Configuration

configuration-de-grbl-v1-1

Configuration de GRBL Par Steph, publié le 02 janvier 2018 & mis à jour le 07 février 2018 - 2227

Préparation de GRBL

Avant de “téléverser” GRBL dans votre Arduino, certains paramétrages s’imposent.

Le fichier "grbl.h"

Ce fichier est situé dans “/libraries/grbl” de votre dossier d’installation de l'IDE Arduino.

“grbl.h” est le fichier qui synthétise tous les appels aux autres fichiers en procédant à des “include”. On y touche pas !

Le fichier "config.h"

Ce fichier est situé dans “/libraries/grbl” de votre dossier d’installation de l'IDE Arduino.

C’est ici que le paramétrage de GRBL en accord avec votre machine va avoir lieu. Il est donc important d’être très attentif à cette partie de l’installation.

C’est un fichier de 413 lignes mais avec beaucoup de commentaires. Les lignes commençant par deux barres obliques (slasches) sont commentées et donc invalidées. il suffit d'ôter les “slashes” pour qu'elles deviennent valides.

Par exemple, vers la ligne 105 on peut modifier le cycle de homing.

Pour un homing sur les axes X et Y seulement et les deux en même temps :

On commente les deux lignes suivantes :

#define HOMING_CYCLE_0 (1«Z_AXIS) REQUIRED: First move Z to clear workspace.

#define HOMING_CYCLE_1 [1)] OPTIONAL: Then move X,Y at the same time.

Et on décommente celle-ci :

#define HOMING_CYCLE_0 [2)] NOT COMPATIBLE WITH COREXY: Homes both X-Y in one cycle.

Attention à bien vérifié et surtout a sauvegarder votre fichier avant toute modification…

Le fichier "report.c"

Ce fichier est situé dans “/libraries/grbl” de votre dossier d’installation de l'IDE Arduino.

Ici c’est juste pour le fun ! Mais attention aux erreurs suite à l’enlèvement inopportun d’un caractère obligatoire… Un point-virgule, des guillemets, une parenthèse… Et hop une erreur à la compilation ! Bref sauver le fichier d’origine avant toute manipulation.

Vous pouvez par exemple remplacer la ligne N° 151 : printPgmString(PSTR(“\r\nGrbl ” GRBL_VERSION “ ['$' for help]\r\n”));

Par celle-ci : printPgmString(PSTR(“\r\nMa CNC sous grbl ” GRBL_VERSION “ \r\n”));

Ce qui affichera en message d’accueil : “Ma CNC sous Grbl 1.1f”.

Mais bon, cela ne sert pas à grand-chose… Par contre la modification de ce fichier vous permet de “franciser” tous les messages de la console. les messages sont en général les mots ou pharses entre quotes (“Home”, “Pgm End”…). Attention toutefois à ne pas faire d'erreurs de syntaxe et de ne pas utiliser de caractères spéciaux…

Installation de GRBL

Une fois les fichiers modifiés par vos soins, il faut bien entendu recompiler tout ce petit monde. Pour cela, rien de bien compliqué :

```
  Connectez votre Arduino à votre PC via l’USB.
  Configurez le dans l’IDE Arduino (Type de carte et N° de port).
  Ouvrir "Fichiers/exemples/grbl/grblUpload".
  Cliquez sur la coche verte de vérification de code (En haut à gauche).
  Si tout se passe bien, la barre du bas sera verte et vous aurez un message favorable. Toutefois, une partie en rouge vous alerteras sur le fait que la mémoire est quasi pleine... C'est normal, GRBL 1.1 utilise la mémoire au maximum de ses possibilités sur les UNO et NANO.
```

Il ne reste plus qu’à téléverser GRBL dans l’Arduino.

Paramétrage de GRBL

Maintenant que GRBL est chargé dans l'Arduino avec la bonne configuration, il reste à faire le paramétrage.

