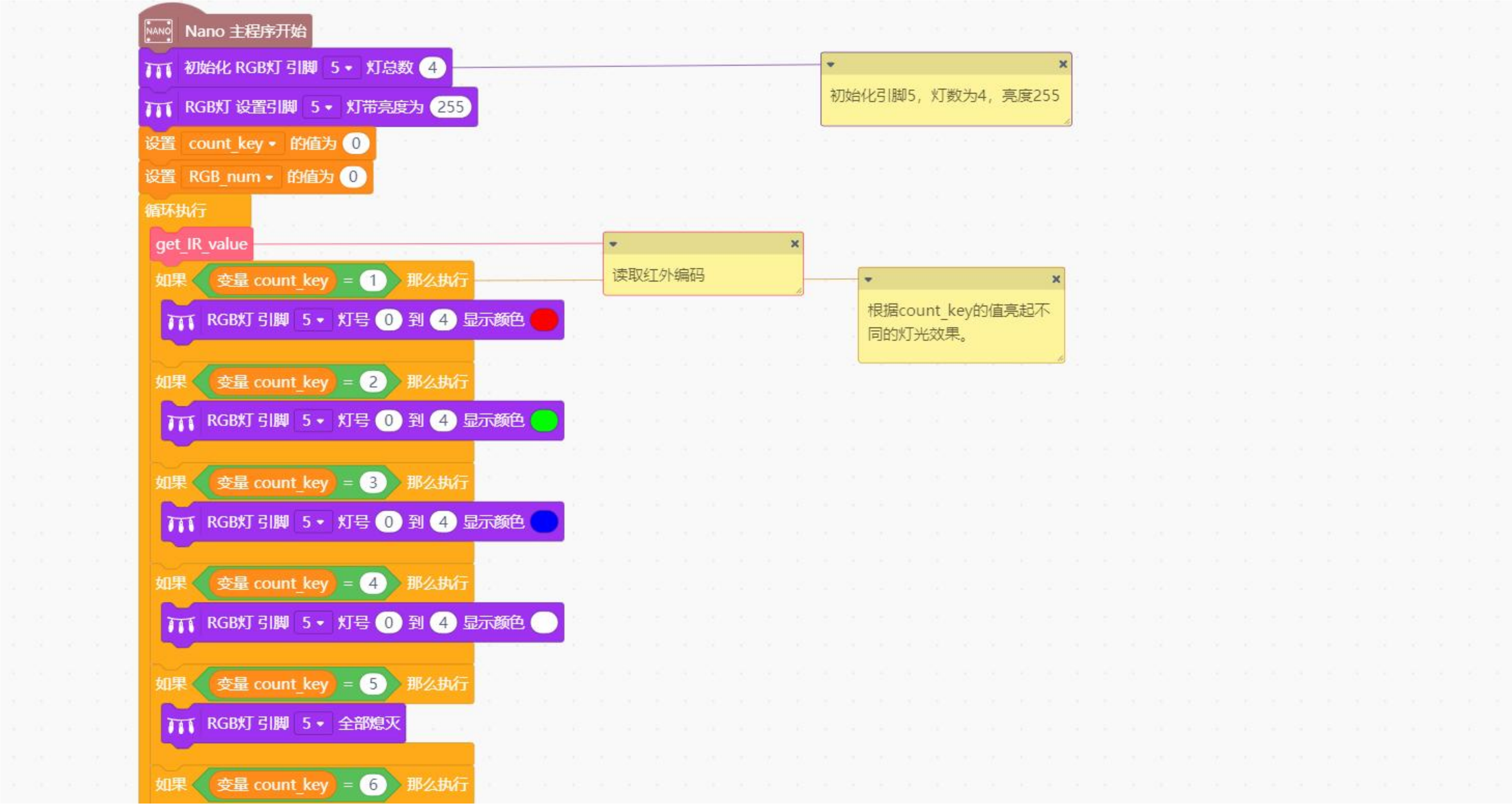


RGB灯 引脚 4 ▾ 灯号 0

RGB灯 引脚 4 ▾ 移动 0

RGB灯 引脚 4 ▾ 循环移动

The complete programming is as follows:



