Connect and Control WS2812 RGB LED Strips via Raspberry Pi

If you are looking for RGB LED strips, you will probably come across the WS2812 or the WS2801. These two LED strips are very different, but both can be controlled with the Raspberry Pi. After using the latter in a previous tutorial and using it in our Ambilight, this tutorial is about using the Raspberry Pi WS2812 RGB LED Strip.

The models WS2812B and WS2811 are also compatible and can be addressed with this guide.

It must be said that one should not be fooled by the “smaller” model number of the WS2801. This has – apart from the price – a few advantages, which will be discussed in more detail below.

Components

A plug adapter makes connecting the power cable to the power supply very easy.

Anyone who has already used an RGB LED strip (such as Ambilight) can continue to use the accessories. This includes the following in addition to a Raspberry Pi*:

- For beginners: power supply* + connector* (see picture right)
- For experienced: switching power supply 5V 10A* + power cord*
- Jumper cable*, ebay
- Breadboard*

The maximum power of the power supply depends on the number of LEDs. According to the data sheet, an LED under full load (= maximum brightness) needs about 60mA. At 5m and 30 LEDs / m, this is 9 amps. So, a power supply, which has up to 10A is suitable. If you have more LEDs in use, you may need more than one power supply (more on that later).

In addition, of course, the actual RGB LED strip of the type WS2812 (B) or WS2811 (also called NeoPixel) is required. These are available in three different versions, which differ in the number of LEDs per meter:

1. LED Strip with 30 LEDs per meter*
2. LED Strip with 60 LEDs per meter*
3. LED Strip with 144 LEDs per meter*

The length should be chosen depending on the project to be implemented. But mostly 5m rolls are a bit cheaper. My **recommendation** is the middle version with **60 LEDs/m**. These have a higher density of lights and thus a higher brightness. In terms of price, however, they are still cheaper than a similarly long WS2801 strip with only 32 LEDs

NeoPixel WS2812B / WS2811 vs. WS2801 on a Raspberry Pi

With only one data line, the frequency must be higher to carry the same amount of data.

In a previous tutorial we already saw how to control a WS2801 RGB LED strip. That stip differs from the WS2812B or WS2811 and has some advantages, which I would like to list here first:

- Having **two data lines** requires fewer calculations.
- As a result, a **higher frequency** can be achieved.
- The **brightness** of the WS2801 is higher.
- It is also possible to control the WS2801 and play music, which is not possible with the WS2812 (more on this below)

Now the question arises, why one should still use a WS2812 RGB LED strip on the Raspberry Pi, if the WS2801 nevertheless brings some advantages? The answer is quite simple: the **cheaper price**. In comparison, one meter of WS2801 is quite expensive, which is why you want to handle it more sparingly. At relatively low prices for several feet of a WS2812 strip, even a large projects can be realized without a huge budget.

However, it must be said that it is unfortunately not possible to simultaneously play sounds via the Raspberry Pi onboard sound card and control the strip. This is because the Raspberry Pi is not a real-time system like the Arduino or ESP8266 is. For the sound reproduction PWM is used, which is also needed to control the WS2812 on the Raspberry Pi. Using both at the same

time is not possible, which is why someone who relies on the sound reproduction should just take the WS2801.

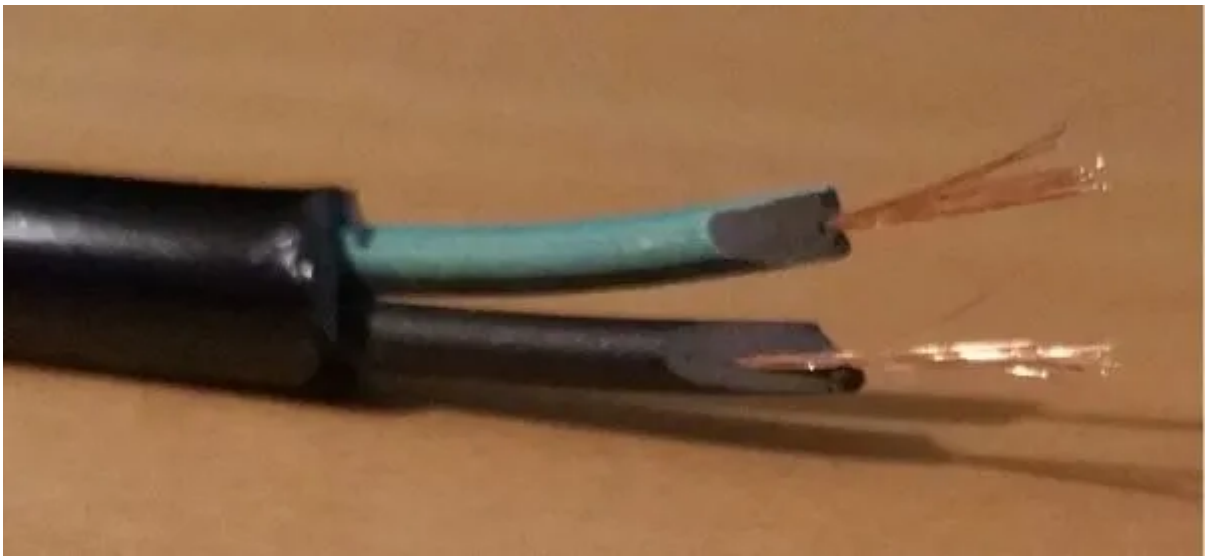
Connecting the NeoPixel WS2812 to the Raspberry Pi

