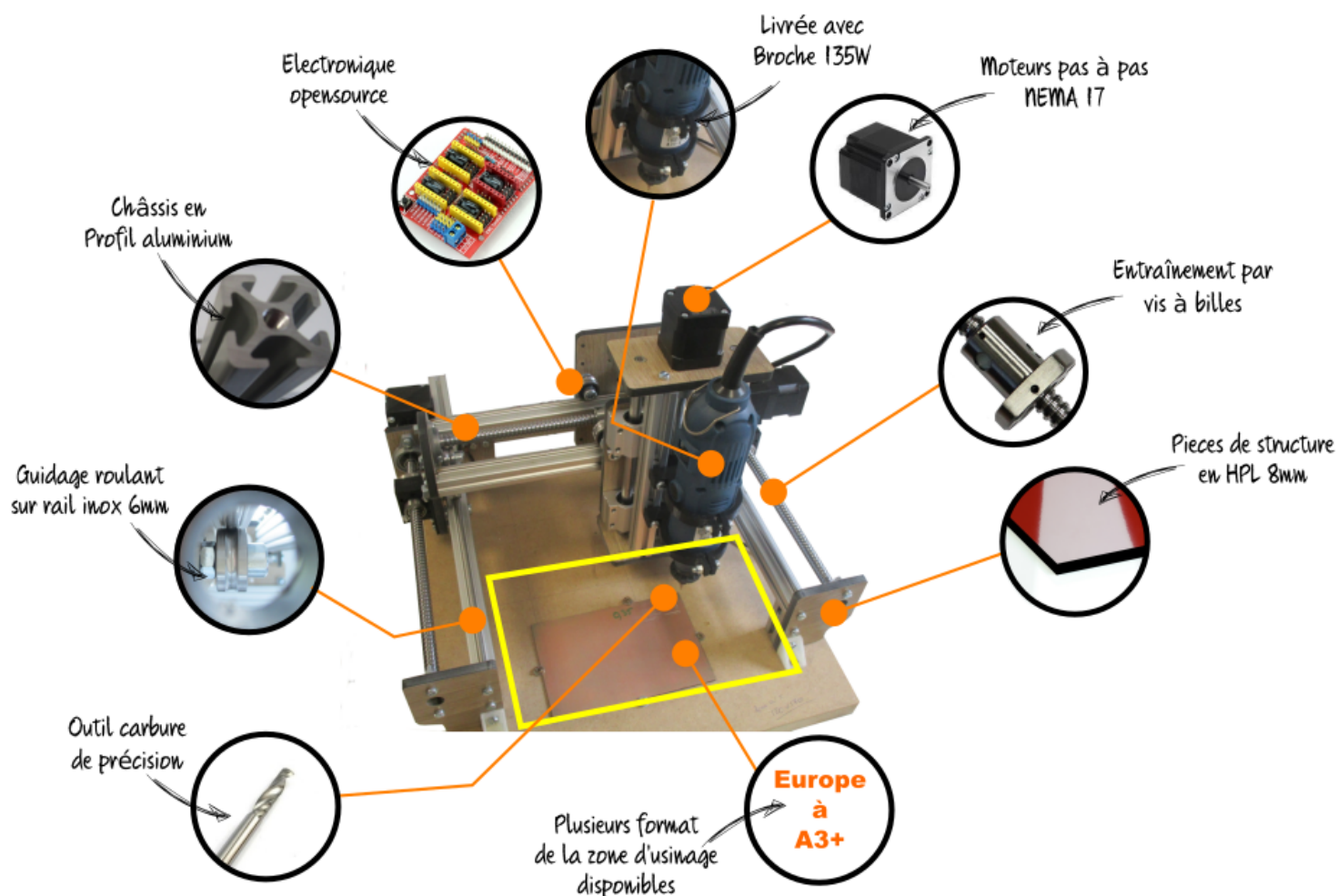


# Open Maker Machine PLUS : installation du firmware Smoothieware

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



**NOTE : TOUTE LA PROCEDURE QUI SUIT EST A FAIRE « HORS TENSION » :**  
la carte électronique sera ici alimentée uniquement par le câble USB.

La mise sous tension et les tests de mise en route ne seront fait qu'une fois le firmware programmé (voir tuto « prise en main ») .

