



Association ORE

Ouverture Rencontres Evolution · FabLab 21

ARDUINO

Lampe connectée

Fiche tutoriel · 12 pages

Contents

1	Travail pratique	3
1.1	Exercice introduction Arduino	3
1.2	Introduction à la carte esp 8266	3
1.3	Introduction à la découpe LASER et dernière face de la lampe	5
1.4	Introduction à l'impression 3D et support de la lampe	7
1.5	Montage de la lampe.	8
1.5.1	Liste des pièces nécessaires	8
1.5.2	Instruction de montage	8
2	Informations utiles	10
2.1	Installation librairies de carte	10
2.2	Installation librairies .h	11
2.3	Guide de survie fonction FastLED.h	11
2.3.1	La fonction FastLED.show()	11
2.3.2	fill_solid()	11
2.3.3	Coloré une LED spécifique donnée.	12
2.3.4	Différentes manière de donner une couleurs	12

1 Travail pratique

Arduino à récemment eu une mise à jour, vous pouvez la faire si elle vous a proposé à la 1ère connexion si elle n'est pas encore faite sur votre ordinateur.

1.1 Exercice introduction Arduino

Cet exercice permet simplement de comprendre comment fonctionne Arduino avec une LED simple si vous n'avez jamais fait d'Arduino jusqu'à aujourd'hui.

Arduino permet de connecter des objets à une différente carte utilisant le même logiciel. Ici nous utiliserons des esp8266.

Afin de comprendre comment fonctionne une carte on va d'abord la connecter en USB à l'ordinateur et on va connecter une résistance et une LED simple à cette carte.

On connecte un côté de la LED à la borne GND, et du côté de la résistance à la borne 3V afin de faire fonctionner le tout. Normalement la LED devrait s'allumer car la borne 3V fonctionne comme une pile qui envoie en permanence du courant.

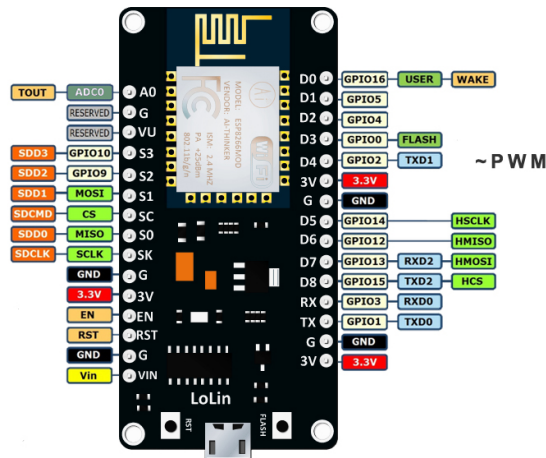
On va maintenant connecter le fil qui était sur 3V à la prise D0 et recopier le petit code suivant.

```
1  #define ledPin D0
2  void setup() {
3      pinMode(ledPin,OUTPUT);
4  }
5
6  void loop() {
7      digitalWrite(ledPin,HIGH);
8      delay(500);
9      digitalWrite(ledPin,LOW);
10     delay(500);
11 }
12 |
```

Ce code permet de faire clignoter la LED avec la fonction digitalWrite() qui impose ce que l'on envoie comme courant sur le pin en question afin d'allumer ou éteindre la LED.

1.2 Introduction à la carte esp 8266

Si les noms des connexions de votre carte ne sont pas reconnues par le logiciel d'Arduino, vous pouvez utiliser les numéros après GPIO pour la connexion qui correspond sur l'image suivante . Si il n'y a pas de numéros GPIO correspondant alors connexion:



