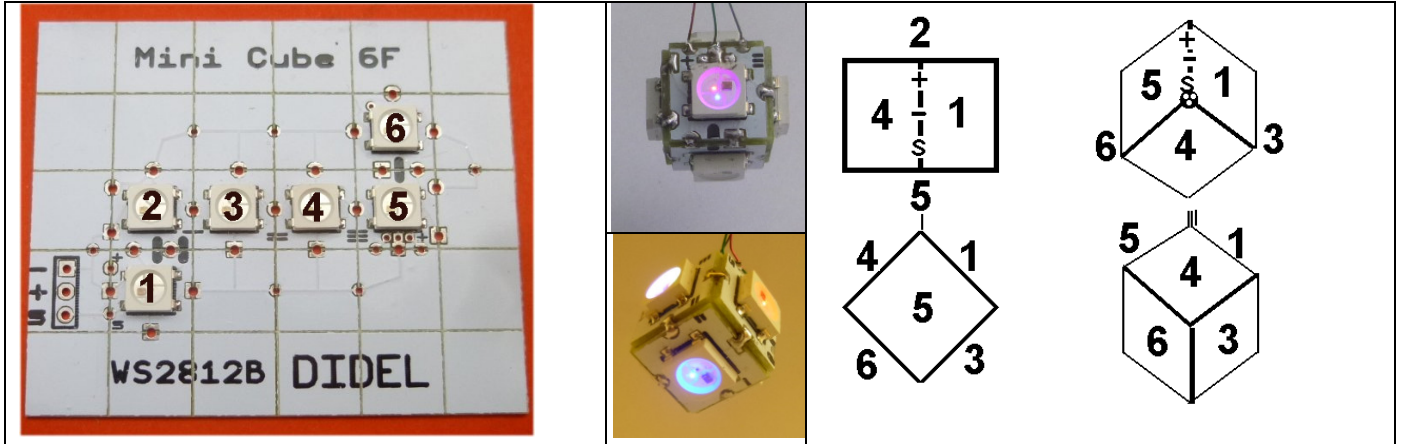


## Mini Cube6 software

Cube6 is a strip of 6 WS2812B RGB smart LEDs. Its fun to get special visual effects, and for this you have to know where LEDs are. Decide first how you will hang your mini cube.



## The WS2812B

You do not really need to understand the WS2812B internal structure and all the interesting points about color coding, RGB or Hue representation, eye logarithmic VS LED linear response. If you are interested, see [www.didel.com/RGBstrips.pdf](http://www.didel.com/RGBstrips.pdf) or search on the Web.

## NeoPixel lib

You can use NeoPixel, there is a lot of info on the Web. Install NeoPixel, select the good options, define the pin and set N\_LEDS to 6, and you are ready to load programs.

## Is a buffer required?

NeoPixel uses a buffer in which you modify LEDs states and then call the `strip.show();` function to transfer all the pixel.

This allows to modify randomly a LED, but this mean that each time you modify a LED, you have to specify its number. Usually, a `for` loop modify all the LEDs value in sequence before calling the `strip.show();`. Random access is seldomly used.

## Didel WS28.h compact lib

We have a different approach with the WS28 lib.

First, you will save a lot of typing and download time.

Second, you will understand the programs you write and be able to add more easily functions.

Third, its a simple start. Noting to install, Simple 1k programs to start with.

A WS2812B LEDs follows a simple rule:

<p>I transfert my 24-bit register to the PWM counters and I reset my internal state when there has been no signal for 300 microseconds (on new WS2812B).                  If I receive bits, I keep for myself the first 24                  Then I forward the next ones.                  Picture shows the inputs of the first four LEDs on a 4-track scope, having send 6x24 pulses to set the colors.                  If you send only 24 pulses, only the first pixel is modified.</p>	
---	--

The transfer functions are easy to write. One need a S8() function that transfers a byte. One call three times that function to build the RGB (r,g,b); function. The circuit need to get the color byte values in the order g – r – b . The RGB function is hence

```
void RGB(byte rr, byte gg, byte bb) {
  S8(gg); S8(rr); S8(bb);
}
```

S8 is a critical "bit-bang" function that cannot be interrupted. Arduino delay(), which uses interrupt, can anyway be used because WS28.h assign the name delay on delMs . delMs () is a blocking function (like Arduino delay) based on a waiting loop.

The functions delMs(), S8() and SetupWS28() are defined in the WS28.h include file.