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

### Principe général

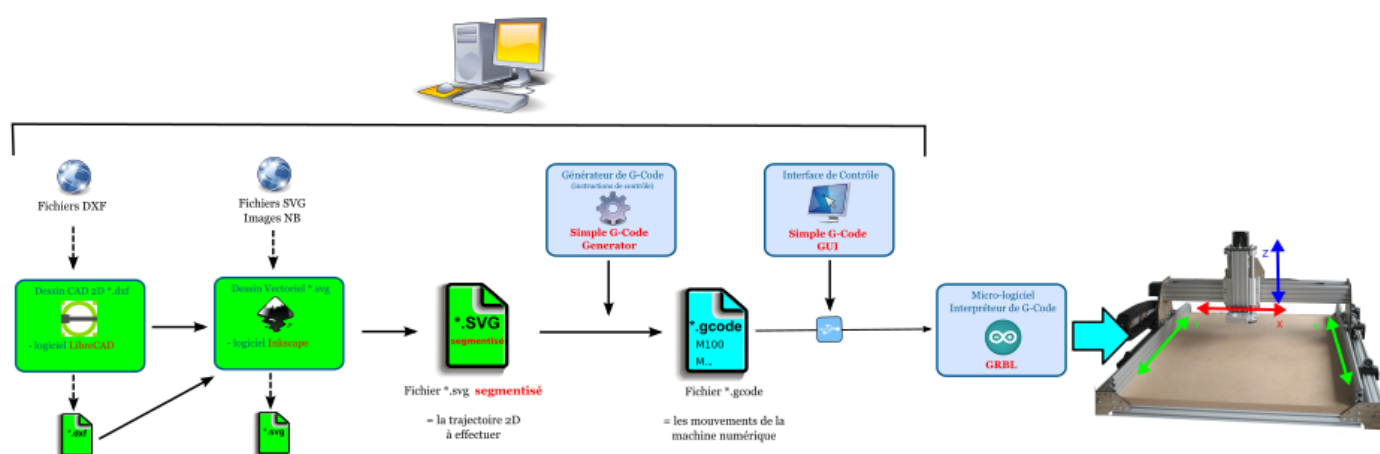
Le principe général de contrôle de l'Open Maker Machine, commun aux imprimantes 3D opensource notamment, est le suivant :

- 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 envoyés 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  |
|---------------------------------|--|-----------------------------------|
| Décodeur de G-Code              | <a href="#">SmoothieWare</a>                                     |                                   |
| Interface graphique de contrôle | <a href="#">Simple G-Code GUI pour GRBL</a>                      | Interface GRBL                    |
| Générateur de G-Code            | <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><br><a href="#">Openscad</a> (mode projection) |
| <b>Conception graphique 3D</b> | <a href="#">Freecad</a><br><a href="#">Openscad</a> | <a href="#">Blender</a>  |

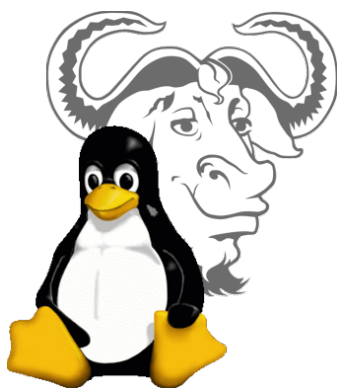
## Avant de commencer

Que ce soit pour la Emotronic ou pour la Smoothieboard, la procédure est sensiblement la même.

## Pré-requis

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)



## Smoothieware : principe général

### Un firmware pour microprocesseur puissant

Le microprocesseur est un ARM Cortex A3 32bits cadencé à 120Mhz.

Les cartes à smoothieware embarque un ARM Cortex A3 : ceci permet de disposer de beaucoup plus de puissance de calcul tout en restant sur une base micro-contrôleur (à la différence d'un mini-pc qui est un véritable SOC – système on Chip)

Voici un rapide comparatif entre une Arduino UNO et un ARM Cortex :

#### Catégorie microcontrôleur

8 bits

32 bits

#### Cadence microcontrôleur :

16Mhz

120Mhz

#### FLASH

32K

**RAM**

4K

64K

**Chargé par carte SD**

SmoothieWare est un code en C qui est compilé préalablement et mis sur une carte SD que la carte Emotronic ou SmoothieBoard va charger à la mise sous tension et/ou reset.