Dans GRBL, toutes les dimensions sont en millimètres. Si vous comptez en pouces, pensez à la conversion…

Le changement des valeurs des paramètres peut se faire de plusieurs manières.

Via UGS ou un autre programme d'interface avec GRBL

Connecter vous à l'Arduino, et cliquez sur “Paramètres/Paramètres Firmware/GRBL”. Vous obtenez la fenêtre suivante :

Double cliquez sur la valeur à changer, modifier la et faite enregistrer. Une fois terminé, fermer cette fenêtre, GRBL est paramètré.

Via l'IDE Arduino

La commande “$$” lancée depuis le terminal série affiche les données par défaut :

Taper sur la ligne de commande en haut de la fénêtre, le paramètre à changer et sa valeur. Par exemple : “$1=30”. Puis cliquez sur le bouton “envoyer”, le paramètres est changé. faites de même avec chaque paramètres à changer. A chaque fois taper “$$” afin de vérifier la prise en compte de la nouvelle valeur du paramètre.

Les différents paramètres à modifier ou pas

Paramètre

Défaut

Explication

-$0

1. 10
    1. Durée du signal de pas en µS
    2. A paramétrer éventuellement en fonction de vos drivers pas-à-pas. Avec des impulsions trop longues, des erreurs peuvent survenir à vitesse élevée. 10 est une bonne valeur par défaut.

$1

25

Retard de la désactivation des moteurs aprés une instruction en mS.

C'est le temps que GRBL gardera les moteurs verouillés, avant de les désactiver. Pour certains moteurs, cela peut engendrer des pertes de pas. Si c'est votre cas, il suffit de passer ce paramètre à

255 et vos moteurs ne seront jamais désactivés.

$2

0

Inversion du signal de pas (Masque binaire).

Par défaut, l'impulsion du signal de pas débute sur un front montant. Puis aprés la durée définie par le paramètre $0, on repasse en niveau bas. En règle générale, il n'y a pas à changer la valeur par défaut de ce paramètre.

Valeurs du paramètre de 0 à 7, entre parenthèses est représentée la valeur binaire du paramètre

```
  0 (00000000) - Aucune inversion
  1 (00000001) - Inversion de l'axe X
  2 (00000010) - Inversion de l'axe Y
  3 (00000011) - Inversion des axes X et Y
  4 (00000100) - Inversion de l'axe Z
  5 (00000101) - Inversion des axes X et Z
  6 (00000110) - Inversion des axes Y et Z
  7 (00000111) - Inversion des 3 axes
```

$3

0

Inversion du signal de direction (Masque binaire).

Par défaut, GRBL considère que les axes de déplacent dans la direction positive lorsque la broche “direction” est au niveau bas et inversement.

Sur certaines machines il est nécessaire d'inverser la direction d'un ou plusieurs axes. Par exemple, pour une table translatée on devra inverser l'axe Y. Pour ce faire, on passera le paramètres à 2

Valeurs du paramètre allant de 0 à 7. Pour choisir la valeur, se reporter au paramètre $2.

$4

0

Inversion du signal d'activation “Enable” (Booléen).

Par défaut, la broche d'activation est au niveau haut pour désactiver et au niveau bas pour activer.

Au besoin, on pourra inverser cela en passant le paramètre à 1. Nécéssité de redémarrer pour la prise en compte de ce changement.

$5

0

Inversion du type de contact des fin de courses (Booléen).

Par défaut, les broches de limites sont au niveau haut via une résistance “pull-up” interne à l'Arduino. Les contacts de fin de course étant normalement ouverts “NO”; lorsqu'ils sont actionnés la broche limites correspondante passe au niveau bas.

Pour installer des fins de courses NF (normalement fermé), il faut passer ce paramètre à 1. Attention, il est nécessaire dans ce cas de rajouter des résistances de “pull-down” externe pour chaque broche de limite. Nécéssité de redémarrer pour la prise en compte de ce changement.

$6

0

Inversion de la broche de sonde (Booléen).

