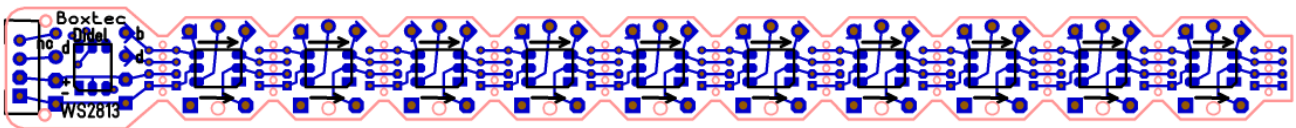


Experiments with WS2013 RGB mini strip

You can buy RGB WS2013 strips from many vendors, load demo programs, try to understand how to modify the demos. For many, the more LEDs, the better. Here for the few, be happy understanding a lib, and learn programming in C.

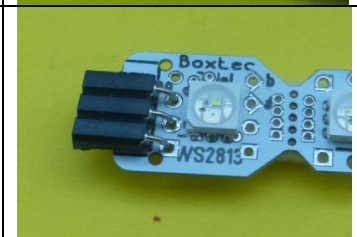
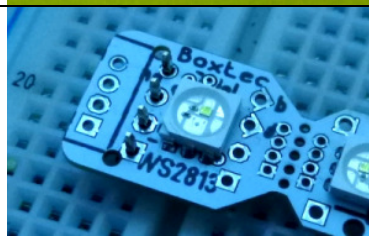
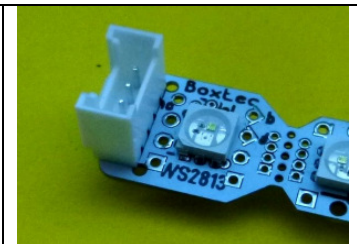
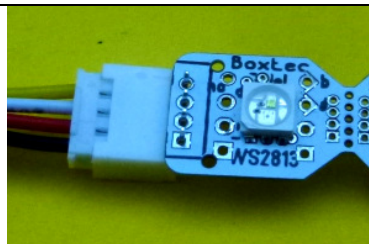
The WS2013 Ministrip has 2 objectives

- 1) Experimenting: You get one PCB with easy 4-pins 2.54mm pitch connectors, breadboard compatible, and one 2mm Grove compatible connector. The strip has 11 leds, enough to test light shows and rainbows.
- 2) Doing special led arrangements and wearable designs.
The 11 LEDs are breakable as you wish. There are two options for the wiring. Since we know sewing is critical, we promote the use of flat cable. Your choice!



Connect the strip

You need three wires toward your card. Gnd, +5V or +3.3V, and an output, let us say pin13 (A0). On the strip, you have choice for a Grove connector, or a 3-pin header (male on bottom if you plan a breadboard, female on top if you will use jumper wires, or direct wires (and have a connector or pins at the other end).



Which driver?

For the WS2813, there are several drivers on the net. The WS2813 is compatible with WS2812 and most if not all of the self-clocking LEDs libraries (not RGBW and sometimes colors are swapped).

Adafruit Neopixel is widely used and documentation is well done:

<https://learn.adafruit.com/neopixel-painter/test-neopixel-strip>

Nice if you buy special LED arrangements. It takes 2k+ bytes of program memory, and 3 variables for every LED.

Didel developed a set of files you may like if you are concerned about code size and download time.

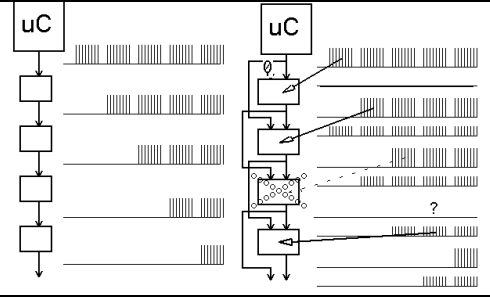
A simple strip of less than 256 LEDs can be controlled adding a library of 200 bytes, 500 bytes with Hue. You can use the small processors of the AVR Tiny family. The files are written in portable C, not using Arduino functions (GCC compiler). See

www.didel.com/WS2813Lib.pdf (github soon).

More on the WS2813

The WS2813 solves partly the reliability problem. An additional input on every LED takes the clock of the previous circuit. If the normal signal is not active when it should, the signal is taken from the previous LED and the LED can light if good. Two consecutive bad LEDs will break the chain.

The first LED receives only one self-clocking signal. The next ones needs 2 lines, in addition to the power lines.



Voltage is 5V, but the strips work well at 3.7V lipo voltage and down to 2.5V (with the blue getting weak).

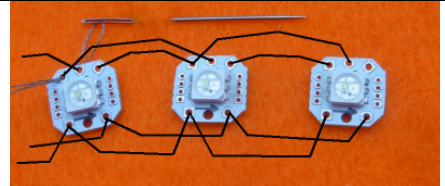
Current is 5 to 80mA for 10 LEDs, depending on light intensity, plus 20-40mA for an Arduino AVR328 board. Maximum intensity is only required outside.

You can fold a little the strips without breaking it. We put the copper on both side: the copper is compressed on one side, it is stretched on the other side and may break. After first bent, add solid glue and do not bend again.