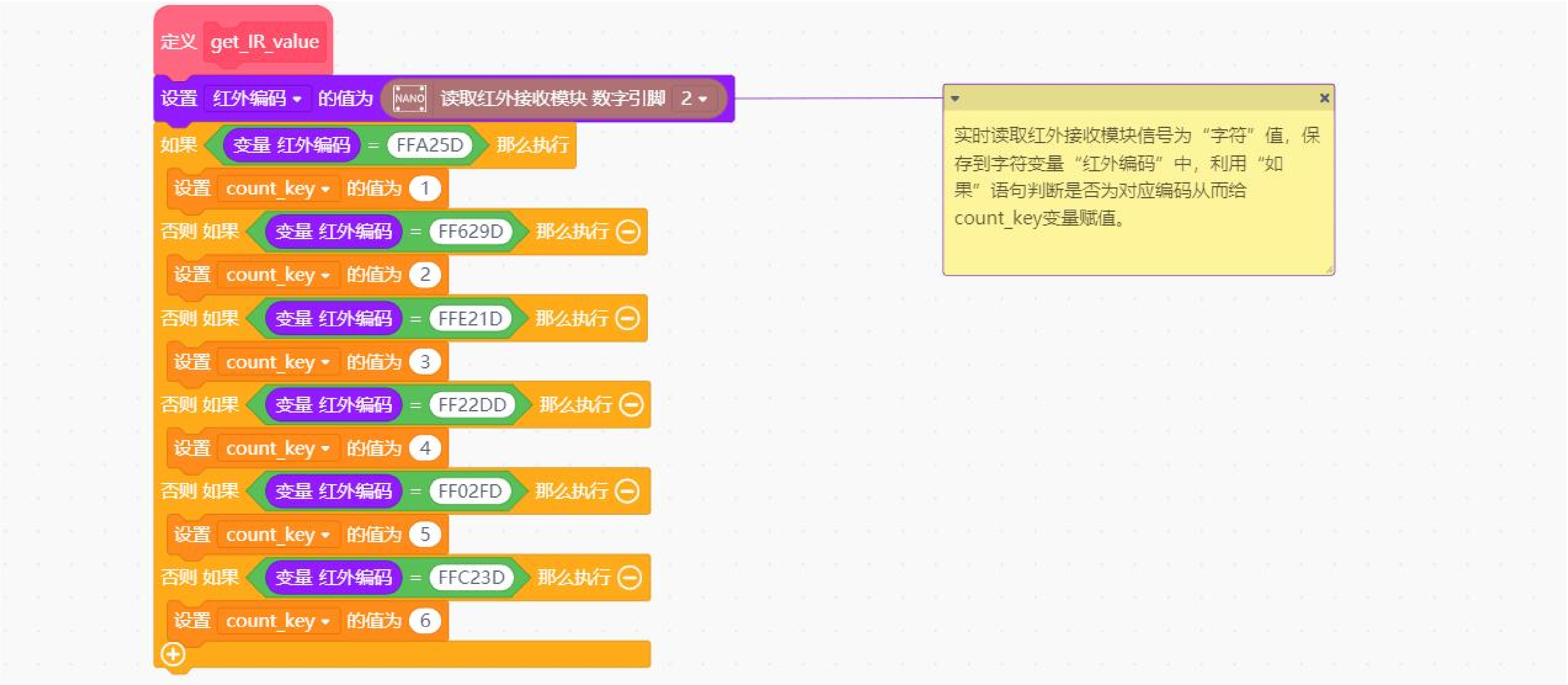
```
Nano 主程序开始
初始化 RGB灯 引脚 5 ▾ 灯总数 4
RGB灯 设置引脚 5 ▾ 灯带亮度为 255
设置 count_key ▾ 的值为 0
设置 RGB_num ▾ 的值为 0
循环执行
  get_IR_value
  如果 变量 count_key = 1 那么执行
    RGB灯 引脚 5 ▾ 灯号 0 到 4 显示颜色 🔴
  如果 变量 count_key = 2 那么执行
    RGB灯 引脚 5 ▾ 灯号 0 到 4 显示颜色 🟢
  如果 变量 count_key = 3 那么执行
    RGB灯 引脚 5 ▾ 灯号 0 到 4 显示颜色 🔵
  如果 变量 count_key = 4 那么执行
    RGB灯 引脚 5 ▾ 灯号 0 到 4 显示颜色 ⚪
  如果 变量 count_key = 5 那么执行
    RGB灯 引脚 5 ▾ 全部熄灭
  如果 变量 count_key = 6 那么执行
```

