

## Civade.com

« [DIY: Réalisation d'un éclairage de broche VFD à - Firefox- Synchroniser ses marque pages, mots de »](#)

### Piloter une CNC par une Arduino avec Grbl : multiplateforme et USB !

- mercredi 15 juin 2011
- [Lien permanent](#)
- Par [jphi](#)
- Category [Technologies](#)
- Mots-clés [arduino](#), [cnc](#), [embarqué](#), [firmware](#)

Les **imprimantes 3D** (Reprap, dérivés de Reprap, Makerbot) utilisent l'**Arduino** comme contrôleur embarqué. Ce choix avait été fait pour des raisons de simplicité technique (faible cout d'une Arduino, connexion USB, facilité de programmation, etc.). Plusieurs années plus tard, cela a donné lieu à tout un écosystème de variantes, re-développement, ou nouvelles implémentations d'un micrologiciel pour Arduino ayant toujours le même objectif : Interpréter du Gcode et piloter des moteurs pas à pas.

Du coup cela offrirait la possibilité de piloter une **CNC** par un micro ordinateur qui n'est pas nécessairement un PC, et ne dispose en aucun cas de port parallèle. Le mac devient donc un candidat idéal...


Toutefois, les imprimantes 3D n'ont pas besoin de rapidité, et un très petit sous ensemble du langage est implémenté, et il n'y a souvent rien pour faire des arcs de cercle.... De plus, ces micrologiciel souffre de gros problèmes de performance, car il ne sont pas optimisé pour tirer parti des fonctionnalités matérielles du processeur de l'Arduino. Ce ne sont donc à priori pas des candidat idéaux pour la réalisation de l'électronique d'une **CNC** un tant soit peu rapide. Mais nous allons tenter de montrer le contraire....

Edit: Un billet sur une version plus récente de GRBL est disponible : <http://www.civade.com/post/2014/01/02/Arduino-et-GRBL-l-incontournable-solution-pour-piloter-une-petite-CNC>.

### Introduction: la CNC sur arduino



Arduino est une petite carte électronique open hardware, avec son environnement de développement open source, qui permet très simplement de développer en C et de tester en envoyant par un lien USB sur la plateforme. Déjà abordée à plusieurs reprises dans ce blog ( <http://www.civade.com/tag/arduino> ), cette plateforme permet très facilement, grâce à une fantastique bibliothèque de code, de faire pratiquement n'importe quoi ... a condition de faire tenir son programme dans les 32Ko de flash et les 2Ko du processeur ATMEGA368.

C'est aussi facilement abordable pour un programmeur qui veut se mettre au 'matériel' grâce à la simplicité de mise en œuvre des entrées sorties, qu'à un électronicien qui veut se mettre au logiciel grâce aux nombreux exemples livrés ou disponibles sur l'Arduino Playground ( <http://www.arduino.cc/playground/> ( <http://www.arduino.cc/playground/>)  ( <http://www.arduino.cc/playground/>) ).


Petite visite en vidéo avant de voir le détail:

#### Pilotage de CNC avec une Arduino et Grbl



### Choix du micrologiciel :

Bien sur dans un aussi petit processeur, il n'est pas question de disposer d'une interpolateur de Gcode haut de gamme comme celui intégré dans EMC2 ou Mach3, mais plus d'avoir un compromis acceptable. Par contre, cela permettrait de s'affranchir des contraintes de temps réel, si difficile à obtenir sur PC. En effet, tout le calcul de pas et d'accélération étant confié à un processeur dédié, indépendant, et n'ayant pas à gérer l'environnement graphique de l'utilisateur, cela simplifier l'approche.

J'ai testé plusieurs solutions d'interpolateur de Gcode pour Arduino, et la plus rapide et fonctionnelle semble Grbl ([http://dank.bengler.no/-/page/show/5470\\_grbl](http://dank.bengler.no/-/page/show/5470_grbl) ( [http://dank.bengler.no/-/page/show/5470\\_grbl](http://dank.bengler.no/-/page/show/5470_grbl))  ( <http://dank.bengler.no/-/page>

/show/5470\_grbl) ).

Entièrement écrit en C , et optimisé pour tenir parti de fonctionnalités intégrées à l'ATMEGA368 (timers, pwm, etc..), Il est paramétrable par la liaison série (accélération, nombre de pas par mm pour chaque axe, etc..) et reçoit le Gcode par le même canal.

Il a été sélectionné par Jonathan Ward, du département 'Machines That Make' du MIT pour son projet 'MTM Snap' ( [http://mtm.cba.mit.edu/machines/mtm\\_snap-lock/](http://mtm.cba.mit.edu/machines/mtm_snap-lock/) ) et par Synthetos comme moteur d'une shield commerciale de moteurs pas à pas nommée Grblshield ( <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ) ( <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ) ).

Simen Svale Skogsrud, l'auteur du logiciel, a fait un travail remarquable d'optimisation. Il est toutefois à l'heure de l'écriture de cet article, en plein déménagement (info Twitter!), et ne peut travailler sur le projet depuis une paire de mois. Cette info serait anodine si il n'y avait pas quelques bugs bloquants dans la dernière version, seule à supporter les arcs de cercles.

Heureusement Riley Porter et Alden Hart (Synthetos) ont apporté les corrections requises au logiciel et les ont proposées à Simen pour intégration. Il y a donc fort à parier que les prochaines versions 'Officielles' de Grbl intégreront ces correctifs.

Dans les corrections faites :

- L'accélération : très buggée sur Grbl 0.6b, et très fonctionnelle sur la version de Synthetos
- Une assignation des broches légèrement différente;
- La gestion de la désactivation des moteurs lorsqu'il n'y a pas de commandes en cours d'exécution (enable);
- Une gestion de polarité de la broche enable inversée (et cela ne vas pas nécessairement nous rendre service... voir plus loin);
- Une gestion des broches 'step' inversée également (même remarque que précédemment);
- Divers petits 'bug fixes'.