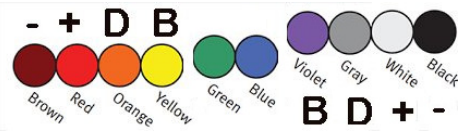
Using flat cable to spread the LEDs

If you break, sand the extremity to smooth the hole marks. The numbers on the back are not really usefull, Leds can be connected in any order. You find on the Web how to sew the leds. We marked the fixing pads at best to help you.



We believe flat cable is the best for all kind of constructions: reliable and easy to solder with a jig.

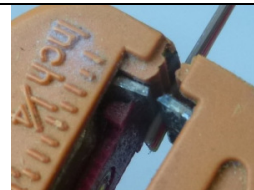
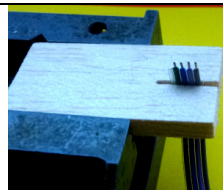
Flat cable is easily available and cheap. Thickness is 1mm and the colors are great. Use only wires 1 to 4 and 5 to 8, and have the ground on wire 1 (brown) or 5 (green). Wire black – white – gray-violet is another option.



Note, on the PCB, the square pad is always the Gnd. The + pad is octagonal

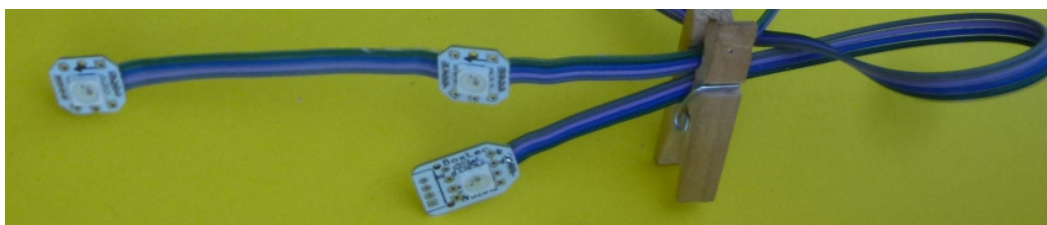
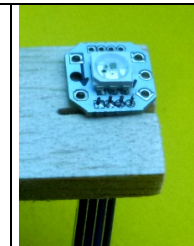
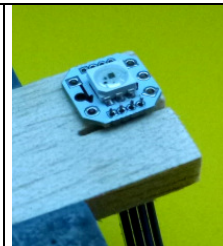
The jig and its use

The jig is a simple piece of balsa wood with a slot of 0.9mm. Flat cable slides in with enough friction to stay in place. Fix it on a vice, or build a support with more than 3cm below.



You need a good wire-stripper and strip a maximum of 2mm. What is crucial is to keep the strand of wires together, dia 0.8mm.

Now you see the trick. Position the flat cable and put the LED on top, well centered. If one brin is folded, you have better to cut and strip again. Solder the 4 wires.



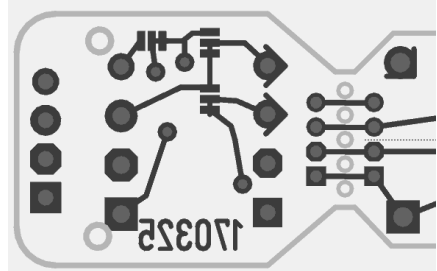
Special utilizations

You are not satisfied with the first LED on the ministrip ? You do not want that first LED active ? The Grove or 2.54mm connector is good, but you need e.g. a rather long flat cable to go to a first LED ?

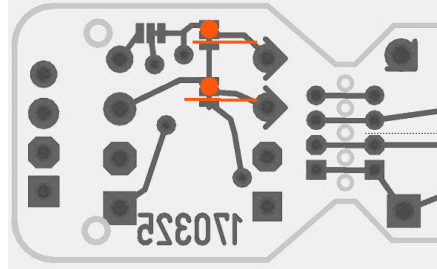
Look at the bottom of the strip beginning there are straps (solder jumpers) and we have planned 3 situations.

By default, the input connector is 3 pins and the first LED is active. Info is transmitted further over the 4 wires.

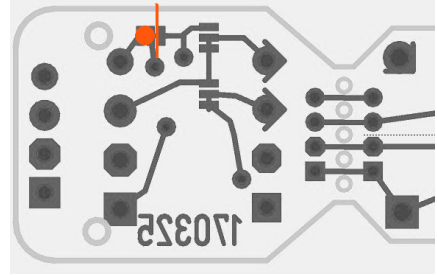
If the strip is "far away", say 50cm or more, the first LED close to the controller has the advantage to shape the signals accordingly to the LED requirements. Microcontrollers signals do not like to have long wires.



The LED can be deactivated by moving two straps. Cut two connections with a scalpel and add two drops of solder.



The strip is the continuations of a previous strip. You can just connect the 4 wires. In this case, wire #4 is ignored, the first LED has the Bin signal connected to Gnd. I will work, but if that LED get dead, the strip will be cut. Move the strap and get the Bin signal from the previous strip.



Note: It is indeed not required to assign a zero on the B input. Any stable signal seems to be accepted.