

# grbl

## v0.9j

### O que é? Para que serve? Como configurar?

\$0=10 (step pulse, usec)  
\$1=25 (step idle delay, msec)  
\$2=0 (step port invert mask:00000000)  
\$3=6 (dir port invert mask:00000110)  
\$4=0 (step enable invert, bool)  
\$5=0 (limit pins invert, bool)  
\$6=0 (probe pin invert, bool)  
\$10=3 (status report mask:00000011)  
\$11=0.020 (junction deviation, mm)  
\$12=0.002 (arc tolerance, mm)  
\$13=0 (report inches, bool)  
\$20=0 (soft limits, bool)  
\$21=0 (hard limits, bool)  
\$22=0 (homing cycle, bool)  
\$23=1 (homing dir invert mask:00000001)  
\$24=50.000 (homing feed, mm/min)  
\$25=635.000 (homing seek, mm/min)  
\$26=250 (homing debounce, msec)  
\$27=1.000 (homing pull-off, mm)  
\$100=314.961 (x, step/mm)  
\$101=314.961 (y, step/mm)



**RODRIGO CONRADO**

## Quem somos

Atividade Maker é um projeto com o propósito de divulgar o estilo de vida "faça você mesmo" (DIY – Do It Yourself), aqui vamos divulgar projetos de Marcenaria, Eletrônica, Robótica, Arduino, Impressão 3D, CNC entre outros, Somos apaixonados pelo estilo Faça Você Mesmo.

**Atividade Maker - Construa o Seu Mundo, Você Pode!**

## Como entrar em contato?

Você pode entrar em contato pelos seguintes meios:



[www.atividademaker.com.br](http://www.atividademaker.com.br)



[www.facebook.com/atividademaker](https://www.facebook.com/atividademaker)



[www.youtube.com/c/AtividadeMakerOficial](https://www.youtube.com/c/AtividadeMakerOficial)



[contato@atividademaker.com.br](mailto:contato@atividademaker.com.br)



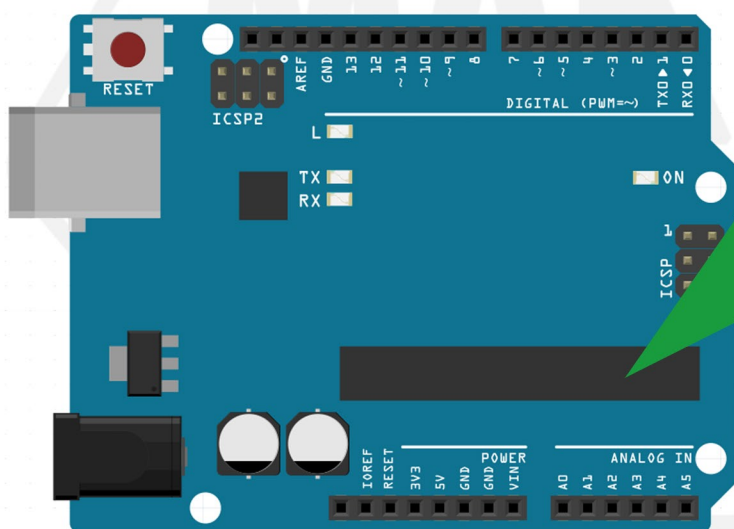


## Sobre o Grbl

O Grbl é grátis, open source (código aberto), software de alto desempenho para controlar o movimento de máquinas que se movem, que fazem as coisas, ou que fazem as coisas se moverem, e será executado diretamente em um Arduino.

Muitas impressoras 3D de código aberto usam Grbl em seus núcleos, sendo adaptado para uso em centenas de projetos, incluindo máquinas de corte a laser, CNC entre outros. Devido ao seu desempenho, simplicidade e requisitos de hardware, o Grbl tornou-se um fenômeno de código aberto.

