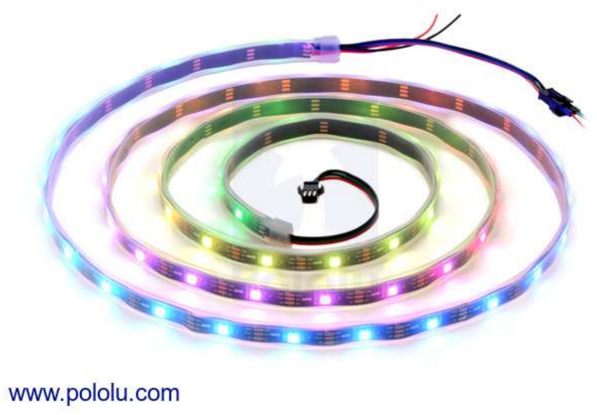
Addressable RGB 60-LED Strip, 5V, 2m (WS2812B)



Pololu item #: 2547
Brand: **Unspecified**

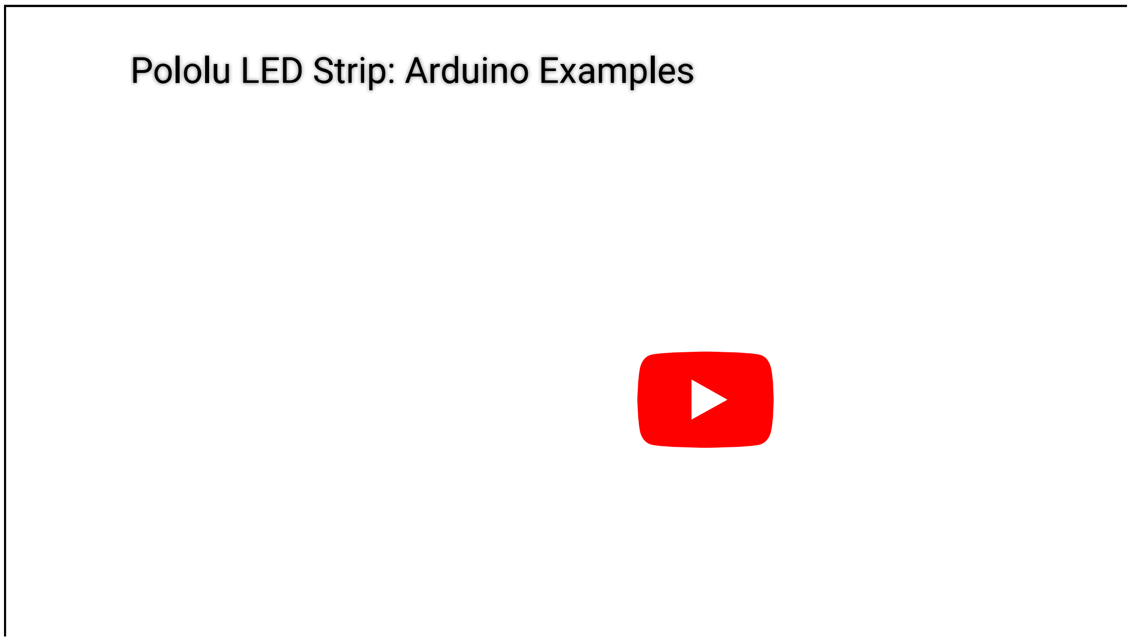
This product has been discontinued.

Please consider the similar **Addressable RGB 60-LED Strip, 5V, 2m (SK6812)**, which features the same interface and color order and can generally be used as a drop-in replacement.

This **2-meter** long strip contains **60 RGB LEDs** that can be individually addressed using a one-wire interface, allowing you full control over the color of each RGB LED. The flexible, waterproof strip runs on 5 V and can be chained with additional WS2812B strips to form longer runs or cut apart between each LED for shorter sections.

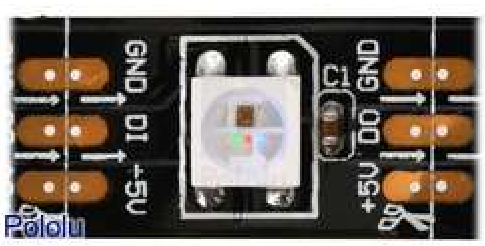
Alternatives available with variations in these parameter(s): length RGB LED density **Select variant…**

Description Specs (7) Pictures (11) Resources (7) FAQs (1) On the blog (2)



Overview

These flexible RGB LED strips are an easy way to add complex lighting effects to a project. Each LED has an integrated driver that allows you to control the color and brightness of each LED independently. The combined LED/driver IC on these strips is the extremely compact WS2812B (essentially an improved WS2811 LED driver integrated directly into a 5050 RGB LED), which enables higher LED densities. In the picture on the right, you can actually see the integrated driver and the bonding wires connecting it to the green, red, and blue LEDs, which are on at their dimmest setting.