Par défaut, la broche de sonde est au niveau haut via une résistance “pull-up” interne à l'Arduino. Quand le capteur indique un niveau bas, par mise à la masse de la sonde, GRBL l'interprète comme déclenchée.

Pour inverser cela, il faut passer ce paramètre à 1. Attention, il est nécessaire dans ce cas de rajouter une résistance de “pull-down” externe. Nécéssité de redémarrer pour la prise en compte de ce changement.

$10

25

Rapport de situation en temps réel (Masque binaire).

Détermine ce que doit répondre GRBL en temps réel lors de l'envoi de la commande “?”.

Par défaut GRBL renvoie : Son état, la position “machine” et la position “travail”. trois autres informations sont disponibles et peuvent être utilisées dans le cadre d'un interfaçage avec GRBL.

```
Tampon du planificateur
Tampon de la liaison série Rx
Etat des limites (Ordre : ZYX).
```

Valeurs du paramètre en fonction du choix des réponses demandées. les valeurs sont à additionner, par exemple pour obtenir les positions “machine” (1), “travail” (2) et le tampon série (8), le paramètre prendra la valeur : 11

```
1 - Position "machine".
2 - Position "travail".
4 - Tampon du planificateur.
8 - Tampon liaison série.
16 - Etat des limites.
```

$11

0.010

Ecart de jonction en mm.

Cet écart est utilisé par le gestionnaire d'accélération. En diminuant cette valeur on diminue l'accélération dans les courbes et inversement. Avec une valeur trop élevée on risque la perte de pas. 10 est une valeur optimale.

$12

0.002

Tolérance des arcs en mm.

Les commandes G2/G3 de GRBL subdivise les arcs en de multiples petits segments. Probablement n'aurez vous jamais à changer cette valeur trés faible et en dessous de la précision de la plupart des machines.

$13

0

Unités des rapports d'état (Booléen).

Par défaut, GRBL utilise les mm pour générer ces rapports en temps réels (Paramètre $ 10). Pour qu'il les livre en inches il suffit de passer ce paramètre à 1.

$20

0

Activation des limites logicielles (Booléen).

Ce paramètre est un dispositif de sécurité évitant à votre machine de dépasser les valeurs physiques de vos axes de travaul (Paramètres $130 à $132).

A chaque réception d'une information Gcode, GRBL vérifie si ces limites sont dépassées ou non. Si elles sont dépassées, il stoppera tout (Y compris la broche et le lubrifiant) et génèrera une alarme. la position de la machine est conservée et aprés déverrouillage de l'alarme le programme peut continuer.

Afin que cela fonctionne il est impératif que le “homing” soit activés (Paramètre $22) et les dimensions d'axes correctement remplies (Paramètres $130 à $132).

$21

0

Activation des limites physiques (Booléen).

Ce paramètre fonctionne sensiblement comme le paramètre $20 mais utilise des contacts fin de course.

Lorsqu'un contact est actionné, GRBL stoppe tout immédiatement et passe en mode alarme vous abligeant à vérifier la machine et à tout réinitialiser.

Il est plus que conseillé d'utiliser des procédés de prévention contre le parasitage électrique.

Si on positionne un fin de course à chaque bout des axes, il conviendra de les relier en parallèle si ce sont des contacts normalement ouverts “NO” ou en série si ce sont des contacts normalement fermés “NF”.

Un déclenchement de limite physique est considéré comme un évènement critique ou les moteurs pas-à-pas sont arrêtés immédiatement et auront probablement perdus des pas. GRBL ne peut alors plus garantir sa position et rentre dans une boucle d'alarme infinie vous obligeant à vérifier la machine et le réinitialiser.

$22

0

Cycle de homing (Booléen).

Ce cycle est utilisé pour connaitre avec exactitude la position de la machine a chaque nouvelle session GRBL. Avec le homing, vous avez toujours le point 0 de référence machine.

Pour mettre en oeuvre le homing, il est nécessaire de disposer de fin de courses positionnés par défaut au point les plus “haut” de chaque axes.

