Grbl doc en français

Intro

Grbl est une alternative sans compromis, haute performance et économique au contrôle de mouvement basé sur un port parallèle pour le fraisage CNC. Il fonctionnera sur un Arduino (Duemillanove / Uno) tant qu'il porte un Atmega 328.

Le contrôleur est écrit en C hautement optimisé utilisant toutes les fonctions intelligentes des puces AVR pour obtenir une synchronisation précise et un fonctionnement asynchrone. Il est capable de maintenir jusqu'à 30 kHz d'impulsions de contrôle stables et sans gigue.

Il accepte le code g conforme aux normes et a été testé avec la sortie de plusieurs outils de FAO sans problème. Les arcs, les cercles et les mouvements hélicoïdaux sont entièrement pris en charge, ainsi que toutes les autres commandes g-code primaires. Les fonctions macro, les variables et la plupart des cycles prédéfinis ne sont pas supportés, mais nous pensons que les interfaces graphiques peuvent faire un bien meilleur travail en les traduisant de toute façon en code g.

Grbl inclut une gestion complète de l'accélération avec anticipation. Cela signifie que le contrôleur examinera jusqu'à 18 mouvements dans le futur et planifiera ses vitesses à l'avance pour offrir une accélération en douceur et des virages sans à-coups.

Versions de GRBL

Versions de GRBL

G-Codes pris en charge dans v1.1

- G0, G1: Motions linéaires
- G2, G3: Mouvements d'arc et d'hélice
- G4: Habiter
- G10 L2, G10 L20: Définir les décalages de coordonnées de travail
- G17, G18, G19: Sélection du plan
- G20, G21: Unités
- G28, G30: Aller à la position prédéfinie
- G28.1, G30.1: Définir une position prédéfinie
- G38.2: Sonder
- G38.3, G38.4, G38.5: sondage
- G40: Modes de compensation du rayon du cutter OFF (Uniquement)
- G43.1, G49: Décalages de longueur d'outil dynamique
- G53: Déplacer en coordonnées absolues
- G54, G55, G56, G57, G58, G59: Systèmes de coordonnées de travail
- G61: Modes de contrôle de chemin
- G80: Annuler le mode mouvement
- G90, G91: Modes de distance
- G91.1: Modes de distance de l'arc IJK
- G92: Décalage de coordonnées

- G92.1: Effacer les décalages du système de coordonnées
- G93, G94: Modes d'avance
- M0, M2, M30: pause et fin du programme
- M3, M4, M5: Contrôle de la broche
- M7 * , M8, M9: Contrôle du liquide de refroidissement
- M56 * : Commande de neutralisation de mouvement de stationnement

(*) indique les commandes non activées dans config.h par défaut.

Doc utilisateur

Utiliser Grbl en anglais, mais avec images

GRBL c'est quoi?

Astuces idees

Config GRBL

config

Utiliser Grbl

zeevy a modifié cette page 12 days ago · 24 révisions

Cette page wiki est destinée à fournir diverses instructions sur l'utilisation de Grbl. N'hésitez pas à contribuer et à maintenir cette page à jour! Mise en route (pour les nouveaux utilisateurs.)

Après avoir clignoté Grbl sur votre Arduino, la connexion à Grbl est assez simple. Vous pouvez utiliser l'IDE Arduino lui-même pour vous connecter à Grbl. Expérimentez ou jouez avec, juste pour voir si vous l'aimez. Les autres programmes de port série, tels que CoolTerm ou PuTTY, fonctionnent également très bien. Les instructions sont à peu près les mêmes.

```
  Ouvrez l'IDE Arduino et assurez-vous que votre Arduino avec Grbl est
connecté à votre port USB.
  Sélectionnez le port série de l'Arduino dans le menu Outils, comme vous le
feriez normalement avec un Arduino.
  Ouvrez la «fenêtre de série» dans le menu Outils.
  Si vous utilisez Grbl v0.9 ou ultérieur, assurez-vous de changer la
vitesse de transmission de 9600 à 115200.
  Une fois ouvert, vous devriez voir un message de bienvenue Grbl comme Grbl
v0.Xx ['$' for help] . Cela signifie que tout est bon! Vous êtes connecté!
  Assurez-vous de changer le menu déroulant "Pas de fin de ligne" en "Retour
chariot". Si vous utilisez un autre programme de port série, vous devez
faire de même.
  Si vous n'avez pas reçu le message de bienvenue ou certains caractères
tronqués, assurez-vous que le débit en bauds est réglé sur 9600 (ou 115200
```

```
pour v0.9 +).
```

De là, vous pouvez simplement commencer à envoyer des commandes G-code à Grbl, et il les exécutera pour vous. Ou, vous pouvez taper $ pour obtenir de l'aide sur certaines des commandes spéciales de Grbl ou comment écrire certains paramètres de votre machine dans la mémoire EEPROM de Grbl.

Lorsque vous avez commencé à vous sentir à l'aise avec G-code / CNC et que vous êtes prêt à exécuter un programme G-code complet, nous vous recommandons d'utiliser l'une des nombreuses interfaces graphiques que les utilisateurs ont écrites pour diffuser vos programmes G-code. et exploiter pleinement toutes les capacités de Grbl.

NOTE: Consultez le wiki de ShapeOko. Il a la liste la plus à jour et complète des interfaces graphiques Grbl. Comment diffuser des programmes G-Code à Grbl [bCNC] (https://github.com/vlachoudis/bCNC)

Capture d'écran bCNC

Un expéditeur de code-g avancé complet pour GRBL. bCNC est un programme multiplateforme (Windows, Linux, Mac) écrit en python avec des dépendances externes minimes. L'expéditeur est robuste et rapide capable de fonctionner correctement avec du matériel ancien ou lent comme Rasperry PI (comme il a été validé par le mainframe GRBL sur les tests lourds).