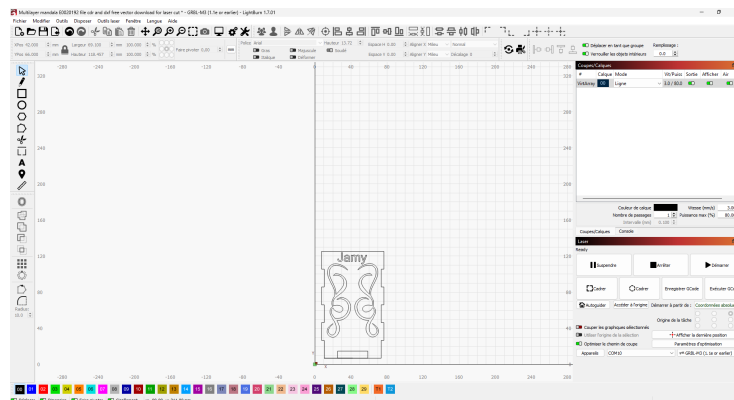
1. Verifier que la carte "Node MCU 1.0 (ESP-12E Module)" est bien disponible dans la liste de carte. Si ce n'est pas le cas alors suivre les instructions dans le "2.1.Installer librairies de carte" vers la fin du document.
2. Une fois cela fait ajouter la librairie "FastLED.h" dans le gestionnaire de librairie Si vous ne savez pas comment faire alors regarder la partie "2.2. Ajouter des librairies" en fin de document.
3. Copier sur votre ordinateur et ouvrir le document "Ruban_exemple" dans votre logiciel Arduino.
4. On va maintenant connecter les rubans de LED au cartes en prenant soin de bien connecter les liens fils au bonne prise du ruban. (5V sur VIN, GND sur G, et D0 sur D0).
5. Utiliser la liste de fonctions de base "2.3 Guide de survie: fonction de FastLED", ainsi que la base de code fournie, afin de réaliser les fonction suivantes:
 - (a) Faire que le ruban soit entièrement bleu.
 - (b) Faire que le ruban clignote de bleu à vert sur toute la longueur du ruban.
 - (c) Pur que l'on est les 3 1ères LED qui s'allume de 3 couleurs différentes. et le reste du ruban éteins.
 - (d) Modifier ce scénario afin que ces 3 couleurs se déplacent sur le ruban avec un déplacement toute les secondes.

- (e) (question bonus) Modifier votre précédent en utilisant la fonction modulo(utilisable avec le symbole%), que le déplacement des couleur soit sans interruption et donc qu'il réapparaisse en début de ruban une fois qu'ils sont arrivé au bout.
- (f) (question bonus 2) Enfin refaire ce scénario afin , cette fois , que les 3 couleur "rebondisse" sur les bords du ruban une fois arriver à la fin et reparte dans le sens inverse.

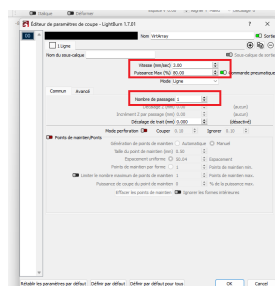
1.3 Introduction à la découpe LASER et dernière face de la lampe

la dernière face de la lampe devrais avoir la même forme que les autres, mais on va vous permettre de la personnalisé en y ajoutant votre nom.

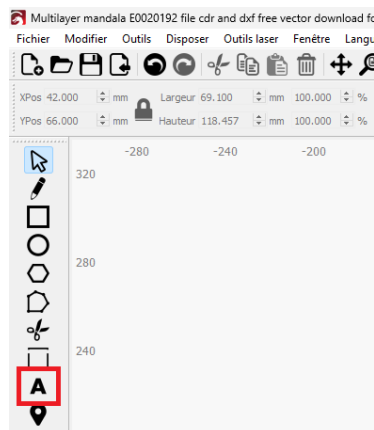
Dans le dossier stage sur votre bureau, doit se trouver un fichier "FACES LAMPES.dxf". Vous pouvez l'ouvrir dans le logiciel lightburn, vous devriez avoir l'écran suivant d'afficher.



Appuyer sur le bouton Vit/Puiss (puissance et vitesse) et vérifier bien que la vitesse est à 3.00, la puissance à 80.00 et le nombre de passage à 1.

