



GRBL c'est quoi?

GRBL est un logiciel pour contrôler le mouvement de machines qui font “des trucs”. Si le mouvement [Maker](#) était une industrie, GRBL en serait le standard.

La plupart des imprimantes 3D Open-Source sont basées sur GRBL. Il a été adapté pour être utilisé dans des centaines de projets, comprenant des machines à découpe laser, des écritoires manuels automatisés, perceuses, peintre de graffiti et machines à dessins bizarroïdes... En raison de ses performances, de sa simplicité et de sa frugalité en besoins matériels, GRBL a grossi en un vrai petit phénomène Open Source.

Origine

Vous avez peut-être déjà lu ce nom ici où là, si vous vous intéressez à la CNC amateur et à la manière de les contrôler. GRBL est un contrôleur logiciel libre, open-source, haute performance, écrit en C hautement optimisé pour tourner sur un [Arduino](#) “standard” (Uno).

Le créateur, [Simen Svale Skogsrud](#) explique sur son blog que lorsqu’il commanda sa première fraiseuse pilotée par ordinateur, en 2007, il a été perplexe quant à la manière de la contrôler. La pratique courante à l’époque (qui n’a que peu changée!) était de sauver une vieille boîte beige avec un port parallèle et d’utiliser cela pour envoyer les impulsions aux moteurs. Cela lui apparut juste dépassé. Il voulait un simple système embarqué sur la machine qui dialoguerait en USB avec un ordinateur portable. Ce sont exactement les mêmes raisons qui m’ont poussées à choisir Arduino + GRBL comme solution principale.

Choix du Contrôleur

Simen à choisi l’[Arduino](#) pour plusieurs raisons. La plus importante étant que c’était, et c’est toujours, le système embarqué le plus populaire dans la communauté du DIY (Do It Yourself, Faites-le Vous Même). Un ordinateur d’une simplicité touchante.

Pourtant, c’est une machine chétive au regard de la tâche à effectuer. Ses 2kb de mémoire sont risibles, même par rapport au standard des années 80.

Voici ce que GRBL doit faire avec ces 2Kb:

- - Analyser G-Code, un langage ordinateur cryptique né dans les années 50 pour décrire les

actions idéalisées d'une fraiseuse

- - Construire un modèle de ces actions et les traduire en une séquence de mouvements physiquement possible pour une machine donnée.
- - Exécuter ces mouvements en envoyant un flot continu de pulsions haute fréquence aux moteurs pas à pas qui déplacent effectivement l'outil.

Faire tenir tout ça dans si peu d'espace est un vrai tour de force. Les premières versions de GRBL ont été offertes en Open-Source par Simen en 2009. Depuis 2011, c'est **Sungeun K. Jeon Ph.D.** (@**chamnit**) qui est le leader du projet.

GRBL, pour qui?

Pour ceux qui font du fraisage et ont besoin d'un contrôleur simple pour leur système basé sur l'omniprésent Arduino. Ceux qui haïssent d'encombrer leur atelier avec un PC-Tour juste pour le port parallèle. Bricoleurs/Hackers qui veulent un contrôleur écrit en compact et modulaire C comme base de leur projet.

Fonctionnalités

GRBL est prêt pour la production non intensive. Nous l'utilisons pour tous nos fraisages, depuis Laptop où PCs, utilisant d'excellentes interfaces utilisateur. Il est écrit en C optimisé pour utiliser les fonctions intelligentes des puces Atmega328, pour obtenir un timing précis et des fonctions asynchrones. Il est capable de maintenir un taux de pas supérieur à 30kHz, et délivre un courant propre de pulsations de contrôle.

GRBL est pour l'usinage 3 axes. Pas d'axe de rotation (pas encore!) - Juste X, Y et Z.

L'interpréteur [G-Code](#) implémente un sous-ensemble du standard NIST rs274/ngc et est testé au travers d'un grand nombre d'outils sans problèmes. Mouvements linéaires, circulaires et hélicoïdaux sont supportés sans problèmes.

- G-Codes supporté avec **v0.9i**
 - **G38.3, G38.4, G38.5**: Sondage
 - **G40**: Modes de compensation du rayon de coupe
 - **G61**: Modes de contrôle du chemin de l'outil
 - **G91.1**: Modes Distance d'Arc IJK
- G-Codes supportés dans **v0.9h**
 - **G38.2**: Sondage
 - **G43.1, G49**: Décalage dynamique de longueur d'outils
- G-Codes supportés dans **v0.8** (and **v0.9**)
 - **G0, G1**: Mouvements Linéaires
 - **G2, G3**: Mouvements en Arcs et Hélicoïdaux
 - **G4**: Creuser
 - **G10 L2, G10 L20**: Réglage des décalages du travail
 - **G17, G18, G19**: Sélection de plan
 - **G20, G21**: Unités
 - **G28, G30**: Va en position pré-définie
 - **G28.1, G30.1**: Réglage de position pré-définie
 - **G53**: Mouvement en coordonnées absolues
 - **G54, G55, G56, G57, G58, G59**: Système des coordonnées du travail
 - **G80**: Abandonne le mode de Mouvement

- **G90, G91:** *Modes de distance*
- **G92:** *Décalage des coordonnées*
- **G92.1:** *Efface le décalage du système de coordonnées*
- **G93, G94:** *Modes de taux de débit*
- **M0, M2, M30:** *Pause et termine le programme*
- **M3, M4, M5:** *Contrôle de la broche*
- **M8, M9:** *Contrôle de refroidissement*

La plupart des options de configuration peuvent être réglées en fonctionnement et sauvées en mémoire (eeprom) entre les sessions et même conservées entre différentes versions de GRBL lorsque vous mettez à jour le firmware.

Gestion de l'accélération

Dans les débuts, les contrôleurs [Arduino](#) n'avaient pas de planification d'accélération, et ne pouvaient pas tourner à pleine vitesse sans difficultés. La gestion constante de l'accélération par GRBL avec son planificateur d'avance a résolu cette issue et a été répliquée partout dans le monde du micro-contrôleur CNC. GRBL utilise intentionnellement un planificateur d'accélération constante simplifié, largement suffisante pour une utilisation amateur. Grâce à cela, le temps de développement est utilisé sur les algorithmes de planification et sur l'obtention de la certitude que nos mouvements sont solides et fiables.

Si vous voulez en savoir plus sur GRBL et les fonctionnalités des dernières versions, je vous invite à [vous reporter au Wiki original](#), dont cet article est une traduction partielle.