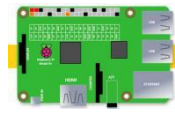
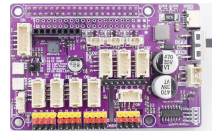




Lesson 9 How to Control WS2812 LED

In this lesson, we will learn how to control WS2812 LED.

9.1 Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.0	1	
3 pin cable	1	
WS2812 RGB LED	1	

9.2 Introduction of WS2812 LED

WS2812 LED module is a low-power RGB tri-color lamp with integrated current control chip. Its appearance is the same as a 5050LED lamp bead, and each element is a pixel. The pixel contains an intelligent digital interface data latch signal shaping amplifier driving circuit, and also contains a high-precision internal oscillator and a 12V high-voltage programmable constant current control part, which effectively guarantees that the color of the pixel light is highly consistent.



WS2812 LED is a very commonly used module on our robot products. There are three WS2812 LEDs on each module. Pay attention to the direction of the signal line when connecting. **The signal line needs to be connected to the "IN" port of WS2812 LED after being led from the Raspberry Pi. When the next WS2812 LED needs to be connected, we connect a signal wire drawn from the "OUT" port of the previous WS2812 LED with the "IN" port of the next WS2812 LED.**

When using the Raspberry Pi to install the driver board Adeept Robot HAT V3.0, the WS2812 LED can be connected to the WS2812 interface on the Adeept Robot HAT V3.0 using a 3pin cable.

We use a third-party library [rpi_ws281x] to control the WS2812 LED. You can learn about it via https://github.com/richardghirst/rpi_ws281x.

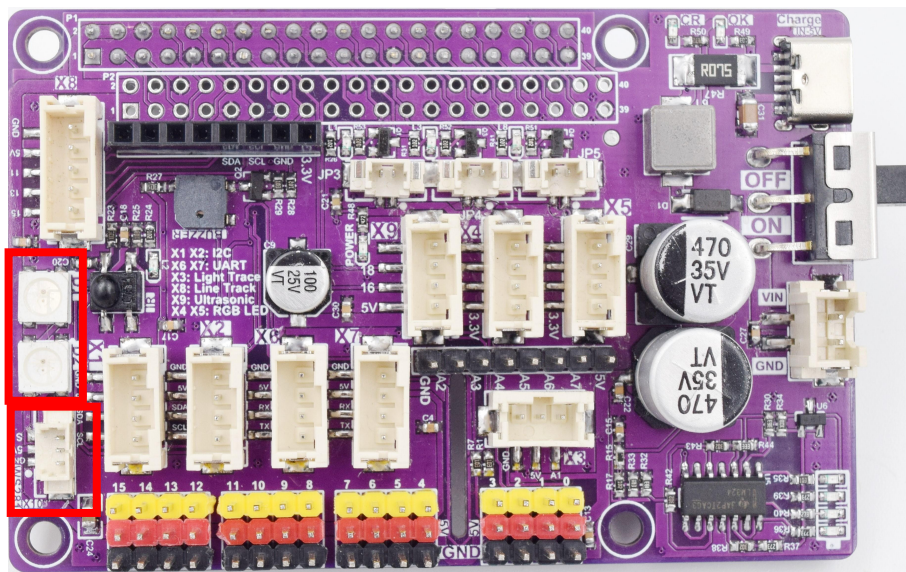
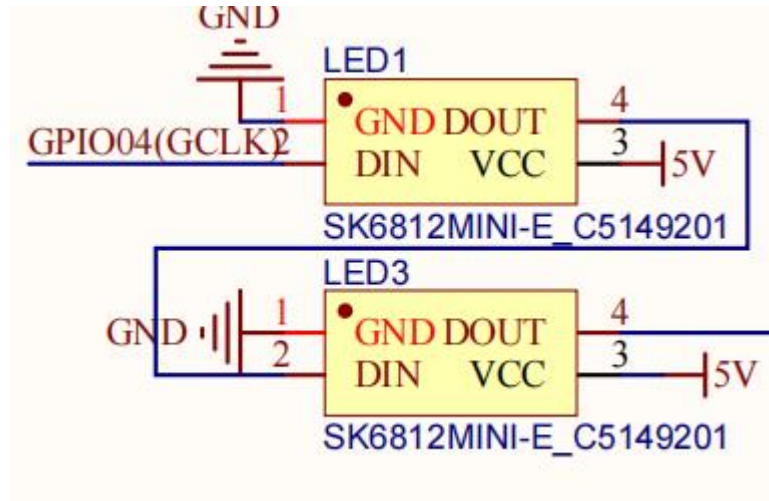
If you connect the WS2812 LED to the WS2812 interface of Adeept Robot HAT V3.0, the signal line is equivalent to connecting to the GPIO 12 of the Raspberry Pi.

9.3 Wiring diagram (Circuit diagram)

When the WS2812 LED is in use, the IN port needs to be connected to the WS2812 port on the Adeept Robot HAT V3.0 driver board.

