

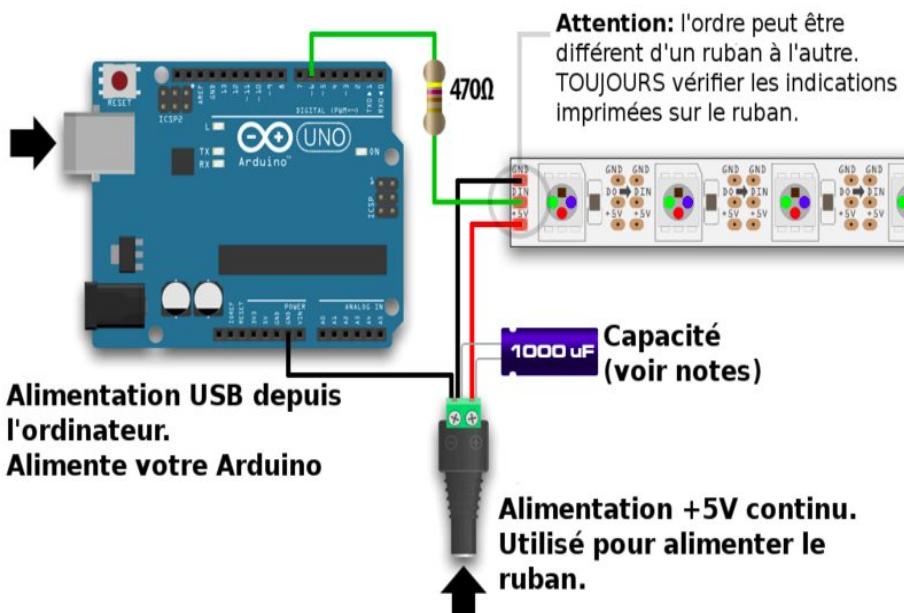
Compte rendu de la soirée du lundi 5 décembre 2016 à L'atelier MJC-Fablab de Château-Renault

Nous étions 6 à nous retrouver à 20h autour des tables de la MJC.

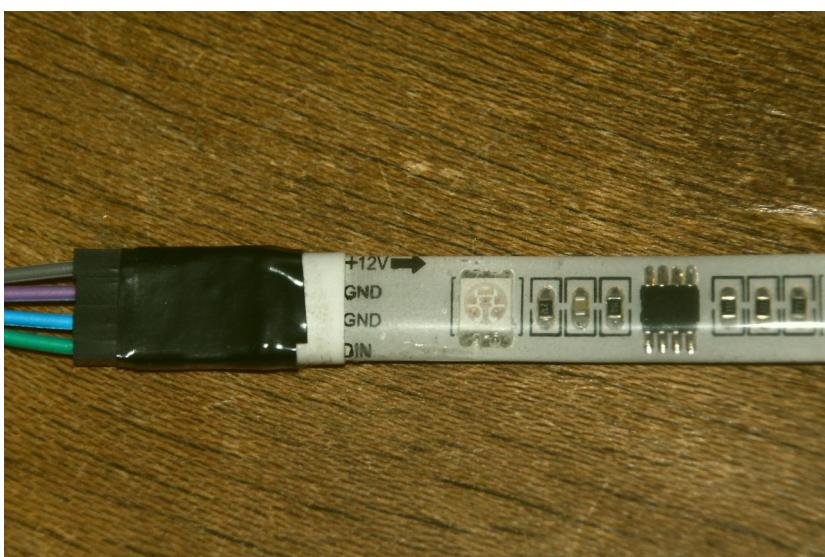
Michel a repris les côtes du plateau de protection de l'imprimante 3D pour le transport.
Quand la pièce sera opérationnelle, on mettra les plans en ligne.

j'ai présenté un montage pour la fin d'année consistant à commander un ruban de leds RGB WS2812 par un Arduino via une librairie Neopixel de Adafruit