**Accès par câble USB : 2 modes possibles**

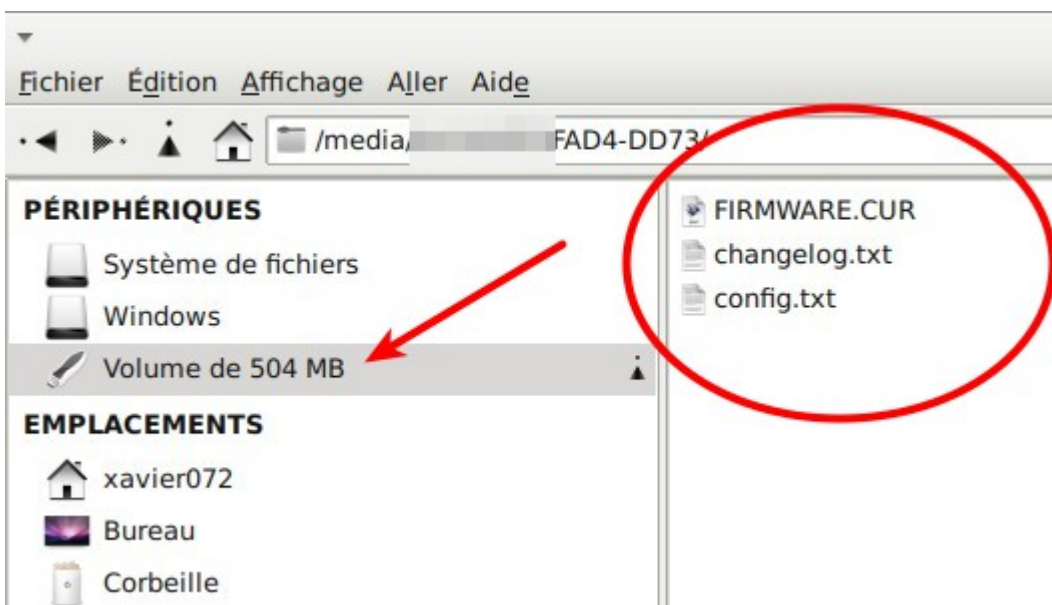
Lorsque l'on connecte la carte Emotronic ou Smoothieboard sur le port série, on a 2 mode de fonctionnement possible :

- le mode « volume » : la carte apparaît comme un disque ou une clé USB à laquelle on pourra avoir accès pour notamment la configuration
- le mode « série » classique qui correspond à la création d'un port série sur le système qui va permettre de communiquer avec la carte.

**POINT CLE : Il est impossible d'avoir les connexions simultanément ! Soit on accèdera à la carte comme une clé USB ou un disque, soit on communiquera avec. En clair, il faudra fermer tous les accès à la carte pour pouvoir communiquer par le port série et inversement fermer toute communication série avant d'accéder au disque.**

**Smoothieware : comment configurer le smoothieware, en bref :****Configurer le firmware****Accéder à la carte**

Connecter la carte au PC via le port USB. A ce moment là, les LEDs de la carte clignotent indiquant le lancement et après quelques secondes, on dispose d'un nouveau disque dans le gestionnaire de fichier :



Monter et ouvrir le volume : on voit qu'il contient plusieurs fichiers, notamment :

- firmware.cur

- config.txt

Si il y en a d'autres, on peut les supprimer.

## Configurer la carte

Supprimer le fichier config.txt qui est sur la carte.

Puis [télécharger ici le fichier de configuration pour l'OMM PLUS](#)

Enregistrer le sur votre disque dur et copier/coller le fichier sur le volume ouvert précédemment.

