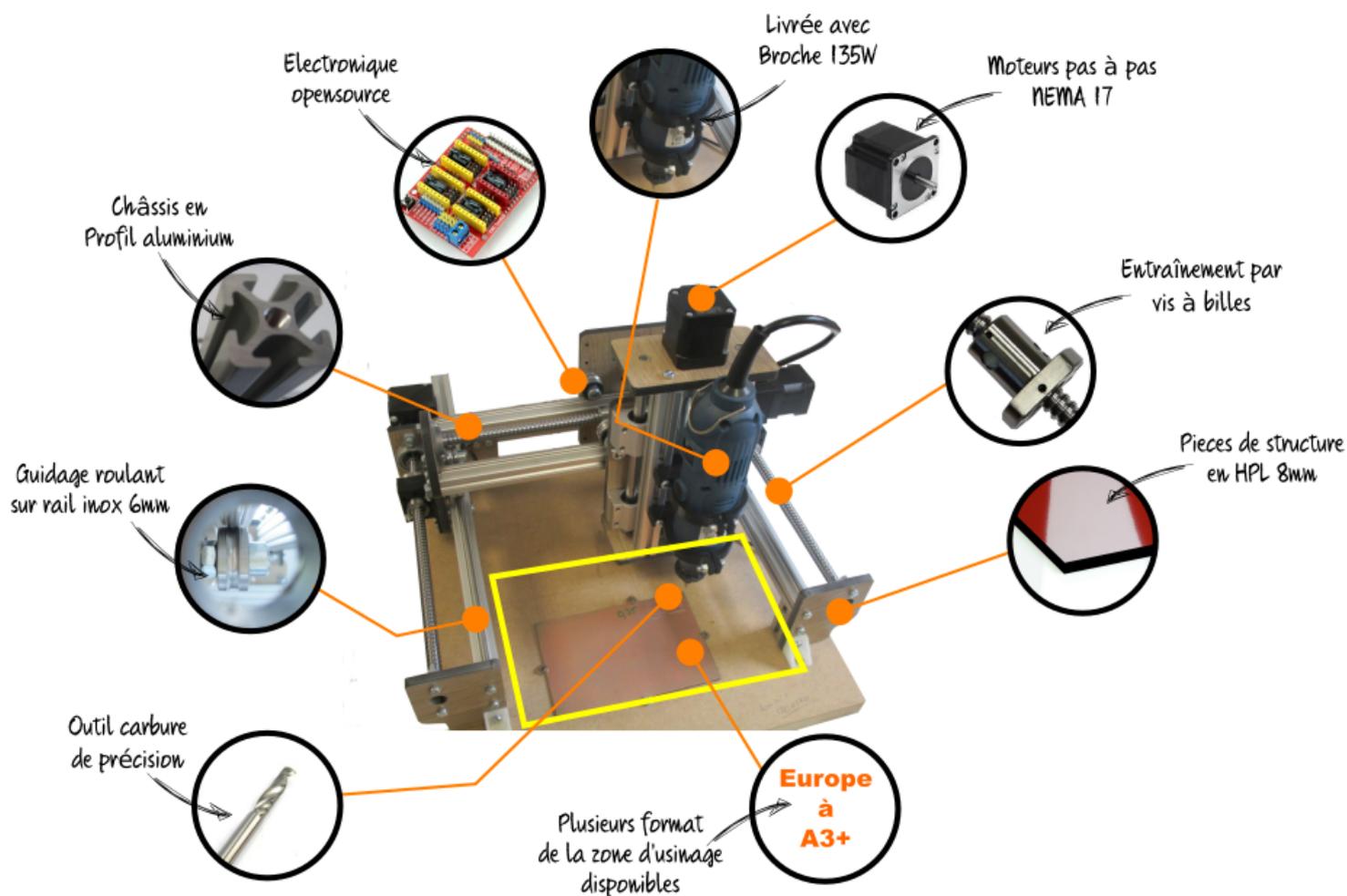


# Open Maker Machine PLUS : Première mise sous tension

Par X. HINAULT – [www.mon-club-elec.fr](http://www.mon-club-elec.fr) – Octobre 2016 Juin 2017 - Tous droits réservés - Licence [Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/) BY NC SA



On présume donc ici que l'on dispose d'une Open Maker Machine PLUS montée et câblée avec une carte électronique opérationnelle : soit Arduino+CNC shield avec GRBL, soit Emotronic/Smoothieboard avec Smoothieware.

## Contenu du document

Ce document est la suite logique de :

- la doc de montage mécanique
- la doc de câblage électronique
- la doc d'installation du firmware.

**Si vous n'avez pas encore réalisé ces étapes préalables, ce n'est pas la peine d'aller plus loin... !**

Ce document contient la procédure de première mise sous tension en mode manuel.

## Respecter les règles de sécurité

Comme pour tout appareillage, l'utilisation de l'Open Maker Machine PLUS impose le respect de quelques règles de sécurité, notamment :

- **protection électrique** satisfaisante et sans risque pour l'utilisateur
- présence d'un **bouton d'arrêt d'urgence** sur l'alimentation générale
- ne pas laisser approcher les enfants et les mineurs de la machine en action et sous tension
- bonne installation de la machine sur un plan dégagé et stable
- disposer de l'ensemble des accessoires utiles de façon ordonnée et facile d'accès

En cas d'utilisation de la broche (moteur pour outil)

- casque anti-bruit
- lunettes de protection
- aspiration des poussières

Pour de plus amples précisions, voir [la documentation de sécurité](#)

## Pour comprendre : la chaîne logicielle de contrôle de l'Open Maker Machine PLUS

### Principe général

Le principe général de contrôle de l'Open Maker Machine, commun aux imprimantes 3D opensource notamment, est le suivant :

