6.4 Color RGB light

1. Learning goal:

Learn about the WS2812RGB light control principle and learn to control RGB lights to show different effects.

2. Experimental phenomena:

After power is turned on, the RGB light shows the effect of breathing light. Each time the button is pressed, the effect will be switched.

The first time is pressed for the breathing light, the second time is the marquee, and the third time is the water light. The fourth time is the a colorful light.

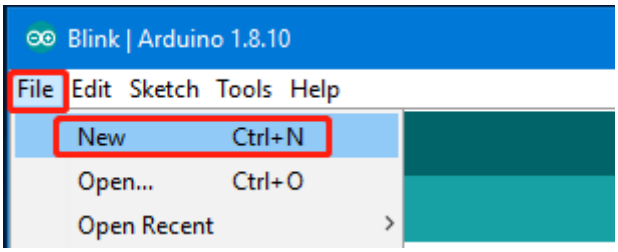
3. About WS2812B

WS2812B is an intelligent external control LED light source integrating control circuit and light-emitting circuit. Its appearance is the same as a 5050 LED lamp bead. It looks like a normal RGB lamp, and each component is a pixel.

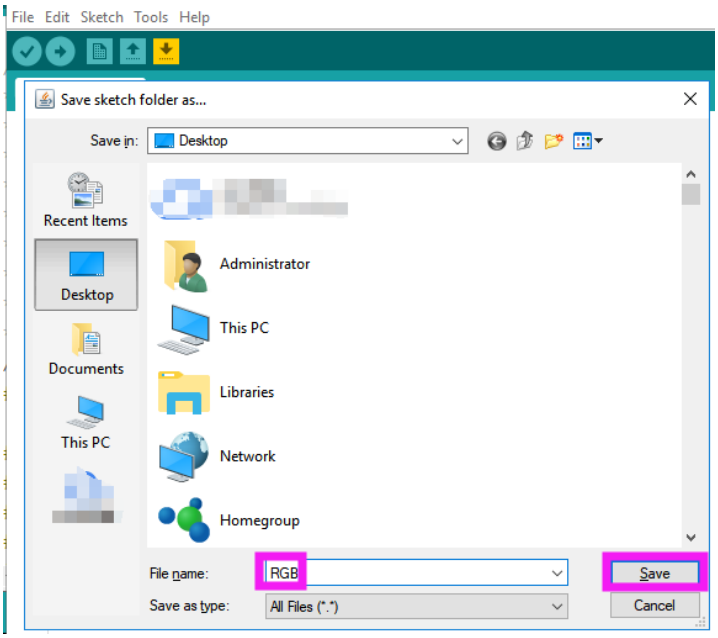
The WS2812 requires only one signal line to control all LEDs on the strip. Multiple strips can be easily extended in series.

4. Create new project

4.1 Click【File】-->【New】.



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



4.3 We can see that there is a Serial folder with **RGB.ino** on the computer desktop.
4.4 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.

5.Programming

5.1 Import the library Adafruit_NeoPixel.h of the WS2812 programming light and initialize it. The first parameter is the number of lights currently controlled, the second parameter is the control pin of the RGB lamp, and the third parameter is the set frequency.

```
//Import library
#include <Adafruit_NeoPixel.h>

#define KEY_PIN 8    //Define Key pin
#define RGB_PIN 9    //Define the RGB pin
#define MAX_LED 4    //4 RGB lights

//Initialize WS2812 programming RGB
Adafruit_NeoPixel strip = Adafruit_NeoPixel(MAX_LED, RGB_PIN, NEO_RGB + NEO_KHZ800);

/*RGBEffects*/
int RGB_Effect = 0;     //1:Breathing light,2:Marquee,3:Water light,4:Color light,5:1
```

5.2 Add the keyscan() function, each time press the function button RGB_Effect add 1, control to switch the RGB light effect.

```
void keyscan()
{
  int val;
  val = digitalRead(KEY_PIN); //Read the digital 8-port leve
  if (val == LOW)             //When the button is pressed
  {
    delay(10);                  //Delayed debounce
    val = digitalRead(KEY_PIN); //Read button status again
    while (val == LOW)
    {
      val = digitalRead(KEY_PIN); //Third read button status
      if (val == HIGH)            //Determine if the button
      {
        RGB_Effect++;
        return;
      }
    }
  }
}
```

5.3 New create clearRGB() function, clear RGB light.

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

5.4 New create showRGB() function, The RGB lamp is displayed. The first parameter selects the RGB bit. When it is greater than or equal to the maximum value, it displays all. The second display color R value. The third is the G value of the display color, and the fourth is the B value of the display color.

```
void showRGB(int num, int R, int G, int B)
{
  uint32_t color = strip.Color(G, R, B);
  if (num > MAX_LED - 1)
  {
    for (int i = 0; i < MAX_LED; i++)
    {
      strip.setPixelColor(i, color);
    }
  }
  else
  {
    strip.setPixelColor(num, color);
  }
  strip.show();
}
```

