Interface to driver for WS2812B-LEDs

Mattscheibe, January 2021

There are quite a few libraries for driving strands of WS2812B light emitting diodes. Every LED pixel is made up of the cheapest individually addressable type, WS2812B.

You can find more drivers for microcontrollers (e.g. "fastled" for the Arduino platform) than for the Raspberry Pi micro computer. Some always present the strand of LEDs as one linear strand. Users therefore have to do calculations "by hand" if they arrange the strand in several stripes glued together to form a matrix. If in a matrix it would be more convenient to address pixels by their row and their column index instead of by their absolute position in the strand. It would be even more convenient if it would be possible to glue and solder in any sort of possible order, with every stripe running in counterdirections, for instance, allowing for easy connections at the sides, and the entry point of the Din driving signal at any one of four possible position (any one of four corners of the matrix rectangle). (One might, for example, want to use 3 x 3 LEDs as a tic-tac-toe game—cf. to 1).

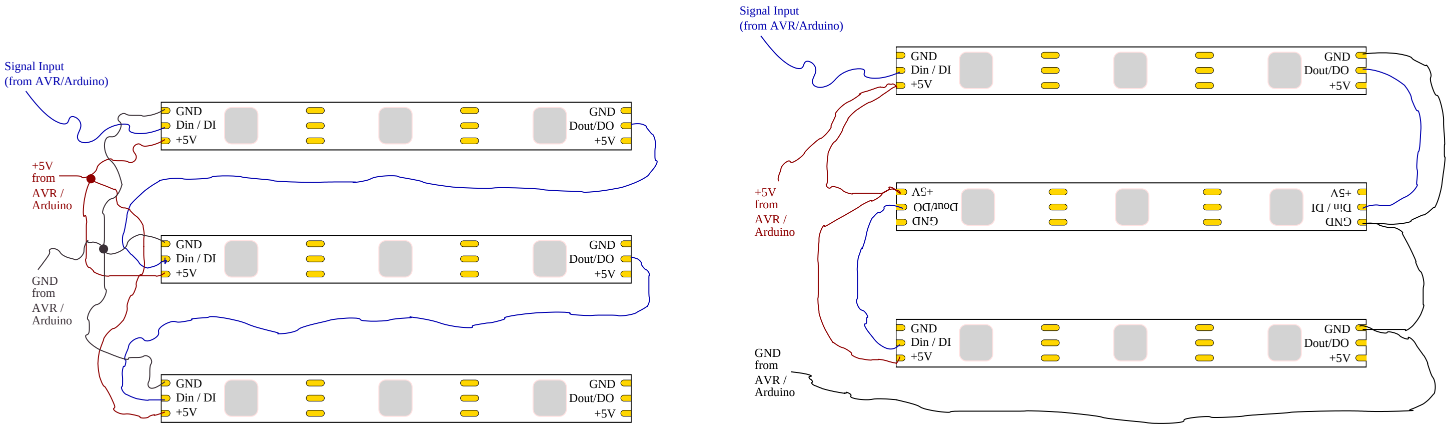


Figure 1: two examples for wiring: 3x3 matrix, signal input at top left position, mounted horizontally

If you glue all stripes in the same direction (example to the left), every Dout at the end of one stripes needs to traverse the matrix to get back to Din of its successor strip. If you glue all stripes in alternating directions (flip over every second one), it is easier to get from Dout of one stripe to Din of the next stripe. Din and Dout of successor stripe are always on the same side of the matrix, not on opposing sides. The ws2812b_rpi library presented here makes it easy to address every pixel in such a matrix just by using its row and its column index. In this figure, the scheme to the left is of type 5 and the scheme to the right, with alternating directions, is of type 4 (cf. back of this page). Both have their input signal on the top, to the left.