Par defaut le cycle déplace en premier l'axe Z afin de dégager l'espace de travail puis simultanément les axes X et Y. Le premier axe en butée attend le second. Avant de pouvoir bouger un axe, GRBL vous oblige à effectuer un cycle de homing “$H”.

$23

0

Inversion de directiobn de homing (Masque binaire).

Par défaut, GRBL voit les fins de courses dans le sens positif pour les trois axes. Si vous preferez une autre postion de homing, il suffit de modifier ce paramètre et bien entendu d'avoir des fins de courses positionnés en conséquence…

Pour changer, de manière avancée, le paramètrage du homing il est nécessaire de modifier le fichie “config.h” puis de compiler GRBL et de le téléverser de nouveau dans l'Arduino

Valeurs du paramètre de 0 à 7, entre parenthèses est représentée la valeur binaire du paramètre

```
  0 (00000000) - Aucune inversion (En haut de l'axe Y à droite et broche
haute).
  1 (00000001) - Inversion de l'axe X (En haut de l'axe Y à gauche et broche
haute).
  2 (00000010) - Inversion de l'axe Y (En bas de l'axe Y à droite et broche
haute).
  3 (00000011) - Inversion des axes X et Y (En bas de l'axe Y à gauche et
broche haute).
  4 (00000100) - Inversion de l'axe Z (En haut de l'axe Y à droite et broche
basse).
  5 (00000101) - Inversion des axes X et Z (En haut de l'axe Y à gauche et
broche basse).
  6 (00000110) - Inversion des axes Y et Z (En bas de l'axe Y à droite et
```

```
broche basse).
  7 (00000111) - Inversion des 3 axes (En haut de l'axe Y à gauche et broche
bassee
  ).
```

$24

25

Vitesse de référencement du homing (mm/mn).

C'est la vitesse de déplacement utilisée pour la recherche précise du zéro de chaque axe.

Ce paramètre est à 25 par défaut. On peut en augmenter la valeur, mais en vérifiant à chaque changement que la détection du zéro est bien assurée. Une valeur de 50 semble être optimale.

$25

500

Vitesse de recherche du homing (mm/mn).

C'est la vitesse de déplacement utilisée pour la recherche des fins de courses de chaque axe.

Ce paramètre est à 500 par défaut, cela semble être optimale.

$26

250

Anti-rebond contact (mS).

GRBL introduit un délai dans l'établissement des contacts électrique appelé “anti-rebond”. Cela permet de palier au parasitage électrique et à des déclenchements intempestifs des limites.

Il est toutefois conseillé de garder ce paramètre a environ 15mS et d'ajouter des systèmes “anti-rebonds” sur chaque fin de course (Et sur le bouton d'arrêt d'urgence). Cela peut aller d'un simple condensateur de 0.1µF en parallèle sur le contact à une isolation galvanique par opto-coupleurs.

$27

1

Retrait de sécurité (mm).

Distance de retrait aprés détection du zéro de l'axe. Cela permet d'éviter des déclenchement intempestifs des limites physique. Les bonnes valeurs semblent être de 1 ou 2mm.

$30

1

Vitesse maximum de la broche (tr/mm).

C'est la vitesse maximum à laquelle votre broche peut tourner. Ce paramètre sert à GRBL pour régler le cycle PWM de réglage de la vitesse variable de la broche. La valeur par défaut est 1000 et est à modifier suivant votre type de broche.

$31

1000

Vitesse minimum de la broche (tr/mm).

A priori c'est 0.

$32

0

Mode laser (Booléen).

Dans ce mode on utilise un laser et les commandes G1, G2 ou G3 sont envoyées en continu lorsqu'elle sont programmées avec une vitesse de broche, dans ce cas la puissance du laser. Lorsqu'elle est désactivée, mode CNC, GRBL arr^te le mouvement à chaque changement de vitesse de broche.

$100

250

Pas/mm de l'axe X (pas/mm).