初始化引脚5，灯数为4，亮度255

读取红外编码

根据count_key的值亮起不同的灯光效果。

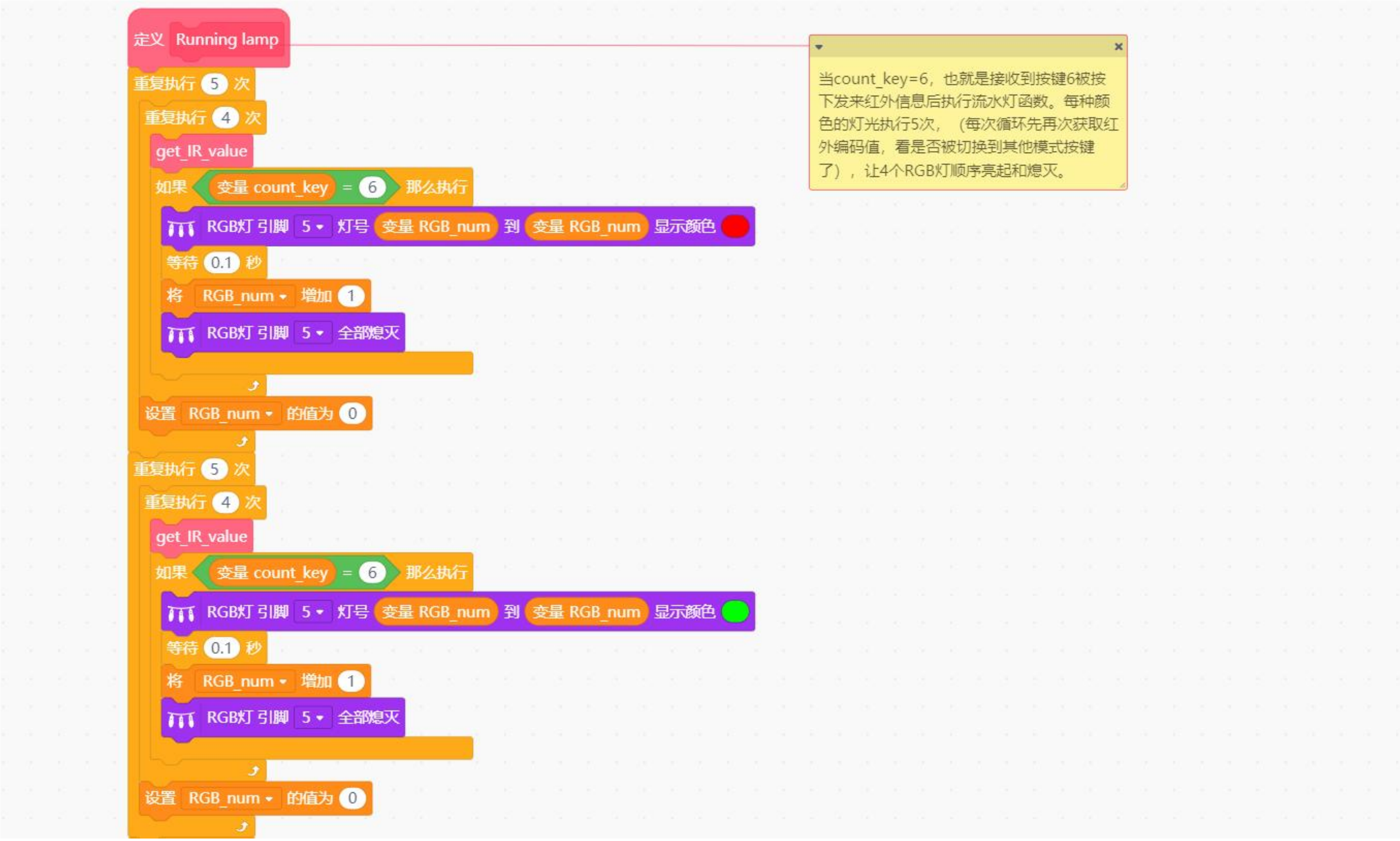
Function to read infrared receiving module information



```
定义 get_IR_value
设置 红外编码 ▾ 的值为 读取红外接收模块 数字引脚 2 ▾
如果 变量 红外编码 = FFA25D 那么执行
    设置 count_key ▾ 的值为 1
否则 如果 变量 红外编码 = FF629D 那么执行 ⊖
    设置 count_key ▾ 的值为 2
否则 如果 变量 红外编码 = FFE21D 那么执行 ⊖
    设置 count_key ▾ 的值为 3
否则 如果 变量 红外编码 = FF22DD 那么执行 ⊖
    设置 count_key ▾ 的值为 4
否则 如果 变量 红外编码 = FF02FD 那么执行 ⊖
    设置 count_key ▾ 的值为 5
否则 如果 变量 红外编码 = FFC23D 那么执行 ⊖
    设置 count_key ▾ 的值为 6
⊕
```