Close up of one segment of a WS2812B-based LED strip, with the red, green, and blue LEDs on at their dimmest setting.

In contrast to the APA102C used in some of our **[other similar LED strips](#)**, which uses a standard SPI interface (with separate data and clock signals), the WS2812B uses a specialized one-wire control interface and requires strict timing. See the bottom of this product page for a more detailed comparison of the WS2812B and APA102C.

Close up of a WS2812B, with the red, green, and blue LEDs on at their dimmest setting.

We offer six different kinds of WS2812 LED strip with different LED densities and lengths. Our strips with **30 LEDs per meter** are available in three lengths:

- **1 meter, 30 LEDs**
- **2 meters, 60 LEDs**

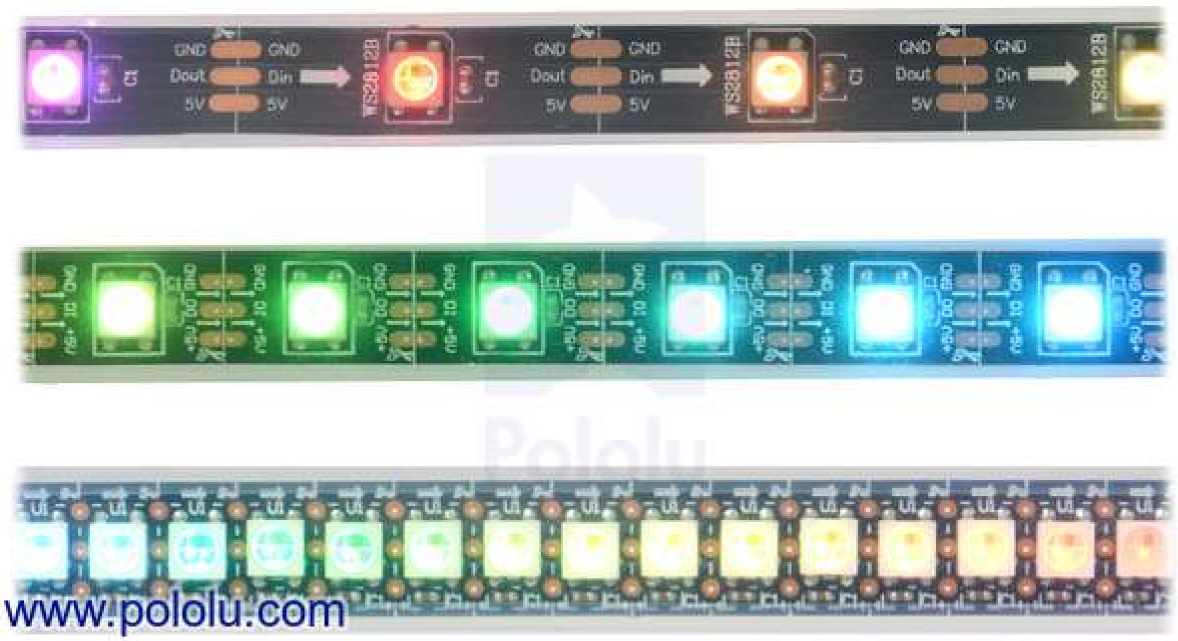
- **5 meters, 150 LEDs**

We also offer denser WS2812 LED strips that have **60 LEDs per meter**:

- **1 meter, 60 LEDs**
- **2 meters, 120 LEDs**

Our **highest-density strip** has its WS2812 LEDs packed together as tightly as possible, resulting in 72 LEDs on a 0.5 meter strip (i.e. **144 LEDs per meter**).

The information on this page applies to all of the WS2812-based LED strips we sell.



LED side of the WS2812B-based addressable LED strips, showing 30 LEDs/m (top), 60 LEDs/m (middle), and 144 LEDs/m (bottom).