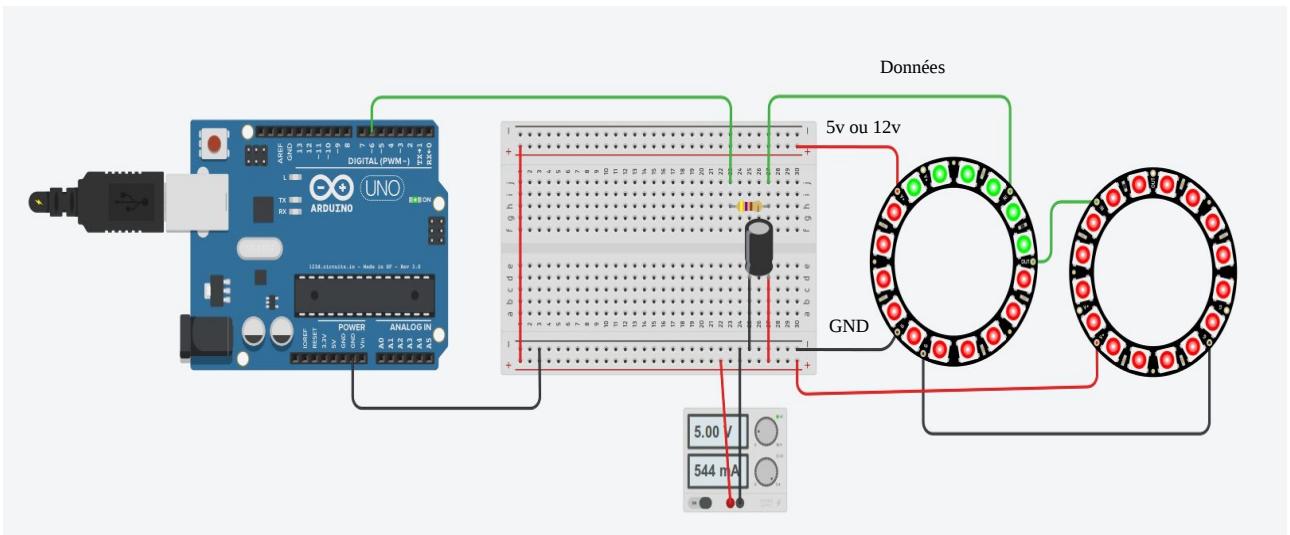
Montage théorique :



mon montage :



raccordement des 4 fils :
+12v ou Vcc
2x GND ou masse
Din pour envoi des données de commandes des leds en série



Un exemple qui permet de tester le circuit

```
#include <Adafruit_NeoPixel.h>
#define PIN      6
#define NUMPIXELS 32

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 500; // delay for half a second

void setup() {
  pixels.begin(); // This initializes the NeoPixel library.
}

void loop() {
  for(int i=0;i<NUMPIXELS;i++){
    pixels.setPixelColor(i, pixels.Color(0,150,0)); // Moderately bright green color.
    pixels.show(); // This sends the updated pixel color to the hardware.
    delay(delayval); // Delay for a period of time (in milliseconds).
  }
}
```

Pour que cela fonctionne , on a besoin :

- 1 Arduino uno ou mega ou nano ...
- 1 ruban de leds RGB
- 1 alimentation externe de 12v (5V pour le schéma) 3A
- 1 breadboard (pour le branchement lors des tests)
- Une série de fils pour les connexions

La dernière version de L'[IDE Arduino](#) (1.6.13 au 5/12/2016)

Contrôler des NéoPixels à "partir de rien" est un sacré challenge. AdaFruit propose une bibliothèque / librairie permettant avant tout de prendre du plaisir et de se concentrer sur les points intéressants

Installer la bibliothèque Neopixel dans l'IDE Arduino

Croquis → Inclure une bibliothèque → gérer les bibliothèques → taper « neopixel » dans le filtre de recherche → cliquer sur « More info » dans la case d'Adafruit Neopixel → cliquer sur « Installer »

Dans « Fichier » → « Exemples » → « Adafruit Neopixel » (dans le bas) → « Simple »

Vous devriez avoir le programme de test « simple » pour vérifier si votre ruban s'allume

Commentaires sur le programme « simple » :

```
#include <Adafruit_NeoPixel.h>
```

On inclus la bibliothèque Adafruit_Neopixel.h

```
#define PIN 6
```

on déclare la broche 6 pour l'envoie des données sur le ruban. On pourrait déclarer une autre broche
On pourrait même commander plusieurs rubans à partir du même Arduino.

```
#define NUMPIXELS 32
```