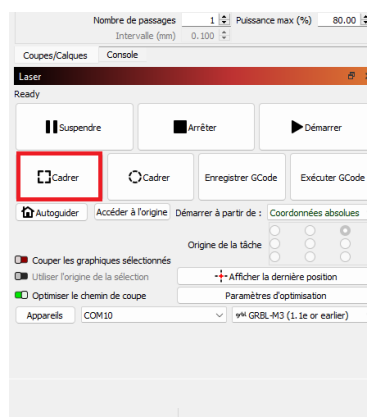


Pour personnalisé votre dernière face avec votre prénom, vous aller cliquer sur l'icone en forme de A à gauche de l'écran puis n'importe où dans le grillage après ça. Vous pouvez alors écrire votre prénom.



Vous pouvez alors régler la taille, et la position du texte afin qu'il tienne sur le haut de la face.

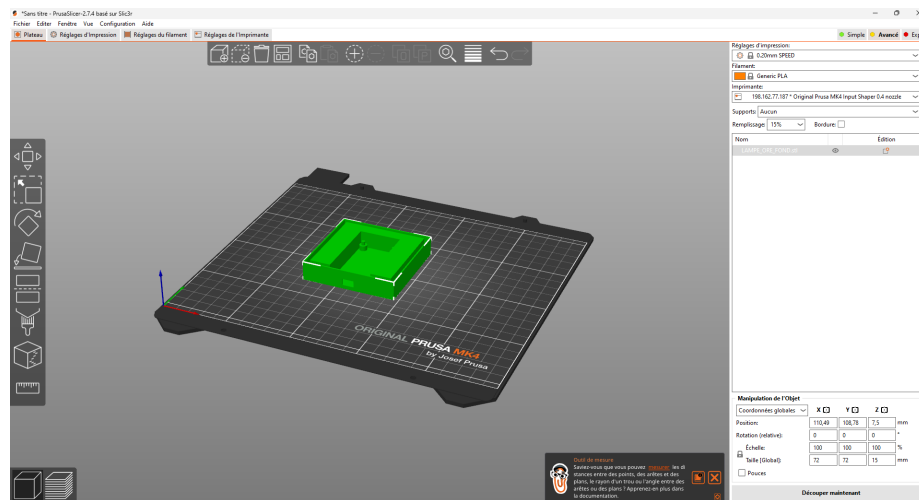
Vous aller alors appuyer sur le cadrer avec l'icône rectangulaire. Et regarder à l'intérieur de la découpe LASER vous allez voir le laser (éteint) faire le tour de la zone de découpe, si sont mouvement ne correspond pas il faut changer les paramètre de lightburn.



Une fois que l'on a bien faire le modèle à découper, bien vérifier les paramètre et que la découpe LASER coupe bien sur la bonne zone, on peut aller appuyer sur démarrer.

1.4 Introduction à l'impression 3D et support de la lampe

Vous aller dans Fichiers Scannés et chercher le fichier STL LAMPE_ORE_FOND.gcode (ou LAMPE_ORE_COUVERCLE.gcode). Et ouvrez ces document avec le logiciel PrusaSlicer. Une image de ce type devrait apparaître.

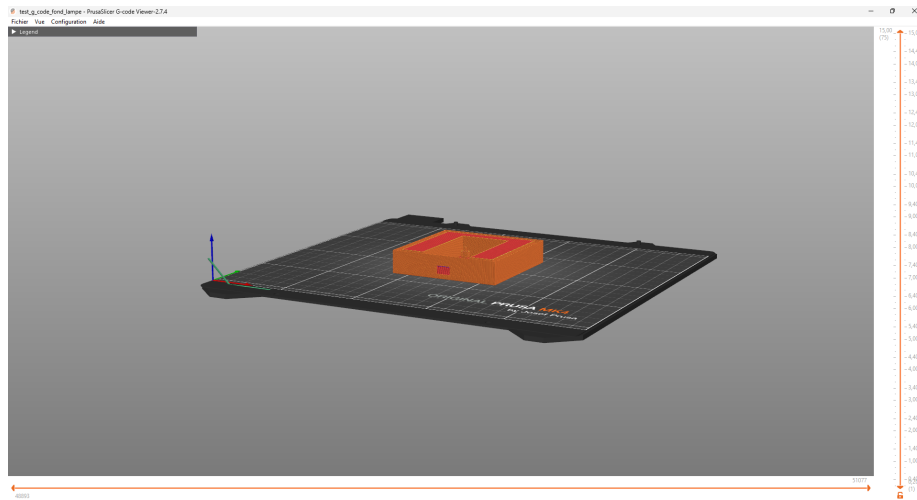


Pour vous déplacer dans le logiciel en 3D vous pouvez utiliser les clic gauche (rotation) et droit (translation) et la molette (niveau de zoom)