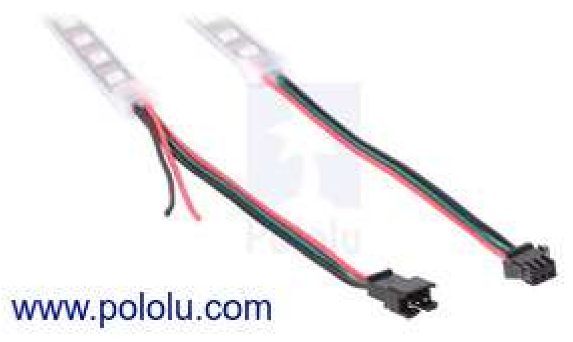
Features and specifications

- Individually addressable RGB LEDs (30, 60, or 144 LEDs per meter)
- 24-bit color control (8-bit PWM per channel); 16.8 million colors per pixel
- One-wire digital control interface
- 5 V operating voltage
- Each RGB LED draws approximately 50 mA at 5 V with red, green, and blue at full brightness
- 12 mm width, 4.6 mm thickness
- Flexible, waterproof silicone rubber sheath (**IP65 protection rating**)

- Includes flexible silicone mounting brackets
- Black strip color
- Power/data connectors on both strip ends for easy chaining, and the input side includes an additional power and ground wire for alternate power connections
- Strips can be cut apart along the lines between each RGB LED segment to separate them into usable shorter sections
- Example code available for Arduino, AVR, and mbed

Using the LED strip

Each LED strip has three connection points: the input connector, the auxiliary power wires, and the output connector. These can be seen in the adjacent picture, from left to right: auxiliary power wires, input connector, output connector. The strip uses 3-pin JST SM connectors.



The connectors and power wires for addressable LED strips. On the left is the input end of the strip and on the right is the output end.

The **input connector** has three male pins inside of a plastic connector shroud, each separated by about 0.1″. The black wire is ground, the green wire is the signal input, and the red wire is the power line.

The **auxiliary power wires** are connected to the input side of the LED strip and consist of stripped black and red wires. The black wire is ground, and the red wire is the power line. This provides an alternate (and possibly more convenient) connection point for LED strip power.

The **output connector** is on the other end of the strip and is designed to mate with the input connector of another LED strip to allow LED strips to be chained. The black wire is ground, the green wire is the signal output, and the red wire is the power line.

All three black ground wires are electrically connected, and all three red power wires are electrically connected.

A close-up of the JST SM connectors for our addressable LED strips.

Included hardware

These LED strips ship with flexible silicone brackets and screws. Strips with lengths of 1 meter or greater include five brackets and ten screws *per meter*. Our 0.5 meter high-density strip ships with a total of two brackets and four screws. The brackets fit over the waterproof sheath and can be used to mount the LED strip. The LED strip also ships on a plastic reel.

The 1m, 2m, and 5m addressable LED strips include five mounting brackets per meter; the 0.5m strip includes 2 total brackets.

A 2-meter, 60 LED addressable RGB LED strip on the included reel.

Connecting the LED strip

To control the LED strip from a microcontroller, two wires from the input connector should be connected to your microcontroller. The LED strip’s ground (black) should be connected to ground on the microcontroller, and the LED strip's signal input line (green) should be connected to one of the microcontroller's I/O lines. The male pins inside the input connector fit the female terminations on our **premium jumper wires** and **wires with pre-crimped terminals**. If you are connecting the LED strip to a breadboard or a typical Arduino with female headers, you would want to use **male-female wires**.

Controlling an addressable RGB LED strip with an Arduino and powering it from a 5V wall power adapter.

We generally recommend powering the LED strip using the auxiliary power wires. Our **5 V wall power adapters** work well for powering these LED strips and a **DC Barrel Jack to 2-Pin Terminal Block Adapter** can help you make the connection between the adapter and the strip. However, you might need a **wire stripper** to strip off some more insulation from the power wires.

It is convenient that the power wires are duplicated on the input side because you can connect the auxiliary power wires to your 5 V power supply and then the power will be available on the data input connector and can be used to power the microcontroller that is controlling the LED strip. This means you can power the microcontroller and LED strip from a single supply without having to make branching power connections.

Warning: The WS2812B seems to be more sensitive than the TM1804 on our original LED strips. We recommend taking several precautions to protect it:

- Connect a capacitor of at least 100 μF between the ground and power lines on the power input.
- Avoiding making or changing connections while the circuit is powered.
- Minimize the length of the wires connecting your microcontroller to the LED strip.
- Follow generally good engineering practices, such as taking precautions against electrostatic discharge (ESD).
- Consider adding a 100 Ω to 500 Ω resistor between your microcontroller's data output and the LED strip to reduce the noise on that line.

