

# Rapport de séance

## Séance 1 – 13/12/2019

### Travail effectué

Au cours de cette séance, nous avons commencé à construire le cube. Nous avons ainsi commencé une des sept plaques qui le composeront.

Nous sommes partis d'une plaque restante du projet « Kubo » de l'année dernière. Le cube de l'année dernière étant de dimension 8x8x8, nous avons dû démonter entièrement la plaque récupérée, afin de l'adapter à nos dimensions (7x7x7), Après quoi nous avons commencé la construction de notre plaque.

Nous avons commencé par l'armature métallique sur laquelle viendront se placer les LEDs. Il a fallu pour ça réutiliser l'armature du projet de l'année dernière, la déplier, enlever les résidus métalliques restés collés à la suite des soudures avec des LEDs puis la poncer afin d'assurer une bonne conductivité lorsque nous aurions branché les LEDs, et enfin la replier et souder les différentes parties entre elles afin que ses dimensions soient adaptées à notre projet. Je me suis chargé d'enlever les résidus métalliques à l'aide d'un fer à souder, et de replier l'armature aux dimensions désirée.

Une fois l'armature créée, il a fallu commencer à placer et souder les LEDs RGB. Pendant cette phase, nous avons rencontrés plusieurs difficultés. La principale étant que plusieurs des LEDs sont défectueuses, ou ne marchent tout simplement plus. Chaque LED placée sur la plaque doit ainsi être testée et vérifiée avant d'être soudée, afin d'éviter des problèmes. Pour cela, M. Masson nous a fourni une carte Arduino sur laquelle se trouve un programme permettant simplement d'éclairer chaque LED branchée en rouge, puis vert, puis bleu, puis une couleur aléatoire. Nous avons ainsi soudé 10 LEDs à l'armature au cours de cette séance, en alternant la personne responsable de la soudure régulièrement.