Em 2009, Simen Svale Skogsrud (<http://bengler.no/grbl>) agradeceu a comunidade de código aberto, escrevendo e liberando as primeiras versões do Grbl a todos (inspiradas no Arduino Gcode Interpreter por Mike Ellery). Desde 2011, Grbl está a avançar como um projeto de código aberto voltada para a comunidade sob a liderança pragmática da Sungeun K. Jeon Ph.D. (@chamnit).



Placa Arduino Uno ou Compatível

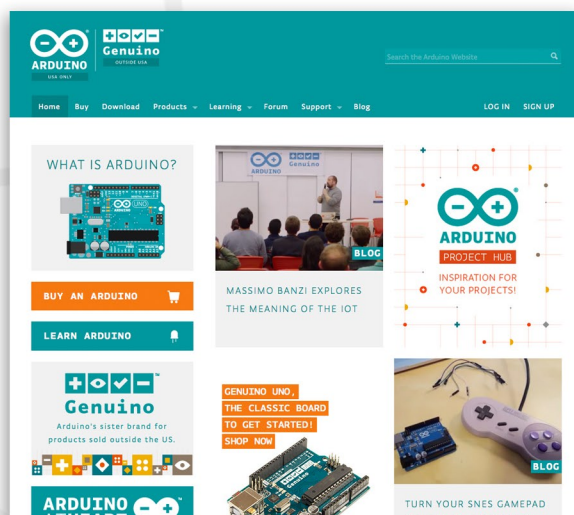
O Grbl é instalado dentro do Arduino Uno ou compatível.



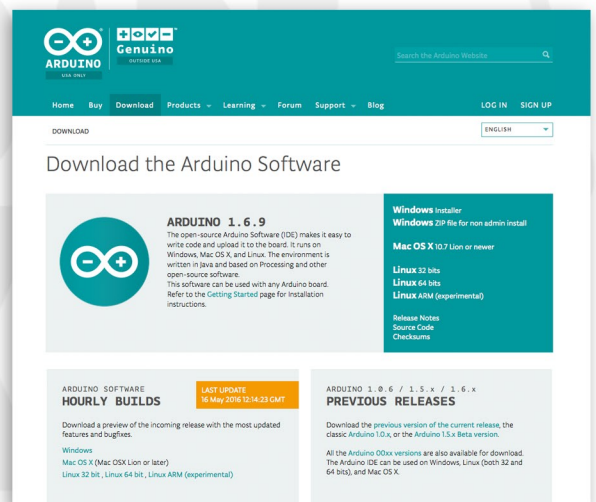


## Primeiros Passos - Arduino

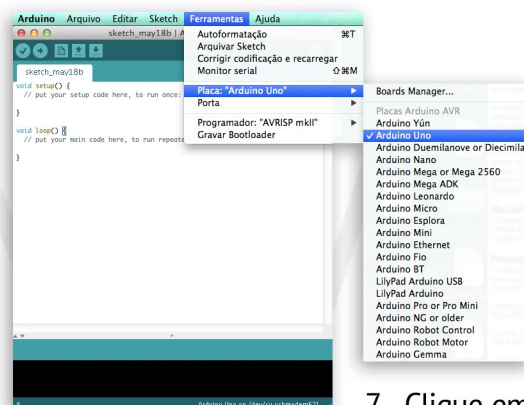
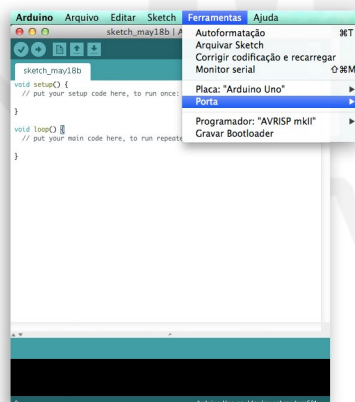
Vamos instalar o Grbl no Arduino através da IDE do Arduino, para isso você precisa acessar o site **www.arduino.cc** e efetuar o download da IDE para o seu sistema operacional.



- 1 - Acessar [www.arduino.cc](http://www.arduino.cc);
- 2 - Clicar em Download;
- 3 - Clicar no seu sistema operacional para efetuar o download;
- 4 - Faça a instalação conforme seu sistema;



- 5 - Abra a IDE Arduino;
- 6 - Clique em **Ferramentas > Porta** e selecione a porta serial.