If the strip does get damaged, it is often just the first LED that is broken; in such cases, cutting off this first segment and resoldering the connector to the second segment brings the strip back to life.

Making a custom cable

If you do not want to use our **premium jumper wires** to connect to the LED strip's input, it is possible to make a custom cable.

One option for making a custom cable is to cut off the unused output connector on the last LED strip in your chain. This can then be plugged into the input connector of the first LED strip. The wires on the output and input connectors are 20 AWG, which is too thick to easily use with our crimp pins and housings, but you could solder the wires to header pins.

Alternatively, you can get your own JST SM connectors and make a custom cable using those. The parts you would need to get are the SMP-03V-BC and the SHF-001T-0.8BS, which are described in the **SM Connector datasheet** from JST. These can be purchased from several places, and we got them from **Heilind**. You will also need some 22–28 AWG **stranded wire** and a **wire stripper**. We do not know of a great way to crimp wires onto the JST crimp pins, but we were able to successfully do it using our **narrower crimping tool** and **pliers**. (With the wider crimping tool, it is hard to avoid crimping parts of the pin that should not be crimped.) Before crimping, use pliers to bend the outer set of tabs a little bit so that they can hold on to the insulation of the wire. This makes it easier to position the crimp pin and the wire. Next, you should be able to follow the instructions on the **crimping tool product page** to crimp the wire. After that, you will probably need to squeeze the crimp pin with pliers to get it to fit into the JST plug housing. On the other end of the cable you could make a custom connector using our **crimp pins** and **crimp connector housings**, which will allow you to plug it directly into a breadboard or 0.1″ header pins.

Current draw and voltage drop

Each RGB LED draws approximately 50 mA when it is set to full brightness and powered at 5 V. This means that for every 30 LEDs you turn on, your LED strip could be drawing as much as 1.5 A. Be sure to select a power source that can handle your strip's current requirements.

There is some resistance in the power connections between the LEDs, which means that the power voltage near the end of the strip will be less than the voltage at the start of the LED strip. As the voltage drops, RGB LEDs tend to look redder and draw less current. This voltage drop is proportional to the current through the strip, so it increases when the LEDs are set to a higher brightness.

We tested the current draw and voltage drop of some LED strips by setting all the LEDs to full brightness, and these were the results:

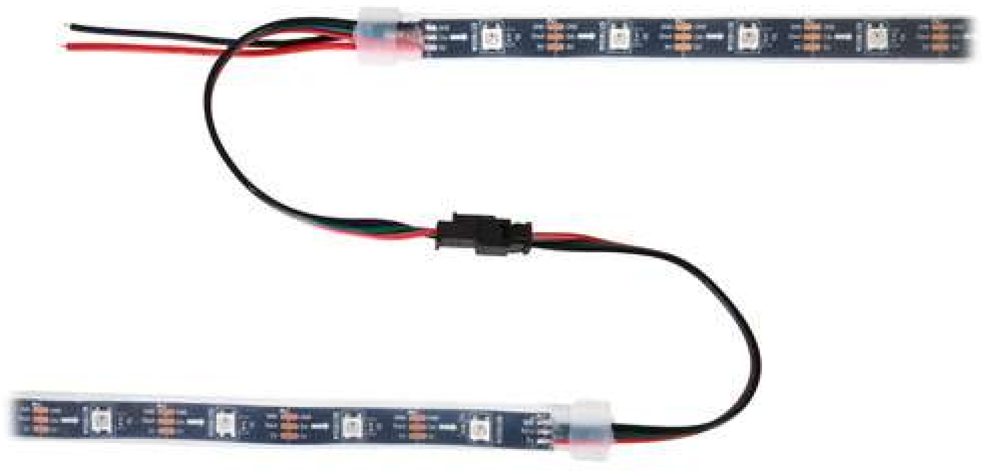
- The 30 LED 1 m strip drew 1.5 A and had a voltage drop of 0.2 V.
- The 60 LED 2 m strip drew 2.9 A and had a voltage drop of 0.8 V.
- The 150 LED 5 m strip drew 4.1 A and had a voltage drop of 2.0 V.
- The 60 LED 1 m strip drew 3.0 A and had a voltage drop of 0.6 V.