Four lines depends on the cube wiring:

```
#define bP 0 // bit number in the selected port
#define PORT PORTC // the name of the AVR38 port
#define DDR DDRC // the name of the direction port
#define Npix 6
```

Note that Arduino digitalWrite() is far too slow to be used.

To write the 6 colors of the cube, e.g. red, red, green, green, blue, blue, and test how the intensity can vary, the complete program is

```
// Cube1.ino
#include "WS28.h" // define pin used
void setup () {
  SetupWS28();
}

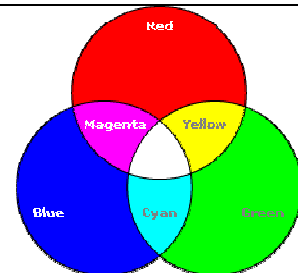
byte br=0; // brightnes for all colors)
void loop () {
  RGB(br,0,0); RGB(br,0,0);
  RGB(0,br,0); RGB(0,br,0);
  RGB(0,0,br); RGB(0,0,br);
  br++;
  delMs (10); // must be grater than 300 us
}
```

The cycle lasts 256x10ms = 2.5s

You notice the LEDs are very quickly bright when the intensity increases. Eye sensitivity is logarithmic. Test the LogRGB () function of **Cube2.ino** (load from [www.didel.com/Cube6.zip](http://www.didel.com/Cube6.zip)). Note there are only 32 intensity steps. Look at [www.didel.com/RGBstrips.pdf](http://www.didel.com/RGBstrips.pdf) for details. **Cube3.ino** compare 2 nextby LEDS.

## Color Names

```
#define red br,0,0
#define green 0,br,0
#define blue 0,0,br
#define yellow br,br,0
#define cyan 0,br,br
#define magenta br,0,br
#define white br,br,br
#define black 0,0,0
```



These definitions are part of the WS28.h file.

It is more easy to set the colors of the cube, in their 1,2,..6 order. For the simple sequence tested on **Cube4.ino** program, the program loop is:

```
RGB(red); RGB(green); RGB(blue); RGB(yellow); RGB(cyan); RGB(magenta); delMs(500);
RGB(green); RGB(blue); RGB(yellow); RGB(cyan); RGB(magenta); RGB(red); delMs(500);
RGB(blue); RGB(yellow); RGB(cyan); RGB(magenta); RGB(red); RGB(green); delMs(500);
RGB(yellow); RGB(cyan); RGB(magenta); RGB(red); RGB(green); RGB(blue); delMs(500);
RGB(cyan); RGB(magenta); RGB(red); RGB(green); RGB(blue); RGB(yellow); delMs(500);
RGB(magenta); RGB(red); RGB(green); RGB(blue); RGB(yellow); RGB(cyan); delMs(500);
```

## Special functions

How do you plan to hang your cube? If from the tip, LEDs 1,4,5 are "together".

To have them of the same color , and the 3 below of a different color, on can write the function

```

#define top cyan
#define bot yellow
void TopBot () {
  RGB(top); RGB(bot); RGB(bot); RGB(top); RGB(top); RGB(bot);
  delMs(500);
}

```

Think how to define the evolution of colors, test and improve. It is both a programming and artist work.

## Hue

Our perception of the color is of "rainbow" type. RGB is related to eye and LED technology. HSB (Hue-Saturation-Brightness) is a way to simplify our coding.

Saturation is a parameter we can set at maximum.

Brightness is the second parameter. First parameter to go through the rainbow colors:

0= red orange yellow green blue indigo violet = 255.

There are several formulas to compute the r g b values from the hue value.

On WS28.h Hue() function is simple, the AVR does not divide efficiently (see again [www.didel.com/RGBstrips.pdf](http://www.didel.com/RGBstrips.pdf)).

Test program is **Cube5.ino**. The loop is

```

byte br = 4;
void loop () {
  for (byte i=0; i<256; i++) {
    Hue(i,16); Hue(i,16); Hue(i,br); Hue(i,br); Hue(i,br); Hue(i,br);
    delay (10);
  }
  delay (1000);
}

```

## Note on WS2812B timing

Early 2017, the manufacturer modified the reset timing of the WS2812B, not changing its name. All the Cube6 are using the new circuit with a 300 us reset time. The previous WS2812B has a 6us reset time, short compared to calculation time between sending 2 pixels. Hue function does not work on old WS2812B. If you use WS28.h on existing strips, test them with the DelayTest.ino program you find on [www.didel.com/Cube6.zip](http://www.didel.com/Cube6.zip). If the 6 first LEDs are always on, it is good.