### Documents

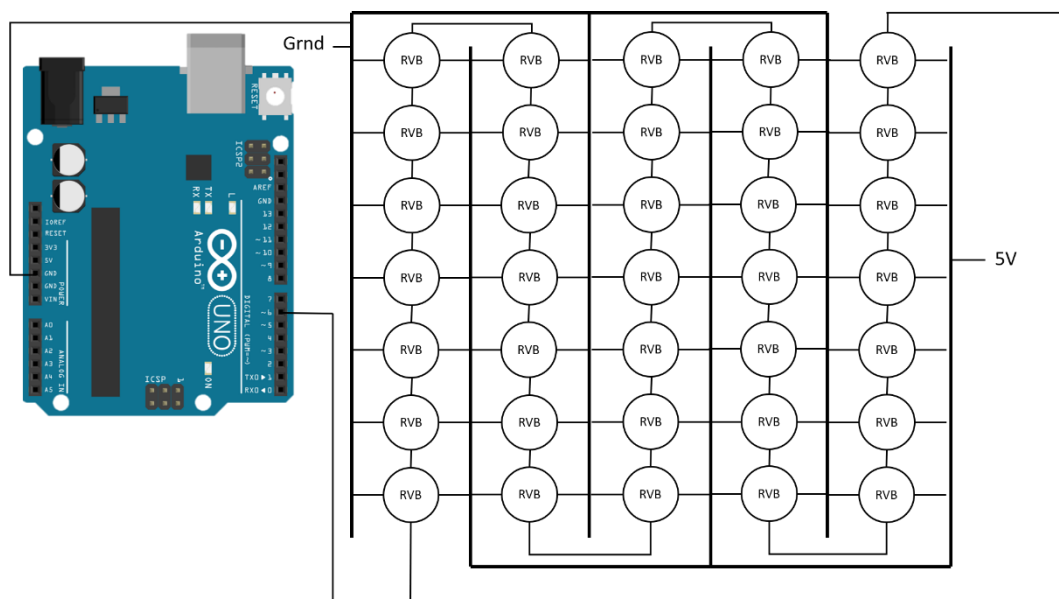
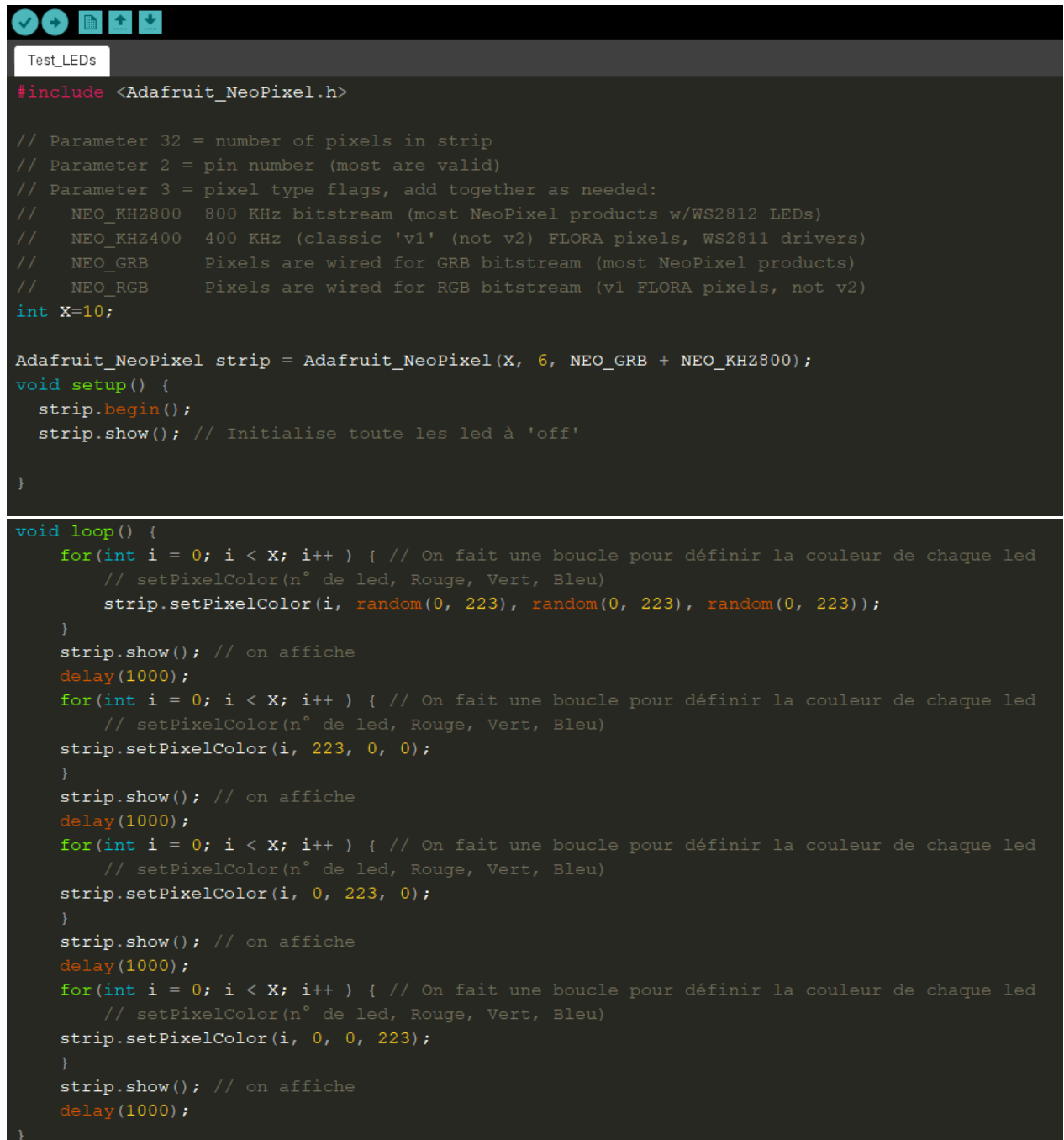


Schéma d'une plaque du cube.



```

Test_LEDs

#include <Adafruit_NeoPixel.h>

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
int X=10;

Adafruit_NeoPixel strip = Adafruit_NeoPixel(X, 6, NEO_GRB + NEO_KHZ800);
void setup() {
  strip.begin();
  strip.show(); // Initialise toute les led à 'off'
}

void loop() {
  for(int i = 0; i < X; i++ ) { // On fait une boucle pour définir la couleur de chaque led
    // setPixelColor(n° de led, Rouge, Vert, Bleu)
    strip.setPixelColor(i, random(0, 223), random(0, 223), random(0, 223));
  }
  strip.show(); // on affiche
  delay(1000);
  for(int i = 0; i < X; i++ ) { // On fait une boucle pour définir la couleur de chaque led
    // setPixelColor(n° de led, Rouge, Vert, Bleu)
    strip.setPixelColor(i, 223, 0, 0);
  }
  strip.show(); // on affiche
  delay(1000);
  for(int i = 0; i < X; i++ ) { // On fait une boucle pour définir la couleur de chaque led
    // setPixelColor(n° de led, Rouge, Vert, Bleu)
    strip.setPixelColor(i, 0, 223, 0);
  }
  strip.show(); // on affiche
  delay(1000);
  for(int i = 0; i < X; i++ ) { // On fait une boucle pour définir la couleur de chaque led
    // setPixelColor(n° de led, Rouge, Vert, Bleu)
    strip.setPixelColor(i, 0, 0, 223);
  }
  strip.show(); // on affiche
  delay(1000);
}

```

Programme permettant de tester le fonctionnement des LEDs