# Configuration et Réglages de la cnc

Configuration de grbl avec Cnc Shield et autres joyeusetés...

---

**EN COURS DE RÉDACTION**

---

## Zone de travail

Trois paramètres pour configurer la zone de travail (le volume où la broche peut accéder) selon les axes X, Y et Z (en mm)

```
$130=300.000
$131=400.000
$132=100.000
```

## Moteurs pas à pas

Pour avoir un déplacement précis et fluide il est nécessaires de configurer correctement les drivers moteurs (micro pas) en fonction des moteurs (pas/tour) et des caractéristiques mécaniques de la machines (pas de vis sans fin). Avec ces données nous pourrons configurer grbl en fonction de notre matériel.

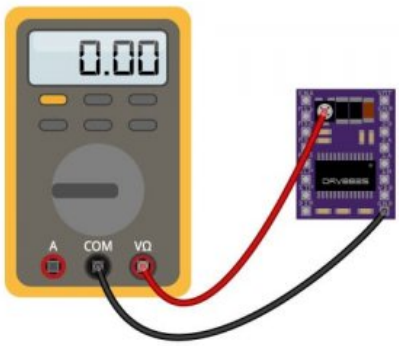
- Câblage bipolaire série

- VDC 5,27V,
- 0,85A/phase,
- 1,8°/pas

---

**Réglage du courant**

Les drivers moteurs permettent de contrôler le courant maximale qui pourra être transmis au moteurs. Il est possible de régler ce courant **Imax** directement sur le circuit en réglant le potentiomètre miniature et en mesurant la tension entre la pin de test (Vref) et la masse. Pour cela il faut un voltmètre et un petit tournevis plat puis alimenter le drivers sans envoyer de commande de rotation au moteur.



Limite de courant

```
Imax = Vref(V) x 2
Vref = Imax /2 = 0,85 /2 = 425mV
```

---

**Calcul du nombre de pas / tour**

Pour avoir une précision théorique de 1,87 micron nous avons réglé les drivers moteur à 1/8 de pas.

- Caractéristique des moteurs : 1,8°/pas soit 360°/1,8 = 200 pas tour
- Configuration des drivers moteur DRV8825 : jumper en position M0 et M1 soit 1/8 pas
- Pas de vis : 3mm

```
pas/mm =(pas par tour du moteur x micros-pas)/ pas du filetage(en mm)
pas/mm = (200 x 8) /3 = 533,333
```

Nous pouvons maintenant modifier les valeurs des paramètres de grbl :

```
$100=533.333
$101=533.333
$102=533.333
```

---

**Impulsions par pas**

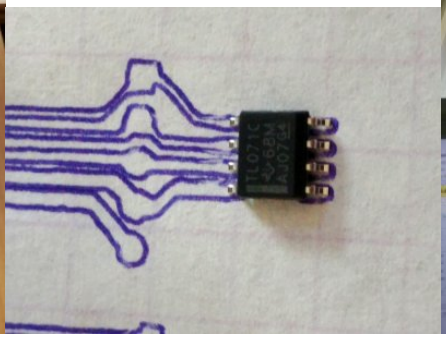
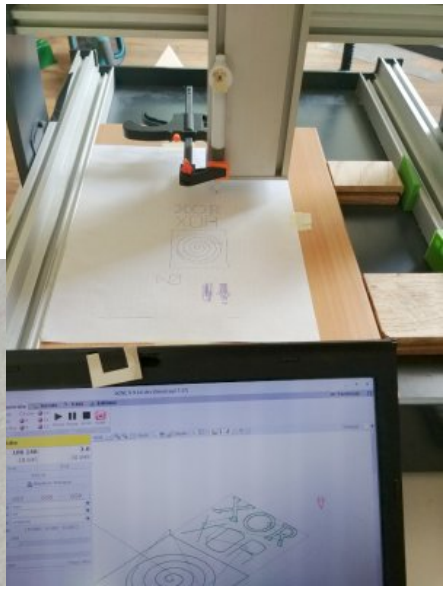
N'ayant aucune information sur la doc technique du moteur ST4118M1206 de nanotec, il apparaît dans un drivers +/- dédié à celui-ci (Datatsheet drivers nanotec pour moteur ST4118 page 8) que les impulsions doivent être supérieur à 2 µs

à tester de 2 > 10 µs ... ça change pas grand chose pour le moment

```
$0=6
```

**Étalonnage**

Pour vérifier la précision des déplacements il est nécessaire de mesurer “précisément” les déplacement selon les 3 axes. Pour cela un crayon du scotch, un reglet et du papier feront l'affaire.