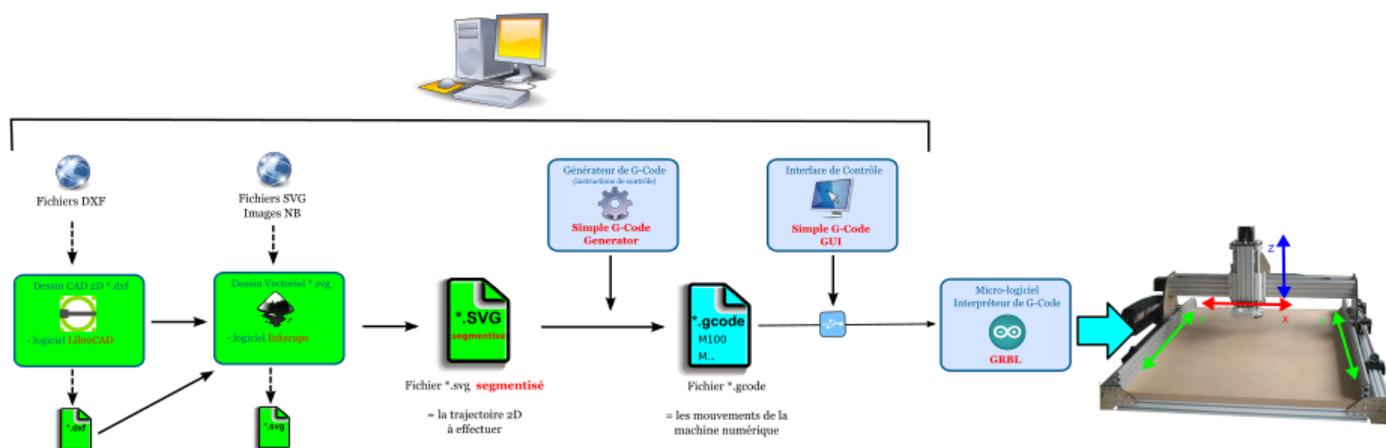
- à partir d'une pièce conçue dans un logiciel, en 2D (format SVG) ou en 3D (format STL) selon les cas...
- ... l'ensemble des mouvements nécessaires de la mécanique 3 axes pour créer la pièce vont être calculés et traduits en G-Code (retenez ce « gros-mot »), un langage simple de programmation d'automate
- les instructions ainsi obtenues vont ensuite être envoyées via le port série vers la machine qui va les exécuter fidèlement.

### Structure de la chaîne logicielle utilisée

Logiquement, l'Open Maker Machine PLUS, pour être mise en œuvre, nécessite une chaîne logicielle qui est open-source (libre en fait) incluant :

- un **décodeur de G-Code** : micro-logiciel (ou firmware) programmé dans la carte Emotronic une fois pour toutes. Ce « décodeur de G-Code », comme son nom l'indique va décoder le G-Code reçu par la machine sur le port série. Autrement dit, il va transformer une chaîne reçue sur le port série (par exemple « G01 X10.0 Y10.0 ») en un mouvement de la machine (positionnement de l'outil en coordonnées X=10mm et Y=10mm).
- une **interface graphique de contrôle** (interface « homme-machine ») qui va permettre :
  - le contrôle manuel de la machine via le port série (USB),
  - l'ouverture d'un fichier de G-Code et l'envoi de son contenu vers l'Open Maker machines via le port série (USB)
- un **générateur de G-code** : logiciel graphique qui à partir soit d'un fichier STL (dessin 3D décrivant la surface d'un objet), soit d'un fichier SVG (dessin 2D vectoriel), va permettre :
  - de générer le G-Code (=les mouvements machine à exécuter)

- en prenant en compte les paramètres voulus (diamètre d'outil, vitesse de déplacement, etc.)
- un logiciel de **conception graphique** :
  - soit **2D** : qui permettra de facilement générer un parcours outil 1 passe en créant un simple dessin vectoriel (fichier de type SVG)
  - soit **3D** (fichiers de type STL) : qui permettra de créer :
    - soit des objets 2D en épaisseur pour générer des parcours outils multi-passes,
    - soit même de véritables objets 3D pour générer des parcours outils complexes de sculpture 3D,



## Synthèse des solutions concrètes retenues en pratique

Comme pour tout autre thématique, plusieurs solutions logicielles sont disponibles et je vous présente ici une synthèse résumant les choix conseillés ainsi que les alternatives possibles :

Fonction	Solution conseillée (décrite ici)	Solutions alternatives possibles
<b>Décodeur de G-Code</b>	<a href="#">SmoothieWare</a> ou GRBL	
<b>Interface graphique de contrôle</b>	<a href="#">Simple G-Code GUI pour GRBL/Smoothieware</a>	Universal GCode Sender
<b>Générateur de G-Code</b>	<a href="#">Simple G-Code Generator</a> ou <a href="#">Pycam</a>	Plugin Inkscape ± Slic3R adapté ?
<b>Conception graphique 2D</b>	<a href="#">Inkscape</a>	<a href="#">LibreCAD</a> <a href="#">OpenScad</a> (mode projection)
<b>Conception graphique 3D</b>	<a href="#">Freecad</a> <a href="#">OpenScad</a>	<a href="#">Blender</a>

## Notions de G-Code

Même si il n'est pas question de faire de vous un expert en G-Code, il est cependant utile et intéressant en pratique de savoir à quoi cela correspond et comment sont construites les commandes types de G-Code.

Le G-Code est un langage industriel simple de contrôle d'automates.

### Principe général

Chaque commande de G-Code est construite sous forme :

## **CODE X888.888 Y888.888 Z888.888 F888.888**

où :

- CODE définit l'action à accomplir
- chacun des éléments suivant le CODE est optionnel.

### **Les lettres « clés » utilisées et leur signification**

#### **Lettres clés de commande principales**

- Commande G : commande préparatoire, choix de modes
- Commande M : fonction machine

#### **Lettres clés de paramètres de commandes**

- Commande X,Y,Z : commande de position absolue
- Commande F : commande de vitesse d'avance
- Commande S : vitesse de rotation
- Commande P : durée pause

### **Les instructions essentielles**

- **G00** X88.8 Y88.8 Z88.8 F88.8 : mouvement rapide vers les coordonnées X,Y,Z, à la vitesse F en mm/sec
- **G01** X888 Y88.8 Z88.8 F88.8 : mouvement linéaire vers les coordonnées X,Y,Z, à la vitesse F en mm/sec
- **G04** P888 : pause en seconde
- **G28** X Y Z : home du ou des axes précisés
- **G90** : mode coordonnées absolues
- **G91** : mode coordonnées relatives
- **G92** : Fixe la position courante interne

### **Exemple**

Commande G-Code :

**G01 X10.0 Y25.0 F6 ;**

Explication :

**G01** : mouvement linéaire

**X10.0 Y25.0** : vers les coordonnées X=10.0mm et Y=25.0mm

**F6** : à la vitesse de 6mm/sec

### **Contenu d'un fichier de G-Code**

Un fichier de G-Code va simplement contenir une succession d'instructions de G-Code à la « queue leu leu » à raison de 1 commande par ligne. Voici un extrait d'un tel fichier généré automatiquement :

```
G01 F6.00000
```

```
G01 Z10.0000
```

```
G04 P3
```

```
G01 X0.2826 Y100.0320
```