- The 120 LED 2 m strip drew 4.7 A and had a voltage drop of 1.4 V.

The voltage drop was computed by measuring the voltage difference between ground and power on the input end of the strip, then doing the same measurement on the output end, and subtracting the two values.

Chaining

Multiple LED strips can be chained together by connecting input connectors to output connectors. When strips are chained this way, they can be controlled and powered as one continuous strip. Please note, however, that as chains get longer, the ends will get dimmer and redder due to the voltage drop across the strip. If this becomes an issue, you can chain the data lines while separately powering shorter subsections of the chain.



Two addressable RGB LED strips connected.

We recommend chains of LEDs powered from a single supply not exceed 180 total RGB LEDs. It is fine to make longer chains with connected data lines, but you should power each 180-LED section separately. If you are powering each section from a different power supply, you should cut the power wires between the sections so you do not short the output of two different power supplies together.

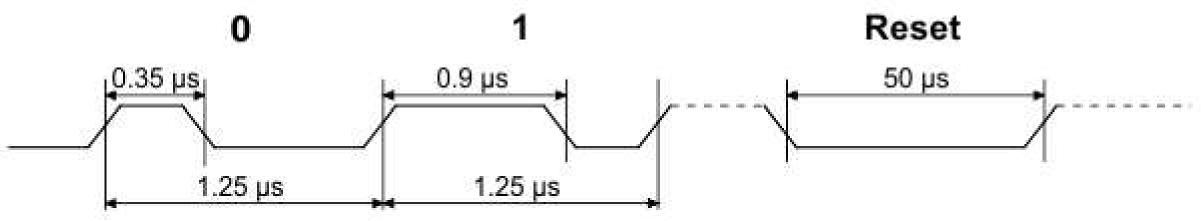
Cutting

The LED strip is divided into segments, with each segment containing one RGB LED. The strip can be cut apart on the lines between each segment to separate it into usable shorter sections. The data connection is labeled **DO**, **Dout**, **DI**, or **Din**, the positive power connection is labeled **5V**, and the ground connection is labeled **GND**. Each LED in the picture below is at the center of its own segment; there are little scissors drawn on the PCB silkscreen where the segments can be cut.

Protocol

These LED strips are controlled by a simple, high-speed one-wire protocol on the input signal line. The protocol is documented in the **WS2812B datasheet** (266k pdf) and also below.

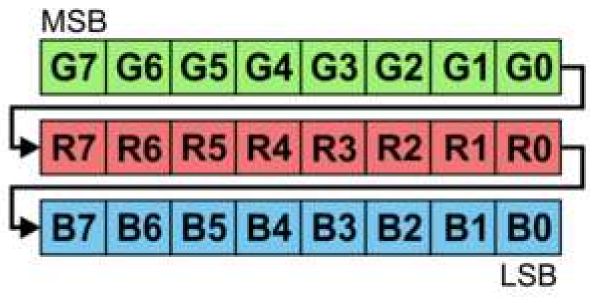
The default, idle state of the signal line is low. To update the LED colors, you need to transmit a series of high pulses on the signal line. Each high pulse encodes one bit: a short pulse (0.35 μs) represents a zero, while a long pulse (0.9 μs) represents a one. The time between consecutive rising edges should be 1.25 μs (though in our tests, the strips worked with cycle times up to approximately 6 μs). After the bits are sent, the signal line should be held low for 50 μs to send a reset command, which makes the new color data take effect (note: it is possible for low pulses as short as 6 μs to trigger a reset). The pulse widths do not have to be precise: there is a threshold that determines whether the pulse is a 0 or a 1, and a wide range of pulse widths on both sides of the threshold will work.



SK6812/WS281x RGB data timing diagram.

The color of each LED is encoded as three LED brightness values, which must be sent in GRB (green-red-blue) order. Each brightness value is encoded as a series of 8 bits, with the most significant bit being transmitted first, so each LED color takes 24 bits. The first color transmitted applies to the LED that is closest to the data input connector, while the second color transmitted applies to the next LED in the strip, and so on.

To update all the LEDs in the strip, you should send all the colors at once with no pauses. If you send fewer colors than the number of LEDs on the strip, then some LEDs near the end of the strip will not be updated. For example, to update all 30 LEDs on a 1-meter strip, you would send 720 bits



24 bits represent the color of one SK6812/WS281x LED in an addressable RGB LED strip.

encoded as high pulses and then hold the signal line low for 50 µs. If multiple strips are chained together with their data connectors, they can be treated as one longer strip and updated the same way (two chained 1-meter strips behave the same as one 2-meter strip).