Caractéristiques:

```
  interface simple et intuitive pour les petits écrans
  importation de fichiers g-code et dxf
  expéditeur rapide de code g (fonctionne bien sur RPi et ancien matériel)
  configuration de l'espace de travail (boîte de dialogue pour les commandes
G54..G59)
  boutons configurables par l'utilisateur
  Évaluation de la fonction g-code avec extension du temps d'exécution
  Sondage facile:
      sondage simple
      Centre de recherche avec un anneau de palpage
      mise à niveau automatique , sondage Z et mise à niveau automatique en
modifiant le code g lors de l'envoi.
      affichage de la carte de couleur en hauteur
      expansion manuelle de changement d'outil et sondage automatique de
longueur d'outil
  Divers outils:
      base de données configurable par l'utilisateur des matériaux, des
fraises, du stock
      base de données des propriétés des matériaux, stock, fraises, etc.
      Fonctions de FAO de base (profilage, empochage, perçage)
      Plugins g-code utilisateur:
          générateur de bol
          générateur de boîte de joint de doigt
          générateur de pignon droit simple
          générateur de spirograph
          surface aplatie
          ...
  Éditeur de code G et affichage
```

```
      affichage graphique du code g, et espace de travail
      déplacer et éditer graphiquement g-code
      code de réorganisation et optimisation rapide du mouvement
      déplacement, rotation, mise en miroir du code g
  pendentif web à utiliser via des téléphones intelligents
```

Expéditeur universel de code G (UGS) [Java multiplate-forme]

Fenêtre principale UGSVisualiseur UGS

Une interface graphique complète, développée par @wwinder, qui diffuse, visualise le code G, et possède une fonctionnalité complète de contrôle et de retour d'informations pour les fonctionnalités de niveau supérieur de Grbl. Il est écrit en Java, ce qui signifie qu'il peut être exécuté sur n'importe quelle machine compatible Java, y compris le RaspberryPi! Le groupe Grbl travaille en étroite collaboration avec ce projet et recommande vivement d'utiliser cette interface graphique. Si vous rencontrez des problèmes ou si vous souhaitez demander plus de fonctionnalités de l'interface graphique, @wwinder a fait un excellent travail en remplissant de nombreuses demandes. Chevalet [Contrôleur CAD + CAM + Grbl sur navigateur]

Capture d'écran de chevalet

Easel est un projet Web développé par Inventables spécifiquement pour X-Carve, Carvey + Grbl. Il s'agit d'un package tout-en-un pour la conception (y compris les importations SVG), la génération de parcours et le contrôle des machines. Il a également un magasin d'applications où les développeurs tiers peuvent faire des applications qui peuvent importer dans Easel. La documentation pour créer une application est ici (https://discuss.inventables.com/c/easel/app-development). En plus des outils de conception 2D, Easel vous permet de prévisualiser vos parcours en 3D avant de les envoyer à votre machine. Vous pouvez également importer G-Code dans Easel et l'utiliser comme expéditeur. Le chevalet est en développement constant par l'équipe Inventables. Vous pouvez demander des fonctionnalités ou signaler des problèmes via le bouton de commentaires dans l'application ou sur le forum Inventables www.inventables.com/forum . GRBLweb [Navigateur Web]

GRBLweb ui

GRBLweb est un expéditeur et contrôleur GCODE basé sur le Web pour GRBL. Plusieurs périphériques série peuvent être connectés pour contrôler plusieurs machines.

Il y a aussi une image de Raspberry Pi pré-construite basée sur Ragrbian exécutant GRBLweb disponible ici . GrblPanel [Windows]

GrblPanel

Géré par un informaticien à la retraite @gerritv, GrblPanel est une interface graphique qui implémente les fonctions et fonctionnalités les plus avancées des machines de production telles que Haas, Fanuc, etc. Tous les outils nécessaires pour configurer et exécuter un travail de fraisage sont soigneusement organisés et conçus pour être facilement accessible sur la base des flux de travail acceptés depuis des décennies dans les ateliers d'usinage. GrblPanel ne fonctionne actuellement sous Windows via .Net v4.5, mais sera finalement mis à jour pour une utilisation multiplateforme via Mono . grblControl [Windows / Linux]

screenshot_heightmap_original

Application GUI pour machines CNC à base de GRBL avec visualiseur G-Code.

Fonctions supportées:

```
  Contrôler une machine cnc basée sur GRBL via les commandes de la console,
les boutons sur le formulaire, le pavé numérique.
  Surveillance de l'état de la machine cnc
  Chargement, édition, sauvegarde et envoi de fichiers G-code à cnc-machine.
  Visualisation des fichiers G-code.
  Axe Z autoleveling pour fraisage PCB.
```

SourceRabbit GCode Sender [Java multiplate-forme]

screenshot_heightmap_original

SourceRabbit GCode Sender est un expéditeur multi-plateforme G-Code compatible GRBL écrit en Java. Il dispose d'une interface utilisateur hautement optimisée et asynchrone (pilotée par les événements) et d'une communication USB vers série. Il peut également être utilisé sur des ordinateurs dotés d'une petite quantité de RAM et de CPU. Scripts de streaming Python (Officiellement supporté par Grbl) [Multi -plateforme]

REMARQUE: Si vous rencontrez des difficultés avec le streaming sur Grbl, nous vous demanderons d'utiliser ce script de diffusion en continu Python pour éliminer l'interface graphique que vous utilisez comme source du problème. Avant de poster sur le sujet des problèmes, veuillez utiliser ce script pour exécuter votre programme G-code.