**G01 Z0.0000**

**G01 X100.0330**

**G01 Y0.2817**

**G01 X0.2826**

**G01 Y100.0320**

**G01 Z10.0000**

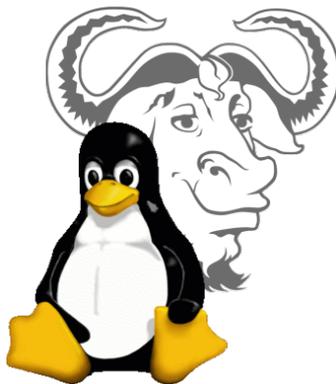
Plusieurs remarques :

- un fichier de G-Code n'est qu'un fichier texte facile à éditer et/ou à générer
- un fichier de G-Code sera le plus souvent généré automatiquement...
  - par un logiciel dédié (appelé générateur de G-Code) à partir d'un fichier 2D ou 3D
  - et potentiellement par n'importe quel petit programme personnalisé dédié
- ... mais il est également possible de créer manuellement de petits fichiers de G-Code simples ou des séquences personnalisées d'actions à réaliser (perçage)
- les instructions de G-Code seront envoyées une à une vers l'Open Maker Machine PLUS via l'interface de contrôle

### **Pré-requis : un système informatique opérationnel**

On présume ici :

- que vous disposez d'un ordinateur avec **un système opérationnel installé**
  - idéalement de type **Gnu/Linux** (=celui pour lequel l'ensemble des logiciels utilisés est fonctionnel), possiblement sous Windows ou Mac OsX (80 % des logiciels utilisés le seront, reste 20 % qu'il vous faudra adapter à votre système).
  - Je conseille idéalement un système Gnu/linux de type Debian ou Ubuntu :
    - une bonne solution « clés en main » est [notre distribution « maison »](#) (une Debian Testing) sous XFCE (voir : )
    - ou bien Xubuntu : <http://xubuntu.org/getxubuntu/>
- avec **le logiciel [Arduino installé](#) (le terminal série du logiciel Arduino sera utile)**



### **Installation du logiciel Arduino sous Gnu/Linux**

Pour installer Arduino sous Gnu/Linux (Debian, Xubuntu) : dans un terminal :

```
sudo apt-get install arduino
```

une fois fait, brancher carte Arduino et ouvrir l'IDE Arduino : si le port série n'est reconnu, fermer l'IDE et à nouveau dans un terminal faire :

```
sudo usermod -a -G dialout $USER
```

```
sudo chmod a+rw /dev/ttyACM0
```

Ouvrir à nouveau l'IDE : le port série doit être détecté à présent.

Note : On pourra ensuite, si on préfère, télécharger la dernière version Arduino sur le site officiel et la dézipper pour l'exécuter de façon autonome.

## Première mise sous tension

### Matériel nécessaire

Pour contrôler l'Open Maker Machine, on a besoin :

- de l'**Open Maker Machine PLUS** évidemment, montée et câblée conformément à la documentation de montage,
- ainsi que l'**alimentation 220V - 12V/250W** correctement connectée à l'Open Maker Machine PLUS
- deux **multiprises** dont au moins l'un d'entre eux avec interrupteur



- d'un **ordinateur de contrôle** (pc de bureau, netbook ou portable) idéalement sous Gnu/Linux : noter que l'ordinateur de contrôle pourra être une petite configuration (type dualcore - 1 ou 2 Go de RAM - HDD de 30Go) si on souhaite utiliser un poste dédié sous Gnu/Linux.



- d'un câble USB « classique » mâle A- mini-USB



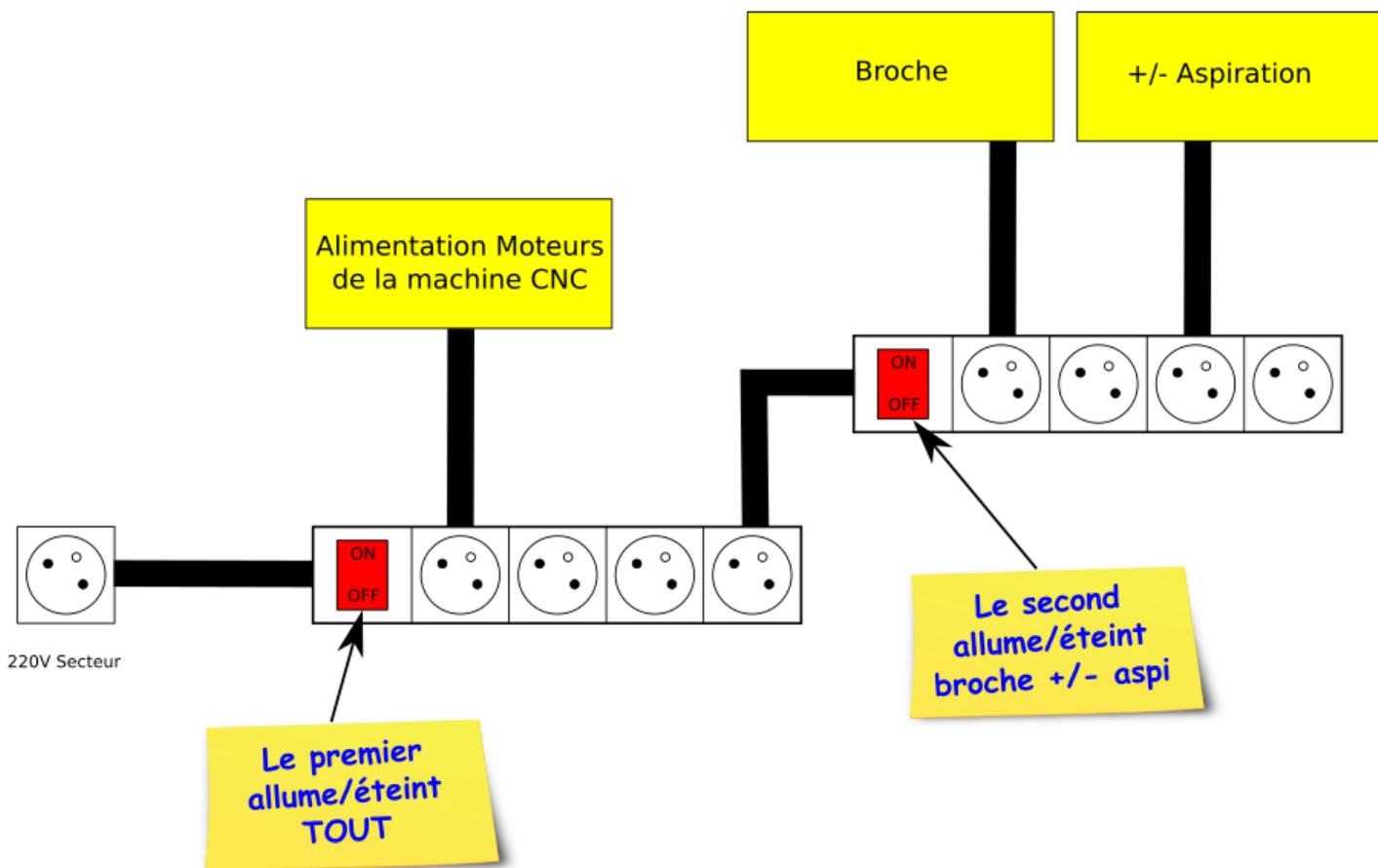
## **Circuit sur secteur à réaliser (en restant hors tension)**

---

A présent, on va pouvoir mettre en place la connexion sur le secteur, **SANS METTRE L'OPEN MAKER PLUS SOUS TENSION** cependant à ce stade.