- 7 - Clique em **Ferramentas > Placa** e selecione Arduino Uno;

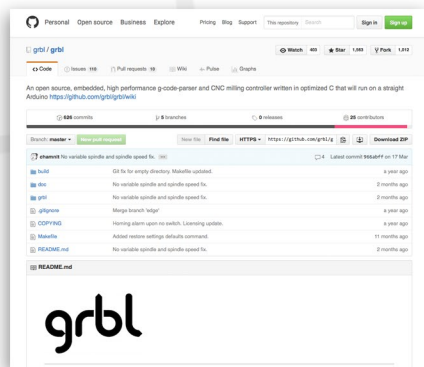






## Primeiros Passos - Grbl

Após a instalação da IDE Arduino e suas configurações você vai precisar efetuar o download do Grbl para que possa ser enviado para a memória do Arduino.



1 - Acesse: <https://github.com/grbl/grbl> e faça o download do Grbl clicando no botão **Download ZIP**;

2 - Descompactar o arquivo em um local de sua preferencia;

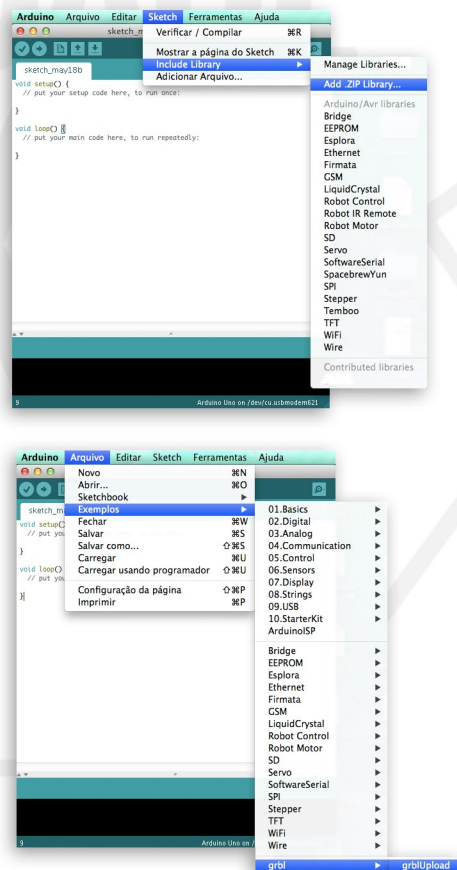
3 - Abra o IDE Arduino;

4 - Clique em **Sketch > Include Library > add .Zip Library** e selecione a pasta que você descompactou o Grbl;

5 - Conecte o cabo USB no seu computador e na sua placa Arduino;

6 - Clique em **Arquivos > Exemplos > e escolha o grbl > grblUpload**;

7 - Depois clique para transferir o código para o Arduino.





## Primeiros Passos - Terminal Serial

Pronto agora você já tem o Grbl instalado dentro do Arduino, você já pode fechar a IDE do Arduino pois não iremos mais utilizá-la.

Agora é preciso escolher um **Terminal Serial** para que possamos utilizar o Grbl.

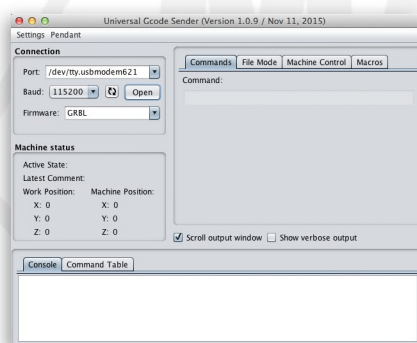
Você encontra uma lista com várias opções no seguinte endereço:

**<https://github.com/grbl/grbl/wiki/Using-Grbl>**

Iremos utilizar em nosso tutorial o terminal serial, **Universal Gcode Sender** na versão **1.0.9**.