Inclus avec le code source et officiellement supporté par Grbl, deux scripts de streaming Python sont fournis pour illustrer des méthodes de streaming simples et plus complexes qui fonctionnent bien multiplate-forme. Ces scripts ne prennent pas entièrement en charge toutes les fonctionnalités de Grbl, mais sont plutôt conçus comme un moyen de comparer ou de dépanner d'autres variétés de jardin ou d'interfaces graphiques récemment écrites. Ceux-ci se trouvent dans le dossier 'script' du dépôt principal. Remarque: Les scripts de diffusion en continu nécessitent l' installation du module pySerial .

```
  Installez le module pySerial .
  Télécharger le script Python simple_stream.py .
  Ouvrez le script dans un éditeur de texte brut et modifiez la ligne
suivante pour refléter votre système:
```

s = serial.Serial('/dev/tty.usbmodem1811',9600)

```
  Au lieu de /dev/tty.usbmodem1811 (Mac), vous devez mettre le nom de
périphérique du port série de votre Arduino. Ce sera différent pour chaque
machine et système d'exploitation. Par exemple, sur un système Linux, cela
ressemblerait à / dev / ttyACM0 . Ou sur une machine Windows, cela peut
ressembler à COM3 .
  Le script recherche et lit gcode à partir d'un fichier nommé grbl.gcode ,
vous devez créer ce fichier et y mettre le gcode que vous voulez exécuter.
Ou simplement changer ce nom dans le script selon vos besoins.
  Ouvrez une fenêtre de terminal / commande et changez les répertoires à
l'emplacement du script Python et exécutez le script Python avec la commande
suivante:
```

./simple_stream.py (Mac / Linux) python simple_stream.py (Windows)

```
  Vous devriez maintenant voir le code g en train d'être transmis à grbl
avec les messages 'ok' et votre machine devrait commencer à bouger.
```

L'autre, stream.py script de streaming plus avancé a des arguments de ligne de commande et ne nécessite pas de modifier le script lui-même, contrairement à simple_stream.py . La principale différence est que stream.py utilise un schéma de comptage de caractères pour s'assurer que le tampon de lecture en série de Grbl est plein, ce qui crée effectivement une autre couche tampon au-dessus de la file d'attente interne de Grbl. Ceci permet à Grbl d'accéder et d'analyser le prochain bloc de code G immédiatement à partir du tampon de lecture en série, plutôt que d'attendre l'envoi et la réponse 'ok' dans le script simple_stream.py . Ceci est très utile pour les mouvements, comme les courbes, qui ont des segments de ligne courts très rapides qui peuvent provoquer une insuffisance de mémoire tampon, ce qui peut conduire à des étranges mouvements de mouvement. En d'autres termes, il assure un mouvement plus lisse. Utilisez ce script, si vous n'avez pas peur de la ligne de commande ou si vous éprouvez des mouvements étranges. Autres interfaces graphiques grblUI

Une interface utilisateur graphique simple: https://github.com/jgeisler0303/grblUI . Programmé en Java, en utilisant rxtx pour la communication série. Devrait théoriquement fonctionner sous Linux, Mac et Windows. Apparemment quelques problèmes sur Mac. Tous les commentaires, astuces et astuces appréciés (Problèmes ou Wiki dans grblUI). Découvrez le pot prêt à l'emploi dans les téléchargements. grblgui

Un Streamer G-Code graphique: https://github.com/cody82/grblgui . Programmé en Java, en utilisant rxtx pour la communication série et OpenGL 2.0 pour le rendu.

Caractéristiques notables:

```
  Il affiche la durée du travail et le temps restant à compléter en minutes.
  Il affiche la vitesse actuelle.
  Vous pouvez basculer le maintien du flux et entrer les commandes G-Code.
  Il affiche graphiquement l'état du tampon sur le parcours d'outil!
```

En développement:

```
  Simulez le processus de fraisage et affichez le modèle résultant.
```

CNCinfusion [Windows]

Actuellement en cours de développement en C # https://github.com/nm156/CNCInfusion Gcode Sender [Windows]

https://github.com/downloads/OttoHermansson/GcodeSender/gcodesender.exe
http://www.contraptor.org/forum/t-287260/gcode-sender-program LaserGRBL [Windows]

GUI optimisé au laser pour Grbl http://lasergrbl.com

```
  Interface simple et minimale conçue pour Grbl v1.1
  jpg, bmp, png Importation d'image (vectorisation d'image, niveaux de gris,
tramage 1bit)
  Aperçu graphique 2D pour la gravure / découpe (avec cartographie en
```

```
niveaux de gris)
  Commande de remplacement facile à utiliser
  Boutons définis par l'utilisateur, pouvoir pour vous!
  Configuration de Grbl Import / Export
  Décodage des codes de configuration, d'alarme et d'erreur pour Grbl v1.1
(avec info-bulle de description)
  Bouton de prise d'origine, bouton d'attente d'alimentation, bouton de
reprise et bouton de réinitialisation Grbl
  Aperçu du temps de travail et projection en temps réel
  Jogging (pour toute version Grbl)
  Remplacements de flux (pour Grbl> v1.1) avec interface facile à utiliser
  Multilingue: anglais, italien, allemand, français, espagnol, danois et
brésilien
```

LaserGRBL_preview

GRBL-Traceur [Windows]

https://github.com/svenhb/GRBL-Plotter

```
  Soutenir GRBL 1.1 (et 0.9 également)
  Exporter / importer les paramètres spécifiques à la machine (joystick,
boutons)
  Contrôler un 2ème matériel GRBL
  Echange d'outils
  Boutons définis par l'utilisateur
  Joystick comme contrôle
  Importation / création et conversion en GCode
      de SVG Graphics
      du texte (en Hershey Font)
      conversion de la Z-Dimension en axe Z (routeur) ou broche on / off
(laser) ou broche-vitesse (RC-Servo PWM)
```

Grbl Overseer [Multiplatform Desktop + Android]

Interface utilisateur conviviale avec plusieurs tâches de planification

```
  Interface utilisateur simple, conviviale et conviviale
  Vue 3D des travaux et de la position actuelle de l'outil
  Planifier plusieurs tâches, chacune a son origine spécifiée
  Exécute automatiquement une simulation avant la production et compile
toutes les erreurs
  Commandes de jog adaptatives (plus vous appuyez longtemps, plus vite ça
va)
  Console série intelligente, les messages / réponses GRBL sont regroupés
avec la commande correspondante
  Barre supérieure intelligente, indiquant toujours le statut actuel du
GRBL. La couleur d'arrière-plan change avec l'état pour permettre une
lecture facile de l'état même loin de l'appareil
  Éditeur intégré pour la configuration GRBL
  Multiplateforme (testé sous Windows, Linux et Android)
```

```
  Prise en charge de l'interface USB / série sur les appareils Android 3.1
avec API USB
  Support Grbl> = v1.1
```

Capture d'écran :