Cette valeur est propre à cahque axe de la machine, elle est définie par las caractéristiques de l'entrainement des axes. C'est le nombre de millimètres parcourus à chaque révolution du moteur.

La formule de calcul est la suivante : (Pas par révolution du moteur x nombre de micropas) / millimètres parcourus par révolution.

Typiquement le “pas par révolution du moteur” est de 200, le micropas de 1,2,4,8,16,32… les millimètres parcourus par révolution sont en fonction du pas de votre vis sans fin par exemple.

$101

250

Pas/mm de l'axe Y (pas/mm).

Cette valeur est propre à cahque axe de la machine, elle est définie par las caractéristiques de l'entrainement des axes. C'est le nombre de millimètres parcourus à chaque révolution du moteur.

La formule de calcul est la suivante : (Pas par révolution du moteur x nombre de micropas) / millimètres parcourus par révolution.

Typiquement le “pas par révolution du moteur” est de 200, le micropas de 1,2,4,8,16,32… les millimètres parcourus par révolution sont en fonction du pas de votre vis sans fin par exemple.

$102

250

Pas/mm de l'axe Z (pas/mm).

Cette valeur est propre à cahque axe de la machine, elle est définie par las caractéristiques de l'entrainement des axes. C'est le nombre de millimètres parcourus à chaque révolution du moteur.

La formule de calcul est la suivante : (Pas par révolution du moteur x nombre de micropas) / millimètres parcourus par révolution.

Typiquement le “pas par révolution du moteur” est de 200, le micropas de 1,2,4,8,16,32… les millimètres parcourus par révolution sont en fonction du pas de votre vis sans fin par exemple.

$110

500

Déplacement maximal possible axe X (mm/mn).

Le moyen le plus simple pour régler ce paramètre est d'envoyer une commande G0 X100 (Ou autre) et de tester en augmentant le paramètre par petit pas. Lorsque vous atteindrez le seuil admissible de votre moteur, celui-ci se bloquera. Ensuite diminuer la valeur de 15% environ et c'est réglé.

$111

500

Déplacement maximal possible axe Y (mm/mn).

Le moyen le plus simple pour régler ce paramètre est d'envoyer une commande G0 X100 (Ou autre) et de tester en augmentant le paramètre par petit pas. Lorsque vous atteindrez le seuil admissible de votre moteur, celui-ci se bloquera. Ensuite diminuer la valeur de 15% environ et c'est réglé.

$112

500

Déplacement maximal possible axe Z (mm/mn).

Le moyen le plus simple pour régler ce paramètre est d'envoyer une commande G0 X100 (Ou autre) et de tester en augmentant le paramètre par petit pas. Lorsque vous atteindrez le seuil admissible de votre moteur, celui-ci se bloquera. Ensuite diminuer la valeur de 15% environ et c'est réglé.

$120

10

Taux d'accélération de l'axe X (mm/mn).

De manière simple, une valeur faible se traduit par un mouvement lent et une valeur élevée accélère le mouvement. pour le réglage, on procédera de manière similaire au paramètre 110.

$121

10

Taux d'accélération de l'axe Y (mm/mn).

De manière simple, une valeur faible se traduit par un mouvement lent et une valeur élevée accélère le mouvement. pour le réglage, on procédera de manière similaire au paramètre 111.

$122

10

Taux d'accélération de l'axe Z (mm/mn).

De manière simple, une valeur faible se traduit par un mouvement lent et une valeur élevée accélère le mouvement. pour le réglage, on procédera de manière similaire au paramètre 112.

$130

200

Déplacement maximum sur l'axe X (mm).

C'est la course maximum de bout à bout pour l'axe X, qui est bien sur spécifique à chaque machine. C'est utilisé par la vérification des limites logigicielles du $20

$131

200

Déplacement maximum sur l'axe Y (mm).

C'est la course maximum de bout à bout pour l'axe Y, qui est bien sur spécifique à chaque machine. C'est utilisé par la vérification des limites logigicielles du $20

$132

200

Déplacement maximum sur l'axe Z (mm).