Adeept Robot HAT V3.0 has 2 WS2812 LEDs on board, which are located at the front of the GPIO12 pin. When you use the WS2812 Port interface to connect one or more WS2812 LED

modules, the first and second (0 and 1) in the code control the two WS2812LEDs on the board. Starting from the 3rd one is used to control the extended WS2812LED light. (2,3,4,...)



Note: Currently this module is not compatible with Raspberry Pi 5. You need to wait for the WS1812 official update dependency library to be compatible with Raspberry Pi 5. Some of the latest hardware versions of Raspberry Pi 4 may also have incompatibility issues. This requires waiting for the official update of WS2812.

9.4 How to control WS2812 LED

Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

```
cd adept_rasptank2/examples/
```

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adept_rasptank2/examples/
pi@raspberrypi:~/adept_rasptank2/examples $
pi@raspberrypi:~/adept_rasptank2/examples $
```

3. View the contents of the current directory file:

```
ls
```

```
pi@raspberrypi:~/adept_rasptank2/examples $ ls
01_LED.py      03_servo.py   05_ws2812.py  07_lineTracking.py
02_buzzer.py  04_motor.py   06_ultra.py
```

4. Enter the command and press Enter to run the program:

```
sudo killall python3
```

```
pi@raspberrypi:~/adept_rasptank2/examples $
pi@raspberrypi:~/adept_rasptank2/examples $ sudo killall python3
pi@raspberrypi:~/adept_rasptank2/examples $
```

If an error is reported when running the 06_ws2812.py program, please check the Q&A of this tutorial.

```
sudo python3 05_ws2812.py
```

```
pi@raspberrypi:~/adeept_rasptank2/examples $ sudo python3 05_ws2812.py
```

Note: When you run the command on Raspberry Pi 5, you will get the following information.

```
WS2812 officially does not support Raspberry Pi 5 for the time being, and the WS2812 L
ED cannot be used on Raspberry Pi 5.
```

5. After running the program successfully, you will observe that the WS2812 alternately flashing lights of different colors.
6. When you want to terminate the running program, you can press the shortcut key "**Ctrl + C**" on the keyboard.

9.5 Main code program

For the complete code, please refer to the file [05_ws2812.py](#).

```
1. import time
2. from rpi_ws281x import *
3. import argparse
4.
5. # LED strip configuration:
6. LED_COUNT      = 3      # Number of LED pixels.
7. LED_PIN        = 12      # GPIO pin connected to the pixels (18 uses PWM!).
8. #LED_PIN       = 10      # GPIO pin connected to the pixels (10 uses SPI /dev/spidev0.0).
9. LED_FREQ_HZ    = 800000 # LED signal frequency in hertz (usually 800khz)
10. LED_DMA        = 10     # DMA channel to use for generating signal (try 10)
11. LED_BRIGHTNESS = 255    # Set to 0 for darkest and 255 for brightest
12. LED_INVERT     = False   # True to invert the signal (when using NPN transistor level shift
    )
13. LED_CHANNEL    = 0      # set to '1' for GPIOs 13, 19, 41, 45 or 53
14.
15. def setup():
16.     parser = argparse.ArgumentParser()
```

```
17. parser.add_argument('-c', '--clear', action='store_true', help='clear the display on exit')
18. args = parser.parse_args()
19.
20. # Create NeoPixel object with appropriate configuration.
21. strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ, LED_DMA, LED_INVERT, LED_BRIGHTNESS, LED_CHANNEL)
22. # Initialize the library (must be called once before other functions).
23. strip.begin()
24.
25. # Define functions which animate LEDs in various ways.
26. def colorWipe( R, G, B):
27.     """Wipe color across display a pixel at a time."""
28.     color = Color(R,G,B)
29.     for i in range(strip.numPixels()):
30.         strip.setPixelColor(i, color)
31.         strip.show()
32.
33. def run():
34.     colorWipe(0, 0, 255) # blue.
35.     time.sleep(2)
36.     colorWipe(0, 255, 0) # green.
37.     time.sleep(2)
38.     colorWipe(255, 0, 0) # red.
39.     time.sleep(2)
40.     colorWipe(0, 0, 0) # All lights off.
41.     time.sleep(1)
42.
43.
44. def check_rpi_model():
45.     _, result = run_command("cat /proc/device-tree/model |awk '{print $3}'")
46.     result = result.strip()
47.     if result == '3':
48.         return 3
49.     elif result == '4':
50.         return 4
51.     elif result == '5':
52.         return 5
53.     else:
54.         return None
55.
56. def run_command(cmd=""):
```