实时读取红外接收模块信号为“字符”值，保存到字符变量“红外编码”中，利用“如果”语句判断是否为对应编码从而给count_key变量赋值。

running water lamp function

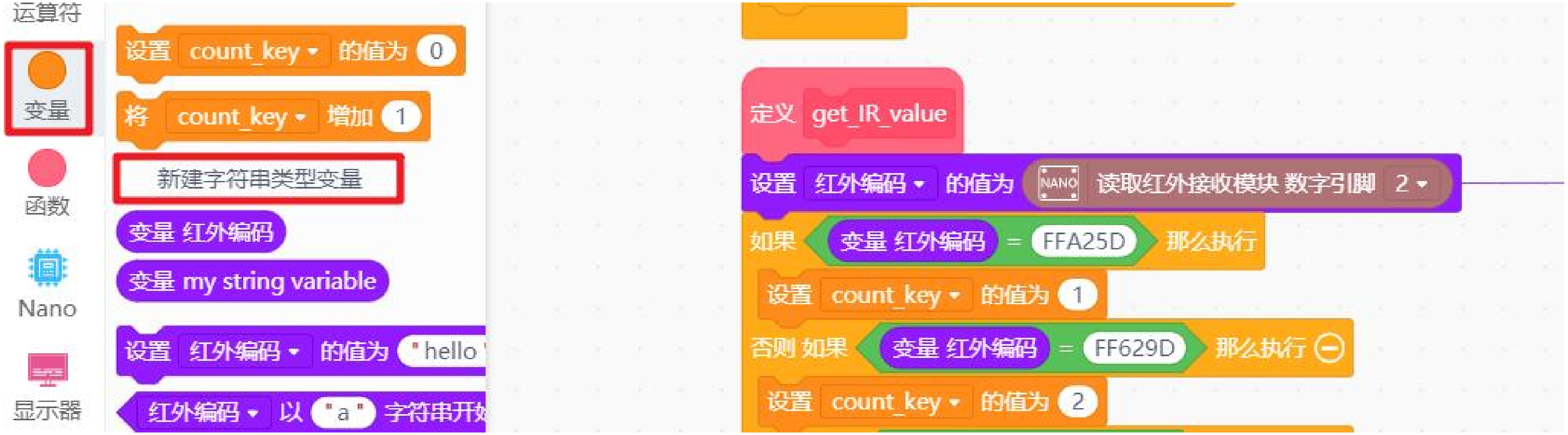


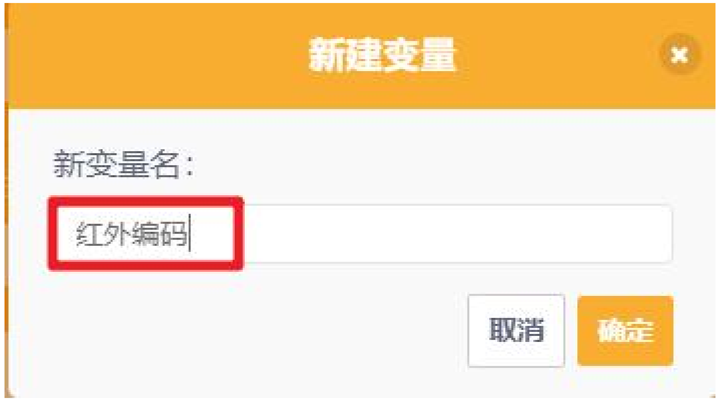
```
定义 Running lamp
重复执行 5 次
  重复执行 4 次
    get_IR_value
    如果 变量 count_key = 6 那么执行
      RGB灯 引脚 5 灯号 变量 RGB_num 到 变量 RGB_num 显示颜色 🟥
      等待 0.1 秒
      将 RGB_num 增加 1
      RGB灯 引脚 5 全部熄灭
  设置 RGB_num 的值为 0
重复执行 5 次
  重复执行 4 次
    get_IR_value
    如果 变量 count_key = 6 那么执行
      RGB灯 引脚 5 灯号 变量 RGB_num 到 变量 RGB_num 显示颜色 🟩
      等待 0.1 秒
      将 RGB_num 增加 1
      RGB灯 引脚 5 全部熄灭
  设置 RGB_num 的值为 0
```