**Le point clé et essentiel : prévoir un arrêt d'urgence dédié pour l'Open Maker Machine PLUS,** ce qui est réalisé au minimum en utilisant un multi-prises avec interrupteur sur lequel sera connectée l'Open Maker Machine PLUS. Idéalement, l'utilisation d'un vrai bouton d'arrêt d'urgence est recommandé :

L'idéal est de **connecter en série les 2 multiprises sur le 220V et de connecter sur le premier l'alimentation principale moteur et sur le second la broche  $\pm$  aspiration**. De cette façon, le premier interrupteur assure l'extinction totale moteurs + broches et le second permet de gérer uniquement la broche  $\pm$  aspiration :



Il faudra veiller à **ne pas connecter l'ordinateur de contrôle sur le même multiprise**, de façon à pouvoir laisser ou mettre hors tension l'Open Maker machine indépendamment du poste de contrôle.

**Je ne saurai trop insister sur la nécessité d'avoir un interrupteur commun d'arrêt immédiat de l'alimentation des moteurs et de la broche !**

## Connexion à l'Open Maker Machine PLUS

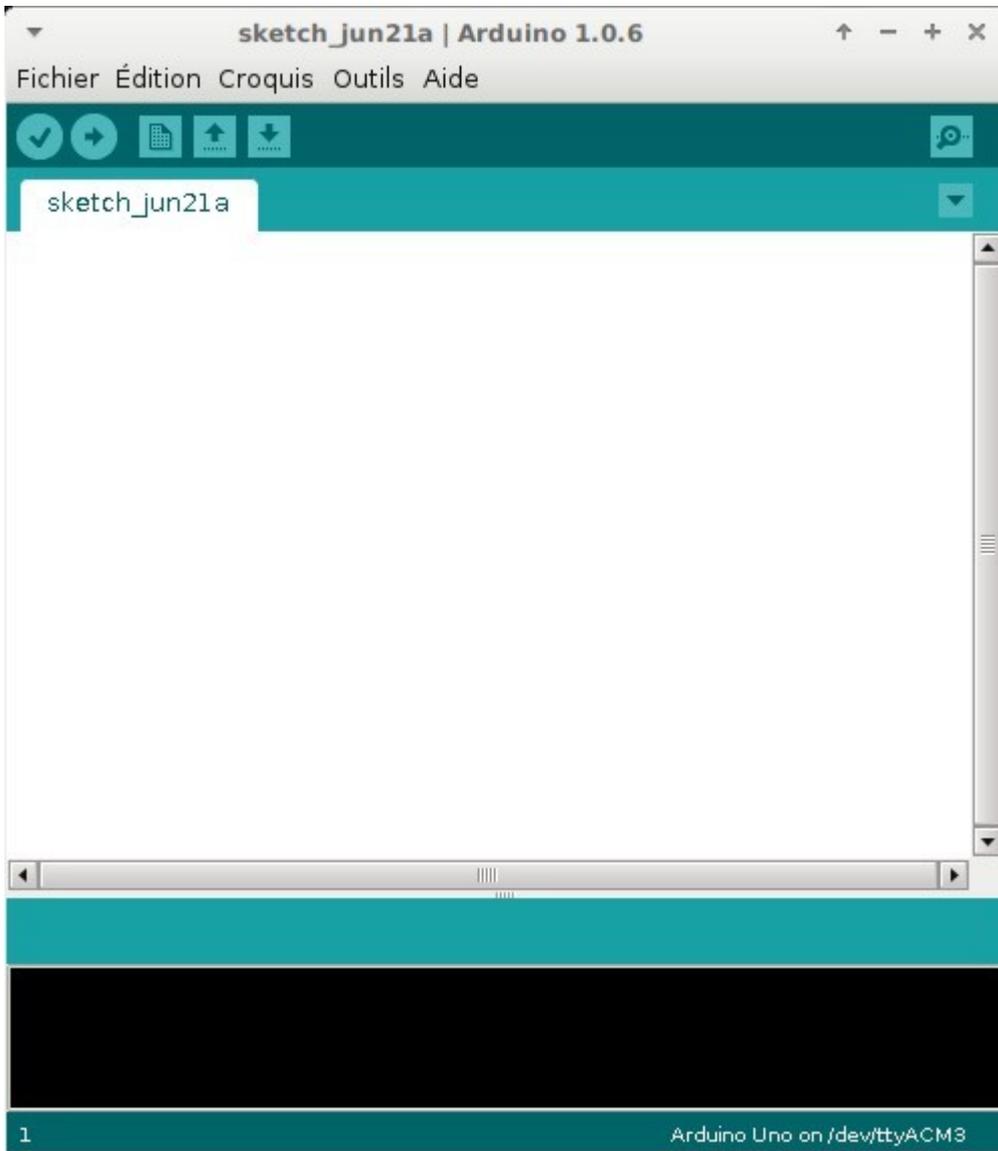
### Besoin d'un terminal série

Pour communiquer avec l'Open Maker Machine PLUS, on a besoin d'un « Terminal Série » qui va être capable d'envoyer et de recevoir des chaînes de caractères à la machine.