5.5 New create waterfall_RGB() function, which to show the water light effect, the color is green. if you need to modify the color, you can modify it in the showRGB() command. The variable w has been set to a static variable. It is only assigned 0 when the first initialization.

```
void waterfall_RGB()
{
  static int w = 0;
  clearRGB();
  showRGB(w, 0, 255, 0);
  w++;
  if (w >= 4)
    w = 0;
}
```

5.6 New create marquee_RGB() function, which to show the marquee effect.

```
void marquee_RGB()
{
  static int m = 0;
  clearRGB();
  switch (m)
  {
  case 0:
    showRGB(m % 4, 255, 0, 0);
    break;
  case 1:
    showRGB(m % 4, 255, 90, 0);
    break;
  case 2:
    showRGB(m % 4, 255, 255, 0);
    break;
  case 3:
    showRGB(m % 4, 0, 255, 0);
    break;
  case 4:
    showRGB(m % 4, 0, 255, 255);
    break;
  case 5:
    showRGB(m % 4, 0, 0, 255);
    break;
  case 6:
    showRGB(m % 4, 255, 0, 255);
    break;
  case 7:
    showRGB(m % 4, 255, 255, 255);
    break;

  default:
    break;
  }
  m++;
  if (m >= 8)
    m = 0;
}
```

5.7 New create breathing_RGB() function, which to show the breathing light effect.

```
void breathing_RGB()
{
  static int a = 0, b = 0, c = 0;
  if (a == 0)
  {
    b += 2;
    if (b >= 250)
      a = 1;
  }
  else
  {
    b -= 2;
    if (b <= 5)
    {
      a = 0;
      c++;
    }
  }
  switch (c)
  {
  case 0:
    showRGB(MAX_LED, b, 0, 0);
    break;
  case 1:
    showRGB(MAX_LED, b, b, 0);
    break;
  case 2:
    showRGB(MAX_LED, 0, b, 0);
    break;
  case 3:
    showRGB(MAX_LED, 0, b, b);
    break;
  case 4:
    showRGB(MAX_LED, 0, 0, b);
    break;
  case 5:
    showRGB(MAX_LED, b, 0, b);
    break;
  case 6:
    showRGB(MAX_LED, b, b, b);
    break;

  default:
    break;
  }
  c %= 7;
}
```

5.8 New create colorfull_RGB() function, which to show the colorful light effect.

```
void colorfull_RGB()
{
  static int rgb_r = 254;
  static int rgb_g = 0;
  static int rgb_b = 0;
  if (rgb_r == 254 && rgb_g < 254 && rgb_b == 0)
    rgb_g += 2;
  else if (rgb_g == 254 && rgb_r > 0 && rgb_b == 0)
      rgb_r -= 2;
  else if (rgb_g == 254 && rgb_b < 254 && rgb_r == 0)
    rgb_b += 2;
  else if (rgb_b == 254 && rgb_g > 0 && rgb_r == 0)
    rgb_g -= 2;
  else if (rgb_b == 254 && rgb_r < 254 && rgb_g == 0)
    rgb_r += 2;
  else if (rgb_r == 254 && rgb_b > 0 && rgb_g == 0)
    rgb_b -= 2;

  showRGB(MAX_LED, rgb_r, rgb_g, rgb_b);
}
```

5.9 Set the pin mode and clear the RGB light display in the setup() function.

```
void setup() {
  // put your setup code here, to run once:
  pinMode(RGB_PIN, OUTPUT);         //Set the RGB p
  pinMode(KEY_PIN, INPUT_PULLUP);   //Set the butto

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

5.10 In the loop() main loop function monitor button, each time press k1 button to switch the display effect.

The millis() function is the time to get the chip from boot to current, in milliseconds. The advantage of using the millis() instead of the delay() is that the system can still run normally, and the action of the normal monitor button.

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

  switch (RGB_Effect)
  {
  case 1:     //breathing light
    nowTime = millis();
    if (nowTime - oldTime >= 20)
    {
      oldTime = nowTime;
      breathing_RGB();
    }
    break;

  case 2:   //marquee
    nowTime = millis();
    if (nowTime - oldTime >= 150)
    {
      oldTime = nowTime;
      marquee_RGB();
    }
    break;
  case 3:   //water light
    nowTime = millis();
    if (nowTime - oldTime >= 300)
    {
      oldTime = nowTime;
      waterfall_RGB();
    }
    break;

  case 4:   //colorful light
    nowTime = millis();
    if (nowTime - oldTime >= 10)
    {
      oldTime = nowTime;
      colorfull_RGB();
    }
    break;

  case 5:   //Turn off light
    showRGB(MAX_LED, 0, 0, 0);
    RGB_Effect = 0;
    break;

  default:
    break;
  }

}
```