Before we connect the Raspberry Pi to the WS2812 LED Strip, we finish the power supply. If your strip has less than 20-30 LEDs, external power is not required. If it has more LEDs, the power of the Pi, however, is no longer enough and an external power supply is required.

The power supply itself depends on the selected type. For a power adapter with additional DC adapter (beginners), only the DC adapter needs to be connected, which will be followed by the power cables of the WS2812B. Once you have chosen this method, you can jump to the next point. A switched-mode power supply (for experienced users), however, is a little more complex.

Before, however, the note: **When working with 230V caution is necessary because of mortal danger! Get help from an electrician or get back to the safe version of the normal power supply! Perform all work with a separate network connection.**

First, we need a standard power cable that can handle 10A. Mostly on the bottom of the plug is a corresponding note (230V, 10A). We cut this cable carefully with a utility knife or similar.It contains two or three cables with different colors. These cables must also be carefully stripped.



Power cord with separated insulation and two inner cables (black => L, blue => N).

My cable has only two inner cables, but that's enough. For connection to the switching power supply, the colors of the inner cables serve as orientation. The switching power supply has the connections "L", "N" and "PE" or a grounding symbol.

- The **black** or **brown** inner cable comes to the outer conductor "**L**".
  The **blue** inner cable comes to the neutral conductor "**N**".
  The **green-yellow** inner cable is connected to the protective conductor "**PE**" or **earthing symbol**.

The voltage should be about 5V.

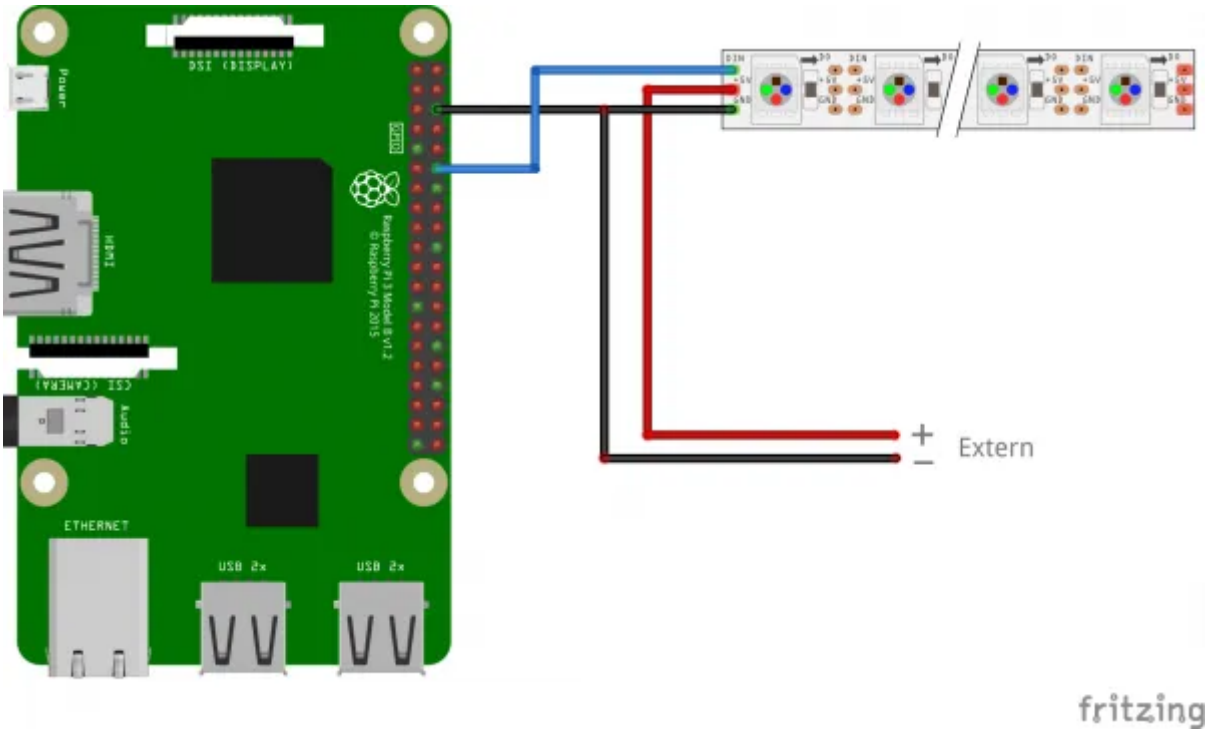
After the locking screws have been tightened, the power cord can be plugged into the socket and the voltage at the power supply can be measured with a multimeter. Although this is not mandatory, but can be made as a hedge. VCC is connected to V+ and GND to COM. If the voltage is not exactly 5V, this is not a problem.