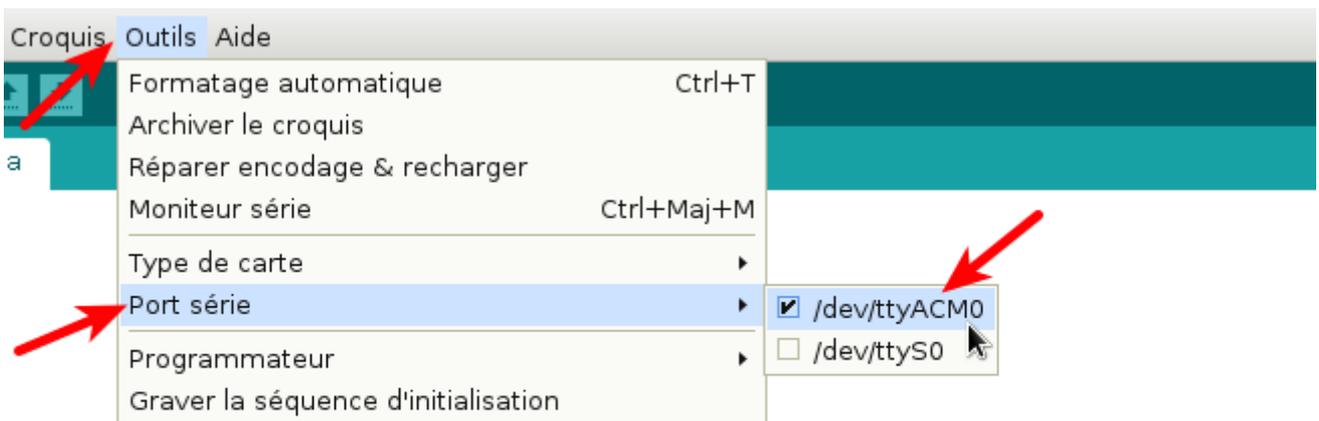
La façon la plus simple d'avoir un terminal série est d'utiliser celui du logiciel Arduino. Commencer par lancer le logiciel Arduino dont on ne va utiliser ici que le Terminal Série

### Vérifier le port de la carte

Lancer le logiciel Arduino :



et vérifier que la carte de l'Open Maker Machine PLUS est correctement détectée et que le port série est bien sélectionné via le menu **Outils > Port série > sélectionner le port série de la forme /dev/ttyACM0** sous Gnu/Linux, sous la forme COM5 sous Windows, etc.

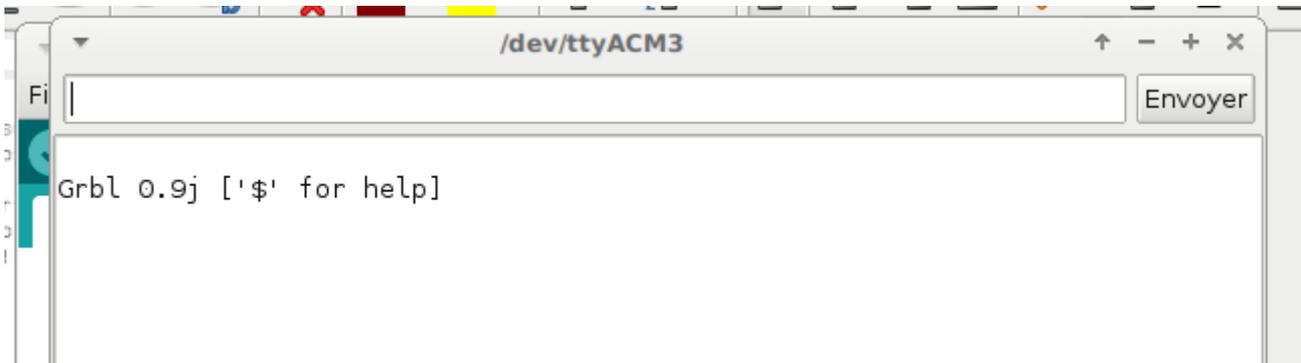


## Ouvrir le Terminal Série

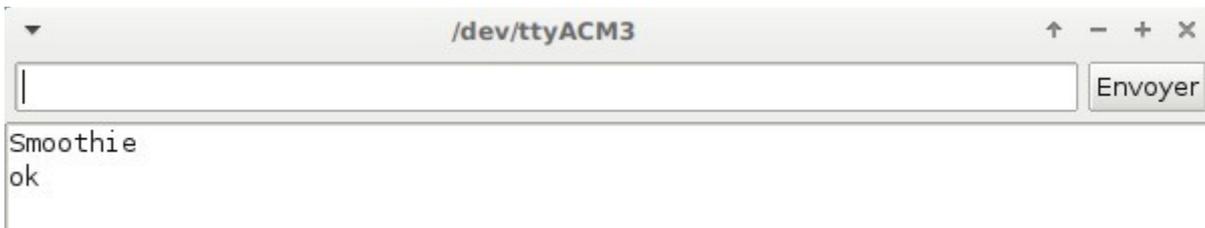
Une fois fait, se connecter avec le terminal Arduino : il suffit pour cela de l'ouvrir via le menu Outils > Moniteur Série ou bien en faisant CTRL+MAJ+M ou bien en cliquant sur :



On obtient la fenêtre du Terminal qui peut afficher des messages selon le firmware. Par exemple avec GRBL :

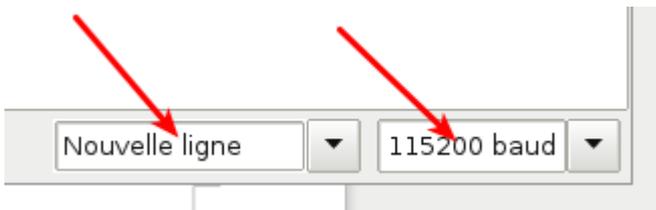


Ou bien avec Smoothieware :



## Vérifier les paramètres de communication

Vérifier qu'on a bien configuré le terminal :



## Test des moteurs de l'Open Maker Machine PLUS par le Terminal Série

### Ce que l'on va faire ici

On va commencer par **tester un à un le sens de rotation des différents moteurs** et nous allons corriger au besoin le câblage des moteurs.

La procédure va consister :

- à envoyer un ordre de mouvement au moteur de chaque axe,
- d'observer le sens de rotation
- et si il le faut, à mettre la machine hors tension, à inverser le sens de connexion du moteur, à remettre sous tension
- et à revérifier le bon sens de rotation en envoyer un nouvel ordre de mouvement.

## Règle de sécurité importante

En cas d'anomalie de fonctionnement lors de la phase de test (bruit anormal, mouvement en force, etc.), **soyez prêt à mettre hors tension l'alimentation des moteurs A TOUT MOMENT** pour éviter d'abîmer votre machine.