---

# Fin de courses Z

Par défaut dans grbl la pin configurée pour le fin de course Z est la pin12 or sur le schield CNC c'est la pin 11 .

Il faut donc modifier config.h, pour cela il suffit de commenter la ligne 339 du fichier.

```
// #define VARIABLE_SPINDLE
```

---

#### Compiler grbl

Pour que cette modification soit effective il est nécessaire de compiler grbl avec la nouvelle modification.

- Télécharger les sources de grbl
- Faire les modifications nécessaires dans config.h

```
cd grbl
sudo make grbl.hex
```

---

#### Téléverser le fichier dans l'arduino

```
sudo PROGRAMMER="-c arduino -P /dev/ttyACM0" make flash
```

# Prise d'origine / homing

## Ajout d'options

Quelques modifications du fichier config.h

ligne 124 autoriser prise d'origine par axe : $HX, $HY, and $HZ > pratique pour tester séparément le retour maison

```
#define HOMING_SINGLE_AXIS_COMMANDS // Default disabled. Uncomment to enable.
```

ligne 127 prise d'origine mise à 0 des axes > lors de la prise d'origine la mise à 0 des coordonnées est automatique

```
#define HOMING_FORCE_SET_ORIGIN // Uncomment to enable.
```

ligne 224 pas d'alarme à la mise en route

```
#define FORCE_INITIALIZATION_ALARM // Default disabled. Uncomment to enable.
```

## Configuration du placement des fins de courses

Pour activer ce l'on appel le homing il faut configurer le placement de nos fin de courses pour que la machine aille dans la bonne direction à leurs recherche.

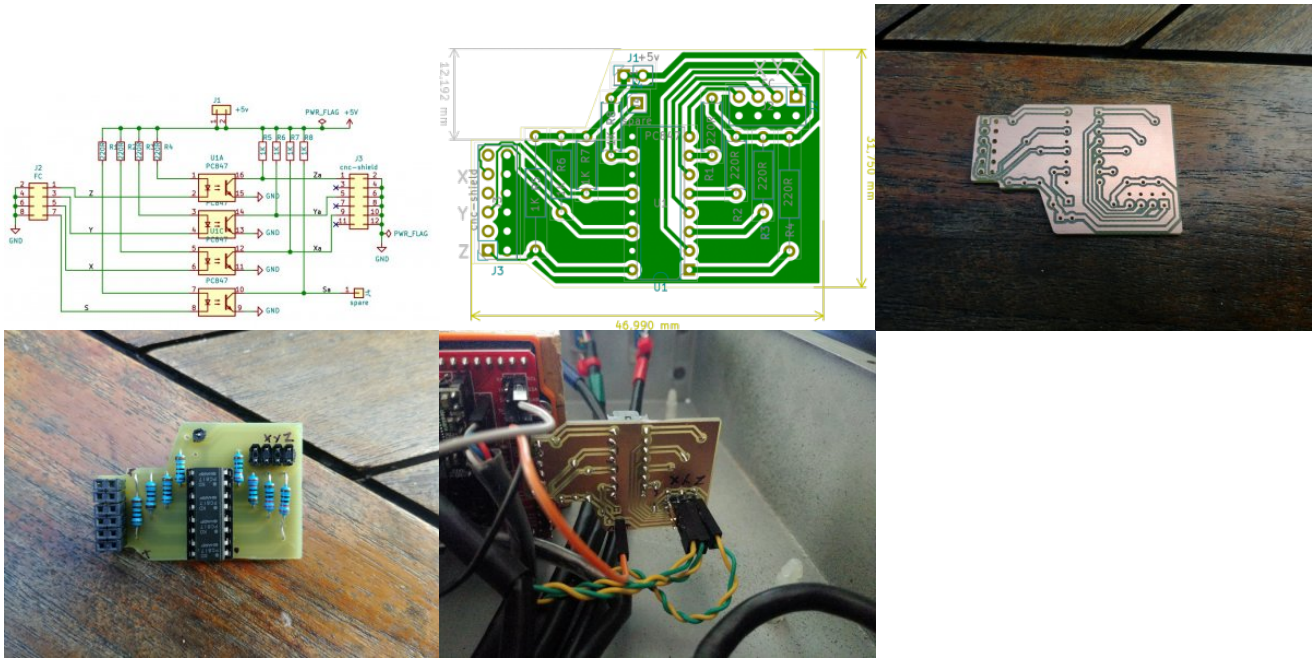
En ce qui nous concerne les fin de course sont positionnés en X-, Y- et Z+ lorsque la broche est en face de nous.

```
$23=3
```