On déclare le nombre de NéoPixel chainés dans le ruban/strip. Dans cet exemple il est fixé à 32,
Modifiez ce paramètre pour qu'il corresponde au nombre de pixels que vous
utilisez (32 correspond au nombre de neopixel du test)

```
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
```

On déclare un objet « Neopixel » qui s'appelle « pixel »

Paramètre 1 (NUMPIXELS) = Le nombre de NéoPixels chainés

Paramètre 2 (PIN) = No de broche de données (Sur votre Arduino, la plupart convient)

Paramètre 3 = Type de pixel (flags/drapeaux), a combiner ensemble en fonction du besoin:

NEO_KHZ800 flux de données à 800 KHz (plupart des NéoPixel basé sur les LEDs w/WS2812)

NEO_KHZ400 flux de données à 400 KHz (Pour les Pixels classiques 'v1' FLORA (pas les V2) pilotés par WS2811)

NEO_GRB Pixels sont raccordés en flux de donnée GRB (GRB=Green,Red,Blue=Vert,Rouge,Bleu - la plupart des produits

NEO_RGB Pixels sont raccordés en flux de donnée RGB (RGB=Red,Green,Blue=Rouge,Vert,Bleu – Pixel

```
pixels.begin();
```

On initialise la librairie Neopixel

```
void loop() {
    for(int i=0;i<NUMPIXELS;i++){
        pixels.setPixelColor(i, pixels.Color(0,150,0)); // Moderately bright green color.
        pixels.show(); // This sends the updated pixel color to the hardware.
        delay(delayval); // Delay for a period of time (in milliseconds).
    }
}
```

on initialise une boucle for qui balaie tous les neopixel de 0 à 31 (32)

```
for(int i=0;i<NUMPIXELS;i++){
```

On contrôle la valeur d'un Neopixel

```
pixels.setPixelColor(i, pixels.Color(0,150,0));
```

avec i :

Le premier argument — i dans cet exemple — est le numéro du pixel le long du ruban, commençant à 0 pour le plus proche d'Arduino. Si vous avez un ruban/chaine de 32 pixels, ils sont numérotés de 0 à 31 (comme c'est couramment le cas en informatique). De nombreux emplacement du code utilise une boucle for , utilisant la variable compteur de la boucle comme argument identifiant le n° de pixel... cela permet de donner une valeur à plusieurs pixels

(0,150,0)

les 3 arguments suivants définissent la couleur du pixel en indiquant le niveau de luminosité en rouge, vert et bleu.
red : Niveau de luminosité rouge , 0 est la plus tenue (éteint) et 255 la luminosité maximale.
green: Niveau de luminosité vert , 0 est la plus tenue (éteint) et 255 la luminosité maximale.
blue : Niveau de luminosité bleu , 0 est la plus tenue (éteint) et 255 la luminosité maximale

`pixels.show();`

La fonction `show()` fait une mise-à-jour de tout le ruban/chaine en une seule fois. Même si cette étape supplémentaire peu paraître ennuyeuse, c'est en fait une bonne chose. Si chaque appel à `setPixelColor()` avait un effet immédiat, votre animation présenterait des sursauts plutôt que faire des mise-à-jours du ruban de façon homogène

Recommandations :

Lorsque vous connectez des NéoPixels sur n'importe quel source d'alimentation active ou microcontrôleur, TOUJOURS CONNECTER LA MASSE/GND (-) AVANT QUOIQUES CE SOIT D'AUTRE. Et a l'inverse, toujours déconnecter la masse/GND en dernier

Démarrez votre Arduino IDE. Sélectionnez ensuite le programme de démonstration Strandtest via le menu Fichier > Exemple > Adafruit_NeoPixel > strandtest

tableau des couleurs des leds RGB

[ICI](#)