**Cette règle de base est et restera valable à tout moment lorsque vous utiliserez votre Open Maker Machine PLUS : c'est le moyen le plus rapide et le plus sûr de ne rien abîmer !**

**Soyez prévenu : un problème surviendra au moment où vous ne vous y attendez pas et éteindre la machine devra être un réflexe !**

## Conseil d'ami

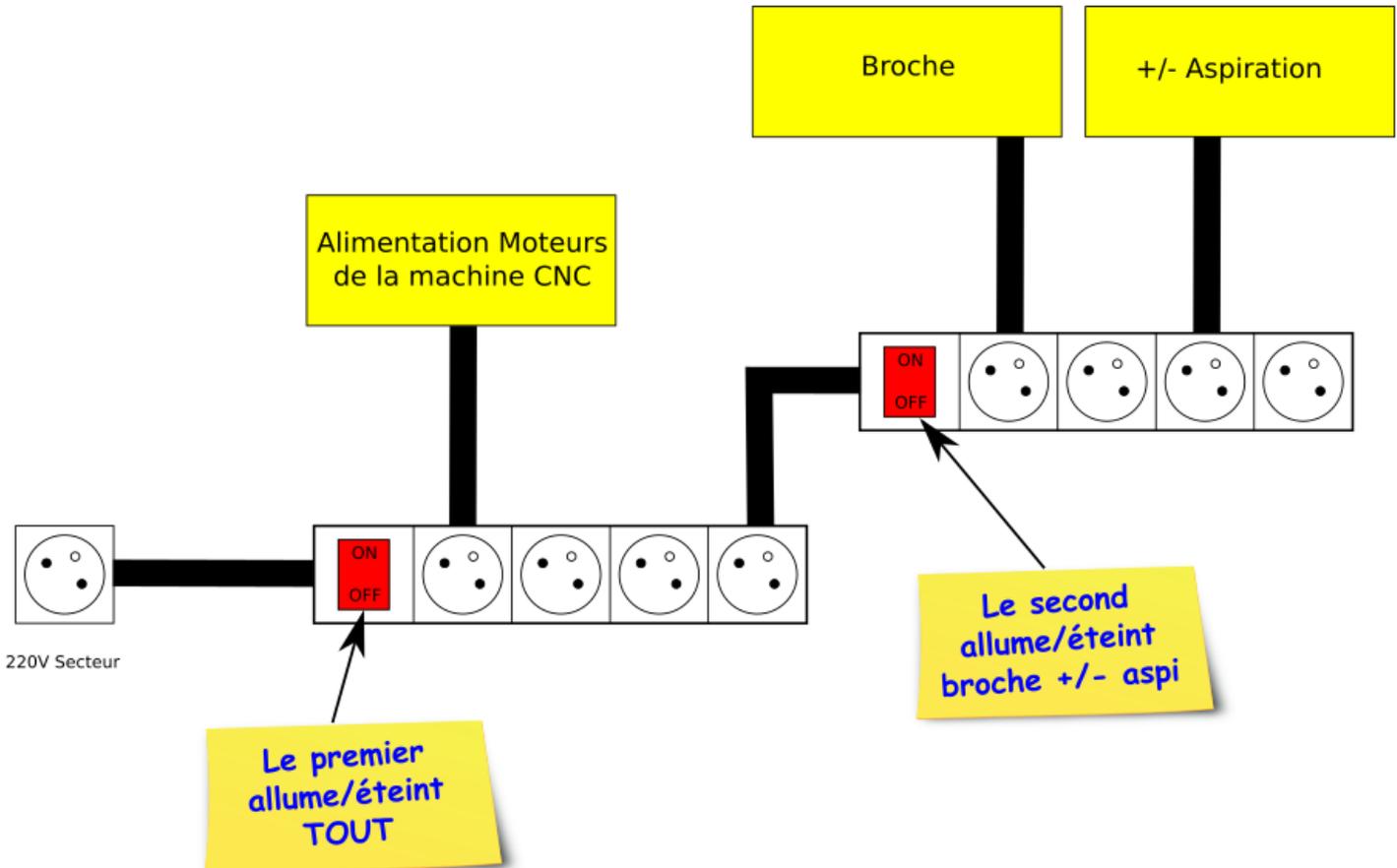
Avant de passer la suite, placez **manuellement les différents chariots X, Y, Z en position médiane = un peu à distance des endstops.**, histoire qu'un mauvais mouvement n'entraîne pas de casse !

## Prêt ? Allez on se lance !

Vous êtes prêts ? Le grand moment est arrivé : on va faire la première mise sous tension.

Allumez le multiprise de l'alimentation principale mais laissez la broche éteinte.

**Ceci a pour effet de mettre les moteurs sous tension** : les moteurs sont bloqués en rotation (si on essaie de les faire tourner manuellement).

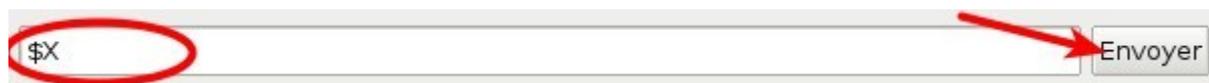


## Premiers ordres de mouvement

---

Si ce n'est déjà fait, on va ouvrir le Terminal Série.

Saisir une première commande qui permet de « débloquer » la machine pour permettre l'envoi d'ordres manuellement. Saisir dans le champs d'envoi du Terminal Série la commande \$X puis clic sur envoi :



On obtient :

```
[Caution: Unlocked]
```

```
ok
```

### **Test du X**

Saisissez :

```
G01 X10 F600
```

Le moteur X doit se déplacer de 10mm en positif, c'est à dire en s'éloignant du endstop.

Ensuite saisissez :

```
G01 X0 F600
```

Le chariot doit revenir à sa position initiale

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question. Une fois fait, rallumez l'alimentation et recommencez l'envoi des ordres de mouvement.

**IMPORTANT : NE JAMAIS LAISSER L'ALIMENTATION PRINCIPALE ALLUMÉE SI ON DEBRANCHE UN MOTEUR**

### **Test du Y**

Saisissez :

```
G01 Y10 F600
```

Le moteur Y doit se déplacer de 10mm en positif.

Ensuite saisissez :

```
G01 Y0 F600
```

Le chariot doit revenir à sa position initiale

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question. Une fois fait, rallumez l'alimentation et recommencez l'envoi des ordres de mouvement.

### **Test du Z**

Saisissez :

```
G01 Z10 F600
```

Le moteur Z doit se déplacer de 10mm en positif.

Ensuite saisissez :

```
G01 Z0 F600
```

Le chariot doit revenir à sa position initiale

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question. Une fois fait, rallumez l'alimentation et recommencez l'envoi des ordres de mouvement.

## Test des endstops

---

**ATTENTION : SOYEZ PRET A ETEINDRE L'ALIMENTATION SI VOS ENDSTOPS NE DECLENCENT PAS +++**

Une fois fait, saisir l'instruction **\$H**

L'axe X va aller chercher son endstop. Pour la première fois faites-le manuellement.

S'il se déclenche bien, resaisir \$H et le laisser se déclencher tout seul.

Faire la même chose pour le endstop Y.