## Ajout d'un circuit anti-parasite pour les fin de courses

Après plusieurs essais de “homing” et quelques renseignements pris sur divers forum la carte cnc-shield-v3 est sujette au parasitage. En effet les moteurs pas à pas génère du bruit électrique qui active de manière impromptu les fins de courses. Il est alors impossible de faire un cycle de prise d'origine complet : les fin de courses s'activant avant d'arriver à leurs emplacement.

D'après la documentation de grbl un circuit permet de réduire ces parasites. Voilà ce que ça donne :



Pour pouvoir activer la prise d'origine il faut aussi activer les fins de courses par sécurité (Hard Limit)

Pour le moment à cause du bruit parasite des moteurs les capteurs de fin de courses ne permettent pas la prise d'origine. Cela met en alarm grbl qui croit qu'un fin de course et activé avant la fin du cycle....

Il faudra sans doute faire un circuit pour filtrer ces parasites :

```
$21=1
```

La prise d'origine facilite les déplacements selon un point connu par la machine et permet donc en cas d'arrêt d'un travail de pouvoir reprendre depuis le dernier point.

```
$22=1
```

https://github.com/gnea/grbl/wiki/Set-up-the-Homing-Cycle

https://lebearcnc.com/grbl-homing-et-fins-de-course/

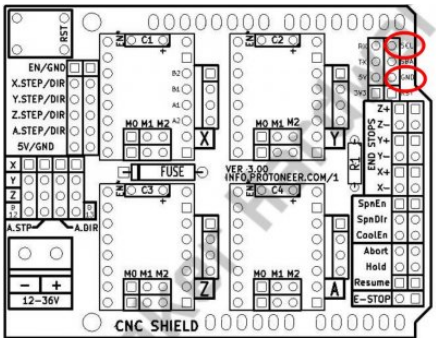
https://www.instructables.com/id/End-Stop-Limit-Switch-Problems/

# Nivelage automatique

### Connexion au cnc-shield

- connexion entre SCL et GND

SCL correspond à la pin A5 de l'arduino



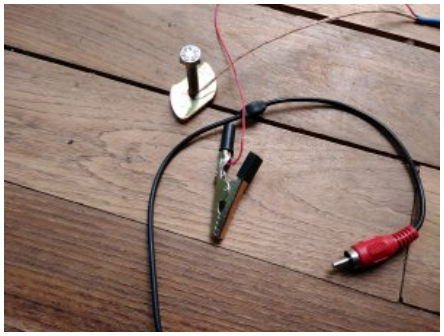

- Modifier $6 si besoin

---

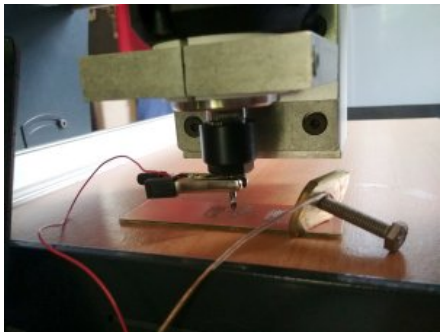
### La sonde

- mise en place d'un connecteur RCA à l'empacement de la led pour relier la dite sonde

- fabrication d'une sonde “minute”