## Pré requis :

- Un PC sous Windows ou Linux, ou un MAC
- L'environnement de développement Arduino
- Un port USB de libre, pour connecter l'Arduino
- 3 contrôleurs pour moteurs pas à pas ou une shield Arduino spécialisée (ex: <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ) ( <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ) ). Dans mon cas, je vais utiliser 3 petits contrôleurs à base de contrôleurs Toshiba TB6560 : les PUTB6560 de hwml.com, montés à partir d'un CI acheté chez MCWidgets ( <http://www.mcwidgets.com/PUTB6560-PCB> ) ( <http://www.mcwidgets.com/PUTB6560-PCB> ) ).

## Configuration :

Il va tout d'abord falloir récupérer la dernière version de l'environnement de développement Arduino sur le site officiel (<http://arduino.cc/en/Main/Software>) et l'installer. Je ne donne pas de mode opératoire détaillé, celui-ci étant déjà très largement documenté.

L'environnement étant installé, nous allons ensuite paramétrer l'ordinateur pour pouvoir utiliser le compilateur C pour AVR en ligne de commande. Cette manipulation est nécessaire, car le programme que nous allons envoyer à l'Arduino n'est pas un sketch Arduino, mais bien un programme en C AVR, optimisé pour le processeur. Mais afin de nous éviter d'installer une chaîne de compilation complète, nous allons utiliser celle qui est installée dans l'environnement Arduino, et qui est utilisée lors de la compilation de Sketches de façon transparente par celui-ci.

Pour ce faire, il va nous falloir ajouter ce chemin à la variable d'environnement \$PATH de notre OS.

**Sur Mac**, éditer le fichier .bash\_profile dans la racine de votre compte et ajouter les lignes suivantes :

```
# pour utiliser l'environnement Arduino comme compilateur avr
export PATH=$PATH:/Applications/Arduino.app/Contents/Resources/Java/hardware/tools/avr/bin
```

puis lancer la main la commande pour éviter de relancer un shell

```
export PATH=$PATH:/Applications/Arduino.app/Contents/Resources/Java/hardware/tools/avr/bin
```

**Sur Windows**, faire :


- Appuyer simultanément sur les touches "Windows" + "Pause"
- aller dans l'onglet "Avancé",
- cliquez sur le bouton "variables d'environnement"
- Modifier la variable d'environnement PATH pour ajouter ";C:\Program Files\arduino-0022\hardware\tools\avr\bin;C:\Program Files\arduino-0022\hardware\tools\avr\utils\bin" à la fin.
- N'oubliez pas le ';' de séparation :

**Sur Linux** éditer le fichier .bash\_profile dans la racine de votre compte et ajouter de quoi appeler le sous répertoire bin contenant les exe du compilateur AVR.

## Récupération du firmware

Le firmware choisi est le fork réalisé par Synthetos de Grbl. La page d'accueil du projet est ici : <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ( <https://www.synthetos.com/wiki/index.php?title=Projects:grblShield> ) ).

Les sources du projet sont à récupérer sur GitHub : <https://github.com/synthetos/grblShield> ( <https://github.com/synthetos> )

[/grblShield](https://github.com/synthetos/grblShield)  (https://github.com/synthetos/grblShield) . Prendre l'archive TGZ disponible. Le fichier ressemblera à ceci : synthetos-grblShield-bc5e96e.tar.gz. Le décompresser.

## Configuration du firmware

Le micrologiciel que nous allons envoyer sur l'Arduino est dans le sous répertoire synthetos-grblShield-bc5e96e/firmware . Nous allons procéder à quelques modifications / configurations.

- Éditer le fichier *config.h*, passer le baudrate à 38400 bauds :

Avant:

```
#define BAUD_RATE 9600
```

Après:

```
#define BAUD_RATE 38400
```

- Éditer le fichier *settings.c* pour réactiver l'invert mask par défaut. Pour ce faire, commenter la ligne

```
///#define DEFAULT_STEPPING_INVERT_MASK 0x1C    ///@grblshield/
```

et décommenter la ligne

```
#define DEFAULT_STEPPING_INVERT_MASK 0
```

- Éditer le fichier *protocol.c* afin de mettre en place un hack (affichage de la ligne start) afin de rendre grbl compatible avec replicatorG, Cela nous sera utile lors des tests. (en gras, l'ajout)

```
void protocol_init()
{
  beginSerial(BAUD_RATE);
  printPgmString(PSTR("\r\nGrbl " GRBL_VERSION));
  printPgmString(PSTR("\r\nstart"));
  printPgmString(PSTR("\r\n"));
}
```

Et enfin, pour faire en sorte que le signal enable soit actif à 1, dans *stepper.c*, remplacer :

```
static void set_step_events_per_minute(uint32_t steps_per_minute);
void st_wake_up() {
  STEPPERS_ENABLE_PORT &= ~(1<<STEPPERS_ENABLE_BIT);
  ENABLE_STEPPER_DRIVER_INTERRUPT();
}
void st_disable_steppers() {    ///@grblshield/
  STEPPERS_ENABLE_PORT |= (1<<STEPPERS_ENABLE_BIT);
  DISABLE_STEPPER_DRIVER_INTERRUPT();
}
```

Par :

```
static void set_step_events_per_minute(uint32_t steps_per_minute);
void st_wake_up() {
  STEPPERS_ENABLE_PORT |= (1<<STEPPERS_ENABLE_BIT);
  ENABLE_STEPPER_DRIVER_INTERRUPT();
}
void st_disable_steppers() {    ///@grblshield/
  STEPPERS_ENABLE_PORT &= ~(1<<STEPPERS_ENABLE_BIT);
  DISABLE_STEPPER_DRIVER_INTERRUPT();
}
```

- Enfin, il va nous falloir paramétrer notre *Makefile* afin qu'il sache sur quel port série sera vue l'Arduino. Pour ce faire, il va nous falloir :

Remplacer la ligne :

```
PROGRAMMER = -c avrisp2 -P usb
```

par (sous windows)

```
PROGRAMMER = -C "C:\Program Files\arduino-0022\hardware\tools\avr\etc\avrdude.conf" -c stk500v1 -P COM5 -b57600
```

ou (sous OSX)

```
PROGRAMMER = -c stk500v1 -P /dev/tty.usbserial-A7006R7r -b57600
```

Bien sur le nom du port série (derrière option -P) est à remplacer en fonction de votre config...