C'est la course maximum de bout à bout pour l'axe Z, qui est bien sur spécifique à chaque machine. C'est utilisé par la vérification des limites logigicielles du $20

Nota

-

Pour les paramètres $110,$111,$112,$120,$121 et $122

il est conseillé de faire des tests en envoyant du Gcode à “blanc” avant tout usinage. Bien vérifier le fonctionnement avec des déplacements simultanés des trois axes.

Les commandes de GRBL

Les commandes “$” fournissent des contrôles supplémentaires pour l'utilisateur,comme l'affichage de

l'état de l'analyseur G-code, la vérification du mode Gcode ou bien encore l'exécution du cycle de homing.

$# - Afficher les paramètres du Gcode

Ces paramètres stockent les valeurs de décalage pour les coordonnées G54 à G59, pour les positions pré-définies G28 & G30, le décalage G92, G43.1 longueur d'outil et G38.2 sonde.

Ils sont pour la plupart écrits sur l'EEPROM et sont donc persistants. Ceux qui ne sont pas persistants sont : G92, G43.1 et G38.2, ils sont donc réinitialisés à chaque redémarrage.

Pour les persistants, ils sont donc conservés même en cas de reflashage de l'Arduino. Pour les modifier, il conviebnt donc de le faire explicitement.

```
  G54 à G59 sont modifiables via les commandes G10 L2 et G10 L20. L2 par
rapport au zéro machine et L20 en fonction de la position machine.
  G28 est modifiable via la commande G28.1.
  G30 est modifiable via la commande G30.1.
```

En tapant la commande "$#", GRBL renverra un tableau du genre :

```
  [G54:10.000,10.000,5.000]
  [G55:100.000,100.000,65.000]
  [G56:100.000,100.000,0.000]
  [G57:220.000,410.000,0.000]
  [G58:0.000,0.000,0.000]
  [G59:0.000,0.000,0.000]
  [G28:0.000,0.000,0.000]
  [G30:10.000,10.000,0.000]
  [G92:0.000,0.000,0.000]
  [TLO:0.000]
  [PRB:0.000,0.000,0.000:0]
  ok
```

"TLO" représentela longueur d'outil et "PRB" les dernières coordonnées de sondage.

$G - Voir l'état de l'analyseur de Gcode

Cette commande renvoie les modes actifs de l'analyseur G-Code. Grbl y répond par :

```
  GC:G0 G54 G17 G21 G90 G94 M5 M9 T0 F250 S0]
  ok
```

Ce sont ces modes actifs qui détermine la façon dont sera interprétée le prochain envoi de Gcode.

$I - Informations sur la version

A cette commande GRBL renvoie, par exemple :

```
  VER:1.1f.20170801:]
  [OPT:V,15,128]
```

```
ok
```

$N - Afficher les blocs de démarrage

Les blocs de démarrage sont des lignes de Gcode que GRBL exécute automatiquement à chaque démarrage ou réinitialisdation. Par défaut, GRBL stocke deux blocs vide que voici :

```
$N0=
$N1=
ok
```

Il est rès important de faire extrèmement attention aux commandes de mouvements (G0, G1, G2, G3, G28 ou G30) que vous allez stocker dans les blocs de démarrage. En effet ces commandes se dérouleront à chaque démarrage ou reset (Arrêt d'urgence, fins de course...), attention donc aux effets “pervers”.

Il ne faut pas y placer les commandes comme G10, G28.1 ou G30.1 car elles sont persistantes et cela forcerait GRBL à réécrire l'EEPROM de l'Arduino à chaque redémarrage ou reset... L'EEPROM n'est pas prévue pour des cycles répétitifs d'écriture.

Un bloc de démarrage se définit en tapant :“$N0=” suivi du Gcode. Par exemple : $N0=G21 G17 G55, ce qui définiera par défaut des unités en mm, la sélection du plan X/Y et le système de coordonnées de travail pré-défini G55.

