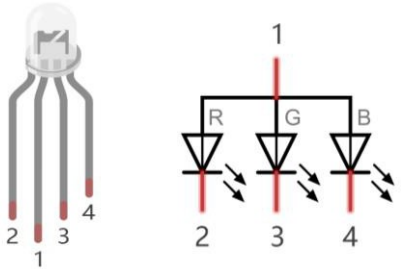
# Lesson 3-Testing WS2812 LED

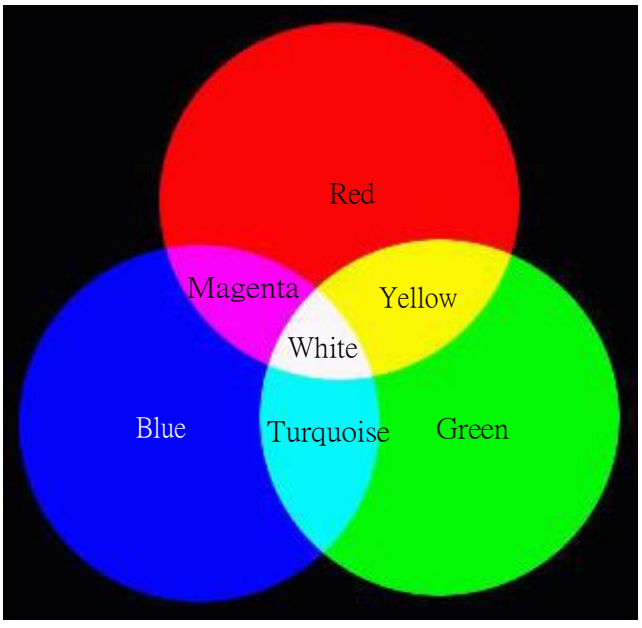
## Knowledege of RGB LED

A RGB LED has 3 LEDs integrated into one LED component. It can respectively emit Red, Green and Blue light. In order to do this, it requires 4 pins (this is also how you identify it). The long pin (1) is the common which is the Cathode (+) or positive lead, the other 3 are the Anodes (-) or negative leads. A rendering of a RGB LED and its electronic symbol are shown below. We can make RGB LED emit various colors of light and brightness by controlling the




3 Anodes (2, 3 & 4) of the RGB LED

Red, Green, and Blue light are called 3 Primary Colors when discussing light (Note: for pigments such as paints, the 3 Primary Colors are Red, Blue and Yellow). When you combine these three Primary Colors of light with varied brightness, they can produce almost any color of visible light. Computer screens, single pixels of cell phone screens, neon lamps, etc. can all produce millions of colors due to this phenomenon.