\$4 et \$5 sont réservés pour régler les vitesses par défaut en usinage (\$4) et en déplacement rapide (\$5);

\$6 sert à régler la résolution des arcs. Ceux ci sont en effet rendus comme de petits segments de droite. La taille de ces segments est indiquée ici` ;

\$7 permet de mettre un masque d'inversion des signaux de pas et de direction pour chaque axe. Pour chacun d'entre eux devant être inversé, prendre la valeur ci dessous, en faire la somme et l'assigner à \$7:

- Pas X : 1
- Pas Y : 2
- Pas Z : 4
- Direction X : 8
- Direction Y : 16
- Direction Z : 32

Ainsi pour inverser la direction de X et Y, il suffirait de mettre 24 (16+8) dans \$7.

\$8 permet de régler l'accélération MAX.

Et enfin, \$9 permet de régler la valeur au dessus de laquelle, lors du passage d'une vitesse à une autre, on passera par le gestionnaire d'accélération au lieu de faire un changement de vitesse instantané.

Après avoir paramétré la carte avec vos valeurs, nous voila prêts pour la suite.

### Câblage de la carte aux contrôleurs pas à pas

Tel que le code a été modifié puis compilé, tous les signaux sont en logique positive, c'est à dire que les impulsions de pas sont données à l'état 1 (5V), idem pour le signal enable, qui valide avec un 5V les déplacements du moteur. Il va donc falloir maintenant raccorder notre carte arduino à un contrôleur pas à pas qui va gérer la puissance . Voici la fonction des différentes broches de l'Arduino (réglages par défaut de la version 0.6b modifiée par Synthetos) :

Fonction	Broche Arduino	Commentaire
Step X	Digital 2	Impulsion de pas axe X
Dir X	Digital 5	Direction axe X
Step Y	Digital 3	" " Y
Dir Y	Digital 6	" " Y
Step Z	Digital 4	" " Z
Dir Z	Digital 7	" " Z
Stepper enable	Digital 8	Activation des tous les axes
Limite X	Digital 9	Contact de limite d'axe X
Limite Y	Digital 10	Contact de limite d'axe Y
Limite Z	Digital 11	Contact de limite d'axe Z
Broche active	Digital 12	Démarre la broche
Direction Broche	Digital 13	Change le sens de rotation de la broche

### Tests avec le serial monitor

Après avoir raccorder nos contrôleurs à l'Arduino (ne pas oublier de raccorder aussi une masse!), les moteurs aux contrôleurs, et alimenté le tout, on va pouvoir passer aux essais. Le plus simple est de la faire toujours avec l'environnement Arduino et le serial monitor. Taper les commandes suivantes pour vérifier le bon fonctionnement :

```
G90
G00 X100 F50
```

Le moteur de l'axe X va faire un déplacement de 100mm à 50 mm / minutes... Pour tester l'axe Y :

```
G00 Y100.0 F400
```


Cette fois c'est le moteur de l'axe Y qui va se déplacer de 100mm, mais à 400 mm par minute. En ainsi de suite pour l'axe Z. Il est ainsi possible de tester la vitesse max, la vitesse d'accélération, etc..

A noter que l'Arduino répond "ok" a chaque fois qu'une commande a été comprise et exécuté. La faible quantité de mémoire ram disponible sur le processeur de l'Arduino ne permet que de mettre une ligne de Gcode en tampon.

Il est donc nécessaire de 'streamer' le gcode à l'aide d'un programme, qui devra attendre le retour de l'Arduino ('ok') avant de renvoyer une autre commande. Pas pratique... Un script Ryby est livré avec la version originale de Grbl, qui a cet usage.

### Tests avec ReplicatorG

Une autre méthode de test est d'utiliser ReplicatorG, le logiciel conçu pour piloter les imprimantes 3D Makerbot. Écrit en Java, et basé sur le code de l'environnement de l'Arduino, ce programme a le bon gout de fonctionner aussi bien sur Windows, Linux et OSX.

La première choses à faire est de le télécharger et l'installer à partir de <http://code.google.com/p/replicatorg/downloads/list> (<http://code.google.com/p/replicatorg/downloads/list>)  (<http://code.google.com/p/replicatorg/downloads/list>) .

Après l'avoir installé, il va nous falloir créer une configuration spécifique pour l'Arduino. Il existe plusieurs 'drivers' dans ReplicatorG. L'un d'entre eux nous intéresse particulièrement puisqu'il est conçu pour passer le Gcode au port série ligne par ligne, en attendant de recevoir le message "ok" avant d'envoyer la ligne suivante.