Si c'est bon GRBL y répond simplement par “Ok”. Au prochain démarrage il enverra le prompt suivant :

```
Grbl 1.1f ['$' for help]
>G21G17G55:ok
```

Pour effacer ce bloc de démarrage il suffit de taper la commande “$N0=”.

$C - Vérifier le mode gcode

Cette commande demande à GRBL de traiter la totalité des blocs Gcode entrant mais sans bouger les axes et ne s'occupant ni de broche ni du refroidissement.

On peut ainsi tester son Gcode gràce à l'analyseur et vérifier si d'éventuelles erreurs sont détectées comme des dépassements de limites par exemple.

$ X - Supprimer le verrouillage de l'alarme

Le mode alarme de GRBL signale en principe un problème critique, comme le déclenchement de fins de course, un arrêt d'urgence...

Si le homing au démarrage est activé, GRBL bloque alors toutes les commandes jusqu'a l'éxécution d'un cycle de homing.

$H - Lancer un homing

Le homing est une commande à part gérée par GRBL conformément aux normes Gcode.

Une fois le homing effectué, vous pouvez déplacer les axes avec les boutons de “JOG” pour approcher

vos coordonnées de travail ou utiliser les positions prédéfinies. Par principe de sécurité, l'axe Z sera maintenu en position haute pour éviter tout accrochage durant les déplacements de positionnement.

$RST

Ces commandes permettent à l'utilisateur de restaurer tout ou partie de l'EEPROM par défaut de l'Arduino. Aprés chacune de ces commandes, GRBL procédera à une réinitialisation. $RST=$

Cette commande efface et restaure les paramètres par défaut “$$”. cela peut permettre de facilement effacer les données spécifiquesentrées par un fabricant et de repartir de zéro. $RST=#

Ici on efface et on remet à zéro, les diverses coordonnées stockées dans l'EEPROM comme G54 à G59, G28.1 et G30.1. cela permet de gagner du temps si on veut reprendre à zéro les divers décalages et positions. $RST=*-

Cette commande supprime et restaure toutes les données utilisées par GRBL : $$, $#, $N et $I.

Par contre cela n'efface rien de ce qui n'est pas GRBL. Pour cela il faut “nettoyer” l'EEPROM via l'IDE Arduino avec un croquis comme celui-ci :

```
#include <EEPROM.h>
void setup() {
   pinMode(13, OUTPUT);
   for (int i = 0 ; i < EEPROM.length() ; i++) {EEPROM.write(i, 0);}
   digitalWrite(13, HIGH);
}
void loop() {}
```

Les commandes en temps réel: ~,!,? Et Ctrl-X

GRBL comporte quatre commandes en “temps réel” qui peuvent être envoyées à tout moment et peu importe les tâches en cours, GRBL y répondra immédiatement. Elles commencent par des caractères spèciaux lus dans la liaison série et interprétées en moins de 5mS.

~ - Cycle start

Cela permet de démarrer un cycle ou de reprendre une commande. Pour que cette commande soit efficace, le tampon de GRBL devra contenir des ordres en attentes.

Cette commande est peu utile car par défaut l“auto cycle” est activé.

! - Feed hold

Cette commande de pause provoque un arrêt du cycle actif via une décélération contrôlée de manière a ne pas perdre la position en cours. le processus pourra être reprit via une commande “~- cycle start”.

? - Current status

Cette commande retourne immédiatement l'état actif de GRBL avec la position en temps réel dans les coordonnées de travail et dans celles machine.

Ctrl-x - Reset Grbl

C'est la réinitialisation logicielle de GRBL. Elle réinitialise GRBL en temps réel mais en conservant la position machine et sans éteindre l'Arduino.

Il est recommandé d'éxécuter cette commande avant de commencer tout travail. La machine partira ainsi sur des bases saines et fera exactement ce qu'on lui demande.

[1)] , [2)]
1«X_AXIS)|(1«Y_AXIS

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:config

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