Capture d'écran de Grbl Overseer

Le développement est toujours en cours, veuillez signaler tout problème que vous rencontrez

Grbl Controller [Application mobile Android]

Interface utilisateur compacte même pour les petits écrans mobiles

```
  Support Grbl> = v1.1
  Prend en charge les remplacements en temps réel, le débit d'alimentation,
la vitesse de broche et le liquide de refroidissement à bascule.
  Position de la machine en temps réel, alimentation, état des tampons.
(vous devez activer les données du tampon dans le rapport d'état via $ 10 =
2)
  Supporte l'envoi de fichiers G-Code depuis un mobile. (Les extensions
supportées sont .gcode .nc et .tap)
  Prend en charge les commandes de texte court.
  Auto ajuste l'axe Z sur la surface de travail en utilisant le sondage
G38.3.
  4 boutons personnalisés hautement configurables.
```

Contrôle de l'axeFichier en streamingSortie de la consoleParamètres Émulateurs série:

Autre que CoolTerm ou PuTTY, les systèmes Linux et Mac ont un grand émulateur en série appelé screen qui est soit intégré ou facilement installable (apt-get install screen) via l'interface du terminal.

Si votre appareil est connecté sur / dev / ttyACM0 (pour Mac, /dev/tty.usbxxxx), tapez screen /dev/ttyACM0 115200 pour vous connecter à l'appareil à 115200 bauds. Là, vous serez connecté à Grbl. Pour quitter

Pour sortir de l'interface écran, appuyez simplement sur Ctrl-a suivi d'un k .

Contrôleur Diba [windows]

capture d'écran

Application GUI pour GRBL 0.9J avec visualiseur 2D G-Code. Contactez-nous: diba.team2017@gmail.com

download: Contrôleur Diba 2.5.01

Zen Toolworks Controller (Zen CNC) (En développement important)

Contactez-nous: info@zencnc.com

Cette application est actuellement disponible uniquement sur Windows Platform (Windows 7 et 10 testé)

L'interface graphique fournit des fonctions très basiques, telles que la lecture numérique, la position définie sur et le déplacement vers. Jogging X, Y et Z, Zéro et Homing. Easy Grbl Modification de paramètres.

Toutes les autres fonctions sont basées sur l'architecture des plugins. Vous pouvez étendre la fonctionnalité en définissant vos propres panneaux et boutons.

Zen Toolworks a également développé une carte de test Grbl, qui peut être utilisée en combinaison avec notre application Zen CNC pour tester toutes les fonctionnalités fournies par Grbl.

Page wiki de ZenCNC

Contrôleur Grbl

Options de l'écran principal du contrôleur Grbl

Nom du port

Ce menu déroulant sera rempli par les ports série candidats. Seuls certains ports sont des ports série réels connectés à Grbl / Arduino.

Le plus simple est de regarder la liste avant de brancher votre connecteur USB (en supposant que vous utilisez un port série USB), fermez le contrôleur Grbl, branchez le connecteur USB, ouvrez le contrôleur Grbl et regardez à nouveau la liste. L'élément qui est nouveau est le port COM souhaité.

Ouvrir - Fermer / Réinitialiser

Ouvrez ou fermez le port COM. L'ouverture du port COM réinitialisera Grbl et définira les coordonnées à 0,0,0. Vous verrez un message de version de Grbl apparaître dans la fenêtre d'état quelques secondes après avoir appuyé sur ce bouton.

La fermeture du port COM a pour effet d'arrêter la dernière opération sur le Grbl. Ce n'est pas un substitut à un arrêt d'urgence approprié, mais c'est un bon moyen d'arrêter rapidement une action. Si aucun interrupteur de fin de course n'est connecté, il n'y a aucune garantie que les moteurs s'arrêteront à temps lorsque vous appuyez sur Fermer / Réinitialiser.

Sur certains systèmes Linux, Grbl Controller doit être démarré avec sudo pour obtenir l'autorisation d'ouvrir le port COM.

Choisir le fichier

Ouvre une boîte de dialogue de sélection de fichier pour choisir votre fichier à envoyer à Grbl. Sur la plupart des systèmes, vous pouvez choisir un fichier pendant l'ouverture du port COM (pour gagner du temps).

Commencer

Va commencer à envoyer le fichier dès que pressé. Si Grbl Controller se connecte toujours avec Grbl, le fichier sera envoyé immédiatement après la fin de l'initialisation.

Arrêtez

Le fichier envoyé sera arrêté en cours de flux. Actuellement, il n'y a aucun moyen de redémarrer un envoi de fichier arrêté.

Si vous avez activé le préchargement agressif dans les options, appuyer sur Stop peut entraîner une attente plus longue pour l'état arrêté en raison des commandes déjà mises en file d'attente dans Grbl.

Barre de progression et durée d'exécution

La barre de progression affiche un pourcentage de lignes envoyées à partir du fichier. L'affichage Runtime affiche en temps réel le nombre d'heures, de minutes et de secondes que le fichier a été envoyé. Le Runtime s'arrête lorsque le fichier a été complètement envoyé, indiquant ainsi combien de temps il a fallu pour envoyer le fichier.

GCode manuel / Commande

Entrez les commandes GCode à exécuter immédiatement.

Il a été observé que Grbl entre parfois dans un état où la spécification de simples commandes X1, Z30, etc. ne fonctionne pas. Vous devez d'abord préface avec un G0 ou un G1. Une fois cela fait, il devrait être possible d'envoyer d'autres commandes avec juste la commande X, Y ou Z directement.

Appuyez sur la touche Entrée pour accepter la commande.

La position saisie sera interprétée en pouces ou en millimètres selon le mode spécifié dans les options

(Grbl v0.8c +).

Machine / travail coordonne LCD

Les coordonnées de la machine sont les coordonnées X, Y, Z référencées à partir du point d'ouverture du port COM vers Grbl. Les coordonnées de la machine sont principalement utiles pour le référencement.

Les coordonnées de travail sont les coordonnées X, Y, Z qui peuvent être réinitialisées à 0 si vous le souhaitez et toutes les commandes référenceront les nouvelles coordonnées de travail.