Enfin, renommez-le **config.txt**

Dans une approche simplifiée, vous n'avez rien de plus à faire sur ce fichier mais sachez qu'il est éditable avec l'éditeur de votre choix si vous voulez le modifier :

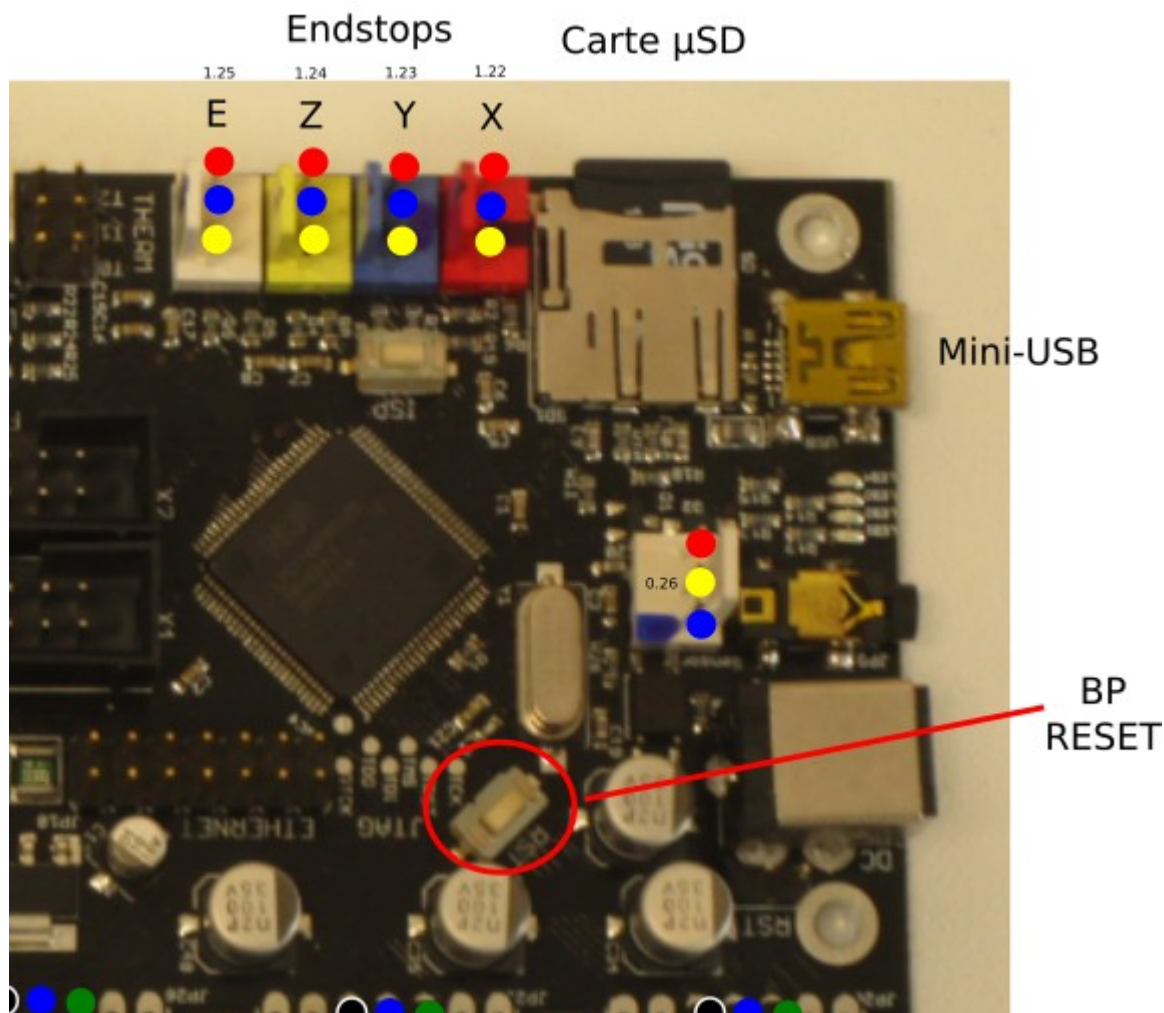
```
config.txt X
32 #minimum_planner_speed ..... 0.0 ..... # sets the minimum planner speed in m
33
34 # Stepper module configuration
35 microseconds_per_step_pulse ..... 1 ..... # Duration of step pulses to stepper
36 base_stepping_frequency ..... 100000 ..... # Base frequency for stepping
37
38 # Cartesian axis speed limits
39 x_axis_max_speed ..... 3000 ..... # mm/min
40 y_axis_max_speed ..... 3000 ..... # mm/min
41 z_axis_max_speed ..... 3000 ..... # mm/min
42
43 # Stepper module pins ( ports, and pin numbers, appending "!" to the number will invert a pin )
44
45 # axe X sur le M1
46 alpha_step_pin ..... 2.0 ..... # Pin for alpha stepper step signal
47 alpha_dir_pin ..... 3.26 ..... # Pin for alpha stepper direction
48 alpha_en_pin ..... 3.25 ..... # Pin for alpha enable pin
49 alpha_max_rate ..... 3000.0 ..... # mm/min
50
51 # axe Y sur le M2 et le M4
52 beta_step_pin ..... 2.1 ..... # Pin for beta stepper step signal
```

