

# keyestudio

## Project 2: WS2812 RGB

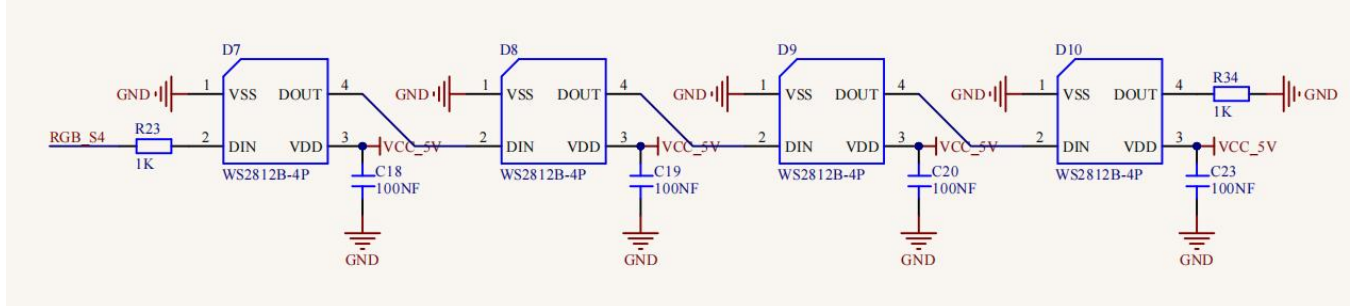
### 1. Description



In this project, we will work to use WS2812 LED to display different colors. Its principle is different from the seven-color LED, we need a pin to control it. This is an intelligent external controlled LED light source integrating control circuit and light emitting

circuit. The appearance of each LED is the same as that of a 5050 LED and each component is a pixel. There are four LEDs on our motor driver board, namely four pixels. Let's learn how to control it to display any color.

### 2. Component Knowledge



The four pixel LEDs are connected in series, and we can control any one of the LEDs with a pin(here we use D10) and make it display any color. The pixel contains an intelligent digital interface data locking signal and amplifying driver circuit as well as a high-precision internal oscillator and a 12V high voltage programmable constant current control part, which

# keyestudio

effectively ensures the color consistency of the pixel light.

The data protocol adopts the communication mode of single-line return to zero code. After the pixel is powered on and reset, the S terminal receives the data transmitted from the controller.

When the first 24bit data extracted by the first pixel, which will be sent to the data latch inside the pixel. This 2812RGB communication protocol and driver has been encapsulated in the bottom layer, we can directly call the interface function to use.

## 3. Test Code

```
/**
 *
 * Keyestudio 4WD Mecanum Robot for Arduino
 * lesson 2.1
 * RGB2812
 * http://www.keyestudio.com
 */
#include <Adafruit_NeoPixel.h>

//Create a class called rgb_2812 to control rgb, there are four LEDs, and pins are connected to D10
Adafruit_NeoPixel rgb_2812 = Adafruit_NeoPixel(4, 10, NEO_GRB + NEO_KHZ800);

void setup() {
  rgb_2812.begin();    //Start rgb2818

  rgb_2812.setBrightness(100); //Initialize the brightness to (0~255)
  rgb_2812.clear();    //Initialize all the NeoPixels to "close" state

  rgb_2812.setPixelColor(0, 255, 0, 0); //The first LED is red
  rgb_2812.setPixelColor(1, 0, 255, 0); //The second LED is green
  rgb_2812.setPixelColor(2, 0, 0, 255); //The third LED is blue
  rgb_2812.setPixelColor(3, 255, 255, 255); //The fourth LED is white
  rgb_2812.show();    //Refresh display
}

void loop() {
}

/**
```

# keyestudio

## 4. Test Result

First, make sure the library files are installed, otherwise the code will fail to compile. Here we use the library file [“Adafruit\\_NeoPixel”](#) , please refer to the [development environment configuration](#) for installation. After uploading the code and turn the DIP switch to the ON end and power on, we can see that the 4 2812 LEDs will respectively be red, green, blue and white.

## 5. Code Explanation

<b>#include &lt;Adafruit_NeoPixel.h&gt;</b>	Import the file rgb_2818
<b>Adafruit_NeoPixel rgb_2812 = Adafruit_NeoPixel(num, pin, NEO_GRB + NEO_KHZ800);</b>	Create an instance to drive the 2812 LEDs, pin = pin number, num = LED number. Here we connect to D10 , 4 LED, that is(4,10)
<b>rgb_2812.begin();</b>	Start rgb2818
<b>rgb.setBrightness(100);</b>	Set the brightness of the LED to 100 and the range is 0 to 255, the larger the value, the brighter the LED will be.
<b>rgb_2812.clear();</b>	Initialize all the NeoPixels to “close” state
<b>rgb_2812.setPixelColor(num, r, g, b);</b>	Set the color of the LED. Num is the LED number and the four LED numbers are 0, 1, 2, and 3 respectively. r, g, and b are the

# keystudio

	colors of the LEDs.
<b>rgb.show();</b>	Refresh display
<b>random(0, 255);</b>	Select a random number from 0 to 255

## 6. Expanded Project: Flowing Water Light

```
//*****  
/*  
  Keystudio 4WD Mecanum Robot for Arduino  
  lesson 2.2  
  RGB2812  
  http://www.keystudio.com  
*/  
#include <Adafruit_NeoPixel.h>  
  
//Create a class called rgb_2812 to control rgb, there are four LEDs, and pins are connected to D10  
Adafruit_NeoPixel rgb_2812 = Adafruit_NeoPixel(4, 10, NEO_GRB + NEO_KHZ800);  
  
void setup() {  
  rgb_2812.begin();    //start rgb2818  
  rgb_2812.setBrightness(150); //Initialize the brightness to (0~255)  
  rgb_2812.clear();    //Initialize all the NeoPixels to "close" state  
}  
  
void loop() {  
  uint8_t r = random(0, 255);  
  uint8_t g = random(0, 255);  
  uint8_t b = random(0, 255);  
  for (uint8_t i = 0; i < 4; i++)  
  {  
    rgb_2812.setPixelColor(i, r, g, b); //The color of the i +1 LED is random(r,g,b)  
    rgb_2812.show();    //Refresh display  
    delay(100); //Wait for a while  
  }  
}  
//*****
```

Once the code has been uploaded successfully, we will see four LEDs showing the flowing light in a random color.