Grbl Controller, à partir de la version 3.1, affiche les coordonnées de Grbl sur les écrans LCD. Il ne calcule jamais ces valeurs à partir de commandes entrées. Il tente de fournir des valeurs raisonnablement à jour d'au moins une fois par seconde lors de l'envoi de fichiers.

Lors de la mise à zéro des coordonnées, il est normal que les coordonnées de travail soient désactivées par une minuscule fraction de 0, telle que 0,001.

Si vous avez activé le mode “pouces” dans les options, l'affichage indiquera pouces, sinon, il affiche normalement des millimètres.

Dernier état

Ceci affiche le dernier état opérationnel connu de Grbl. Cela va normalement montrer Idle ou Run, mais peut contenir d'autres états. Seulement disponible avec Grbl 0.8c +. Axe Ajuster / Jog et Taille de l'étape

Utilisez les boutons pour déplacer les moteurs de la quantité relative, spécifiée par la valeur de la taille de pas.

Si vous avez activé le mode “pouces” dans les options, la taille du pas sera interprétée en pouces, sinon, elle sera interprétée en millimètres.

Coordonnées absolues après ajustement

Si cette option est activée, lorsque vous entrez une commande dans l'entrée Manual GCode, les valeurs sont interprétées en coordonnées de travail absolues.

Par exemple, si vous souhaitez déplacer les coordonnées X à 10 et que l'écran affiche 2, entrez X10 pour votre commande.

Si elle est désactivée, lorsque vous entrez une commande dans l'entrée Manual GCode, la valeur sera relative aux coordonnées actuelles.

Par exemple, si vous voulez déplacer les coordonnées X à 10 et que l'affichage indique 2, entrez X8 pour votre commande.

Le mode que vous choisissez dépend de la manière dont vous préférez travailler avec Grbl Controller.

Broche sur

Active ou désactive le moteur de broche, à condition que vous ayez cette capacité câblée dans votre carte et à un contrôleur de moteur externe. Réinitialiser le contrôle - Soft Reset Grbl

Cela envoie une commande de réinitialisation spéciale à Grbl qui provoque le redémarrage et la réinitialisation. Cela n'affecte pas les coordonnées de la machine, mais elle réinitialise les coordonnées de travail pour qu'elles correspondent aux coordonnées de la machine. Utilisez la position zéro pour restaurer les coordonnées de travail à 0.

Reset Control - Débloquer Grbl

Il existe certaines conditions dans lesquelles Grbl obtient des commandes en conflit ou entre dans un état d'erreur. Si cela se produit, il se verrouillera pour empêcher les commandes de causer des dommages à l'équipement.

Utilisez ce bouton pour déverrouiller Grbl, mais seulement après vous être assuré que le broyeur CNC est en bon état de fonctionnement.

Position zéro

Rétablit simplement les coordonnées de travail à 0, mais est fréquemment utilisé pour restaurer le fonctionnement à partir d'une nouvelle base de coordonnées, par exemple après avoir aligné la broche / le bit sur ce qui est considéré comme la position 0,0,0 sur la pièce.

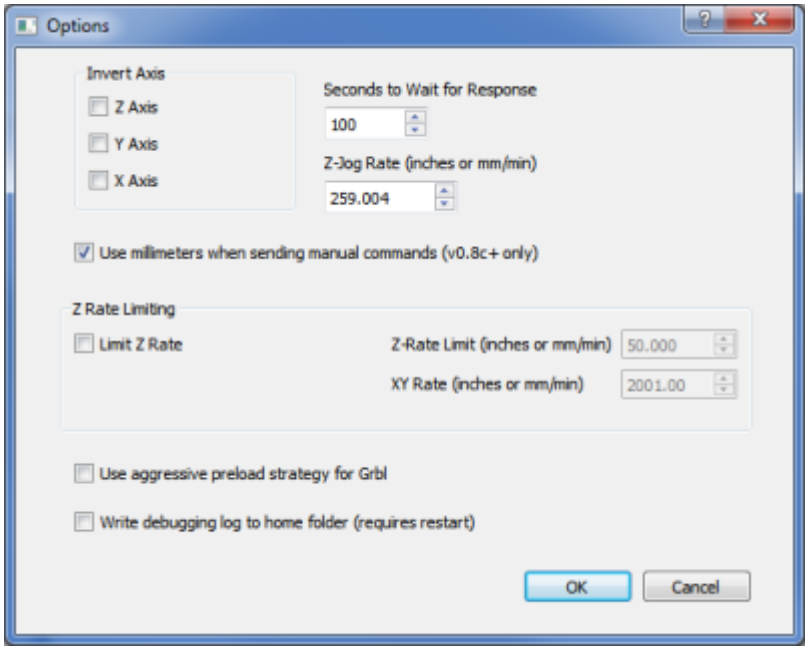
Go Home

Après l'exécution d'un fichier, ceci est utilisé pour déplacer la broche vers le haut et au-dessus de la pièce sur X = 0 et Y = 0. La coordonnée Z sera supérieure de trois mm au point Z le plus élevé pendant l'envoi du fichier.