Les détails de la configuration sont expliqués ci-dessous.

## Relancer la carte

A présent, pour que les changements soient pris en compte, il faut faire un reset sur la carte : pour cela appuyer sur le bouton Reset :





Voilà, c'est fait et vous pouvez passer à l'étape « **Test des moteurs de l'Open Maker Machine PLUS par le Terminal Série** »

**Voilà, c'est fait ! (bon, on vous a mâché le travail en fait...)**

**Bravo ! Votre Open Maker Machine est prête à vous obéir au doigt et à l'oeil !**

### ***Smoothieware : Prise en main détaillée***

Le paragraphe précédent montre comment installer le firmware smoothieware.

Mais vous avez peut-être envie d'en savoir plus ?

## **Présentation de Smoothieware**

Smoothieware est un firmware **écrit en C, opensource, utilisable sur plusieurs bases openhardware** (Smoothieboard, Emotronic, ...). Les caractéristiques importantes de ce firmware sont :

### ***Organisation modulaire***

Dans smoothieware, chaque fonctionnalité est activée / activable sous forme d'un module indépendant du coeur du firmware : ajout d'une tête laser, de endstops, etc. Tout cela passe par l'activation du module correspondant à l'aide de variables dédiées et active les paramètres associés qui sont ignorés si le module n'est pas activé.

### ***Principe général d'utilisation : firmware et fichier de configuration sur carte SD***

Le fichier de config et le binaire compilé sont placés sur une carte SD qui est sur la carte utilisée. A

lancement, le firmware est chargé au démarrage (équivalent EEPROM) en utilisant les paramètres de configuration.

La communication avec la carte se fait par USB (câble mini-USB) et le volume apparaît comme un disque sur le système.

### ***Une configuration des broches originale***

Les broches du microcontrôleur utilisé sont de la forme générale x.yy : ainsi, 2.2 ou 0.19. La documentation de la carte utilisée indique les broches utilisées pour les différentes fonctionnalités hardware. Attention : ce ne sont pas les mêmes pour la Smoothieboard et la Emotronic.

Un code propre original est utilisé pour configurer les broches :

- `!` : inverse le niveau actif de la broche
- `^` : active le rappel au plus

Ainsi, pour activer le rappel au plus et indiquer niveau BAS actif, on fera 2.23 `!^`

### ***Dénomination des axes « à la grecque »***

A la différence des autres firmware où les axes s'appellent X, Y et Z, ici les axes s'appellent **alpha, beta, gamma**.

### ***Couplage étage moteur hardware / axes firmware souple***

Typiquement, la carte associée disposée de 3 à 5 étages moteurs qui peuvent être dénommés M1, M2, M3, M4, M5 par exemple. Chacun de ces moteurs se définit par les broches « hardware » de contrôle, à savoir **step, dir** et **enable**

Il va être possible par simple configuration d'utiliser tel ou tel étage (M1, M2, M3..) indifféremment pour l'axe alpha, beta ou gamma.

A noter que le firmware Smoothieware est très bien documenté mais tout est en anglais...

## **Flasher Smoothieware : Préparer une carte SD pour Smoothieware**

---

### ***Normalement, carte SD fournie « prête à l'emploi »***

Normalement une carte SD est fournie déjà prête. Mettez-la en place dans la carte et connecter le câble USB, ce qui va lancer le chargement.

### ***Si vous voulez préparer vous-mêmes la carte SD : rien de plus simple***

Ceci étant voici comment préparer une carte SD pour Smoothieware.