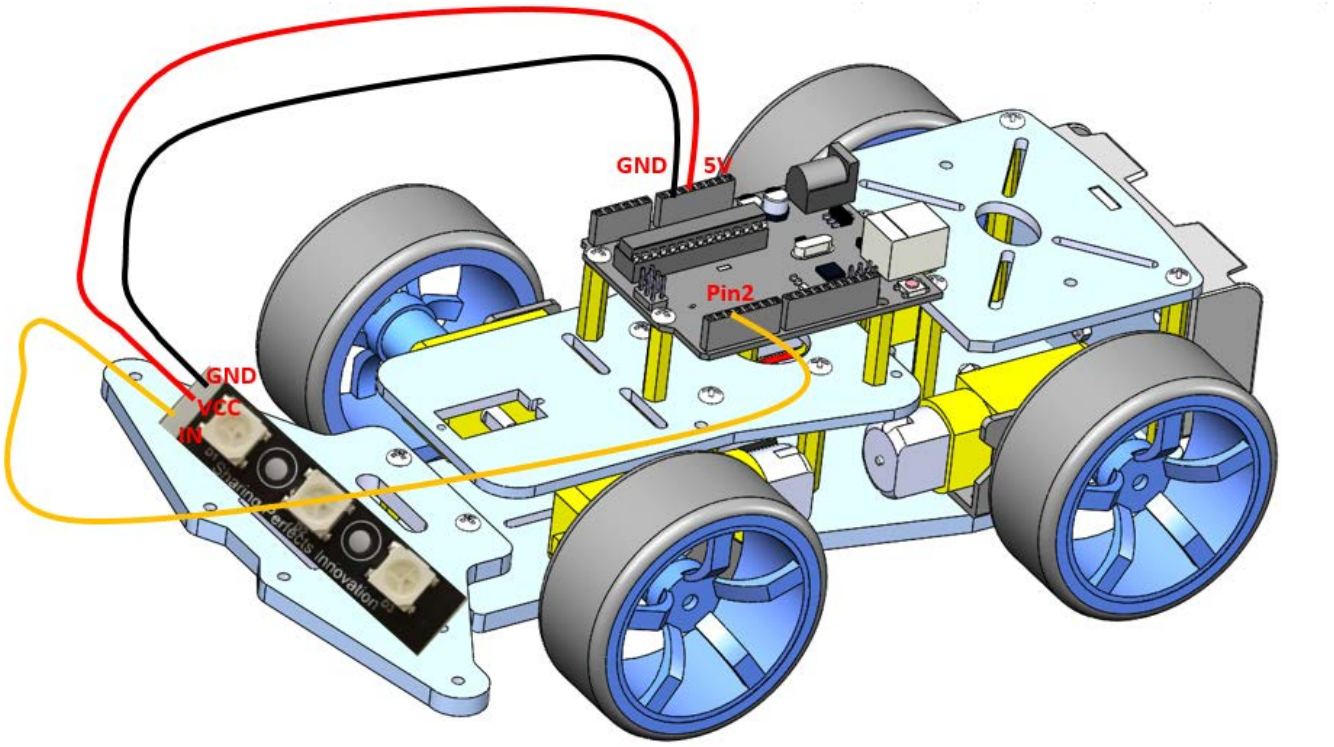



We know from previous section that, control board controls LEDs to emit a total of 256(0-255) different brightness with PWM. So, through the combination of three different colors of LEDs, RGB LED can emit 256^3=16777216 Colors, 16Million colors.

## Circuit connection

Let's take a SW2812 module as an example, this module has 3 RGB LEDs. Fix the SW2812 module on the acrylic expansion board at the front of the car body, and as shown in the figure below, connect the GND of the SW2812 module to the GND of the UNO board with a DuPont line, the VCC of the SW2812 module is connected to the 5V of the UNO board, and the signal pin of the SW2812 module IN is connected to Pin2 of the UNO board

Power supply: It is recommended to connect the computer directly with USB to supply power to the UNO board




## Circuit inspection

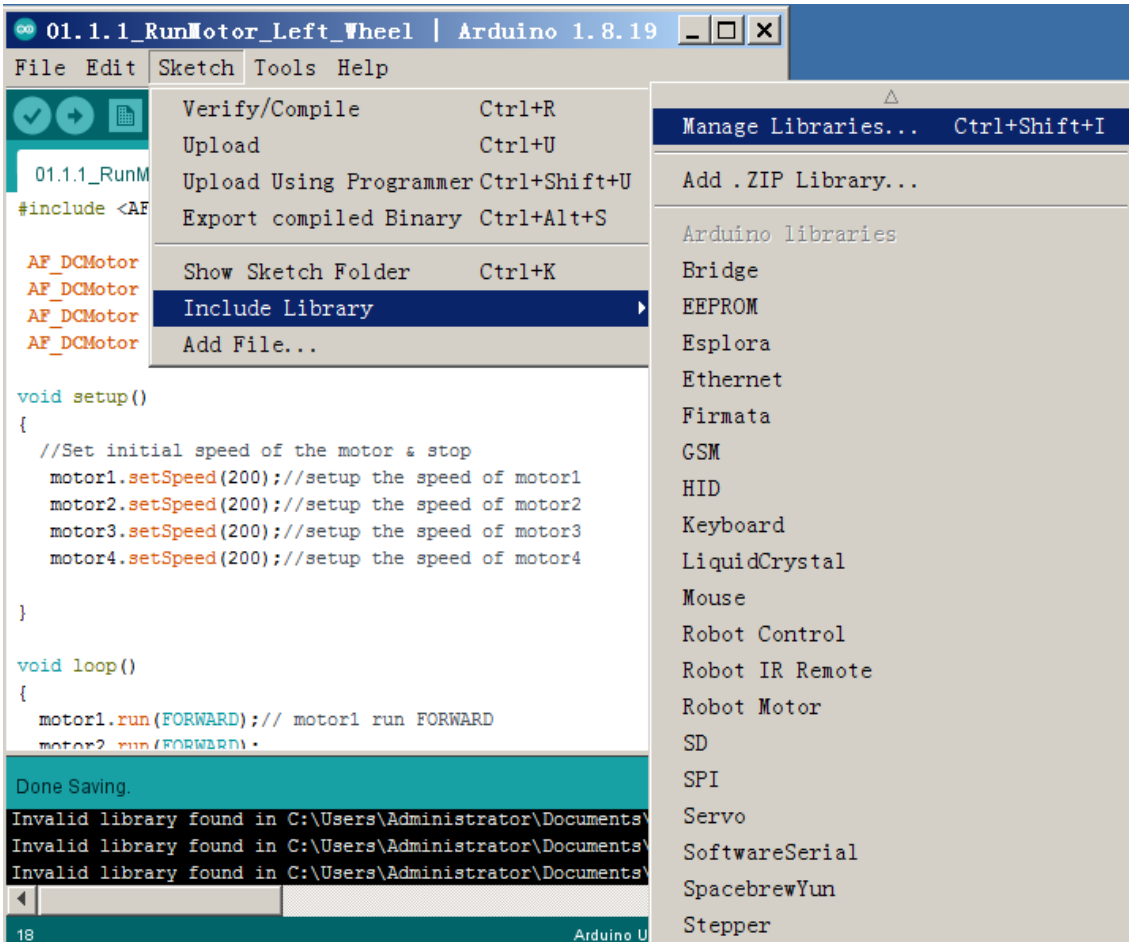
Before powering on, please carefully check whether the connected circuit has open circuit or short circuit, especially GND and 5V, GND and 3.3V, must not be short-circuited.A short circuit can cause high current in your circuit, create excessive component heat and cause permanent damage to your hardware!