Paramètres Grbl

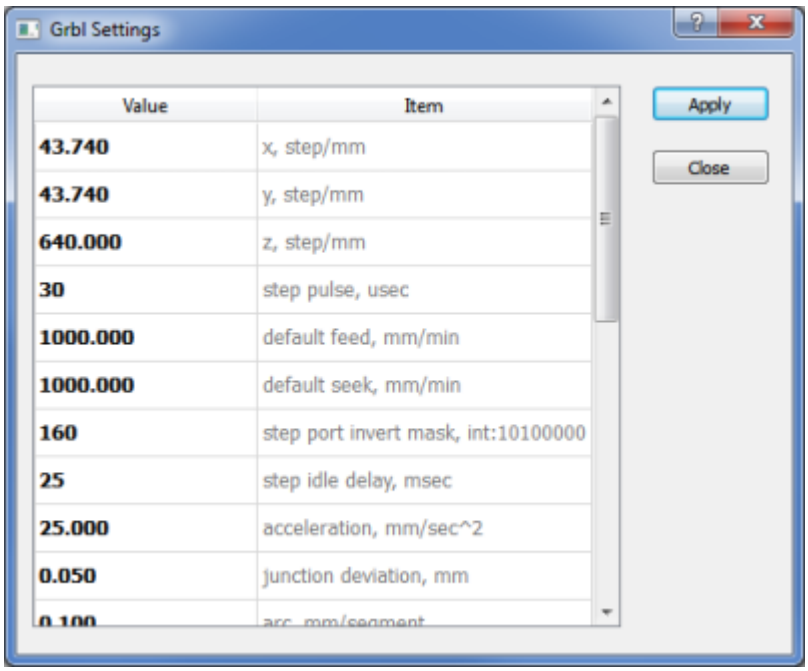
Cela ouvre une boîte de dialogue qui est remplie dynamiquement avec les paramètres réglables disponibles dans le logiciel Grbl. Le nombre de paramètres varie avec la version de Grbl. Tous les paramètres modifiés seront envoyés à Grbl et mettra à jour sa mémoire non volatile.

Options du contrôleur Grbl



Inverser l'axe X / Y / Z

Ces cases à cocher sont fournies dans le cas où vous souhaitez inverser rapidement un axe spécifique. Il est recommandé d'utiliser le “masque d'inversion de port” dans les paramètres Grbl à la place:



Secondes à attendre une réponse

Un envoi de fichier sera annulé si Grbl ne répond pas après ce nombre de secondes. Normalement, vous voulez que ce nombre soit assez élevé, plus de 60 secondes. Un cas particulier où vous pouvez augmenter ce nombre est si vous avez un Shapeoko avec des rails de mètre et faites une traverse lente d'un bout à l'autre.

Vitesse Z-Jog (pouces ou mm / min)

C'est la vitesse d'avance utilisée lorsque vous cliquez sur les boutons haut ou bas pour faire défiler l'axe Z.

260 mm / min est un bon taux de défaut. Ce champ est automatiquement converti si vous passez en mode “pouces”.

Cela n'affecte que les commandes Z-jog. Cela n'affecte pas l'entrée manuelle du code.

Utiliser des millimètres lors de l'envoi de commandes manuelles (v0.8c + uniquement)

Si vous utilisez une version antérieure de Grbl telle que v0.8a, laissez cette option cochée. Sinon, si vous utilisez 0.8c ou plus, vous pouvez choisir de décocher cette option et ensuite toutes les mesures seront en pouces.

Cela inclut les écrans LCD, les commandes données, les pas de pas, etc.

Remarque: Si un fichier est spécifié en pouces et que mm est votre mode, Grbl Controller vous ramènera automatiquement à mm une fois le fichier terminé. L'inverse est également vrai.

Limitation du débit Z

Lors de l'envoi d'un fichier, si la fonction Z-Rate Limiting est activée, toutes les commandes du fichier pour l'axe Z reçoivent un taux d'avance (ou le taux est modifié). L'affectation a lieu au moment de l'exécution. le fichier réel n'est pas affecté.

Limite de taux Z

Si la limitation du débit Z est activée, cette valeur modifie toute commande de l'axe Z du fichier si le débit Z du fichier est supérieur à la limite. Si aucun taux n'est spécifié, ce taux sera attribué. Si la commande de l'axe Z est G0, elle est convertie en G1.

Taux XY

Si un taux Z est spécifié sur la même ligne avec des taux X et / ou Y, la commande est divisée en deux commandes, Z et Y. La commande Z obtient le taux Z tandis que le XY obtient cette valeur de taux XY. Puisque la commande de débit est modale, si une commande XY suit une commande Z sans débit, cette valeur de débit XY est ajoutée à cette commande.

Il est recommandé de définir cette valeur sur la vitesse nominale XY trouvée dans le fichier. Cela peut généralement être trouvé dans la commande XY immédiatement après une commande Z contenant un débit. Ce n'est pas obligatoire, mais un taux inférieur entraînera un achèvement plus lent du travail et un taux plus rapide pourrait être préjudiciable à la coupe.

Utiliser une stratégie de précharge agressive pour Grbl

Si elle n'est pas cochée, Grbl Controller attendra une réponse après chaque commande du fichier avant d'envoyer la commande suivante. Si elle est cochée, elle enverra autant de commandes que nécessaire dans le buffer Grbl, puis attendra une réponse ok / error, après quoi des commandes supplémentaires seront placées dans la mémoire tampon, assurant que la mémoire tampon est toujours au maximum de la capacité possible.

L'activation de la précharge agressive active efficacement la fonction de «gestion de l'accélération» de Grbl pour optimiser les vitesses entre chaque commande afin d'assurer des transitions entre les commandes. J'ai observé des gains de vitesse de 400-600% avec cette fonctionnalité activée car les accélérations et les décélérations qui ont lieu avec le mode normal n'existent plus dans l'axe XY.

Il est fortement recommandé d'effectuer d'abord un essai à sec avec cette fonction activée pour s'assurer que les vitesses sont dans les valeurs attendues pour le trépan, la vitesse du moteur et le matériau fraisé. En effet, les vitesses tendent à être plus élevées que la caractéristique désactivée, ce qui entraîne une charge moyenne plus élevée sur le bit.

Si vous essayez d'appuyer sur le bouton d'arrêt, vous ne verrez pas de réponse immédiate car normalement, jusqu'à 7 ou 8 commandes sont déjà dans le tampon. Il est recommandé d'utiliser le bouton Fermer / Réinitialiser si vous souhaitez arrêter rapidement le fonctionnement. Il est également recommandé d'installer des interrupteurs de fin de course.

Écrire le journal de débogage dans le dossier de départ (nécessite un redémarrage)

L'activation de cette fonctionnalité enverra tous les messages d'état et les diagnostics à un fichier journal appelé GrblController.log dans le dossier de départ de l'utilisateur. Le fichier est effacé chaque fois que Grbl Controller est redémarré.