Each RGB LED receives data on its data input line and passes data on to the next LED using its data output line. The high-speed protocol of the WS2812B allows for fast updates; our library for the Arduino below takes about 1.1 ms to update 30 LEDs, so it is possible to update 450 LEDs faster than 60 Hz. However, constant updates are not necessary; the LED strip can hold its state indefinitely as long as power remains connected.

Implementing the protocol on a microcontroller

Since this LED strip does **not** use a standard protocol, a software bit-banging approach is usually needed to control it from a microcontroller. Because of the sub-microsecond timing, the bit-banging code generally needs to be written in assembly or very carefully optimized C, and interrupts will need to be disabled while sending data to the LED strip. If the interrupts in your code are fast enough, they can be enabled during periods where the signal line is low.

> **Note:** The minimum logic high threshold for the strip data line is 3.5 V, so you should use level-shifters if you want to control these strips from 3.3 V systems. In our tests, we were able to control them with 3.3 V signals from an mbed, but using the strip out of spec like this could lead to unexpected problems.

Sample code

To help you get started quickly, we provide sample code for these microcontroller platforms:

- **PololuLedStrip Arduino library** (also works with our Arduino-compatible **A-star modules**)
- **Example AVR C code**
- **PololuLedStrip mbed library**

Additionally, the **Adafruit NeoPixel library** for Arduino should work with these strips since the NeoPixels are based on the WS2812B.

Comparison with TM1804 LED Strips

These WS2812B-based strips are similar in many ways to our older high-speed TM1804 LED strips (items **#2543**, **#2544**, and **#2545**). The WS2812B's timing parameters are very similar to those of the high-speed TM1804 LED strips, so you can use the same code to control either of them and you can chain one type to the other. However, the two types of strips have different, incompatible connectors, and the order of the red and green channels in the protocol is swapped: the TM1804 colors are sent in red-green-blue order while the WS2812B colors are sent in green-red-blue order.

The TM1804 is just an LED driver and it requires a separate RGB LED to be placed on the strip. Since the WS2812B combines the LED and the driver in a single package, it can be packed more densely, which is why we are able to offer strips with 60 LEDs per meter.

Unlike the TM1804 strips, these LED strips do not have an adhesive backing, but they do include mounting brackets as described above.

Comparison with APA102C LED Strips

Like the WS2812B, the APA102C used in **some of our newer LED strips** also combines an RGB LED and driver into a single 5050-size package, allowing them to be packed as densely as 144 LEDs per meter. However, while the WS2812B uses a one-wire control interface with strict timing requirements (timing requirements so strict that it is typically impractical to have interrupt-based events running on the controlling microcontroller while it is updating the WS2812B LEDs), the APA102C uses a standard SPI interface, with separate data and clock signals, that lets it accept a wide range of communication rates; the trade-off is that two I/O lines are required to control it instead of just one.

The APA102C provides a 5-bit color-independent brightness control that is not available on the WS2812B. This feature can be used to vary the intensity of each pixel without changing its color, and it enables much subtler variations at the low end of the LEDs’ brightness range.

In addition, the APA102C uses a much higher PWM (pulse-width modulation) frequency for controlling each color channel—about 20 kHz, compared to around 400 Hz on the WS2812B. As a result, APA102C LEDs can be less prone to flickering when recorded with a camera and are more suited to applications like persistence-of-vision (POV) displays. (The color-independent brightness is modulated separately at about 600 Hz).

For further comparison of the ICs, see the **WS2812B datasheet** (266k pdf) and **APA102C datasheet** (1MB pdf).

While our WS2812B strips and APA102C strips are physically very similar, they are **not** functionally compatible with each other. The easiest way to tell them apart is to look at the strips’ end connectors and the connections between each LED segment: WS2812B strips have three connections (power, data, and ground), while APA102C strips have four (power, clock, data, and ground). On strips with 30 LEDs/m, you can also check whether “WS2812B” or “APA-102C” is printed next to each LED.

People often buy this product together with:

Wall Power Adapter: 5VDC, 3A, 5.5×2.1mm Barrel Jack, Center-Positive

DC Barrel Jack to 2-Pin Terminal Block Adapter