Você pode efetuar o download no seguinte endereço;

**<https://github.com/winder/Universal-G-Code-Sender>**



- 1 - Conecte o cabo USB no seu computador e em seu Arduino Uno;
- 2 - Clique em **Port** e selecione a porta serial;
- 3 - Clique em **Baud** e selecione 115200;
- 4 - Clique em Open;
- 5 - Se tudo ocorreu bem você deve ver a seguinte mensagem:

**Grbl 0.9j ['\$' for help]**





## Iniciando o Grbl v0.9j

Primeiro, conecte-se ao Grbl usando o terminal serial de sua escolha, em seguida selecione a porta serial onde seu arduino está instalado e defina a taxa de transmissão (Baud) para 115200, clique em Open para se conectar.

Uma vez conectado você deverá visualizar o aviso, que se parece com isso:

```
Grbl 0.9j ['$' for help]
```

## Configurando o Grbl v0.9j

Para visualizar as configurações, digite na linha de comando \$\$ e pressione Enter.

O Grbl deve responder com uma lista das configurações atuais do sistema.

Todas essas configurações são persistentes e mantidas na memória EEPROM, então se você desligar seu Arduino os dados serão mantidos na memória e na próxima vez que você ligar eles estarão do mesmo jeito que você deixou.





## Configurações do Sistema

Você pode alterar qualquer valor de configuração digitando na linha de comando o **\$** mais o número do parâmetro que você quer alterar e informar o valor a ser inserido.

### **\$X = valor (Salva o valor no Grbl)**

O **\$x=valor** salva ou altera um valor de configuração no Grbl, o que pode ser feito manualmente, enviando este comando quando conectado ao Grbl através de um programa terminal serial.

Para alterar manualmente, por exemplo, a opção **\$0** para 10us você deve digitar:

### **\$0 = 10 (depois dar um enter)**

Se tudo correu bem, o Grbl vai responder com um **'ok'** e essa configuração é armazenada na memória EEPROM e ficará lá para sempre ou até que seja alterada novamente. Você pode verificar se o Grbl recebeu e armazenou sua nova configuração corretamente, digitando **\$\$** para ver as configurações do sistema novamente.

**\$0=10** (step pulse, usec)  
**\$1=25** (step idle delay, msec)  
**\$2=0** (step port invert mask:00000000)  
**\$3=6** (dir port invert mask:00000110)  
**\$4=0** (step enable invert, bool)  
**\$5=0** (limit pins invert, bool)  
**\$6=0** (probe pin invert, bool)  
**\$10=3** (status report mask:00000011)  
**\$11=0.020** (junction deviation, mm)  
**\$12=0.002** (arc tolerance, mm)  
**\$13=0** (report inches, bool)  
**\$20=0** (soft limits, bool)  
**\$21=0** (hard limits, bool)  
**\$22=0** (homing cycle, bool)  
**\$23=1** (homing dir invert mask:00000001)  
**\$24=50.000** (homing feed, mm/min)  
**\$25=635.000** (homing seek, mm/min)  
**\$26=250** (homing debounce, msec)  
**\$27=1.000** (homing pull-off, mm)  
**\$100=314.961** (x, step/mm)  
**\$101=314.961** (y, step/mm)  
**\$102=314.961** (z, step/mm)  
**\$110=635.000** (x max rate, mm/min)  
**\$111=635.000** (y max rate, mm/min)  
**\$112=635.000** (z max rate, mm/min)  
**\$120=50.000** (x accel, mm/sec<sup>2</sup>)  
**\$121=50.000** (y accel, mm/sec<sup>2</sup>)  
**\$122=50.000** (z accel, mm/sec<sup>2</sup>)  
**\$130=225.000** (x max travel, mm)  
**\$131=125.000** (y max travel, mm)  
**\$132=170.000** (z max travel, mm)







## Parâmetros do Grbl v0.9j e seus significados

**NOTA:** A numeração de Configurações mudou desde a v0.8c.

### **\$0 = Tamanho do Pulso em microsegundos**

Os drivers controladores de motor são projetados para entender o passo com um determinado tamanho de pulso. Verifique o data sheet ou apenas faça testes com alguns valores. Você precisa de pulsos curtos mas que sejam reconhecidos pelo seu driver. Se os pulsos forem muito longos você pode ter problemas ao executar o sistema em altas taxas de alimentação de pulsos, pois os pulsos de passos podem começar a se sobreporem uns aos outros. Recomendamos algo em torno de 10 microsegundos, que é o valor padrão.