Commencer par récupérer la dernière version compilée :

<https://github.com/Smoothieware/Smoothieware/tree/edge/FirmwareBin>

Note : la version CNC est activée en mode « GRBL » par défaut.

et le fichier de config voulu parmi les exemples possibles :

<https://github.com/Smoothieware/Smoothieware/tree/edge/ConfigSamples>

Notamment, le fichier de config par défaut :

<https://raw.githubusercontent.com/Smoothieware/Smoothieware/edge/ConfigSamples/Smoothieboard/config>

Puis :

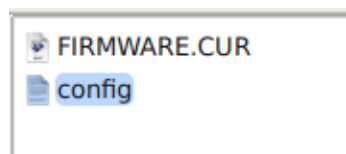
- Créer une partition
- Formater la carte en Fat16
- Copier le firmware.bin
- Et le config.txt

Une fois fait, mettre en place sur la carte et connecter la carte au port USB : les LEDs clignotent et après quelques minutes, le volume apparaît sur le système.

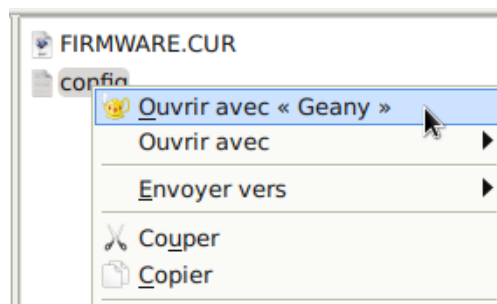
## Configuration de Smoothieware : principe d'accès

### Accès au fichier config.txt sur la carte SD

Une fois la carte utilisée connectée en USB, celle-ci apparaît comme un volume disque classique (fonction MSD). On peut configurer le fichier config.txt en éditant directement le config.txt sur la carte depuis le PC.



Et il est possible de l'ouvrir directement en éditant le fichier :



Pour plus d'info sur la communication USB, voir : <http://smoothieware.org/usb>

### Autre solution : Modifier les paramètres de configuration via un terminal

Une fois connectée en USB, la carte utilisée permet la communication série entre elle-même et le poste de contrôle.

Il est dès lors possible de modifier les paramètres directement via le port série avec l'instruction :

```
config-set sd acceleration 1000
```



voir : <http://smoothieware.org/configuring-smoothie#console-configuration-commands>

**POINT IMPORTANT : NE PAS ACCEDER A LA CARTE SD EN TANT QUE DISQUE PENDANT QU'ON COMMUNIQUE VIA LE PORT SERIE**

## Configuration : les options importantes

Les options de configuration sont nombreuses et variées : <http://smoothieware.org/configuration-options>

**NOTE : Vitesses exprimées en mm/min avec GRBL et Smoothieware, en mm/sec avec Marlin**

### Exemple de configuration de l'axe alpha :

```
# Stepper module pins ( ports, and pin numbers, appending "!" to the number will invert a pin )
alpha_step_pin      2.0      # Pin for alpha stepper step signal
alpha_dir_pin       3.26     # Pin for alpha stepper direction
alpha_en_pin        3.25     # Pin for alpha enable pin
alpha_max_rate      3000.0   # mm/min
```

### Exemple de configuration des endstops

```
## Endstops
endstops_enable      true    # the endstop module is enabled by default and can be
disabled here
delta_homing         false

alpha_min_endstop    1.22!^  # add a ! to invert if endstop is NO connected to
ground
alpha_max_endstop    nc      # NOTE set to nc if this is not installed
alpha_homing_direction home_to_min # or set to home_to_max and set alpha_max
alpha_max            500     # this gets loaded after homing when home_to_max is set

alpha_max_travel     500     # distance max utilisable pour recherche endstop
```

**POINT IMPORTANT : POUR QUE LA NOUVELLE CONFIG SOIT PRISE EN COMPTE, IL FAUT REBOOTER LA CARTE**

(Fermer le répertoire de la carte et éditeur puis appui sur reset ou débrancher/rebrancher USB)

### Bon à savoir :

Pour configurer Smoothieware en mode « GRBL » compatible : [http://smoothieware.org/grbl-mode#using-smoothie-in-grbl\\_mode](http://smoothieware.org/grbl-mode#using-smoothie-in-grbl_mode) avec l'option :