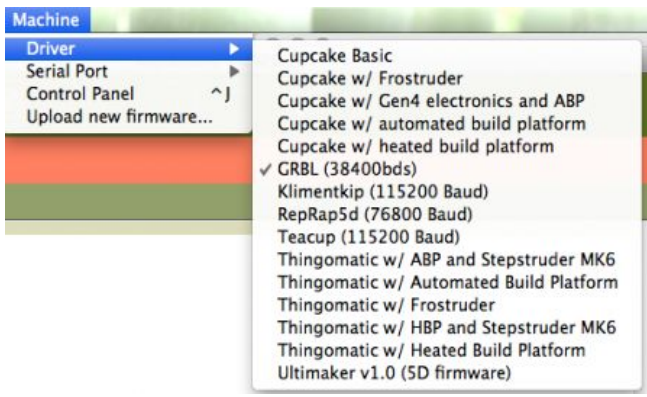
C'est ce driver que nous allons exploiter. Mais pour cela il va nous falloir créer une config de machine. Sur ReplicatorG, elles sont stockées dans le sous répertoire machines, sous forme de fichier XML.

A cet effet, j'ai créé une config : [arduino-grbl.xml](#). Il suffit de copier ce fichier dans le sous répertoire "Machines et lancer ReplicatorG.

Lors du 1er lancement, ReplicatorG indique qu'aucune machine n'est connectée :



Nous allons ensuite sélectionner le pilote que nous venons tout juste d'installer. Pour ce faire, aller dans le menu "Machine", puis "Driver", et sélectionner "Grbl 38400 bauds" :

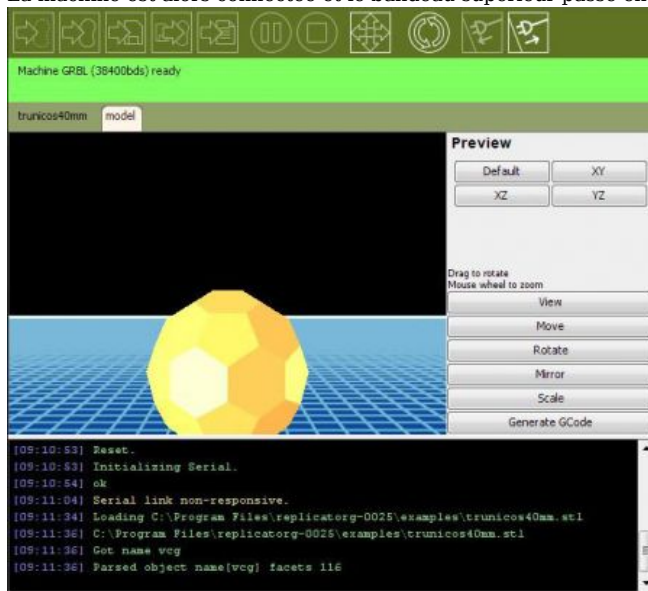


Enfin, la dernière étape consiste à sélectionner le port série sur lequel votre arduino est connectée dans le menu "Machine" puis "Serial Port".

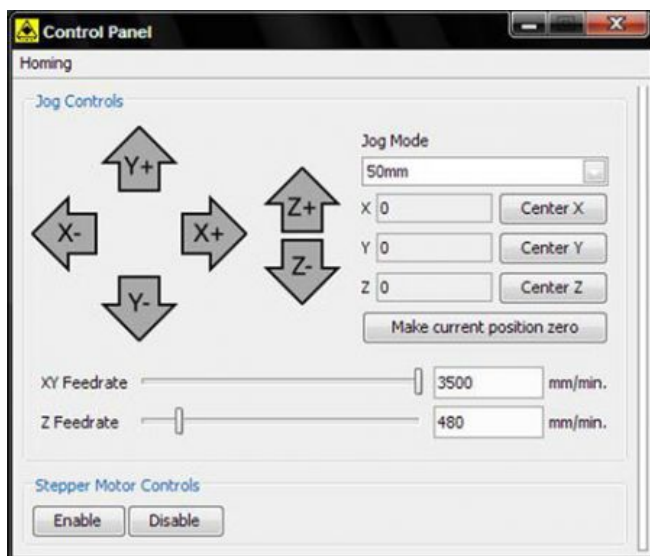


Cliquer ensuite sur le bouton "connect" (avant dernier à droite) afin d'initialiser la connexion avec la machine.

La machine est alors connectée et le bandeau supérieur passe en vert :



L'icone de control panel (les 4 fleches) vous permet ensuite d'accéder à un panneau de contrôle autorisant la commande manuelle



de la machine.

Il y a plus d'explications en images dans la vidéo ci dessus...

## Envoi de programme Gcode en production : solution performante et rapide en python

Comme on l'a vu, si c'est simple de tester du gcode soit en tapant des commandes à la main dans un terminal série, soit en l'envoyant avec ReplicatorG, il va néanmoins nous falloir trouver une solution plus pratique pour la mise en production. Cette solution doit être simple, multiplateforme, et ne nécessiter que peu de connaissance en informatique.

J'ai clairement abandonné le script écrit en Ruby livré avec Grbl. En effet, si Ruby fonctionne nativement sur Mac et Linux, il est très lent sous windows et il n'y a aucune façon simple d'installer la librairie 'serial' nécessaire pour la communication avec l'Arduino. Je suis arrivé au bout, mais quelle galère et quel manque de perf...

Sur le site Contraptor, j'ai trouvé un script nommé 'stream.py' sur [cette page](http://www.contraptor.org/motion-control) (<http://www.contraptor.org/motion-control>) .

Python est natif sur mac et Linux, et assez facile à installer sous Windows. De plus, la librairie serial s'installe très facilement sous Windows puisqu'il s'agit d'un install.exe. Python est donc un bon choix.

Sous Windows, il faut l'installer à partir de <http://www.python.org/download/> (<http://www.python.org/download/>) . Il va également nous falloir installer la librairie serial ( <http://sourceforge.net/projects/pyserial/> ) (<http://sourceforge.net/projects/pyserial/>) ) qui va assurer la communication avec l'Arduino.

Après l'installation sous Windows, les fichiers .py sont automatiquement associés à Python.