Ici, il y a une petite différence entre GRBL et Smoothieware : GRBL ira chercher l'un puis l'autre, Smoothieware cherche les 2 endstops simultanément XY

Une fois la détection des origines faites, saisir la commande (qui mémorise la position courante comme étant le 0,0 machine) :

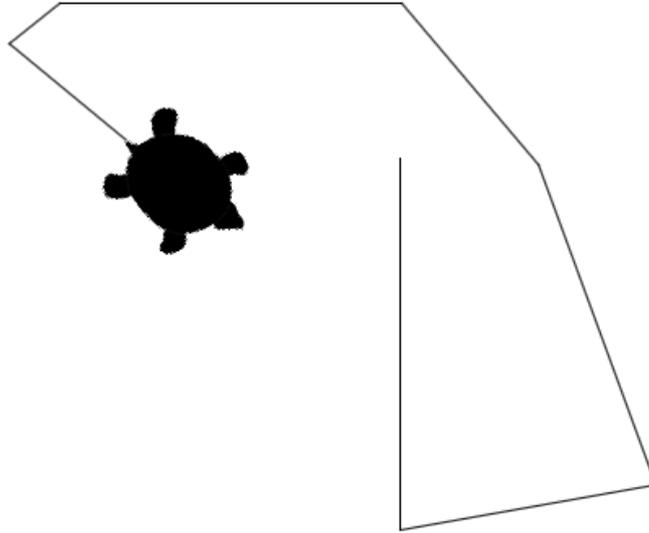
```
G92 X0 Y0
```

Il n'y a pas de endstop sur le Z par défaut : le RAZ du Z est à faire avant de démarrer une découpe.

## Faites-vous plaisir : utilisez l'Open Maker Machine PLUS en mode « Turtle »

---

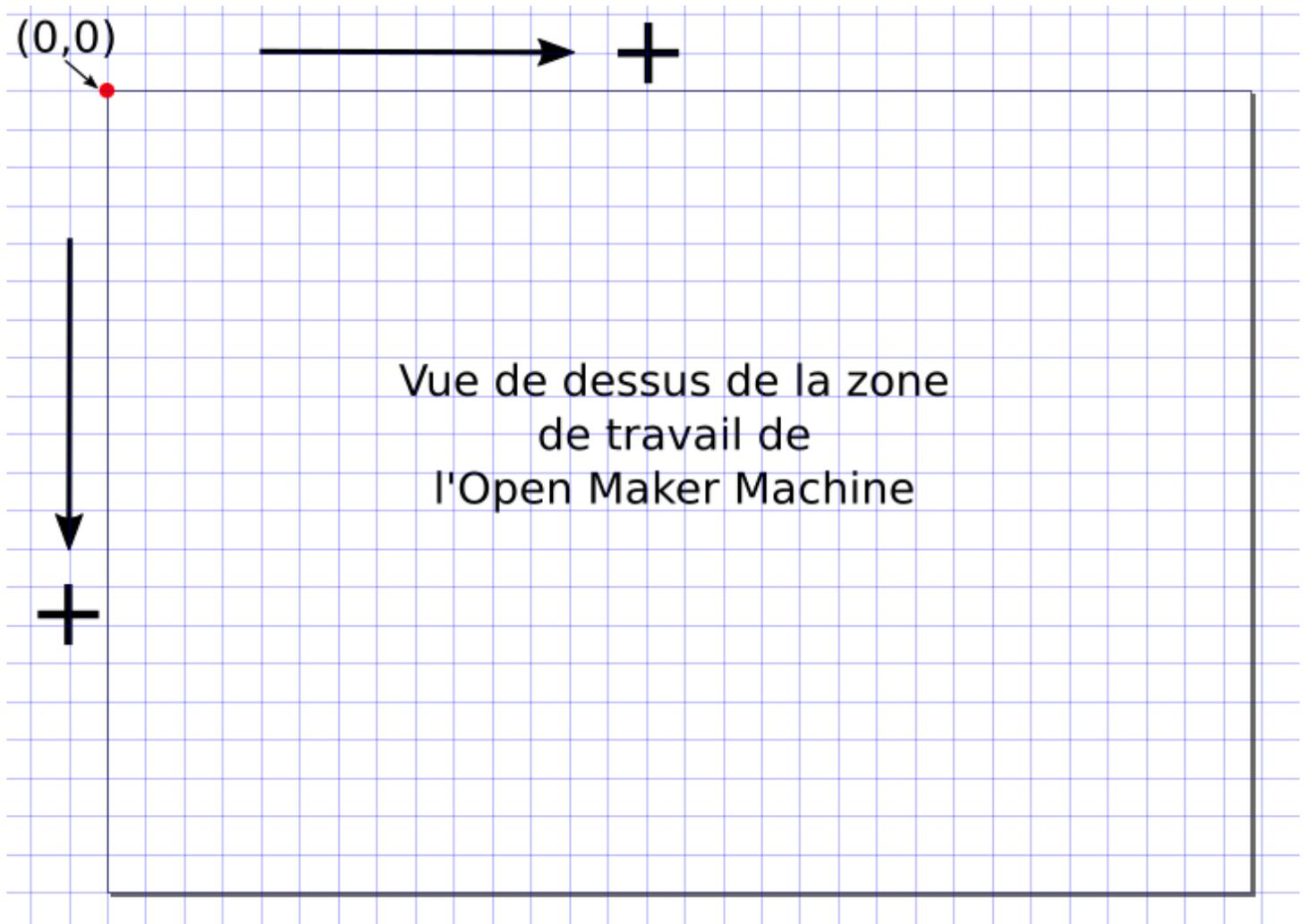
Vous avez probablement utilisé un jour ou au moins entendu parler de « [Turtle](#) », un outil d'initiation à la programmation qui vous permet de tester des actions simples (tourner, avancer, etc.) qui sont appliquées à une petite tortue affichée à l'écran...



Un exemple de dessin avec Turtle

Pour vous familiariser avec votre Open Maker Machine PLUS et vous initier au G-Code, je vous propose à présent de la tester « façon turtle » (même si c'est différent, je vous l'accorde...), c'est à dire en faisant exécuter des ordres de G-Code simples de positionnement. Vous pourrez faire la même chose par la suite avec l'outil crayon, mais vous pouvez d'ors et déjà le faire ici, « just for fun ».