## Program test

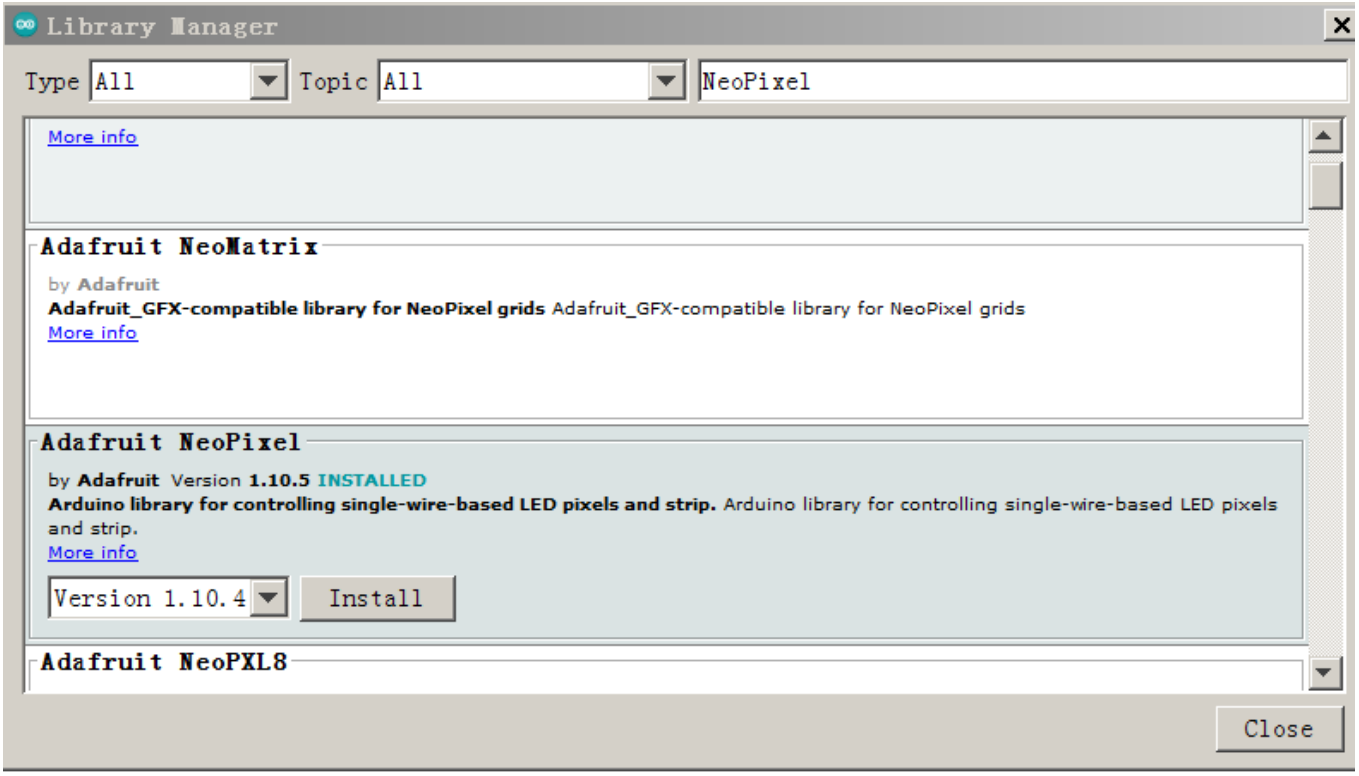
Install Adafruit_NeoPixel.h libraries

In order to establish communication with the sw2812 module, we need to install the Adafruit_NeoPixel.h library first, so that the UNO board can issue commands to control the LED lights of the sw2812 module.

Open to click Arduino IDE，then click “Sketch”>“Include library”>“Manage Libraries...”，Wait for the library manager to download the library index and update the list of installed libraries.



Type "NeoPixel" to filter your search. Look for the Adafruit_NeoPixel library provided by Adafruit. Once found, click the entry, select the latest version, and finally click Install. Wait for the automatic installation to complete, and the SW2812 module library file is ready.




Click "File---"Open" in the IDE interface, select the code in the path "4WD Car Chassis Kit Tutorial\ Sketches\ 03.1_Testing WS2812 LED". After the compilation is successful, connect the UNO board on the 4WD car body to the computer with a USB cable, upload the code.
After the upload is successful, 3 LED lights on the sw2812 module are cyclically changed in the order of red---green---blue---white---off.

You can also set different color changes according to your own preferences.
The corresponding relationship between program setting value and RGB is shown in the following figure


You can find more rgb values and colors here
https://www.rapidtables.com/web/color/RGB_Color.html

## Code as below:

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

#define LED_PIN 2
#define LED_COUNT 3
#define Led_delay 100
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup()
{
  #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
    clock_prescale_set(clock_div_1);
  #endif
  // END of Trinket-specific code.
  strip.begin();           // INITIALIZE NeoPixel strip object (REQUIRED)
  strip.show();            // Turn OFF all pixels ASAP
  strip.setBrightness(50); // Set BRIGHTNESS to about 1/5 (max = 255)
  Serial.begin(9600);
}

void loop()
{
  SW2812_Test();
}

void SW2812_Test()
{
  colorWipe(strip.Color(255,   0,    0), Led_delay); // Red
  colorWipe(strip.Color(  0, 255,    0), Led_delay); // Green
  colorWipe(strip.Color(  0,   0,  255), Led_delay); // Blue
  colorWipe(strip.Color(  255,255, 255), Led_delay); // While
  //rainbow(10);             // Flowing rainbow cycle along the whole strip
```

```
  //theaterChaseRainbow(50); // Rainbow-enhanced theaterChase variant
}

void colorWipe(uint32_t color, int wait) {
  for(int i=0; i<strip.numPixels(); i++) { // For each pixel in strip...
    strip.setPixelColor(i, color);         //  Set pixel's color (in RAM)
    strip.show();                          //  Update strip to match
    delay(wait);                           //  Pause for a moment
  }
}

void theaterChase(uint32_t color, int wait)
{
  for(int a=0; a<10; a++) {  // Repeat 10 times...
    for(int b=0; b<3; b++) { //  'b' counts from 0 to 2...
      strip.clear();         //   Set all pixels in RAM to 0 (off)
      // 'c' counts up from 'b' to end of strip in steps of 3...
      for(int c=b; c<strip.numPixels(); c += 3) {
        strip.setPixelColor(c, color); // Set pixel 'c' to value 'color'
      }
      strip.show(); // Update strip with new contents
      delay(wait);  // Pause for a moment
    }
  }
}

void rainbow(int wait)
{
  for(long firstPixelHue = 0; firstPixelHue < 5*65536; firstPixelHue += 256)
  {
    strip.rainbow(firstPixelHue);
    strip.show(); // Update strip with new contents
    delay(wait);  // Pause for a moment
  }
}

void theaterChaseRainbow(int wait)
```

```
{
  int firstPixelHue = 0;     // First pixel starts at red (hue 0)
  for(int a=0; a<30; a++) {  // Repeat 30 times...
    for(int b=0; b<3; b++) { //  'b' counts from 0 to 2...
      strip.clear();         //   Set all pixels in RAM to 0 (off)
      for(int c=b; c<strip.numPixels(); c += 3)
      {
        int      hue   = firstPixelHue + c * 65536L / strip.numPixels();
        uint32_t color = strip.gamma32(strip.ColorHSV(hue)); // hue -> RGB
        strip.setPixelColor(c, color); // Set pixel 'c' to value 'color'
      }
      strip.show();              // Update strip with new contents
      delay(wait);               // Pause for a moment
      firstPixelHue += 65536 / 90; // One cycle of color wheel over 90 frames
    }
  }
}
```

## What's Next?

THANK YOU for participating in this learning experience!

If you find errors, omissions or you have suggestions and/or questions about this lesson, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://cokoino.com/

Thank you again for choosing Cokoino products.