

## 4.8 Button start RGB

### 1. Learning goal:

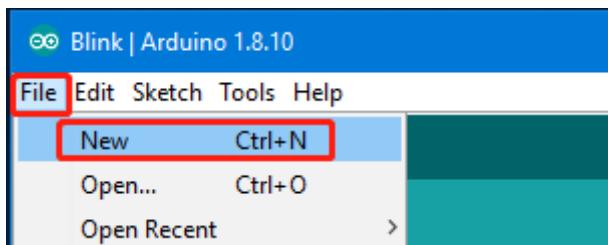
Learn to use the WS2812 programming light library and press the button to activate the RGB light.

### 2. Experimental phenomena:

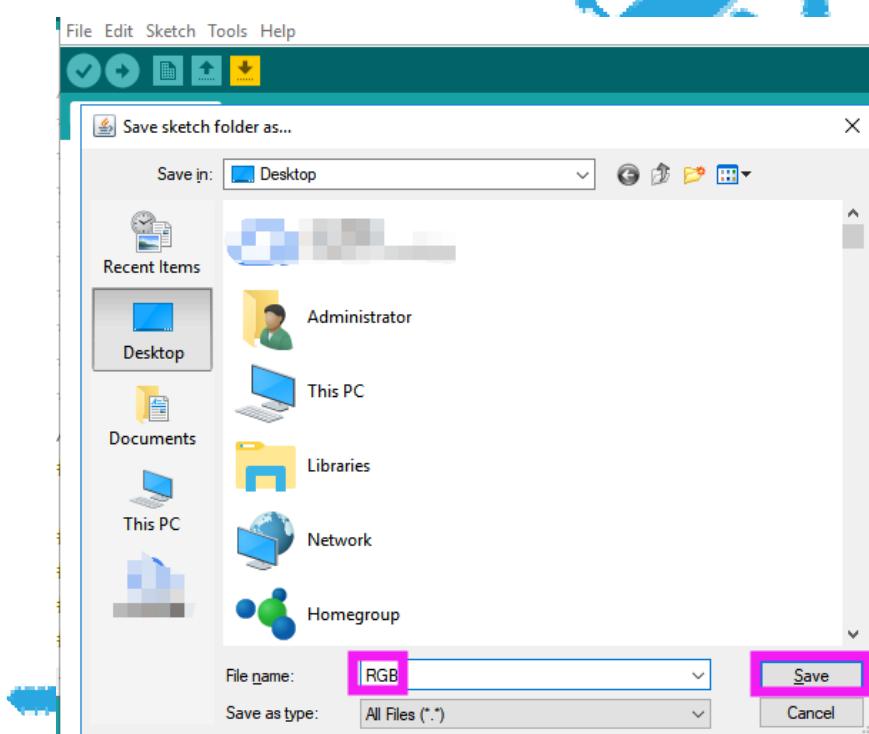
When button is pressed, buzzer will sound, and then the bottom four RGB lights light up the red, green, blue, and white colors every 0.5 seconds, and finally go out.

### 3. Create new project

3.1 Click 【File】-->【New】.



3.2 Press **Ctrl+S** to save and rename RGB. As shown below.



3.3 We can see that there is a Serial folder with **RGB.ino** on the computer desktop.

3.4 We will **RGB.ino** as shown below.

```
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}
```

The setup() function only runs once when the car is turned on or when the reset button is pressed, and the program for initializing the relevant content can be written;

The loop() function is the main loop function of the car and most of the data processing and logic processing are done in this function.

#### 4. Programming

4.1 From the hardware manual, we can know that the pin corresponds to the following

```
#define BUZZER 10      //Define buzzer pin
#define KEY_PIN 8       //Define key pin
#define LED_PIN 5       //Define LED(D9) pin
#define RGB_PIN 9       //Define RGB pin
#define MAX_RGB 4        //4 RGB lights
```

4.2 Import the library of the WS2812 programming light and initialize it. For the method of adding the library file, please refer to **【3.Development Environment Construction】----【3.4 Add additional library files】**

```
1 //Import library file
2 #include <Adafruit_NeoPixel.h>
3
4 #define BUZZER 10      //Define buzzer pin
5 #define KEY_PIN 8       //Define key pin
6 #define LED_PIN 5       //Define LED(D9) pin
7 #define RGB_PIN 9       //Define RGB pin
8 #define MAX_RGB 4        //4 RGB lights
9
10 //Initialize the WS2812 programming light
11 Adafruit_NeoPixel strip = Adafruit_NeoPixel(MAX_RGB, RGB_PIN, NEO_RGB + NEO_KHZ800);
```

4.3 New create clearRGB function, Close RGB light.

```
void clearRGB()
{
    uint32_t color = strip.Color(0, 0, 0);
    for (uint8_t i = 0; i < MAX_RGB; i++)
    {
        strip.setPixelColor(i, color);
    }
    strip.show();
}
```