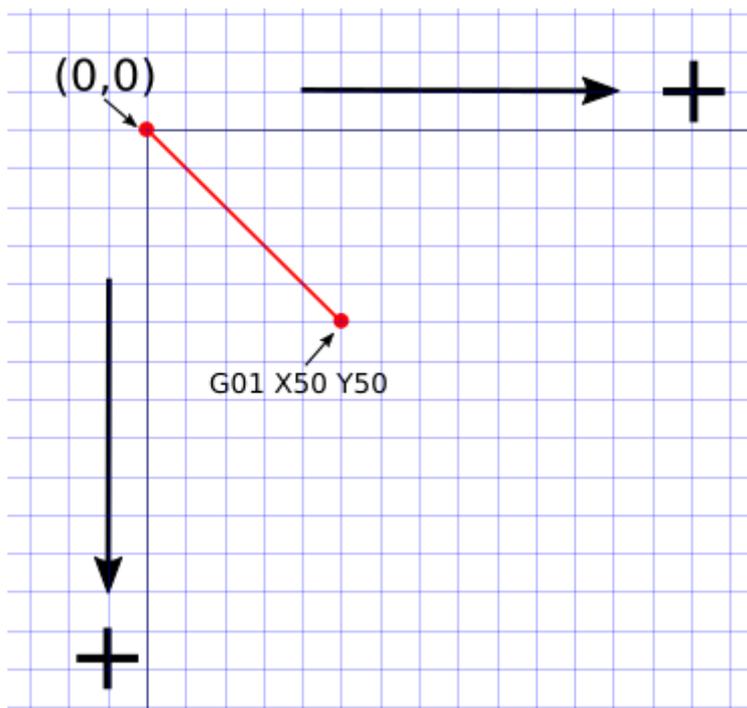
Voici le système de coordonnées de la machine :



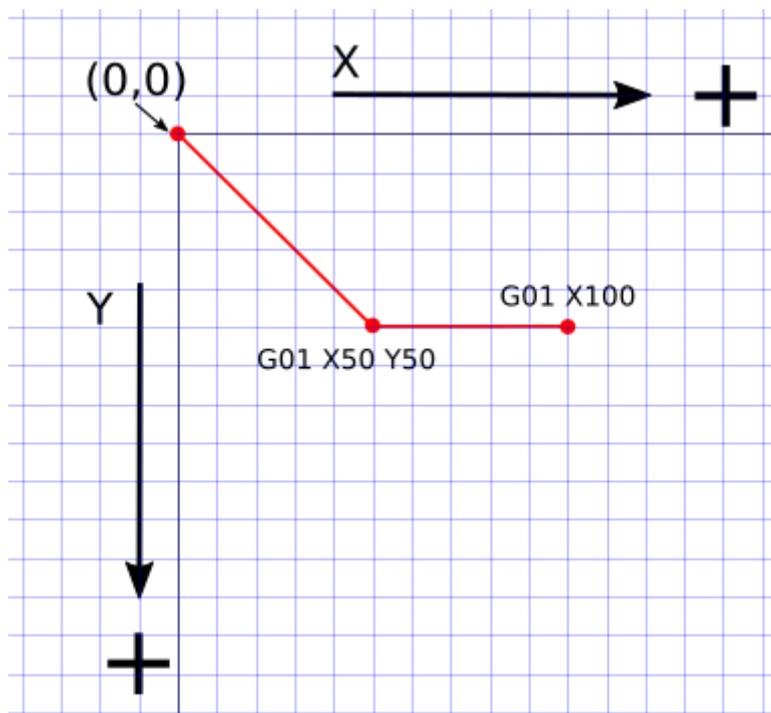
Voici quelques instructions test à saisir dans le terminal série :

Configuration en mode de coordonnées absolues : **G90**

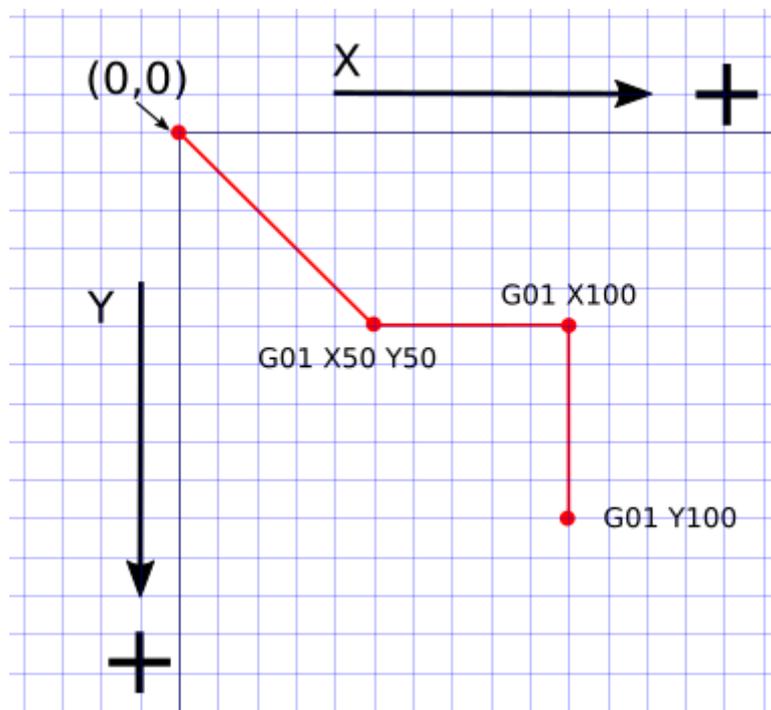
Positionnement linéaire au point X=50mm et Y=50mm à la vitesse de 600mm/min : **G01 X50 Y50 F600**



Positionnement linéaire au point X=100mm et Y courant à la vitesse courante : **G01 X100**



Positionnement linéaire au point X courant et Y=100mm à la vitesse courante : **G01 Y100**



Etc... Prenez le temps de faire vos tests... Une fois terminé, revenez à l'origine avec **G01 X0 Y0**

## Conclusion

En pratique, **il est essentiel de pouvoir « revenir aux fondamentaux »** : indépendamment de toute interface graphique et de toute complication inutile ! Comme vous pouvez le constater, vous pouvez (re)prendre le contrôle de l'Open Maker Machine à partir du seul terminal série du logiciel Arduino.

Cette façon de procéder (mais aussi la possibilité de procéder de cette façon... vous suivez ?) vous donne une grande stabilité dans votre utilisation de la machine, puisque si quelque chose ne se passe pas comme vous le souhaitez ultérieurement avec une interface, un programme, vous pourrez toujours en revenir aux choses simples pour vérifier/trouver d'où vient le problème.

Cela veut dire également que vous pouvez créer votre propre interface de contrôle de l'Open Maker Machine, pour peu que vous soyez en mesure de communiquer avec l'Arduino via le port série. Et là, ça laisse un paquet de possibilités...

## Conclusion

Voilà, vous avez pris en main la machine qui est opérationnelle. A présent, vous allez pouvoir la tester en mode graphique, la façon normale d'utiliser la machine.

Rendez-vous dans le PDF suivant.