```
57. import subprocess
58. p = subprocess.Popen(
59.     cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
60. result = p.stdout.read().decode('utf-8')
61. status = p.poll()
62. return status, result
63.
64. if __name__ == '__main__':
65.     try:
66.         value = 1
67.         rpi_model = check_rpi_model()
68.         if rpi_model == 5:
69.             print("\033[1;33m WS2812 officially does not support Raspberry Pi 5 for the time being, and the WS2812 LED cannot be used on Raspberry Pi 5.\033[0m")
70.             value = 0
71.         else:
72.             setup()
73.             while value != 0:
74.                 run()
75.     except KeyboardInterrupt:
76.         colorWipe(0, 0, 0)
```

10.6 Q&A

- If an error message appears after running the program: *RuntimeError: ws2811_init failed with code -5.(mmap() failed)*

Please use "**sudo**" to run the program.
- If an error message appears after running the program: *RuntimeError: ws2811_init failed with code -3 (Hardware revision is not supported)*

1. Run in the Raspberry Pi command line:

```
cd ../../
```

```
sudo pip3 install rpi_ws281x --upgrade
```

2. Then run the product program:

```
sudo python3 aadept_rasptank2/examples/06_ws2812.py
```

Test whether it can be used. If you can run the program normally, please skip all the steps below. if an error occurs, if an error occurs, please continue with the following operations.

RuntimeError: ws2811_init failed with code -3 (Hardware revision is not supported)

3. Delete the rpi_ws281x library file:

```
sudo pip3 uninstall rpi_ws281x
```

4. Download the libraries needed to manually install the dependent libraries:

```
sudo apt-get update
```

```
sudo apt-get install gcc make build-essential python-dev scons swig
```

5. Download the ws2812 source code:

```
git clone https://github.com/jgarff/rpi_ws281x.git
```

6. View the CPU information of the Raspberry Pi.

```
sudo nano //proc/cpuinfo
```

Find the last few lines in the file


```
Hardware   : BCM2711  
Revision   : b03114  
Serial     : 10000000010aa3e17  
Model      : Raspberry Pi 4 Model B Rev 1.4
```

Information to remember: (used for the file to be modified next)

Revision: **b03114**

Model : Raspberry Pi 4 Model B Rev **1.4**

Exit via "Ctrl + X".

The above information can know the version number of the Raspberry Pi CPU: **b03114**

7. Modify the downloaded rpi_ws281x file.

```
sudo nano rpi_ws281x/rpihw.c
```

Add the following information, **the red part needs to be modified to your Raspberry Pi CPU information.**

```
{  
  
    .hwver = 0xRevision,  
  
    .type = RPI_HWVER_TYPE_PI4,  
  
    .periph_base = PERIPH_BASE_RPI4,  
  
    .videocore_base = VIDEOCORE_BASE_RPI2,  
  
    .desc = "Pi 4 Model B - 2GB  v1.4"  
  
},
```

CPU information "Model: Raspberry Pi 4 Model B Rev 1.4" so write v1.4 in .desc

If your "Model: Raspberry Pi 4 Model B Rev 1.2" write v1.2 in .desc.

For example: (the white part is newly added)

```
#define RPI_MANUFACTURER_MASK (0xf << 16)
#define RPI_WARRANTY_MASK (0x3 << 24)

static const rpi_hw_t rpi_hw_info[] = {
    //
    // Raspberry Pi 400
    //
    {
        .hwver = 0xc03130,
        .type = RPI_HWVER_TYPE_PI4,
        .periph_base = PERIPH_BASE_RPI4,
        .videocore_base = VIDEOCORE_BASE_RPI2,
        .desc = "Pi 400 - 4GB v1.0"
    },
    //
    // Raspberry Pi 4
    //
    {
        .hwver = 0xb03114,
        .type = RPI_HWVER_TYPE_PI4,
        .periph_base = PERIPH_BASE_RPI4,
        .videocore_base = VIDEOCORE_BASE_RPI2,
        .desc = "Pi 4 Model B - 2GB v1.4"
    },
    {
        .hwver = 0xa03111,
        .type = RPI_HWVER_TYPE_PI4,
        .periph_base = PERIPH_BASE_RPI4,
        .videocore_base = VIDEOCORE_BASE_RPI2,
        .desc = "Pi 4 Model B - 1GB v1.1"
    }
};
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
 ^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Save and exit with "Ctrl + X", "Y", "Enter". (文件中已有的树莓派 CPU 版本号是官方已经兼容的)

8. Compile the downloaded code.

```
cd rpi_ws281x/
```

```
sudo scon
```

9. Install rpi_ws281x dependency library.

```
cd python
```

```
sudo python3 setup.py build
```

```
sudo python3 setup.py install
```

10. Update dependent libraries.

```
sudo pip3 install rpi-ws281x --upgrade
```

11. Run the program code:

```
cd ../..
```

```
sudo python3 adeept_rasptank2/examples/06_ws2812.py
```

If the above method is not compatible with your Raspberry Pi version (for example: **c03114**), you need to wait for the compatibility update of ws281x dependent libraries, so this module is temporarily not available. This module needs to be shielded in order not to affect other functions of the robot.