J'ai modifié le script stream.py décrit ci dessus afin :

- De supporter le passage d'un paramètre : le nom du fichier à streamer
- De sortir un message d'information quand le fichier n'est pas passé en paramètres
- De revoir les temps de timeout lors de la connexion à l'Arduino, et la boucle qui vide le buffer de réception du port série.
- De demander l'appui sur une touche à la fin du streaming du fichier afin de pouvoir lire le résultat de l'exécution du programme Gcode
- D'ajouter des commentaires et exemples de paramétrage de la variable 'port\_id' pour mieux comprendre comment utiliser ce script sur PC/Mac/Linux
- De traiter le cas d'un gcode non supporté ("error: Unsupported statement"), qui n'était pas traité dans la version initiale. Quand on rencontre une erreur, désormais on continue l'exécution.

Ma version du stream.py peut être téléchargée ici : [stream.py](#). Voici également un petit fichier gcode pour faire des tests : [test.gcode](#)

Avant utilisation, le fichier stream.py doit être paramétré pour refléter le port série sur lequel est branché l'arduino, et le type de plateforme. Tout est dans les commentaires.

Ensuite, il est très facile à utiliser. Sous linux ou mac, il suffit de lancer la commande suivante pour lancer le streaming :

```
stream.py nom_du_fichier_a_streamer.gcode
```

Sur PC c'est désormais encore plus simple puisqu'il suffit de placer le script python sur le bureau, et de drag'n dropper le fichier gcode dessus... (voir vidéo ci dessus). Le script prendra automatiquement en paramètre le fichier Gcode qu'on lui soumet par ce biais. L'attente d'appui d'une touche à la fin permettra de lire le résultat d'exécution avant la fermeture de la fenêtre.

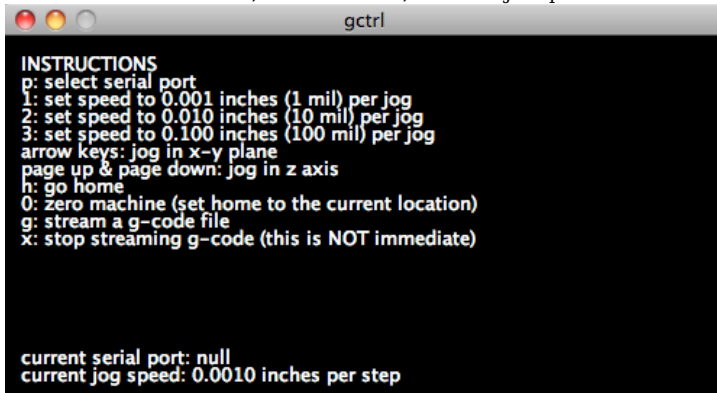
Il y a une démo de fonctionnement sous Linux et Windows dans la vidéo en début d'article.

## Envoi de programme Gcode en production : solution utilisant processing

Jonathan Ward, du MIT, propose une application en Processing pour faire le boulot ([http://mtm.cba.mit.edu/machines/mtm\\_snap-lock/build/software.html](http://mtm.cba.mit.edu/machines/mtm_snap-lock/build/software.html) ). Cela nécessite d'installer tout d'abord Processing, opération simple à partir des archives disponibles sur

l'espace de téléchargement du site éponyme : <http://processing.org/download/> (<http://processing.org/download/>)   
(<http://processing.org/download/>)

L'interface est décevante, en mode texte, aussi ais-je rapidement laissé tombé.




```

gctrl
INSTRUCTIONS
p: select serial port
1: set speed to 0.001 inches (1 mil) per jog
2: set speed to 0.010 inches (10 mil) per jog
3: set speed to 0.100 inches (100 mil) per jog
arrow keys: jog in x-y plane
page up & page down: jog in z axis
h: go home
0: zero machine (set home to the current location)
g: stream a g-code file
x: stop streaming g-code (this is NOT immediate)

current serial port: null
current jog speed: 0.0010 inches per step

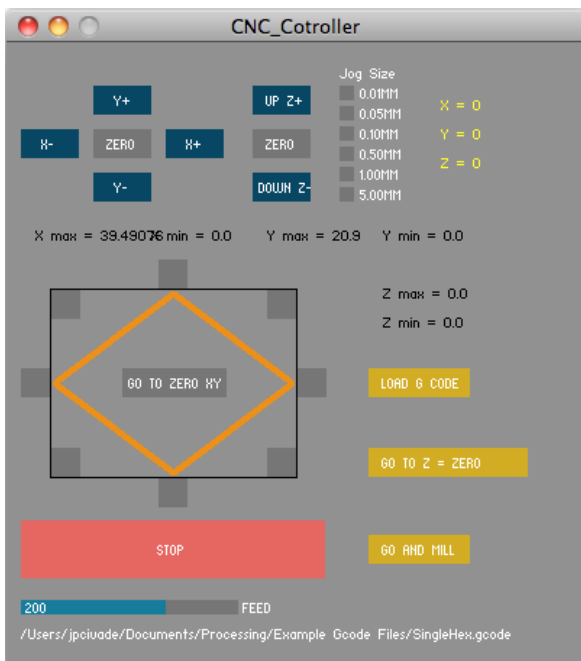
```


Un lecteur du blog, m'a toutefois signalé une article sur une conversion de MF70 réalisée avec GRBL: [http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion.html](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion.html) ([http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion.html](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion.html))  ([http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion.html](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion.html)) . L'auteur a notamment eu la bonne idée de développer un panneau de contrôle également écrit en processing.