La fonctionnalité ne prend pas effet immédiatement. Si vous modifiez cette valeur, vous devez cliquer sur OK, puis quitter Grbl Controller et le redémarrer. Ceci est dû à une anomalie avec la réinitialisation de Log4Qt.

Si vous utilisez une version de version, des messages d'avertissement et d'erreur s'affichent dans la console (si vous en voyez un, par exemple dans Qt Creator). Pour les versions de débogage, vous verrez tous les messages dans la console.

Indépendamment de la génération, tous les messages sont envoyés au fichier journal si la fonctionnalité est activée.

Installer GrblController pour Linux

B. Building GrblController for Linux

Option 1: QtCreator GUI Build and Run

Launch Qt Creator Make sure you have the source code for Grbl Controller: in a terminal window, go back to your home folder: cd ~ mkdir github cd github git clone https://github.com/zapmaker/GrblHoming In QtCreator, open existing project, navigate to

github/GrblHoming and open GcodeSenderGUIthreads.pro Choose the default Desktop mode which should be checked and click Configure Project Click the green arrow to run. This will take a while to build and Grbl Controller should eventually launch.

After running you will need to pick the correct USB serial port. In the drop down of available ports, it may have USB in the name or it may be ttyS0. Extra! Running Ubuntu in VirtualBox with pass-through USB – Only for the Adventurous

If you are running Ubuntu in VirtualBox on Windows, you'll need to do the following:

In VirtualBox, open Settings on your OS and go to USB and click Add (this assumes you installed VirtualBox with USB support). Pick the Arduino entry and then move it to the top of the list. Restart the Linux virtual machine – if Windows 7 you should see a "Device Driver Installation" popup appear with "Unidentified Device" – wait about 5 minutes or so for it to finish. It should find "VirtualBox USB" and say Ready to Use". In a terminal window, cd into the folder that starts with "build…" in the github folder where QtCreator created the executable (tip: don't type the 50 character long folder in by hand, just type a couple of characters and then hit the tab key – it will autocomplete the entire filename). Type sudo ./GrblController It will ask you to type your admin password (this step is required since Linux won't open the serial port if not admin) In the Grbl Controller dropdown, pick ttyACM0 and Open. You should see the friendly greeting of grbl.

Option 2: Command Line

*1. Open terminal window to your GrblHoming folder *2. Type qmake -config release -spec linux-g++ GcodeSenderGUIthreads.pro *3. Type make *4. To launch, type ./GrblController

Since you may not have sufficient permissions to access the executable, you can launch it like so:

gksudo ./GrblController

Or create a shell script to do this for you:

*1. gedit g.sh *2. Enter this text and save

- - #!/bin/sh
- - gksudo ./GrblController

*3. chmod +x g.sh *4. Type this to run: ./g.sh

EN FRANCAIS

B. Installer GrblController pour Linux Option 1: Génération et exécution de l'interface graphique QtCreator

- 1. Lancer Qt Creator
- 2. Assurez-vous que vous avez le code source pour Grbl Controller: dans une fenêtre de terminal, retournez dans votre dossier personnel: cd ~
- 3. mkdir github
- 4. cd github
- 5. git clone https://github.com/zapmaker/GrblHoming
- 6. Dans QtCreator, ouvrez le projet existant, accédez à github / GrblHoming et ouvrez GcodeSenderGUIthreads.pro
- 7. Choisissez le mode Bureau par défaut qui doit être coché et cliquez sur Configurer le projet

- 8. Cliquez sur la flèche verte pour exécuter. Cela prendra du temps à construire et Grbl Controller devrait éventuellement se lancer.

Après l'exécution, vous devrez choisir le bon port série USB. Dans la liste déroulante des ports disponibles, il peut y avoir un port USB dans le nom ou il peut s'agir de ttyS0.

Supplémentaire! Exécuter Ubuntu dans VirtualBox avec pass-through USB - Seulement pour les aventureux Si vous exécutez Ubuntu dans VirtualBox sous Windows, vous devrez effectuer les opérations suivantes:

- 1. Dans VirtualBox, ouvrez Paramètres sur votre système d'exploitation et allez à USB et cliquez sur Ajouter (cela suppose que vous avez installé VirtualBox avec prise en charge USB).
- 2. Sélectionnez l'entrée Arduino, puis déplacez-la en haut de la liste.
- 3. Redémarrez la machine virtuelle Linux - si Windows 7 vous devriez voir apparaître une fenêtre “Installation du pilote de périphérique” avec “Périphérique non identifié” - attendez environ 5 minutes pour la terminer. Il devrait trouver “VirtualBox USB” et dire prêt à l'emploi “.
- 4. Dans une fenêtre de terminal, cd dans le dossier qui commence par “build …” dans le dossier github où QtCreator a créé l'exécutable (astuce: ne tapez pas le long dossier de 50 caractères à la main, tapez simplement quelques caractères puis tapez la touche de tabulation - elle complètera automatiquement le nom de fichier entier).
- 5. Tapez sudo ./GrblController
- 6. Il vous demandera de taper votre mot de passe administrateur (cette étape est requise car Linux n'ouvrira pas le port série si ce n'est pas admin)
- 7. Dans la liste déroulante Grbl Controller, sélectionnez ttyACM0 et Open. Vous devriez voir la salutation amicale de grbl.

Option 2: Ligne de commande

- 1. Ouvrez la fenêtre du terminal dans votre dossier GrblHoming
- 2. Tapez qmake -config release -spec linux-g++ GcodeSenderGUIthreads.pro
- 3. Tapez make
- 4. Pour lancer, tapez ./GrblController

Comme vous ne disposez peut-être pas d'autorisations suffisantes pour accéder à l'exécutable, vous pouvez le lancer comme suit: gksudo ./GrblController Ou créez un script shell pour le faire pour vous:

- 1. gedit g.sh
- 2. Entrez ce texte et enregistrez
  - - #! / bin / sh
  - - gksudo ./GrblController
- 3. chmod + x g.sh
- 4. Tapez ceci pour exécuter: ./g.sh

Configuration GRBLv1.1

GRBL 1.1

LaserWeb4 nécessite au moins GRBL 1.1c. Nous vous recommandons d'utiliser la dernière version de GRBL disponible!