Theoretically, it is also possible to power the Raspberry Pi from this power supply. Some users (according to comments in the WS2801 tutorial) also have done this. For reasons of space, this will not be discussed here.

Connection between Raspberry Pi and WS2812 NeoPixel stripes

If the power supply is set up so far, we connect the Raspberry Pi to the WS2812 RGB LED strip. The (switching) **power supply must first be disconnected from the power**.

Since there is only one data line, we only need one pin (GPIO 18). It is important that the ground connections of the Raspberry Pi and the switching power supply are connected, but not the 5V voltages! Overall, only two cables run from the Raspberry Pi to the WS2812 LED Strip: GPIO 18 (to DIN) and GND to COM of the PSU and GND of the strip.



Schematic connection between Raspberry Pi, WS2812 and the external power supply

Depending on the length of the LED strip, the external power connection should be installed in several places. Ideally, the VCC and GND will be connected in parallel with the switching power supply on approximately every meter (so that many cables will lead to the power supply input).

Note: For lengths less than 1m, the power supply can also be omitted and the input voltage from the RV 5V pin of the Raspberry Pi come.

Preparation & Installation

Before we install the Raspberry Pi library for the WS2812 LEDs, some preparations have to be made:

1. The package sources are updated:

```
sudo apt-get update
```

2. We install the required packages (confirm with Y):

```
sudo apt-get install gcc make  build-essential python-dev git scons sw
ig
```

3. The audio output must be deactivated. For this we edit the file

```
sudo nano /etc/modprobe.d/snd-blacklist.conf
```

Here we add the following line:

```
 blacklist snd_bcm2835
```

Then the file is saved by pressing CTRL + O and CTRL + X closes the editor.

4. We also need to edit the configuration file:

```
sudo nano /boot/config.txt
```

Below are lines with the following content (with Ctrl + W you can search):

```
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
```

This bottom line is commented out with a hashtag # at the beginning of the line:
#dtparam=audio=on

5. We restart the system

```
sudo reboot
```

Now we can download the library.

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

In this directory are on the one hand some C files included, which can be easily compiled. The example code for this is easy to understand. In order to use them in Python, we need to compile them:

```
cd rpi_ws281x/
sudo scons
```

However, in this tutorial we are mainly interested in the Python variant and therefore switch to the Python folder:

```
cd python
```

Here we carry out the installation:

```
sudo python3 setup.py build
sudo python3 setup.py install
sudo pip3 install adafruit-circuitpython-neopixel
```

This will allow us to carry out a first test in the next step.

Test the Raspberry Pi WS2812 RGB LED Strip

In the example folder are some example files, with which the LED strips can be tested. In addition, even two WS2801 LED strips can be independently controlled by Raspberry Pi (multistrandtest.py).

We are initially interested in the simple version. For this we have to complete a few details before the test and therefore we edit the sample file.

```
sudo nano examples/strandtest.py
```

The file looks like this, where we have to state LED_COUNT (number of LEDs to be addressed) and LED_PIN (18 us).

```
Ein Fehler ist aufgetreten. Bitte versuchen Sie es später erneut.
```

Then we save (CTRL + O) and return to the terminal (CTRL + X). Now the file can be executed (passing the path to the compiled files is important):

```
sudo python3 examples/strandtest.py
```

The LEDs of the WS2812 strip should light up as in the video below. In the upper code only a few effects (rainbow, etc.) are defined, however, it is also possible to create further effects analogously (everybody who wants to post his code for effects as a comment is welcome).

Something more

The pHAT is available for Raspberry Pi 3 Model B and Zero.

The WS2812 NeoPixel LEDs are also used in many other projects (mainly designed for Arduino), so you can certainly port some of them.

In addition, the Raspberry Pi Shop Pimoroni made an attachment for the Model B (from version B +) and the Raspberry Pi Zero and calls this unicorn pHAT*. However, this RGB LED matrix uses its own software for operation.

Incidentally, the library I use is also offered by another developer as a server application.

C++ Featured LED led array led display led graphik led matrix LED Strip NeoPixel python WS2801 LED Stripe WS2801B WS2811 WS2812 LED Stripe WS2812B

* Some links may be affiliate links. We get money if you buy something or take an action after clicking one of these links on our site

PREVIOUS ARTICLE
Mehrere Servo Motoren gleichzeitig per Raspberry Pi steuern (PCA9685)

NEXT ARTICLE
6-Gelenkigen Roboter Arm am Raspberry Pi betreiben (6 DOF)

RELATED POSTS