4.4 Create a new showRGB function to display RGB lights. Parameter num: Which lamp is selected, if it is greater than or equal to the maximum value (4), all the lights are on together. The three parameters R/G/B indicate the color RGB value of the indicator light, ranging from 0 to 255.

```

void showRGB(int num, int R, int G, int B)
{
    uint32_t color = strip.Color(G, R, B);
    if (num >= MAX_RGB)
    {
        for (int i = 0; i < MAX_RGB; i++)
        {
            strip.setPixelColor(i, color);
        }
    }
    else
    {
        strip.setPixelColor(num, color);
    }
    strip.show();
}

```

#### 4.5 Initialize the pin mode in the setup() function, and clear RGB light.

```

void setup() {
    // put your setup code here, to run once:
    pinMode(KEY_PIN, INPUT_PULLUP); //Set the button pin to pull-up input mode
    pinMode(LED_PIN, OUTPUT);      //Set the LED pin to output mode
    pinMode(BUZZER, OUTPUT);       //Define buzzer pin
    pinMode(RGB_PIN, OUTPUT);      //Set the RGB pin to output mode

    strip.begin();
    strip.show();
    clearRGB();
}

```

#### 4.6 In the loop () main loop function to detect whether the button is pressed, buzzer will sound, and then the bottom four RGB lights light up the red, green, blue, and white colors every 0.5 seconds, and finally go out.

```

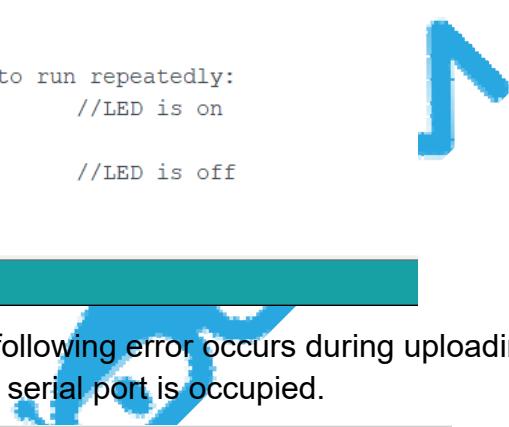
void loop() {
    //put your main code here, to run repeatedly:
    //keyscan
    keyscan();

    if (button_press)
    {
        whistle();
        showRGB(MAX_RGB, 255, 0, 0); //All RGB light become red 0.5s
        delay(500);
        showRGB(MAX_RGB, 0, 255, 0); //All RGB light become green 0.5s
        delay(500);
        showRGB(MAX_RGB, 0, 0, 255); //All RGB light become blue 0.5s
        delay(500);
        showRGB(MAX_RGB, 255, 255, 255); //All RGB light become white 0.5s
        delay(500);
        button_press = false;
    }
    else
    {
        clearRGB();
    }
}

```

### 5. Compiling and downloading code

5.1 After the code is written, press Ctrl+S to save, then click the “√” button to compile. If there is no problem, click “→” to upload (the car must be connected to the computer via the USB cable).

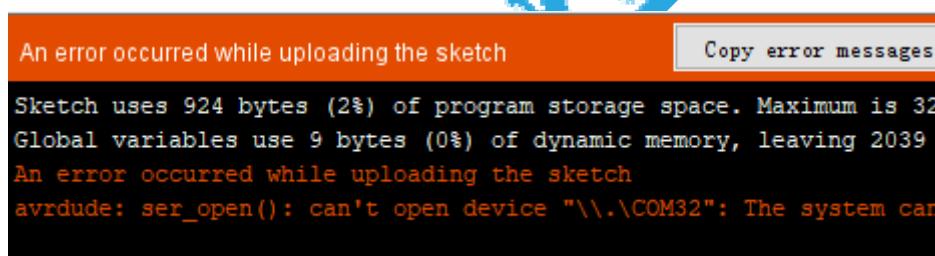


```

File Edit Sketch Tools Help
LED
10 */
11 //Define LED light(D9)pin
12 #define LED_PIN 5
13
14 void setup() {
15   // put your setup code here, to run once:
16   // set LED pin to output mode
17   pinMode(LED_PIN, OUTPUT);
18 }
19
20 void loop() {
21   // put your main code here, to run repeatedly:
22   digitalWrite(LED_PIN, LOW);           //LED is on
23   delay(500);
24   digitalWrite(LED_PIN, HIGH);         //LED is off
25   delay(500);
26 }

```

5.2 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.



**Solution:** Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE 【Tool】-->【Port】.

5.3 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the omniduino omnibus to the library file directory compiled by arduinolIDE.

please refer to 【3.Development Environment Construction】----【3.4 Add additional library files】

```
Adafruit_NeoPixel.h: No such file or directory

RGB:12:10: error: Adafruit_NeoPixel.h: No such file or directory
#include <Adafruit_NeoPixel.h>
^
compilation terminated.

exit status 1
Adafruit_NeoPixel.h: No such file or directory
```