On peut alors modifier 3 types de paramètres, ceux de l'impression, ceux du filaments et ceux de l'imprimante en haut. On peut y modifier des chose comme l'épaisseur du fil, le type de fil, l'épaisseur de remplissage etc.. On va ici laisser les paramètres par défaut.

On peut alors slicer ("découper") le modèle pour cela vous aller appuyer sur "découper maintenant" en bas à droite. Renommer alors le fichier avec un nom que vous pourrez retrouver et enregistrer à un endroit que vous pouvez retrouver.

Vous pouvez maintenant ouvrir le fichier créer à l'étape précédente dans Prusa G-code Viewer,



1. En zoomant sur le modèle de quoi semble-t-il être formé?
2. Pourquoi est ce que cette structure est utile pour de l'impression 3D.

On peut alors se connecter à l'imprimante 3D via son adresse IP, et envoyer le g code dessus qu'on peut ensuite demander à imprimer à n'importe quel moment.

1.5 Montage de la lampe.

1.5.1 Liste des pièces nécessaires

Afin de monter la lampe finale il faut :

- 4 planche de bois découper (dont 1 avec son prénom)
- 1 fond de lampe imprimer en 3D.
- 1 couvercle de lampe imprimer en 3D.
- Une carte Arduino à laquelle est soudée le ruban de LED.
- 1 tube imprimer en 3D

1.5.2 Instruction de montage

1. On colle le tube sur la carte (si c'est pas déjà fais)
2. On met la carte , face vers le bas dans le fond de la lampe en faisant attention à ne pas arracher la prise.

3. On place les planche en bois dans la rainure qui fais le tour du fond de la lampe. En commençant de préférence avec les planche qui sont les plus large en bas afin de simplifier l'ajout des autres planche par la suite.
4. On rajoute le couvercle.

On va maintenant ajouter le code permettant de faire fonctionner la lampe correctement.

Dans les Fichiers Scannés prenez le document "vf" , et ouvrir le fichier Arduino qui se trouve à l'intérieur.

Pour que vous puissiez retrouver facilement la connexion de votre lampe assurez vous que vous modifier bien cette ligne avec votre nom et prénom. Et mettez un mot de passe pour éviter de quelqu'un change la couleur à votre place

```

// Set these to your desired credentials. */
const char *ssid = "lampe-ORE-Nom-Prénom";
const char *password = "";
int r, g, b;

```

Vous pouvez alors envoyer le code sur votre carte , une fois cela fais connecter vous au wifi qui porte votre nom, et déconnectez-vous de toute autre connexion pour éviter tout conflit dans votre téléphone.

Aller dans le navigateur et tenter de vous connecter à l'adresse suivante : 192.168.1.1 . Normalement la page suivante devrais apparaître:



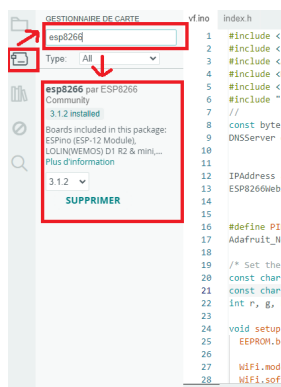
Vous pouvez alors changer la couleur de votre lampe comme vous le souhaitez.

2 Informations utiles

2.1 Installation librairies de carte

Si la librairie de carte esp8266 n'apparaît pas dans votre gestionnaire de carte (2ème icône en partant du haut dans la barre à droite) , il faut alors l'installer manuellement.