### **\$1 = Atraso da Inatividade do Motor em milisegundos**

Toda vez que seus motores completam um movimento e chegam a uma parada, o Grbl vai atrasar o desativamento dos motores por este valor. Ou você pode sempre manter seus eixos habilitados (alimentados de forma a manter a posição), definindo esse valor para o máximo de 255 milisegundos. Mais uma vez apenas para fixar você pode manter todos os eixos sempre habilitados pela configuração \$1=255.

O atraso da inatividade do motor é o tempo de duração que o Grbl irá manter os motores travados antes de desativar. Dependendo do sistema, você pode definir isso para zero e desativá-lo. Em outros, você pode precisar de 25 - 50 milisegundos para garantir que seu eixo chegue a uma parada completa antes de desativá-lo.





## \$2 = Máscara para Inversão do Sinal dos Passos

Este item faz a inversão do sinal de pulso. Por padrão, um sinal começa de nível baixo e vai a alto, depois de um tempo definido em **\$0**, o valor volta para baixo, até o próximo evento do pulso. Quando invertido, o comportamento do pulso muda de alto para baixo depois de um tempo definido em **\$0** volta para alto.

A maioria dos usuários não precisará usar essa configuração, mas isso pode ser útil para determinados controladores CNC que têm exigências peculiares.

Esta máscara para inversão dos passos é um valor que armazena os eixos para serem invertidos com bit. Você realmente não precisa entender completamente como funciona. Você só precisa digitar o valor de configurações para os eixos que deseja inverter. Por exemplo, se você quiser inverter os eixos X e Z, você irá enviar \$2=5.

Valor	Máscara	Inverter X	Inverter Y	Inverter Z
0	00000000	Não	Não	Não
1	00000001	Sim	Não	Não
2	00000010	Não	Sim	Não
3	00000011	Sim	Sim	Não
4	00000100	Não	Não	Sim
5	00000101	Sim	Não	Sim
6	00000110	Não	Sim	Sim
7	00000111	Sim	Sim	Sim

## **\$3 = Máscara para Inversão da Direção**

Esta configuração inverte o sinal de sentido para cada eixo. Por padrão, o Grbl assume que os eixos vão se mover em uma direção positiva quando o sinal de sentido é baixo, e uma direção negativa quando o sinal é alto. Muitas vezes, os eixos não se movem desta forma em algumas máquinas.

Esta definição irá inverter o sinal de sentido dos eixos.

Esta máscara para inversão da direção funciona exatamente como a máscara para inversão dos passos. Para configurar essa opção, você só precisa enviar o valor para os eixos que deseja inverter. Use a tabela anterior. Por exemplo, se quiser inverter apenas a direção do eixo Y, você envia \$3=2

## **\$4 = Inverter o Enable do Controlador**

Por padrão, o enable desativa os motores com nível alto e ativa com nível baixo.

Se você precisar inverter essa condição apenas digite \$4=1 se quiser deixar como padrão deixe \$4=0 (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).



## **\$5 = Inverter os Limites**

Por padrão, os limites estão normalmente em alta com resistência de pull-up interno do Arduino. Quando um limite é baixo, o Grbl interpreta isso como disparado. Para o comportamento oposto, basta inverter os limites digitando \$5=1. Desativar com \$5=0. (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).

NOTA: Se inverter seu limite, você precisará de um resistor pull-down externo para todos os limites para evitar a sobrecarga e fritá-los.