当count_key=6，也就是接收到按键6被按下发来红外信息后执行流水灯函数。每种颜色的灯光执行5次，（每次循环先再次获取红外编码值，看是否被切换到其他模式按键了），让4个RGB灯顺序亮起和熄灭。

Programming Tips: Define "String" Type Variables

Click the " Variable " option, create a new string variable, and add the variable name

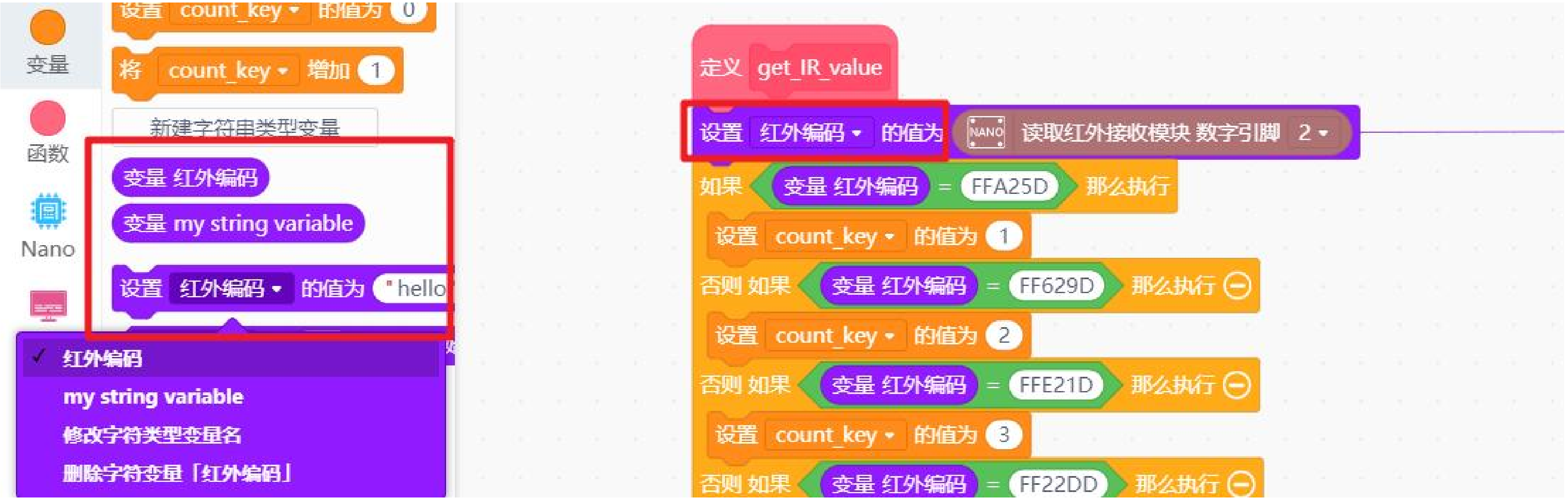

运算符
变量
函数
Nano
显示器
设置 count_key ▾ 的值为 0
将 count_key ▾ 增加 1
新建字符串类型变量
变量 红外编码
变量 my string variable
设置 红外编码 ▾ 的值为 " hello
红外编码 ▾ 以 " a " 字符串开始
定义 get_IR_value
设置 红外编码 ▾ 的值为 NANO 读取红外接收模块 数字引脚 2 ▾
如果 变量 红外编码 = FFA25D 那么执行
设置 count_key ▾ 的值为 1
否则 如果 变量 红外编码 = FF629D 那么执行
设置 count_key ▾ 的值为 2


新建变量

新变量名：

红外编码

Use newly created variables: assign values to newly created variables


变量
函数
Nano
设置 count_key 的值为 0
将 count_key 增加 1
新建字符串类型变量
变量 红外编码
变量 my string variable
设置 红外编码 的值为 "hello
✓ 红外编码
my string variable
修改字符类型变量名
删除字符变量「红外编码」
定义 get_IR_value
设置 红外编码 的值为 NANO 读取红外接收模块 数字引脚 2
如果 变量 红外编码 = FFA25D 那么执行
设置 count_key 的值为 1
否则 如果 变量 红外编码 = FF629D 那么执行
设置 count_key 的值为 2
否则 如果 变量 红外编码 = FFE21D 那么执行
设置 count_key 的值为 3
否则 如果 变量 红外编码 = FF22DD 那么执行