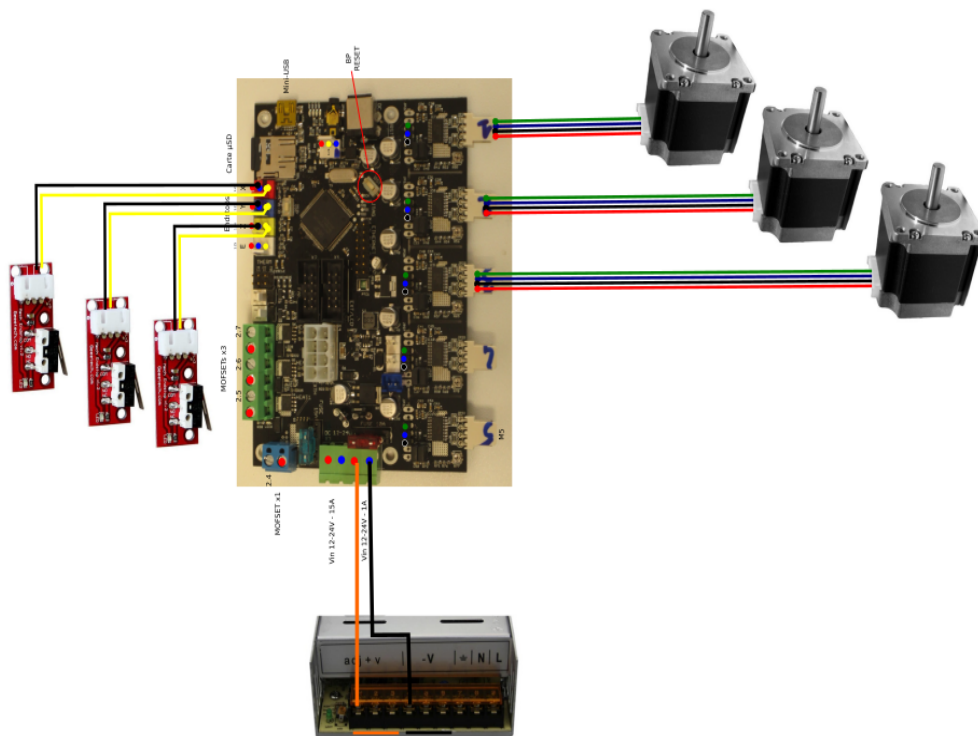
```
grbl_mode true
```

Et ici une page complète de configuration : <http://smoothieware.org/from-grbl?sl=grbl&sl=homings#moving-from-grbl-to-smoothie>

En cas d'utilisation de l'étage M3 avec un étage moteur externe, il faut désactiver le « play led » :

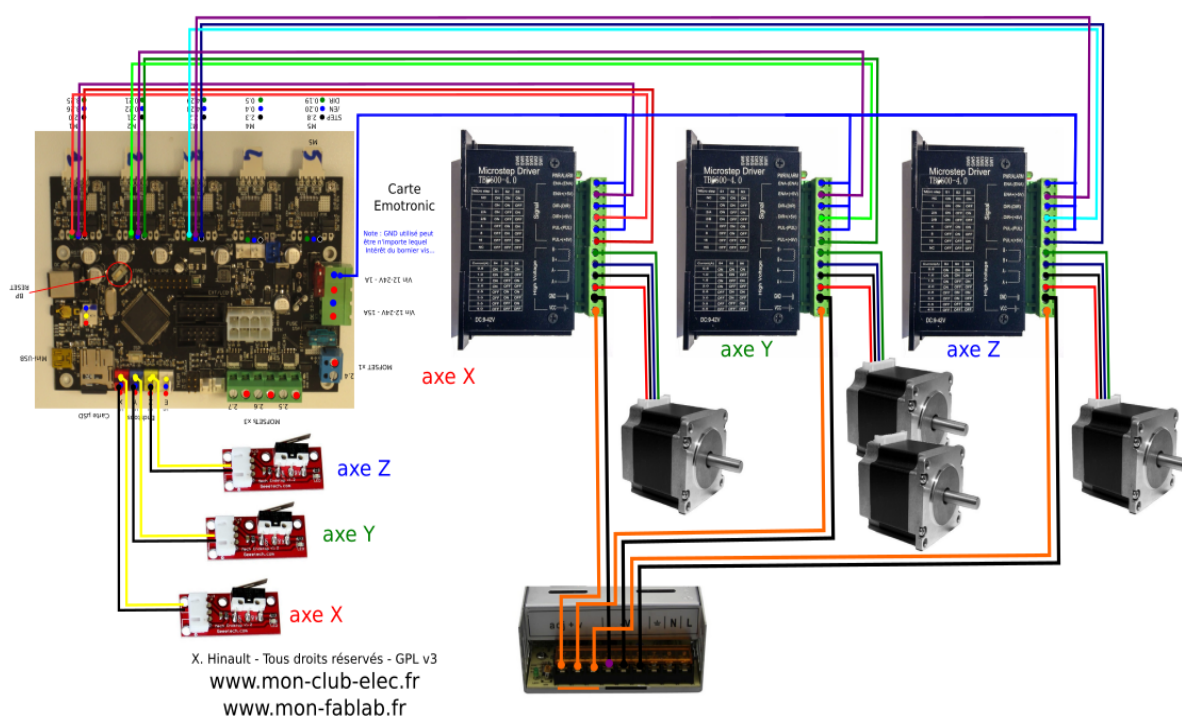
```
play_led_disable      true      # disable the play led
```