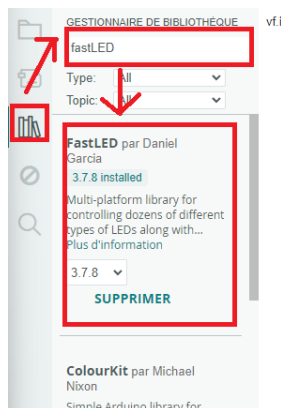
Il faut aller dans fichiers puis préférences. En bas de la fenêtre qui s'ouvre on devrais avoir une barre appelée : "URL de gestionnaire de cartes supplémentaires". Il faut y copier le lien suivant: http://arduino.esp8266.com/stable/package_esp8266com_index.json



Patiencez un peu, et normalement si vous recherchez de nouveau les esp8266 dans le gestionnaire de cartes la bonne librairie devrait apparaître.

2.2 Installation librairies .h

Dans le gestionnaire de librairies (3ème icône en partant du haut dans la barre à droite). Vous pouvez chercher le nom de la librairies qui vous intéresse, ici on cherche FastLED et celle utile au code est la 1ère à ressortir (version 3.7.8 par Daniel Garcia)



2.3 Guide de survie fonction FastLED.h

Dans ce paragraphe on va présenter les principales fonctionnalités de la bibliothèque fastLED et comme les utiliser de façon simple.

2.3.1 La fonction FastLED.show()

On commence par la fonction la plus importante de toute la bibliothèque, en effet cette dernière permet de dire au LED d'afficher tout les changements données par le code, depuis la dernière apparition de la fonction.

Il est donc très important de l'utiliser car sans cela rien ne s'affichera jamais au niveau des LEDs.

2.3.2 fill_solid()

Cette fonction permet à tout un ruban de se mettre d'une seule et unique couleur.

Cette dernière besoin de 3 informations pour fonctionner, quel ruban on doit allumer, le nombre de LED du ruban et la couleur que l'on doit afficher.

Ci dessous quelques exemples de son utilisation.

```
fill_solid(leds, NUM_LEDS, CRGB::Purple);
fill_solid(leds2, NUM_LEDS, CRGB::Orange);
```

Ici on donne la couleurs via son nom en anglais dans les couleurs disponible via CRGB, cependant il existe d'autres manière de donner couleurs à un élément RGB (voir 2.3.4 Différentes façon de donner une couleurs à la fin du document)

2.3.3 Coloré une LED spécifique donnée.

On peut aussi donner la couleur que l'on souhaite à la LED que l'on veut grâce à cette librairie qui permet donc de n'allumer que certains LED spécifique en donnant leur adresse directement, cette adresse correspondant à un nombre.

```
leds[0]=CRGB::Coral;
leds[3]=CRGB::LimeGreen;
leds2[9]=CRGB::Red;
leds2[2]=CRGB::Yellow;
FastLED.show();
```

Attention les adresse des LED commence à 0 et pas à 1, donc leds[0] correspond à la 1ère LED du ruban, ou leds2[9] correspond à la 10ème LED du ruban par exemple

2.3.4 Différentes manière de donner une couleurs

1. Donner la couleur avec CRGB

Dans ce cas de figure, la syntaxe à utilisé dans le code a été donné dans les exemple précédent. On peut trouver une liste des couleurs CRGB à l'adresse suivante en bas de page si besoin de chercher une couleur particulière:

<https://github.com/FastLED/FastLED/wiki/Pixel-reference>

2. Donner la couleur avec 3 nombres

On peut aussi ajouter une couleurs en entrant manuellement les valeurs pour le rouge, le bleu et le vert, ce qui revient à la méthode précédente CRGB sans être limité sur les options. On peut faire varier la valeur de chaque couleur de 0 à 255 soit 256 possibilité. On a alors $256 \times 256 \times 256 = 16777216$ couleurs possible (même si la plupart sont inutiles car trop proche les unes des autres pour qu'on fasse la différentes)

```
leds[8].setRGB(50,100,152);  
leds2[3].setRGB(0,0,255);
```

On devra alors écrire les lignes de la façon suivante: