Lesson 7 Test the ws2812 led module

Table

1. Knowledge of ws2812 led

1.1 Overview：

WS2812B is an intelligently controlled LED that integrates the control circuit and RGB chip in a package of 5050 components.The interior includes intelligent digital ports, data latches and signal amplification drive circuits. It also includes a precise internal oscillator and a voltage programmable constant current control part to effectively ensure that the light color of the pixels is highly consistent.

The data transmission protocol adopts a single NZR communication mode. After the pixel is powered on and reset, the DIN port receives data from the controller, the first pixel collects the initial

24-bit data, and then sends it to the internal data latch, and other data processed by the internal signal amplification circuit are sent to the next one cascade pixels through the DO port. After each pixel is transmitted, the signal is reduced by 24 bits. The pixel adopts self-shaping transmission technology, so that the number of pixel cascading is not limited by signal transmission, but only depends on the speed of signal transmission.

Reset time >280us, no false reset when interrupted;

Support low frequency, with cheap MCU.

The refresh rate is updated to 2KHz, flicker-free, which improves the excellent display effect.

Characteristic:

* The control circuit and LED share the only power supply.

* The control circuit and RGB chip are integrated in a package of 5050 components to form a complete addressable circuit

* The pixel has a built-in signal-shaping circuit. After the waveform is shaped to the next driver, it ensures that the waveform distortion does not accumulate.

* Built-in electronic reset circuit and power failure reset circuit.

* The three primary colors of each pixel can display 256 luminances, completing the display of 16777216 colors, and the scanning frequency is 2KHz.

* The cascade port transmits signals over a single wire.

* If the distance between any two points does not exceed 5m, no additional circuit is required to transmit the signal.

* When the refresh rate is 30fps, the number of cascades should not be less than 1024 pixels.

* Send data at 800Kbps.

* The color of the light is highly consistent and requires no external electronic components, not even capacitors.

1.2 Hardware Introduction:

WS2812B light strip: 30 beads per meter 9w, 60 beads per meter 18w, 144 beads per meter 43W,

voltage: (DC) DC5V, the power consumption of each lamp bead is about 0.3W

Power supply: 5V

Power when each lamp bead is fully lit: 0.3W

Current when each lamp bead is fully lit: 0.6mA

Each chip has four pins, and each pin is defined in the following table

NO.	Pin	Function
1	5V	Positive power supply
2	GND	Negative pole of power supply
3	DI	Data input pin, control data entry of MCU
4	DO	Data output pin, cascading multiple SW2812 outputs, connected to the DI of the next SW2812

cascading wiring method

The DO of the previous chip is connected to the DI of the next chip
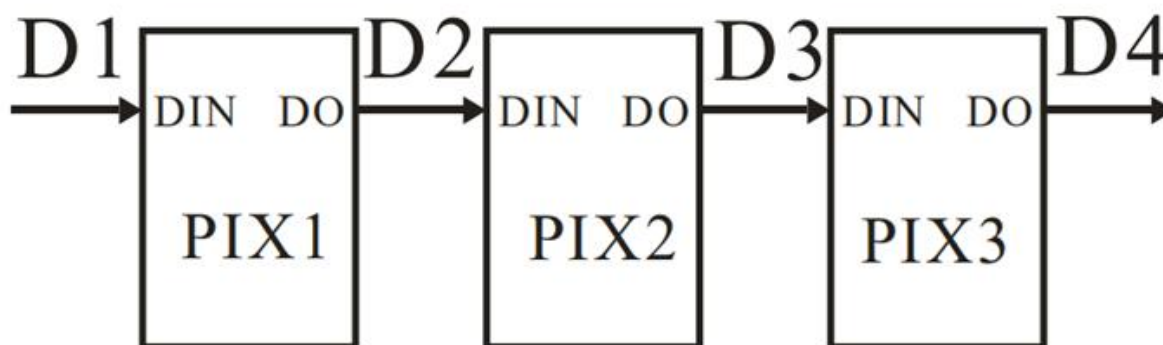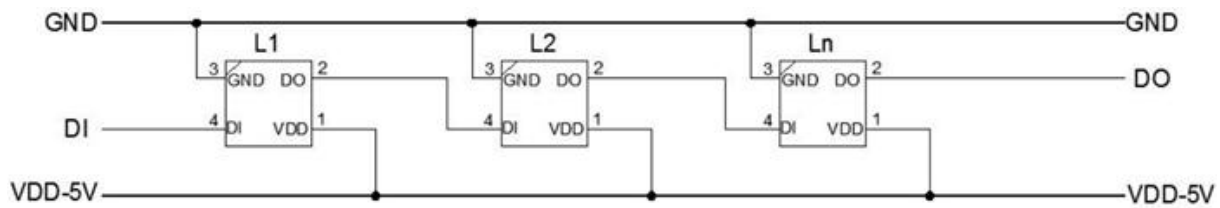
1.3 Control principle:

SW2812 is an RGB chip, so it has three colors of red, green and blue, and each color has corresponding 8 bits. Usually, a pixel is represented by three colors of RGB. For example, #FFFFFF means that the value of R (red) is 255, the value of G (green) is 255, the value of B (blue) is 255, #FFFFFFF is finally displayed as white. So a SW2812 consists of 3 U8s, that is 3*8=24 bits. To determine the color of a SW28122 chip, it is necessary to send 24 bits of data.

Cascade data transmission

The cache of the first screen data

The first 24 bits are received and cached by the first module

The second 24 bits will be forwarded by the first module to the second module and cached

The third 24 bits will be forwarded to the third module by the first and second, and cached

The fourth 24 bits...

The Nth 24 bits...

Reset signal, that is, the real embodiment of the cached data on the display

Second screen data cache

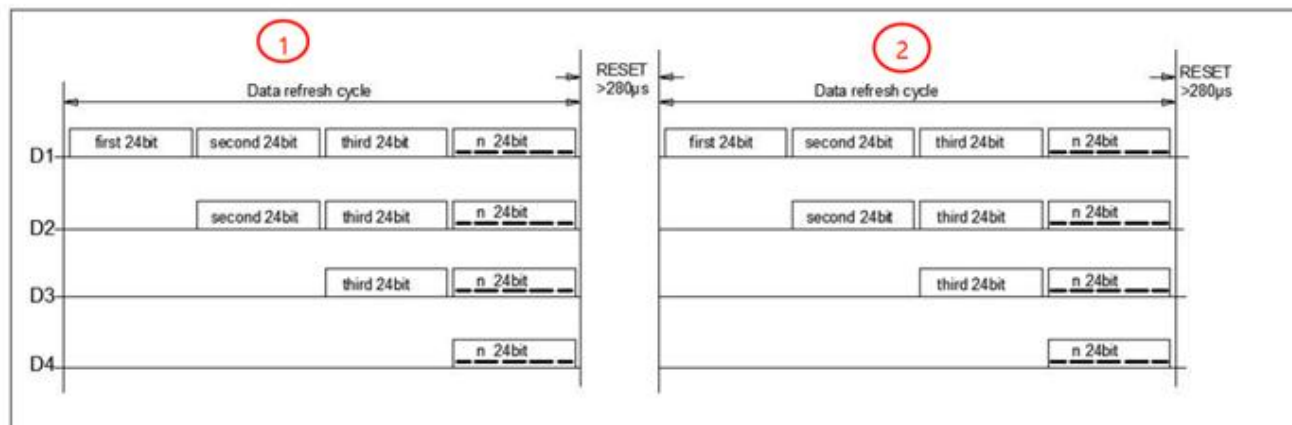The first 24 bits are received and cached by the first module

The second 24 bits will be forwarded by the first module to the second module and cached

The third 24 bits will be forwarded to the lower three modules by the first and second and cached
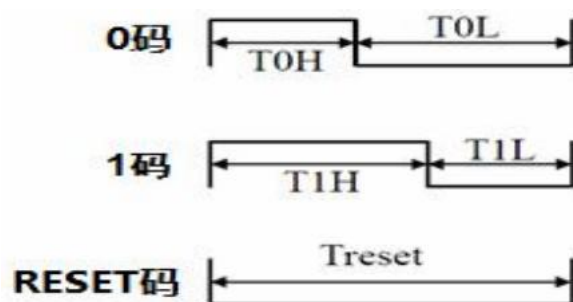
The fourth 24 bits...

Lower N 24 bits...

Reset signal, that is, the real embodiment of the cached data on the display

The meaning of each 24-bit data

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

Data is transmitted in GRB order, the high-order data bits are transmitted first

Support email:cokoino@outlook.com

| T0H	0 code, high voltage time	220ns~380ns
T1H	1 code, high voltage time	580ns~1μs
T0L	0 code, low voltage time	580ns~1μs
T1L	1 code, low voltage time	**580ns~1μs**
RES	Frame unit, low voltage time	>280μs

1.4 About the HSV color model in Adafruit_NeoPixel

In Adafruit_NeoPixel, the RGB color model can be used to control the red, green, and blue lights to synthesize various colors. The HSV color model can also be used to control the hue, saturation, and brightness of the lights to adjust the color.

The advantage of HSV control is that it is more convenient to control the brightness of the light and adjust the color to make it more in line with human intuition.
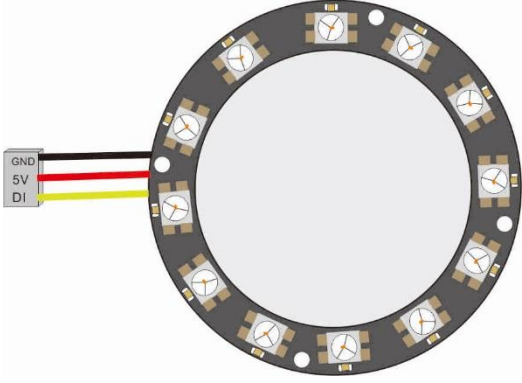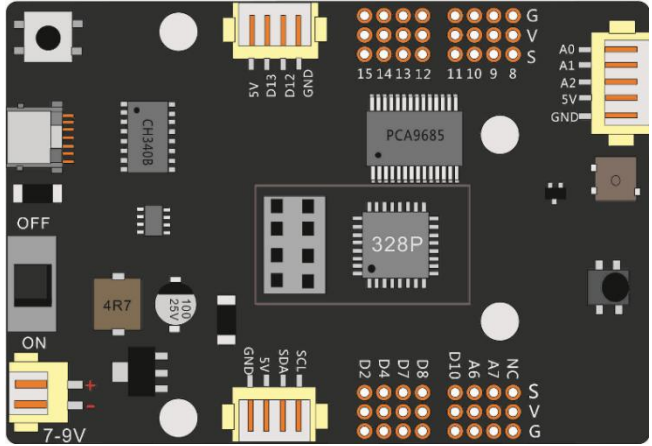
H in HSV: Hue parameter range 0--65535

S in HSV: Saturation parameter range 0--255

V in HSV: Brightness parameter range 0--255
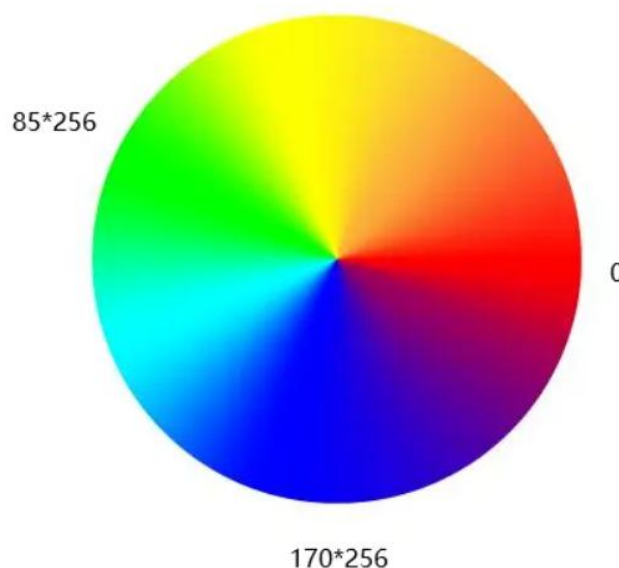
2. What do you need to prepare

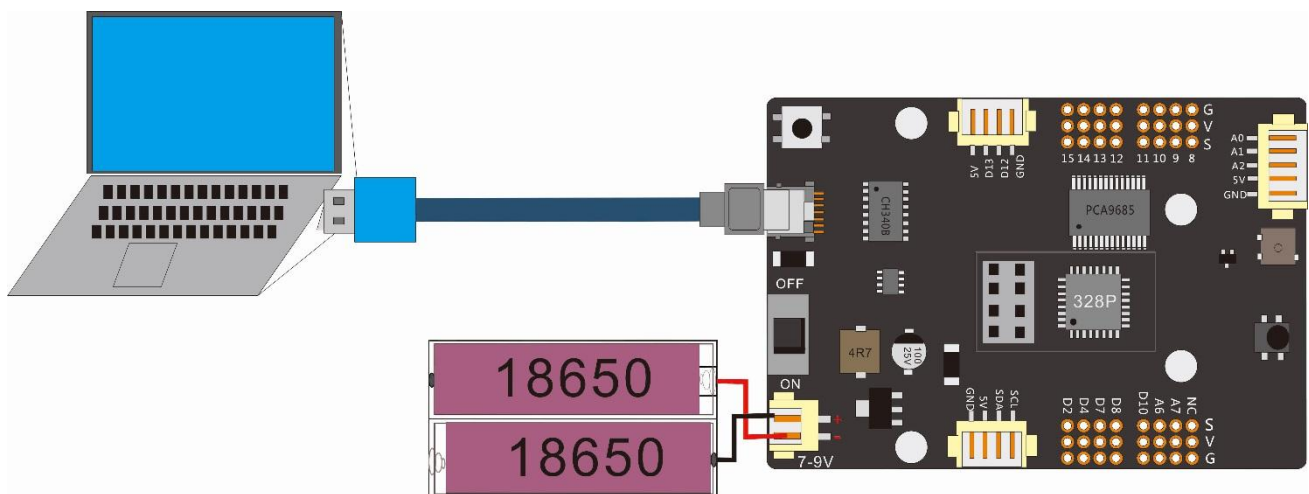| Components	Quantity	Picture
USB cable	1	
WS2812 LED module	1	
Control board	1	
18650 Battery	2	

Remak: 18650 batteries are not included in the kit,please prepared by yourself,thanks!

Support email:cokoino@outlook.com

3. Wiring

Connect the connector of WS2812 LED module to the X8 connector of the control board. The input pin used in this lesson is the D6 pin of Atmega328p. Turn the power switch of the control board to the OFF and connect it to the battery box with two 18650 batteries, then connect the control board to the computer with a USB cable, and turn the power switch to the ON.

Wiring between WS2812 LED module and control board	
Connector of the WS2812 LED module	X7 Connector of the control board
5V	5V
DI	D5
GND	GND

X5 GND 5V A3

X8 GND 5V D6

M1

100 25V

U1

M2
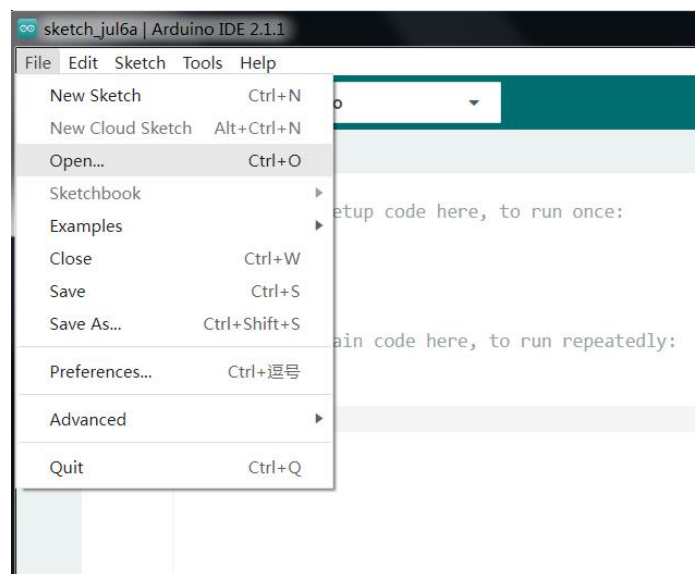
U3

M3

100 25V

X7 D5 5V GND
D1 5V GND

X6 D9 5V GND

M4
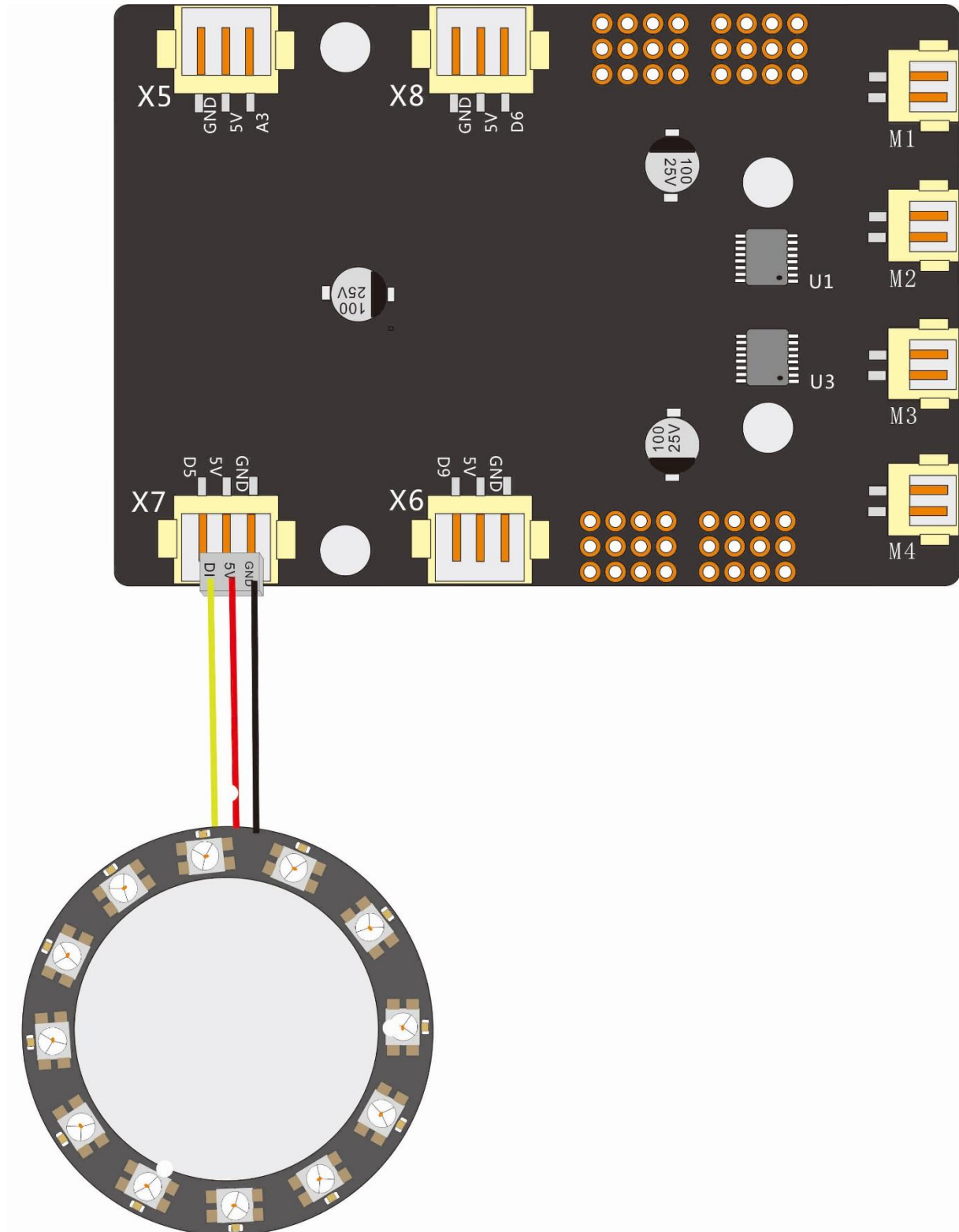
4. Upload the code and test

 The code used in this lesson is placed in this folder："E:\CKK0014-main\Tutorial\sketches"

4.1 Double-click the Arduino IDE shortcut on the desktop to open it

4.2 On the Arduino IDE interface, click "File" --- "open"

4.3 Select the code in the folder named 5_1_2812LED_RGBtest：
E:\CKK0014-main\Tutorial\sketches\5_1_2812LED_RGBtest.
Then click "open"

4.4 Select the board "Arduino UNO"and Port "COM3" (COM port is commonly known as an input output port for a device normally PC which enables communication between Arduino and PC. You can check your arduino com number in device manager, the com port of our arduino board is recognized as COM3 in this tutorial)
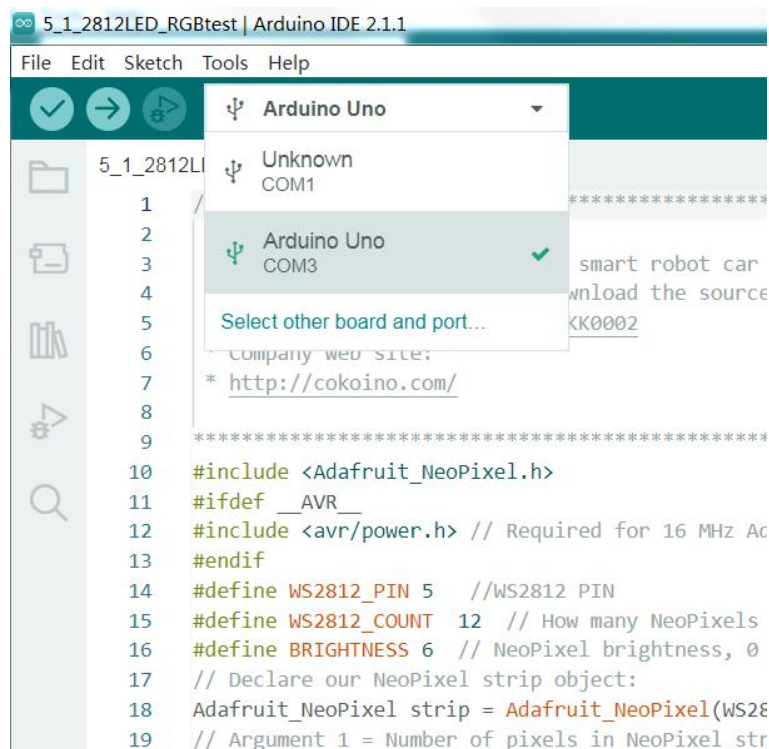
4.5 Install library Adafruit_NeoPixel.h

Installation method: Please refer to Install library Servo.h in Lesson 4

4.6 Click compile button, successfully compiled the code will display "Done compiling"

4.7 Before uploading the code, turn the ESP-01 switch on the control board to the side away from the "ESP-01" silk screen.

4.8 Click upload button, successfully uploading the code will display "Done uploading". When code is uploaded successfully, the program starts to run，you can see that the 12 LEDs on the 2812 LED module are cyclically changing colors from red to green to blue.

5. Code

5-1-2812LED-RGBtest.ino

```cpp
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#define WS2812_PIN 5   //WS2812 PIN
#define WS2812_COUNT  12  // How many NeoPixels are attached to the Arduino?
#define BRIGHTNESS 6  // NeoPixel brightness, 0 (min) to 255 (max)
// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip = Adafruit_NeoPixel(WS2812_COUNT, WS2812_PIN, NEO_GRB +
NEO_KHZ800);
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)
// Argument 3 = Pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)


///////////////////////////////////////////////////////////////////////////
void setup()
{
  // These lines are specifically to support the Adafruit Trinket 5V 16 MHz.
  // Any other board, you can remove this part (but no harm leaving it):
  #if defined (__AVR_ATtiny85__)
    if (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(BRIGHTNESS);
}
///////////////////////////////////////////////////////////////////////////
void loop()
{
  // Fill along the length of the strip in various colors...
  colorWipe(strip.Color(255, 0, 0), 6); // Red
  delay(800);
  //colorWipe(strip.Color(255, 150, 0), 6); // yellow
  //delay(800);
  colorWipe(strip.Color(0, 255, 0), 6); // Green
  delay(800);
  //colorWipe(strip.Color(0, 255, 255), 6); // CYAN
```

```
  //delay(800);
  colorWipe(strip.Color(0, 0, 255), 6); // Blue
  delay(800);
  //colorWipe(strip.Color(180, 0, 255), 6); // purple
  //delay(800);
  //colorWipe(strip.Color(127, 127, 127), 6); // White
  //delay(800);
  //colorWipe(strip.Color(0, 0, 0), 0); // Clear
}


void colorWipe(uint32_t c, uint8_t wait)
{
  for(uint16_t i=0; i<strip.numPixels(); i++) {// For each pixel in strip...
    strip.setPixelColor(i, c);                 //  Set pixel's color (in RAM)
    strip.show();                              //  Update strip to match
    delay(wait);                               //  Pause for a moment
  }
}


//Theatre-style crawling lights.
void theaterChase(uint32_t c, uint8_t wait)
{
  for (int j=0; j<10; j++) {  //do 10 cycles of chasing
    for (int q=0; q < 3; q++) {
      for (int i=0; i < strip.numPixels(); i=i+3) {
        strip.setPixelColor(i+q, c);    //turn every third pixel on
      }
      strip.show();
      delay(wait);
      for (int i=0; i < strip.numPixels(); i=i+3) {
        strip.setPixelColor(i+q, 0);        //turn every third pixel off
      }
    }
  }
}
```

6. Expansion exercises

This lesson also expanded on two code examples to demonstrate the diversity of WS2812 light

display. One is a colored flowing light, and the other is a rainbow light. It should be noted that these

two codes use the FastLED. h library, Please refer to Install library Servo. h in Lesson 4 for the

method to install the FastLED. h library

The code path for the colored flowing light is:

E:\CKK0014-main\Tutorial\sketches\5_2_color_flowing_light

The code path for rainbow lights is:

E:\CKK0014-main\Tutorial\sketches\5_3_Rainbow_light

Please refer to the fourth point of this lesson for the method of uploading code, which will not be

explained in detail here.

7. Any questions and suggestions are welcome

Thank you for reading this document!

If you find any errors and omissions in the tutorial, or if you have any suggestions and questions, please feel free to contact us at:

cokoino@outlook.com

We will do our best to make changes and publish revisions as soon as possible.

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 Amazon Store by search for "**LK

COKOINO**" on Ama zon. We will continue to launch fun, cost-effective, innovative and exciting

products.

Thank you again for choosing Cokoino products.

LK COKOINO