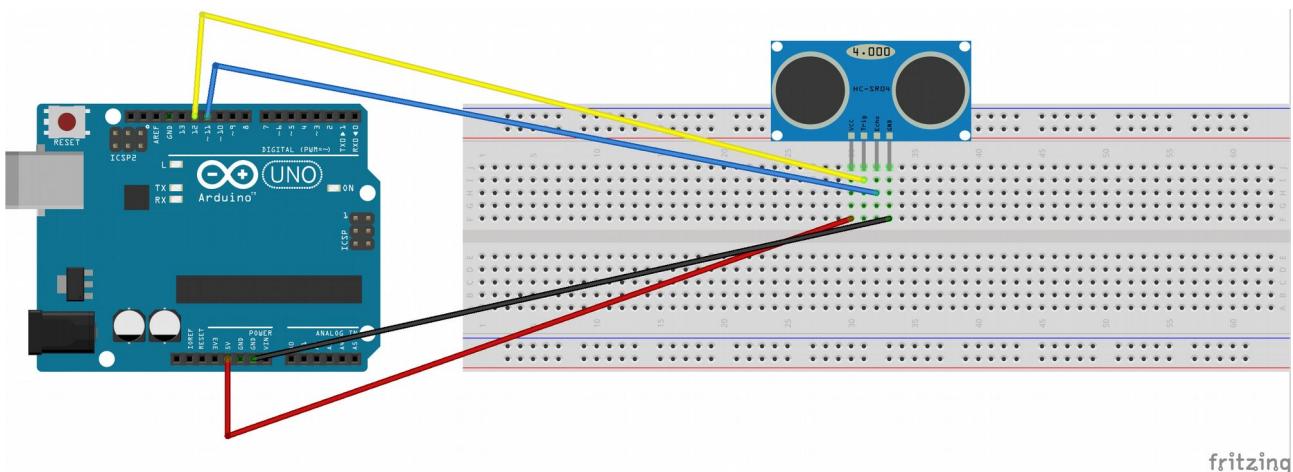
il n'y a plus qu'à personnaliser les couleurs des LEDS pour le sapin

[**Aymeric** a présenté son montage à base d'Arduino et de capteur de distance](#)

Materiel :

1 Arduino Uno
1 breadboard
1 capteur de distance
Des câbles de connexions





Câblage Capteur --> Arduino

- pin 1 Alimentation Vcc --> +5V Arduino
- pin 2 Emission d'ultrasons (Trig) --> pin de sortie **p12**
- pin 3 Réception du signal Echo (signal de sortie TTL en largeur d'impulsion modulée)--> pin d'entrée **p11**
- pin 4 Gnd --> Masse Arduino

```
#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN    11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters).
Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and
maximum distance.

int tmpls_arret = 600;
int tmpls_arret_total;
int tmpls_arret_mini = 10000;
int moyenne;
int nb_echos = 0;

void setup() {
  Serial.begin(9600);
  Serial.println("So... Let's begin !");
  delay(1000);
}

void loop() {
  delay(tmpls_arret);    // Wait 50ms between pings (about 20 pings/sec). 29ms should be the
shortest delay between pings.

  tmpls_arret_total = tmpls_arret_total + tmpls_arret;
}
```

```

Serial.println('tmps_arret_total');

unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
Serial.print("Distance : ");
Serial.print(sonar.convert_cm(uS)); // Convert ping time to distance and print result (0 = outside
set distance range, no ping echo)
Serial.println(" cm");

if(sonar.convert_cm(uS)>1 && tmps_arret_total > tmps_arret_mini)
;
//if(sonar.convert_cm(uS)<50) Serial.println("      ALERTE !!"); //else Serial.println("prout.");
}

}