- mise en place



---

**Procédure**

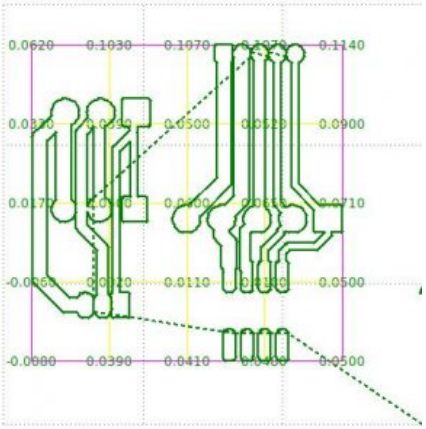
- Préparation

1. Définir X0Y0 comme origine
2. Placer la sonde du la broche et le pcb
3. **Menu Sonde > Sonde**
4. se placer à X0Y0
5. descendre la broche à 5mm de la surface
6. Z=0 > mise à 0 de la référence Z
7. définir Sonde pos -10
8. Icône Ruban > lancer cycle de sondage simple
9. Z=0 > mise à 0 de la référence Z
10. remonter Z de 1mm

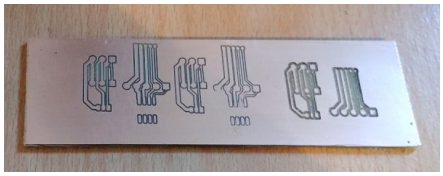
- Nivelage automatique

1. **Menu Sonde > Nivelage automatique**
2. Icône Marges > défini la surface à scanner selon le gcode importé
3. Icône scanner (petit carré violet) > déplacement selon autour des marges prédéfinies pour vérification (ne pas oublié de monté l'axe Z)
4. configurer Zmin -1 / Zmax 1 > le scan débute à 1mm de haut (rapidité) pour aller vers -1mm (sécurité).
5. choisir nombre de prise de niveau N ~5mm d'écart semble d'une bonne précision
6. Icône Scanner (engrenage) > lance la procédure
7. se placer à l'origine X0Y0 (G0 G54 X0Y0)
8. Icône Zero (mire) > validation du plan de nivelage par rapport au point prédéfini X0Y0Z0
9. enlever la sonde
10. lever la broche
11. mettre en marche la broche
12. Lancer la gravure

Voilà ce que nous donne l'écran, à chaque point correspond le niveau mesuré




De droite à gauche les 2 premiers essais en réglant le Z0 à la louche et dernier avec l'auto nivelage. Le résultat est parfait même avec des pistes de 0,25mm et une pointe javelot 90° 0,3mm



---

**Différents type de sonde**

- Métal

https://support.bantamtools.com/hc/en-us/articles/115001829134-Installing-and-Using-the-PCB-Probing-System

http://kingofprotons.blogspot.com/p/pcb-autolevelling.html

plutôt basique

- bois/plastique

http://mikropure.com/height-probe-for-cnc-machine-part-i/

---

**Sources**

https://github.com/vlachoudis/bCNC/wiki/AutoLevel

https://www.instructables.com/id/Setup-a-Touch-Probe-With-a-Arduino-Cbc-Shield-and-/

http://fablab.web-5.org/doku.php/documentation:hardware_drivers_ui_tools:bcnc-autolevel#nivelage_automatique

## Commandes utile

$$ > visualise info de configuration

$X > kill alarm

https://www.tobeca.fr/wiki/doku.php?id=guides:gcode

implémentation du GCODE

## Sources

- pblm z endstop

https://github.com/grbl/grbl/issues/1128

https://forum.protoneer.co.nz/viewforum.php?f=6

https://www.makerstore.com.au/download/publications/CNC-Shield-Guide-v1.0.pdf

https://github.com/gnea/grbl/wiki/Set-up-the-Homing-Cycle

- grbl compilation

https://github.com/grbl/grbl/wiki/Compiling-Grbl

https://github.com/gnea/grbl/wiki/Grbl-v1.1-Configuration

https://www.making-fab.fr/index.php/fraisage-numerique/grbl/10-configuration-grbl