All in all there are 16 possibilities of how a rectangular matrix can be wired and driven.[1] ws2812b_rpi allows you to do the wiring first (without any restrictions of possiblities) and than to easily drive the matrix by choosing the appropriate wiring scheme in the driver's settings (a number between 0 and 15, addressing one of 16 possiblities). You can set colors for every color channel (red, green or blue), either seperately (with three different commands for red, for green or for blue, respectively) or in one single command, setting it to a 24 bit RGB-value (of type 0xFACE42). This RGB value would be the same as is being used for HTML colors, for instance (that "#FACE42" thing), or in Image Processing software like in GIMP (its color picker tool would tell you such values, for example).

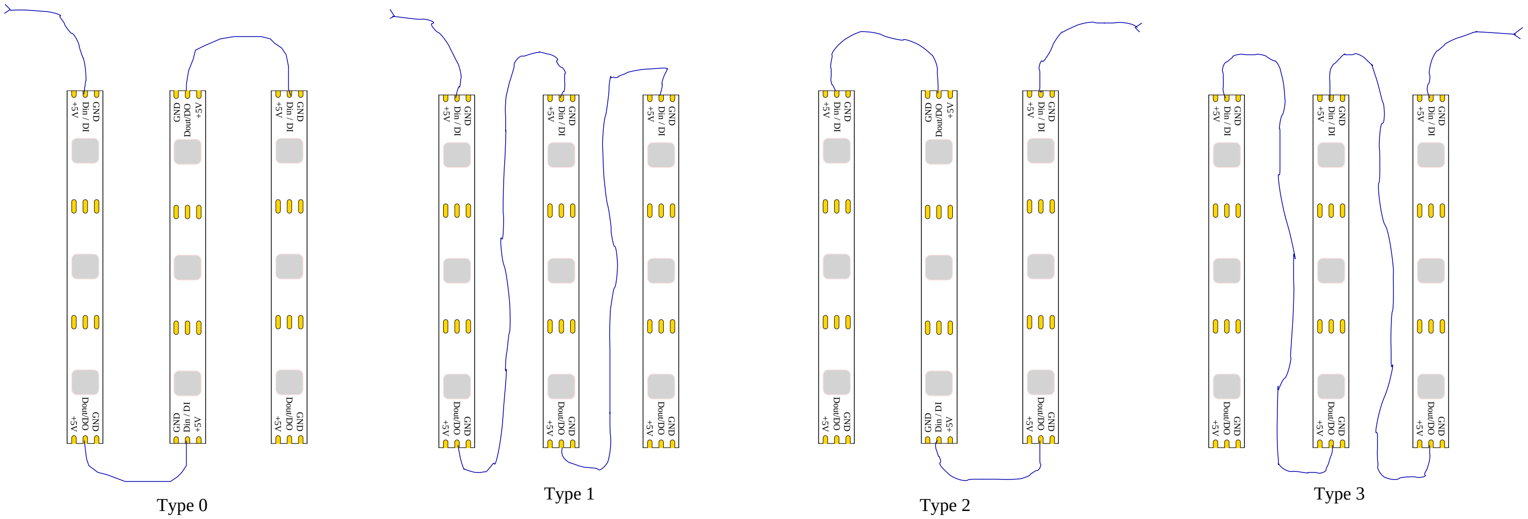
Under https://github.com/Mattscheibe you will find similar libraries for all, AVR controllers, Arduino controllers, and a Raspberry Pi, all with the same API. This will ease migrating a project with LED strands from one platform to another. It would also allow for software development on a RPi (with no turnaround time after program changes) and then one final step of putting your program on a controller to have a standalone, battery powered device. Library names are ws2812b_rpi (for RPi), ws2812b (for AVR controllers) or WS2812b_arduino for AVR-based Arduino compatible boards.

[1] You could drive it (Din connection of strand) either from top left or from bottom left or from top right or from bottom right and you can run the wires either always in the same direction or with changing directions, thus reducing overall wire length.