Les sources sont disponibles ici: [http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion\\_files/Processing\\_CNC.zip](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion_files/Processing_CNC.zip) ([http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion\\_files/Processing\\_CNC.zip](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion_files/Processing_CNC.zip))  ([http://www.thebox.myzen.co.uk/Hardware/CNC\\_Conversion\\_files/Processing\\_CNC.zip](http://www.thebox.myzen.co.uk/Hardware/CNC_Conversion_files/Processing_CNC.zip))

Celui-ci est assez complet et ergonomiquement plaisant à utiliser. Il dispose de contrôles manuels et d'une fonction de streaming de Gcode.

En voici une copie d'écran :



Sa compilation sous processing ne pose qu'un seul problème : il faut préalablement avoir téléchargé la librairie ControlIP5 à partir de <http://www.sojamo.de/controlIP5> (<http://www.sojamo.de/controlIP5>)  (<http://www.sojamo.de/controlIP5>) , et l'avoir décompressé dans un sous dossier 'libraries' du dossier de projets Processing. C'est cette librairie qui va gérer l'affichage graphique du control panel.

Je ne l'ai pas encore testé en conditions réelles, je le ferai dès que je serai à proximité de ma maquette...

D'ici là, bonne bidouille...

## Commentaires

1. Le mercredi 24 août 2011, 10:17 par maturin.j

bonjour,  
ce qui est clair et précis rend plus intelligent, merci.  
vous me faite entrer dans le monde de l'informatique utile,  
ouvrier a la retraite (maigre), j'ai le projet de construire une cnc

je suit votre travail avec intérêts  
cordialement.

2. Le mardi 25 octobre 2011, 15:20 par zozolabo

Super tuto !  
seulement je n'arrive pas à générer des gcodes compatibles avec streamer.py /... une idée ?

3. Le dimanche 25 mars 2012, 12:04 par julkien

salut  
je developpe actuellement un GcodeSender pour windows  
je serais ravis d'avoir ton avis sur l'aplication  
si tu peux contacte moi

4. Le jeudi 19 avril 2012, 08:31 par [jphi](http://www.civade.com) (<http://www.civade.com>)

Super! Mais à le faire, ne serait il pas intéressant qu'il soit multiplate-forme (PC/Mac/Linux)?

5. Le mardi 26 juin 2012, 17:19 par [Patrick](http://www.prof-mpk.fr) (<http://www.prof-mpk.fr>)  (<http://www.prof-mpk.fr>)

[www.prof-mpk.fr](http://www.prof-mpk.fr)

Bonjour,

Je suis prof de productique dans un lycée professionnel dans le 47, j'ai des élèves de bac pro technicien d'usinage sur commandes numériques.

Pour septembre 2012, j'ai prévu de réaliser une imprimante 3D.

Je commence en ce moment l'étude et les plans.

Caractéristiques prévues:

Dimension du plateaux : 300 x 300.

Déplacement en Z, certainement 300 mm aussi.

Motorisation du plateau axe X et Y avec des moteurs pas à pas.

Motorisation de l'axe Z avec aussi un moteur pas à pas.

Guidage du plateau X et Y avec des axes diamètre 12 mm (minimum) et des douilles à billes.

Axe Z: Guidages linéaires à galets

Moteur pas à pas pour l'extrusion.

Plateau chauffant.

Système de refroidissement après injection.

Le module sur l'axe Z sera interchangeable: Module extrusion pour imprimante 3D ou Module de gravure ou...modelage 3D...

Le châssis sera en profilé "ELCOM"

la transmission sera réalisée soit avec des courroies crantées ou des vis hélicoïdales, le choix n'est pas encore défini.

Le pilotage sera réalisé avec un Arduino...

La partie mécanique est réalisable, mais pour la partie "gestion de l'imprimante", c'est une autre histoire, je connais "Arduino" depuis 3 mois, je me renseigne sur le pilotage de l'imprimante sur les forums.

Je pense que j'aurai besoin d'aide pour cette partie.

Arduino + Grbl + ReplicatorG + carte de gestion des moteurs pas à pas.

Il me reste 2 mois pour dessiner le projet, l'usinage se fera au lycée avec mes élèves.

Le financement est réalisé pas le lycée.

Voilà mon projet, si.. il y a des volontaires pour me donner un coup de main dans la partie "gestion de l'imprimante 3D" je suis preneur, merci d'avance..

Patrick du 47.

[www.prof-mpk.fr](http://www.prof-mpk.fr)

6. Le dimanche 29 juillet 2012, 09:46 par MILLASSEAU

Bonjour.

Tous d'abord merci pour vos recherches d'améliorations des cartes "chinoises"de controle CNC.

Je suis actuellement en train de realiser une CNC. Il me reste l'electronique a faire.

J'ai decouvert sur votre site les petits drivers PUTB6560, mais malheureusement ces derniers ne sont plus en vente (ni les PCB). Avez vous le fichier du PCB ou une photo de ce dernier pour le realiser moi-même ?

Merci d'avance.

Luc

7. Le dimanche 5 août 2012, 08:54 par [jphi](http://www.civade.com) (<http://www.civade.com>)

Je ne suis pas sur que les Putb6560 soient adaptés à un pilotage de CNC. En effet, il n'y a aucune logique permettant de les passer à mi courant lorsqu'il ny a pas de commande. Du coup, au repos, les cartes et les moteurs peuvent chauffer (en fonction du courant nominal des moteurs, mais a 3A je confirme, ca chauffe....)

8. Le lundi 12 novembre 2012, 13:32 par johan

Bonjour,

Pour commencer merci pour toutes ces explications et ce travail. Je suis prof de techno et je rêve de transformer ma salle de techno en mini Fablab pour le collège.

Je commence à travailler sur une imprimante 3 réalisé avec la petite cn Charly Robot que nous avons tous dans nos salle de cours. Votre travail clarifie beaucoup de chose pour moi.