## **\$6 = Inverter o Probe**

Por padrão, o probe é realizado normalmente em alta com resistência de pull-up interno do Arduino. Quando o probe é baixo, o Grbl interpreta isso como disparado. Para o comportamento oposto, basta inverter o probe pela digitação de \$6=1. Desativar com \$6=0. (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).

NOTA: Se você inverter o probe, você precisará de um resistor pull-down externo com o probe para evitar a sobrecarga e fritá-lo.





## \$10 = Relatório de Status

Esta configuração determina que o Grbl retorne em tempo real um relatório de status. Por padrão o Grbl irá enviar de volta o seu estado de execução (não pode ser desligado), a posição da máquina, e do trabalho (posição da máquina com coordenadas as compensações e outras compensações aplicadas). Três recursos de relatórios adicionais estão disponíveis, que são úteis para interfaces ou usuários que criam suas máquinas, que incluem o buffer serial RX, o uso do bloco do buffer, e os estados limites (como alto ou baixo, mostrado na ordem ZYX).

Para configurá-los, use a tabela abaixo para determinar quais os dados que você gostaria que o Grbl enviasse de volta. Some os valores dos tipos que você gostaria de ver nos relatórios de status. Este é o valor que você usa para enviar para o Grbl. Por exemplo, se você precisa de posição de máquina e de trabalho, some os valores 1 e 2 e envie ao Grbl \$10=3. Ou, se você precisa da posição da máquina somente e estado dos limites, some os valores 1 e 16 e envie ao Grbl \$10=17.

Em geral, é melhor utilizar essa configuração somente se necessário pois eleva a utilização de recursos desnecessariamente.

Tipo de Relatório	Valor
Posição da Máquina	1
Posição de Trabalho	2
Buffer	4
RX Buffer	8
Limites	16







## **\$11 = Desvio de Junções**

O desvio de junção é usado pelo sistema que controla a aceleração para determinar quão rápido ele pode mover-se através de junções de segmentos gerados pelo programa g-code. Por exemplo, se o seu programa g-code tem uma curva acentuada de 10 graus chegando e a máquina está se movendo na velocidade máxima, esta definição ajuda a determinar o quanto a máquina precisa desacelerar para ir com segurança sem perder passos.

Calcular isso é um pouco complicado, mas em geral, valores mais elevados dão movimentos mais rápidos, enquanto aumenta o risco de perder passos e posicionamento. Os valores mais baixos fazem com que o sistema que controla a aceleração seja mais cuidadoso levando a máquina a se mover mais lentamente. Então se você quer que sua máquina faça as curvas mais rapidamente você deve aumentar esse valor para acelerar o movimento.

## **\$12 = Tolerância de Arco em milímetros**

O Grbl faz círculos G2 e G3, subdividindo-os em linhas menores, tal que a precisão do traçado nunca seja inferior a este valor. Você provavelmente nunca terá de ajustar esta definição, uma vez que 0,002 milímetros está bem abaixo da precisão da maioria das máquinas CNC.

Mas se você achar que seus círculos são realizados lentamente, ajuste essa configuração. Os valores mais baixos vão dar maior precisão, mas pode levar a problemas de desempenho, os valores mais elevados levam a uma precisão inferior, mas pode acelerar o desempenho.







## **\$13 = Relatório em Polegadas**

O Grbl tem um recurso de relatório de posicionamento em tempo real para fornecer um feedback ao usuário de onde a máquina está exatamente nesse momento, bem como, os parâmetros para coordenar as compensações e probe. Por padrão, ele é definido para relatar em mm, mas enviando um comando `$13=1`, o relatório será exibido em polegadas. `$13=0` para voltar para mm.

## **\$20 = Limites Através do Software**

Os limites através do software são um recurso de segurança para ajudar a prevenir que sua máquina tente se deslocar além dos limites disponíveis de curso, fazendo com que bata ou quebre alguma coisa. Ele funciona por saber os limites máximos de cada eixo e sabendo a posição atual de trabalho da máquina, quando um novo código g-code é enviado para o Grbl, ele vai verificar se é possível efetuar o movimento sem causar acidentes excedendo os limites da máquina. Configurando os limites de software o Grbl emitirá um bloqueio imediato de avanço desligando todos os movimentos e exibindo um alarme indicando o problema.