These are the 16 possible schemes of driving and wiring for matrix shapes of a strand of WS2812B LEDs. Those that are easy to solder have every second stripe run in an opposite directions. They are labeled with equal numbers:

```
Type  0: °|_|‾↓   1: °|/|/↓  2: ↑_|‾|_|° 3: ↓\|\|°  4: °-       5: °-       6: -°       7: -°
                                                         |          /         |            \
                                                        -           -          -           -
                                                       |            /           |          \
                                                        →           →          ←           ←
Type  8: .|‾|_↑   9: .|\|\↑  10: ↑_|‾|.  11: ↑/|/|. 12: →      13: →       14: ←       15: ←
                                                       |           \            |          /
                                                        -          -           -           -
                                                         |         \          |            /
°/.=entry point, |(red)=strand, |=cross connection     .-         .-           -.          -.
```

Here is a drawing of the first four types (power supply omitted in these diagrams):



Type 0 Type 1 Type 2 Type 3

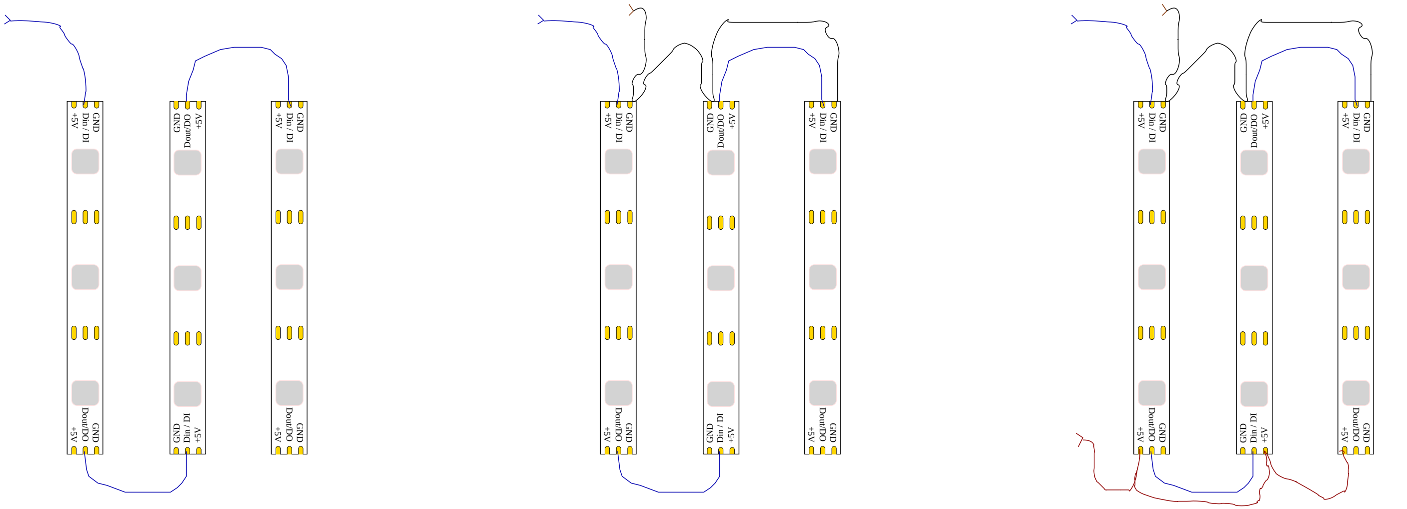
These three steps would probably make your workshop life easier:

* if you need a matrix, glue stripes in alternating directions (flip over every second stripe)

* first solder all signal wires, solder power supply wires at the end

* put all 5V power supply wires on one side of the matrix, all GND power supply wires on the other side of the matrix, in a non-overlapping way

Here are some diagrams:



Step 1: signal wires Step 2a: power supply (GND side) Step 2b: power supply (positive side)

If you glue your stripes in alternating directions, you can put +5V wires on one side and GND wires on the other. They need not touch each other. Still there is space left or wiring every Dout of a stripe to Din of the next strip. That’s why this wiring scheme (alternating directions) is often the most appropriate one.