Merci encore

9. Le samedi 1 décembre 2012, 13:40 par Liberty CARECHE

Bonjour à tous,

J'ai développé une application Terminal avec laquelle on peut envoyer du texte par un port USB. Ce programme envoie soit un texte entier, soit caractère par caractère soit ligne par ligne en gestion Xon Xoff. Tout est configurable. Il mémorise les réglages et permet de sauvegarder tout ou partie des transmissions en fichier texte ou Word. La vitesse de transmission va de 110 Bauds à 256 KBauds.

Je vais par ailleurs développer une application CNC. Les suggestions seront les bien venues, ... besoins... disposition.... etc

Pour obtenir l'application terminal, écrivez moi : [liberty.careche@live.fr](mailto:liberty.careche@live.fr)  
Liberty

10. Le samedi 8 décembre 2012, 09:53 par Liberty Careche

Bonjour,

J'ai déposé il y a deux semaines je crois, un message qui a été censuré sans avis par mail...

Je proposais un Terminal que j'ai développé pour Arduino ou toute autre communication par USB... même d'ordi à ordi... Ce terminal est distribuable gratuitement et je le certifie sans malveillance pour autant que les anti virus Windows sachent faire leur travail.

Comme je n'ai pas été élevé dans une PlayStation, je m'attendais à au moins un message informatif expliquant cette censure, et proposant la correction adéquat comme l'exigent les lois informatiques et la politesse tout simplement.

Cordialement  
Liberty

11. Le samedi 8 décembre 2012, 09:59 par Liberty Careche

J'ajoute que ce Terminal est entièrement configurable autant pour Go/NoGo reset à la connexion que la gestion des CR LF est plein d'autres paramètres,... qu'il enregistre les préférence à la fermeture,... qu'il permet d'enregistrer tles transmissions Entrantes et Sortantes au forma text ou Word... etc

Dites moi si je dois rédiger à nouveau le message d'origine, ou tout simplement considérer que ce blog est érigé à votre gloire, sans espoir d'y voir développer un axe positif pour CNC et/ou Arduino. Si la première proposition est la bonne... he bien bravo pour les infos de ce blog,... joli travail,...

Cordialement  
Liberty

12. Le samedi 22 décembre 2012, 09:39 par [jphi](http://www.civade.com) (<http://www.civade.com>)

Désolé pour avoir tardé à publier les commentaires mais mes activités au Fablab de Montpellier (Labsud) me prennent pas mal de temps et je délaisse un peu mon blog.... En tout cas, voila qui est fait.

13. Le jeudi 24 janvier 2013, 00:02 par Chrisalyon

Bonjour,

Moi aussi, je tiens à vous féliciter ! C'est SUPER intéressant, présenté de façon pragmatique, argumentée...

Je me pose juste quelques questions sur les évolutions des différentes solutions présentées, depuis la rédaction de votre article.

Il semble que Grbl en est à la version 0.8C.

On peut télécharger ReplicatorG version 0040.

Avez-vous suivi ces modifications ?

Les réserves que vous faisiez sur votre article (mi 2011 si j'ai bien compris) à propos de Grbl sont elles réglées ?

ReplicatorG a-t-il évolué ou présente-t-il les mêmes problèmes qu'en 2011.

Avez vous testé "Processing\_CNC" dont vous parlez en toute fin de votre article ?

J'espère que vous pourrez me répondre. Cela m'aiderait bien.

Pour info, j'étais parti sur une cnc pilotée par EMC2 et une liaison parallèle de PC portable... Mais il semble que ce n'est pas gagné (je risque de rencontrer pas mal de soucis). Et là dessus, je découvre grâce à vous, les possibilités qu'offre l'Arduino !

Je tourne déjà comme un lion en cage...

A bientôt.

Christophe.

14. Le jeudi 24 janvier 2013, 00:13 par Chrisalyon

Ah oui, j'oubliais: Grbl permet-il de gérer les jeux? Je m'explique: est-il possible de compenser le défaut mécanique notamment des entraînements par tige filetée / écrou en paramétrant un nombre de pas correspondant au jeu mécanique ?

Merci.

Christophe.

15. Le dimanche 27 janvier 2013, 13:00 par [jphi](http://www.civade.com) (<http://www.civade.com>)

Bonjour,

Non, je n'ai pas retravaillé sur la solution depuis. Peut être que Grbl ou d'autres logiciels ont été mis à jour, mais je n'ai pas regardé. Depuis j'ai beaucoup travaillé avec l'impression 3D, notamment avec le firmware Marlin qui justement intègre.... Grbl.

Il y aura aucun moyen de gérer les jeux avec GRBL, seulement d'exécuter du Gcode (et encore pas tout...). Ceci dit, il me semble que l'auteur de Grbl, en se limitant à une petite Arduino, se pose des limites extrêmes. Une copie d'Arduino 2560 se vend 14 euros livrée sur ebay.... A 4 fois plus de mémoire, d'entrées sorties, etc.. Ca vaudrait sûrement le coup d'étendre GRBL pour ajouter des fonctions non supportées par l'interpréteur actuel sans avoir les limitations mémoires notamment.

En tout cas, je suis vraiment surpris de la fiabilité de Grbl tel qu'utilisé dans Marlin... On fait des impressions 3D pendant des heures et ca ne bouge pas d'un poil...

Bonne continuation dans ton projet!

16. Le lundi 11 février 2013, 22:08 par Claude

Bonjour,

Je cherche à faire la même chose que vous .. et j'ai trouvé votre description excellente, maintenant il me reste un problème : je n'ai pas de soucis avec la partie informatique ou électronique, vos explications sont très claires et je commence à bien gérer l'arduino, par injecter du gcode avec un script python pas de soucis ..

mais comment je passe de mon dessin sur le papier au gcode ??? j'exagère un peu .. mais auriez vous un exemple d'applications (mac ou linux et gratuites) pour la conception des objets ?

je cherche principalement à travailler des surfaces 2D, la 3D serait vraiment un plus ..