```

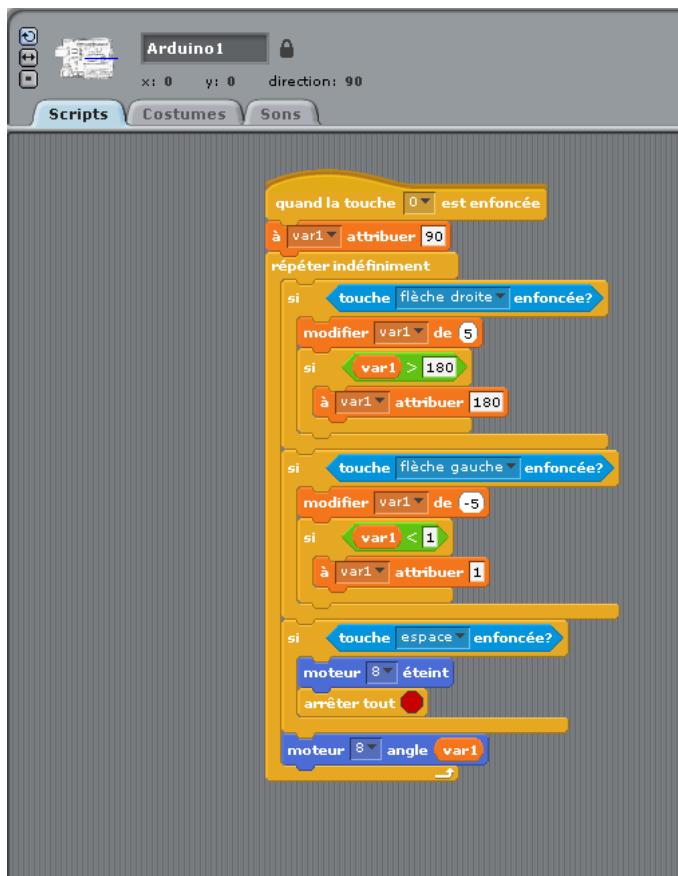
utilisation de la librairie <NewPing.h> [ICI](#)

Un autre exemple [ICI](#)

Ce programme allume une LED RGB cathodes communes en fonction de la distance mesurée

Fernand nous a présenté ce que l'on fait à l'atelier jeunes en programmation du bras robot à l'aide de scratch en lien avec Arduino : S4A et ensuite son interface web domotique sur Arduino méga

Pour la démo S4A voir le compte rendu sur Owncloud du 26 11 2016 [ICI](#)



Des limites apparaissent avec S4A :

- On ne peut que piloter 3 Servo-moteur,
- Le code ne peut pas être transposé en code Arduino
- On est obligé d'avoir un programme dans l' Arduino (S4AFirmware16.ino) pour communiquer entre S4A et Arduino

Les avantages :

- Facilité de prise en main
- Programmation graphique pour ceux que la ligne de commande rebuté
- Très utilisé dans le monde de l'éducation

Pour la démo domotique :



Ce n'est pas exactement la page que Fernand à présenté mais cela y ressemble

Le programme [ICI](#)

On inclue 3 librairies

```
#include <Ethernet.h>
#include <SPI.h>
#include <EEPROM.h>
```

On paramètre le réseau en manuel, ici pour une box free

```
//IP manual settings
byte ip[] = {
    192,168,0,221 }; //Manual setup only
byte gateway[] = {
    192,168,0,1 }; //Manual setup only
byte subnet[] = {
    255, 255, 255, 0 }; //Manual setup only

// if need to change the MAC address (Very Rare)
byte mac[] = {
    0x90, 0xa2, 0xda, 0x0f, 0x27, 0x4c };

//Ethernet Port
EthernetServer server = EthernetServer(80); //default html port 80
```

Dans la fonction : **void printHtmlHeader(EthernetClient client)**

On insère le code html et css pour l'affichage de la page sur le client (navigateur web) le serveur étant l' Arduino dans ce cas

Pour détailler le code complet il nous faudrait plus d'une séance.

Je pense que l'on reviendra plus tard sur une application domotique soit avec Arduino soit avec Raspberry ou les deux ensemble

Pour lundi prochain , on va faire l'inventaire des kits Arduino , pour recenser le matériel disponible afin de pouvoir les utiliser lors des prochaines séances ..

Les programmes .ino sont placés dans le répertoire Owncloud de la séance du 05 12 2016

Infos :

Je (Gérard) serais absent les :

19/12/2016

26/12/2016

02/01/2017

Si vous êtes **plusieurs** à vouloir venir , nous sommes 3 à avoir la clef , il suffit de demander
Gerard

Henri

Michel

Je rappel que l'imprimante 3D peut être empruntée par un membre de l'atelier pour la tester chez lui , on pèsera la bobine de fil en début et en fin d'utilisation , si vous n'avez pas de bobine de fil à vous

Lien pour emprunter l'imprimante 3D [ICI](#)

<https://framaforms.org/emprunt-imprimante-3d-vertex-k8400-1476175797>

Je viendrai peut être un jour en semaine 51 ou 52 pour paramétrer Owncloud sur la box de la MJC

A lundi soir le 12 12 2016 à 20h

Gérard