## Montage de test type : avec étages pololu intégrés



## Montage de test type : avec étages externes

Voici le montage type à réaliser dans le cas d'une CNC (étages externes) :



## Principe d'utilisation

Une fois câblé, il suffit d'envoyer des ordres de G-Code à la carte pour réaliser des actions :

### Test du X

Saisissez :

G01 X10 F600

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

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question.

### Test du Y

Saisissez :

G01 Y10 F600

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

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question.

### Test du Z

Saisissez :

G01 Z10 F600

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

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question.

### **Test de l'extrudeur**

Saisissez :

```
G01 E10 F600
```

Le moteur d'extrudeur devrait tourner de 10mm sauf si la température est trop basse.

Si le déplacement se fait en sens inverse, éteignez l'alimentation des moteurs et inverser « en miroir » le câblage du moteur en question.

### **Test des endstops**

Saisir l'instruction

```
G28 X0
```

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

S'il se déclenche bien, resaisir G28 X0 et le laisser se déclencher tout seul.

Faire la même chose pour le endstop Y. Et pour le Z

Note : le G-Code M119 donne l'état des endstops à l'instant t...

**Note : si on utilise une version CNC du firmware, la commande \$H est reconnue.**

**Pour info, tous les G-Code supportés sont ici : <http://smoothieware.org/supported-g-codes>**

## **Smoothieware : communiquer via la connexion série avec les « commandes console »**

Smoothieware dispose d'un jeu d'instructions spécifiques (hors G-Code) qui permet de naviguer sur la carte SD, etc.

Le première d'entre elle est **help** qui permet d'avoir la liste des autres :

```
version
mem [-v]
ls [[-s]][[folder]]
cd folder
pwd
cat file [[limit]][[-d 10]]
rm file
mv file newfile
remount
play file [-v]
progress - shows progress of current play
abort - abort currently playing file
reset - reset smoothie
dfu - enter dfu boot loader
break - break into debugger
config-get [<configuration_source>] <configuration_setting>
config-set [<configuration_source>] <configuration_setting> <value>
get [pos|wcs|state|status|fk|ik]
get temp [bed|hotend]
set_temp bed|hotend 185
```

```
net
load [file] - loads a configuration override file from specified name or config-override
save [file] - saves a configuration override file as specified filename or as config-override
upload filename - saves a stream of text to the named file
calc_thermistor [-s0] T1,R1,T2,R2,T3,R3 - calculate the Steinhart Hart coefficients for a thermistor
thermistors - print out the predefined thermistors
md5sum file - prints md5 sum of the given file
```

On dispose également de commandes « shell-like » :

```
ls
cd
cat
...
```

Voir ici : <http://smoothieware.org/console-commands>

## Envie d'en savoir plus ?

---

La meilleure source de documentation pour le smoothieware est le site officiel :

<http://smoothieware.org/>