Merci

Claude

17. Le lundi 25 février 2013, 21:36 par [jckarich](http://objcom.blogspot.com) (<http://objcom.blogspot.com>)

Bonjour,

L'année dernière je me suis lancé sur la conception et fabrication d'une petite CNC et le site m'a été super utile.

J'aimerais bien venir faire un tour au labsud un de ses jours pour partager un peu des expériences car j'ai monté également un kit makerbot il y a deux ans et demi et l'année dernière j'ai commencé à monter une prusa mendel toujours en cours.

Merci!

18. Le lundi 11 mars 2013, 12:46 par [jphi](http://www.civade.com) (<http://www.civade.com>)

A Claude : Une piste de démarrage sur <http://inventables.blogspot.fr/2013/01/makercam-web-based-cam-program.html> ... Ça peut vous suffire. C'est néanmoins un vaste sujet si on veut faire des choses plus compliquées. J'utilise perso Sketchup gratuit (<http://www.sketchup.com/intl/fr/download/>) + plugin open source Sketchup > Cambam ([http://www.cerebralmeltdown.com/cncstuff/page2/cambam\\_export/default.htm](http://www.cerebralmeltdown.com/cncstuff/page2/cambam_export/default.htm)) et une licence que j'ai achetée de ce fantastique logiciel CAMBAM (non, je ne touche rien 😊) : <http://www.cambam.info/>

A jckarich : Pas de pb, nous y sommes tous les mardis soirs à partir de 20h30... Les horaires d'ouverture sont sur le site <http://www.labsud.org>

19. Le lundi 27 mai 2013, 20:53 par picachou

Bonjour, je n'arrive pas à connecter ReplicatorG sur ma carte arduino rev3 pourriez vous m'aider merci

20. Le samedi 19 octobre 2013, 20:01 par yanick

bonjour,

je viens de me relancer dans ma CNC.

j'ai bien réalisé la modification des différents fichiers mais je ne sais pas quoi faire après pourriez vous m'aider.

merci

21. Le lundi 21 octobre 2013, 08:26 par [jphi](http://www.civade.com) (<http://www.civade.com>)

La question n'est pas très claire...

22. Le dimanche 29 décembre 2013, 20:09 par laurent

Salut JP,

Merci pour ton tutoriel qui m'a bien aidé à faire mes premiers pas dans le monde du DIY CNC. Voir : <http://www.youtube.com/watch?v=8Y4G3o99-U>

Je ne suis pas parvenu à flasher ma carte Arduino UNO R3 avec la modif que tu proposes (je suppose que ton post commence à dater un peu). Voici ce qui a fonctionné pour moi dans le Makefile :

```
PROGRAMMER = -C "C:\Program Files\Arduino\hardware\tools\avr\etc\avrdude.conf" -c arduino -P COM5 -b115200
```

Pour les tests, j'ai utilisé ça : <https://github.com/winder/Universal-G-Code-Receiver>

C'est très pratique !

Merci encore !

Laurent.

23. Le dimanche 9 mars 2014, 15:07 par gatién

merci pour le tuto, il est génial !

avec, j'ai noté 2 liens qui m'ont marqué (oui oui je suis grand débutant)

<http://blog.protoneer.co.nz/quick-g-code-interpreter/> et <http://blog.protoneer.co.nz/quick-grbl-setup-guide-for-windows-arduino-g-code-interpreter/>

et <https://www.youtube.com/watch?v=107FGGoYX1bg>

je suis quand même bloqué avec un truc idiot...

j'ai des steppers qui réagissent bien, mais l'axe ne tourne pas... (ça bouge dedans, pas dehors) je me dis que c'est une question de fréquence entre les drivers et les moteurs, mais sans savoir si c'est ça ni comment bidouiller le truc.

matériel: easydriver motor controller de chez sparkfun / 42BYH408AA pour les moteurs... j'ai aussi essayé avec des steppers de vieux lecteurs de disquettes et ça coince autant 😞

si vous avez un remède miracle, ou un tips (je me sentirais bien moins frustré)

encore merci en tout cas, je viens de commencer ma CNC !

24. Le samedi 22 mars 2014, 10:49 par NoNo

Bonjour,

Excellent tuto ! J'ai créé une CNC de précision ( pour graver les pcb et accessoirement en 3d dans du bois et l'alu). J'aurais aimé savoir (dsl si ma question semble bête):

- Vous dites que GRBL n'est pas un sketch normal, si je l'upload sur mon arduino uno cela la rend t'elle inutilisable pour en faire autre chose par la suite ?

- J'ai créé moi même un shield pour mes controleur pas a pas mais n'utilise pas les mêmes pin que GRBL, puis-je modifier ces pin dans GRBL ?

Merci d'avance

[25.](#) Le mercredi 2 avril 2014, 13:46 par [jphi](http://www.civade.com) (<http://www.civade.com>)

Nono: Pas vraiment car les pins doivent être sur le même port.. Lire la doc.

[26.](#) Le mercredi 2 avril 2014, 16:14 par Abderrahim

J'étais perdu, grâce à ton explication, je me retrouve.Merci!

[27.](#) Le dimanche 13 avril 2014, 20:54 par icare

Bonjour,

De très bons billets sur la CNC mais on ne peut pas mettre de commentaires sur "Arduino et GRBL - l'incontournable solution pour piloter une petite CNC".



Y aura-t-il une suite à cet article ?

Merci

### **La discussion continue ailleurs**

URL de rétrolien : <http://www.civade.com/trackback/61>

[Fil des commentaires de ce billet](#)

Propulsé par [Dotclear](#)  | Développé et Hébergé par [100 % Net](#)  | [Mentions Légales](#) | [Contact](#)