Veuillez consulter https://github.com/gnea/grbl/wiki/Flashing-Grbl-to-an-Arduino pour obtenir des

instructions sur la mise à jour de votre firmware. Configuration materielle

Pour obtenir de l'aide sur le câblage de votre carte GRBL, veuillez visiter https://github.com/gnea/grbl/wiki/Connecting-Grbl . Ajuster le courant pas à pas

Si vous n'avez pas encore ajusté le courant de vos pilotes pas à pas, faites-le maintenant.

Cela n'a pas de sens de configurer l'accélération et l'avance maximale lorsque le courant pas à pas n'est pas correct! Réglez le courant à une valeur où vous avez un bon couple mais ne surchauffez pas le pilote et le moteur.

Consultez la documentation de votre machine ou des pilotes pas à pas pour savoir comment régler le courant. Les détails peuvent être trouvés sur http://reprap.org/wiki/Pololu_stepper_driver_board . Calculer les pas / mm

Vous devez savoir combien de pas chaque axe doit parcourir pour une distance de 1 mm. Cela dépend de la taille des pully ou du pas de la broche et du micropas.

Il y a une belle calculatrice sur http://www.prusaprinters.org/calculator/#steppermotors .

Configurer les paramètres GRBL

Pour configurer votre carte basée sur GRBL, vous devez être connecté à LaserWeb ou à un programme de terminal. Dans LaserWeb, vous pouvez envoyer des commandes manuellement avec la ligne de commande en bas à droite de l'écran. Après avoir tapé la commande, vous devez appuyer sur ENTERpour l'envoyer à GRBL.

Envoyer $$pour obtenir une liste de la configuration réelle. Pour définir un paramètre, vous devez envoyer $num=val(ex. $ 10 = 0).

Assurez-vous que les valeurs suivantes sont définies:

$10=0 ;send work coordinates in statusReport $30=1000 ;max. S-value for Laser-PWM $31=0 ;min. S-value $32=1 ;Laser Mode on $100=160 ;steps/mm in X, depending on your pulleys and microsteps $101=160 ;steps/mm in Y, depending on your pulleys and microsteps $102=160 ;steps/mm in Z, depending on your pulleys and microsteps $110=10000 ;max. rate mm/min in X, depending on your system $111=10000 ;max. rate mm/min in Y, depending on your system $112=10000 ;max. rate mm/min in Z, depending on your system $120=1000 ;acceleration mm/s^2 in X, depending on your system $121=1000 ;acceleration mm/s^2 in Y, depending on your system $122=300 ;acceleration mm/s^2 in Z, depending on your system $130=400 ;max. travel mm in X, depending on your system $131=300 ;max. travel mm in Y, depending on your system $132=100 ;max. travel mm in Z, depending on your system $$;to check the actual settings

(voir https://github.com/gnea/grbl/blob/master/doc/markdown/settings.md pour plus de détails)

Évaluer l'accélération

Pour trouver les bons paramètres d'accélération, procédez comme suit:

- Chargez notre acceleration_test.jpg (clic droit pour télécharger) dans LaserWeb.
- Ensuite, créez une opération raster et changez la passe à 1 et la vitesse de coupe à 6000 mm / min
- Générez le gcode.

- Cliquez sur l'onglet COM, sélectionnez le port de la machine et connectez-vous.
- Cliquez sur l'onglet JOG. En bas à droite de l'écran, vous voyez la ligne de la console.
- Tapez $120=1500sur la ligne de la console et appuyez sur ENTER. Cela définit votre accélération X sur 1500 mm / s2.
- Exécutez le travail.
- Si l'accélération est trop élevée, vous entendrez / verrez des pas perdus sur les côtés (lors du changement de direction). Réduisez ensuite la valeur d'accélération de 10% et testez à nouveau.
- S'il n'y a pas de pas perdus, augmentez la valeur de 10% et testez à nouveau.
- Si vous avez trouvé la limite où commence la perte de stepps, réduisez la valeur de 15-20% et réglez-la sur $120et $121(axes X et Y).

Évaluer l'alimentation maximale

Pour trouver les paramètres d'alimentation maximum, procédez comme suit:

- Chargez notre feed_test.jpg (clic droit pour télécharger) dans LaserWeb.
- Créez une opération raster et changez la passe à 1 et la vitesse de coupe à 6000 mm / min (= 100 mm / s).
- Générez le gcode.
- Cliquez sur l'onglet COM, sélectionnez le port de la machine et connectez-vous.
- Cliquez sur l'onglet JOG. En bas à droite de l'écran, vous voyez la ligne de la console.
- Tapez $110=30000sur la ligne de la console et appuyez sur ENTER. Ceci définit l'avance X max de GRBL à 30000 mm / min.
- Tapez $111=30000sur la ligne de la console et appuyez sur ENTER. Ceci définit l'avance Y max de GRBL à 30000 mm / min.
- Exécutez le travail.
- Si l'alimentation est trop haute, vous entendrez / verrez des pas perdus pendant les mouvements X. Ensuite, abaissez la vallée d'alimentation (à l'étape 2) de 10% et testez à nouveau.
- S'il n'y a pas de pas perdus, augmentez la valeur de 10% et testez à nouveau.
- Si vous avez trouvé la limite où commence la perte de stepps, réduisez la valeur de 15-20% et réglez-la sur $110et $111(axes X et Y).

Astuces GRBL pour CAMBAM sur Windows 10

Fernand le 07/11/2019

Bonjour j’ai continué de chercher pour GRBLMACHINE rien trouvé sur le net mais sur mon fixe ça marché alors j’ai comparé les 2 ordi et j’ai trouvé ça

Il faut faire un clic droit sur le fichier « Cambam.Cad.Solide.DLL » puis propriété et cocher la case débloqué

Faire de même pour GRBLMachinePlugin.DLL

C’est Windows qui a bloqué pour protéger l’ordinateur

J’espéree que ça va marcher chez toi

From:
http://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
http://chanterie37.fr/fablab37110/doku.php?id=start:cnc:grbl&rev=1610129195

Last update: **2023/01/27 16:08**