5. Compiling and downloading code

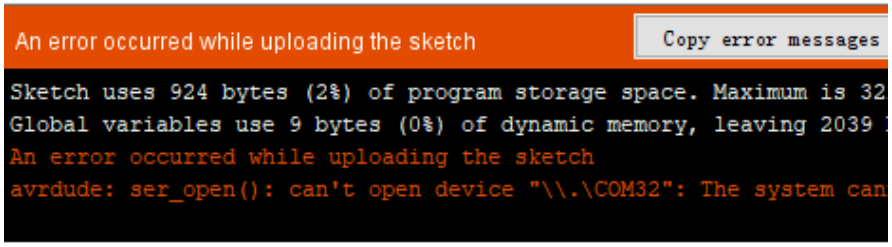
5.1 Open the **RGBEffects.ino** program in the course data, select the serial port and click upload directly (the Omniduino car must first be connected to the computer via the USB data cable).

5.2 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 arduinoIDE.

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

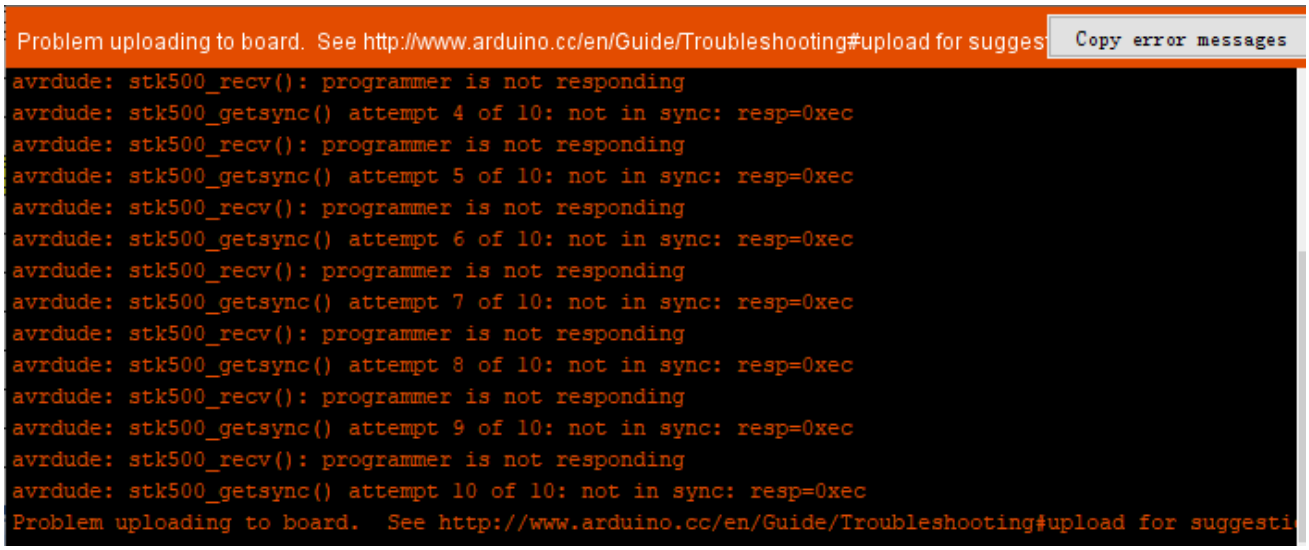
```
Adafruit_PWMServoDriver.h: No such file or directory

CarRun:2:10: error: Adafruit_PWMServoDriver.h: No such file or directory

 #include <Adafruit_PWMServoDriver.h>

          ^~~~~~~~~~~~~~~~~~~~~~~~~~

compilation terminated.

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

5.3 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】.

6.4 After clicking the upload button, the upload is always displayed, but it can't be uploaded successfully for a long time.



Because the uploading program and the WIFI camera communication is realized through the serial port, when the serial port is occupied by the WIFI camera, and the program cannot be uploaded.

Solution:

①Unplug the USB cable, turn off the power of the car, wait for the D2 indicator to go out.

②Then, plug in the USB data cable. At this time, your mobile phone should not connect the WiFi signal of the car.

③You can upload the program to the car according to the normal steps.

④After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. The corresponding experimental phenomenon will appear.

(Tip: If you upload APP control program. After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. Mobile phone connect the car to the WIFI signal, and then open the APP to control.)