- moteurs pas à pas

https://www.lesimprimantes3d.fr/forum/topic/3394-tuto-calcul-des-steps-per-mmunit-une-fois-pour-toutes/

http://osegouin.free.fr/cnc/paps.pdf

- réglage courant drivers

http://gilles.thebault.free.fr/spip.php?article29

https://www.pololu.com/product/2133

https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/

- impulsions driver > moteur

https://www.motioncontroltips.com/faq-pulse-direction-commands-open-loop-stepper-control/

page 7 datasheet driver

- grbl commande

https://github.com/gnea/grbl/wiki/Grbl-v1.1-Commands

- grbl erreur

http://domoticx.com/cnc-machine-grbl-error-list/

- Zone de travail

https://wiki.shapeoko.com/index.php/G-Code#Using_the_Work_Coordinate_Systems

## Firmware

### grbl

grbl Tutoriel LeBearCnc

#### Configuration

Pour le moment :

```
Grbl 1.1g ['$' for help]
[HLP:$$ $# $G $I $N $x=val $Nx=line $J=line $SLP $C $X $H ~ ! ? ctrl-x]
ok
$0 = 10    (Step pulse time, microseconds)
$1 = 25    (Step idle delay, milliseconds)
$2 = 0    (Step pulse invert, mask)
$3 = 0    (Step direction invert, mask)
$4 = 0    (Invert step enable pin, boolean)
$5 = 0    (Invert limit pins, boolean)
$6 = 0    (Invert probe pin, boolean)
$10 = 1    (Status report options, mask)
$11 = 0.010    (Junction deviation, millimeters)
$12 = 0.002    (Arc tolerance, millimeters)
$13 = 0    (Report in inches, boolean)
$20 = 0    (Soft limits enable, boolean)
$21 = 0    (Hard limits enable, boolean)
$22 = 0    (Homing cycle enable, boolean)
$23 = 2    (Homing direction invert, mask)
$24 = 25.000    (Homing locate feed rate, mm/min)
$25 = 400.000    (Homing search seek rate, mm/min)
$26 = 250    (Homing switch debounce delay, milliseconds)
$27 = 1.000    (Homing switch pull-off distance, millimeters)
$30 = 1000    (Maximum spindle speed, RPM)
$31 = 0    (Minimum spindle speed, RPM)
$32 = 0    (Laser-mode enable, boolean)
$100 = 1066,666   (X-axis travel resolution, step/mm)
$101 = 1066,666    (Y-axis travel resolution, step/mm)
$102 = 1066,666    (Z-axis travel resolution, step/mm)
$110 = 1000.000    (X-axis maximum rate, mm/min)
$111 = 1000.000    (Y-axis maximum rate, mm/min)
```

```
$112 = 500.000    (Z-axis maximum rate, mm/min)
$120 = 50.000    (X-axis acceleration, mm/sec^2)
$121 = 50.000    (Y-axis acceleration, mm/sec^2)
$122 = 20.000    (Z-axis acceleration, mm/sec^2)
$130 = 300.000    (X-axis maximum travel, millimeters)
$131 = 400.000    (Y-axis maximum travel, millimeters)
$132 = 100.000    (Z-axis maximum travel, millimeters)
```

#### modifications paramètres

- Renseigner au moins les paramètres $100, $101 and $102 – [X,Y,Z] steps/mm Calculator

```
steps_per_mm = (steps_per_revolution*microsteps)/mm_per_rev
```

- mm_per_rev : distance par tour de la vis hélicoïdale (M8 - 1.25, M10 - 1.5, M12 - 1.75)
- steps_per_revolution : nb de pas nécessaires à une 360° du moteur (360/1.8 = 200)
- microsteps : selon le contrôleur (16 ou 32 avec un DRV8825)

- Pour le sens de rotation des moteurs, il est plus simple d'inverser les câbles
- $5=1 : si les switches de début/fin de courses sont en Normaly Close au lieu de Normaly Open
- $21=1 : pour activer l'utilisation des switches de fin de course
- $22=1 : pour activer l'usage du homing et éventellement $25=5000 pour la vitesse de homing
- $23=0|1|2|3 : pour positionner le point de homing en fonction de la position des switches
- $110 et $111 : vitesse de déplacement max des axes x et y , essayer avec 10000 - Laisser $112 pour z à 500
- $130,$131,$132 : amplitude de déplacement maximum des axes x,y,z en milimètres

### Sonde

Pour la sonde de niveau 0 (probe) utiliser la pin SDL de cncshield qui sur arduino correspond à A5

Test

```
G38.2 F10 Z-100
```

Article extrait de : http://lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://lesporteslogiques.net/wiki/outil/cnc_colinbus-configuration?rev=1587746335**
Article mis à jour: **2020/04/24 18:38**