NOTA: Limites através do software requer que homing seja habilitado e as configurações máximas de cada eixo também, porque o Grbl precisa saber onde ele está. `$20=1` para habilitar, e `$20=0` para desabilitar.





## \$21 = Limites de Hardware

Limites de hardware tem basicamente a mesma função que limites de software, mas usam chaves fim de curso para controlar de forma física os movimentos. Basicamente você conecta algumas chaves (mecânicas, magnéticas ou ópticas) perto do final do curso de cada um dos eixos, ou onde quer que você ache necessário que o movimento seja interrompido. Quando o interruptor dispara, ele irá parar imediatamente todo o movimento e entrar em modo de alarme, o que o obriga a você verificar a sua máquina.

Para utilizar os limites de hardware com o Grbl, os limites são mantidos em nível alto com um resistor pull-up interno, então tudo que você tem a fazer é colocar um interruptor normalmente aberto com o terra e definir limite de hardware com  $\$21=1$ . (Desativar com  $\$21=0$ ) Aconselhamos a tomar medidas de prevenção de interferência elétrica. Se você quiser um limite para ambas as extremidades do curso do eixo, ligue os fios em paralelo assim qualquer um deles irá acionar o limite.

Tenha em mente que um evento de limite de hardware é considerado um evento crítico, onde os motores param imediatamente. O Grbl não terá nenhum feedback sobre onde e qual o posicionamento da sua máquina e vai entrar em um modo de loop infinito ALARME, dando-lhe a oportunidade de verificar sua máquina e forçá-lo a redefinir o Grbl. Lembre-se que é puramente um recurso de segurança.





## \$22 = Ciclo de Homing

Para aqueles que estão iniciados no mundo do CNC, o ciclo de homing é usado para localizar uma posição conhecida e consistente em uma máquina cada vez que você iniciar o seu Grbl entre as sessões. Em outras palavras, você sabe exatamente onde você está em um determinado momento, de cada vez. Digamos que você começa a usinagem de algo ou está prestes a iniciar a próxima etapa em um trabalho e você precisa parar, quando reiniciar o Grbl ele não terá idéia de onde ele estava. Se você tem homing, você tem sempre o ponto de referência zero da máquina, a partir daí o que você tem a fazer é executar o ciclo de homing e retomar de onde parou.

Para configurar o ciclo de homing para o Grbl, você precisa ter interruptores de limite em uma posição fixa, ou então o seu ponto de referência fica confuso. Normalmente, eles são configurados no ponto mais distante em X+, Y+, e Z+ de cada um dos eixos. Por padrão, no ciclo de homing o Grbl move o eixo Z positivo primeiro para limpar a área de trabalho e, em seguida, move os eixos X e Y ao mesmo tempo no sentido positivo.

Quando o homing está habilitado o Grbl vai bloquear todos os comandos g-code até que você execute um ciclo de homing. Nenhum eixos se movimentará, a menos que o bloqueio seja desativado com (\$X).

\$22=1 habilita o homing e \$22=0 volta a desativar.

\$x = desabilita mensagem de alerta e \$h = Procura o homing

**NOTA:** Confira o config.h para mais opções de homing para usuários avançados. Você pode desativar o bloqueio de homing na inicialização, configurar quais os eixos que se movem pela primeira vez durante um ciclo de homing e em que ordem, e muito mais.



## **\$23 = Máscara para Inverter a Direção do Homing**

Por padrão, o Grbl assume que os seus interruptores de limite estão no sentido positivo, o primeiro movimento será no eixo Z positivo, então os próximos movimentos serão simultaneamente nos eixos X e Y positivos até localizar seus limites de curso. Se a sua máquina tem um interruptor de limite na direção negativa, a máscara para inverter a direção do homing pode inverter a direção dos eixos. Ele funciona exatamente como a inversão do passo e da direção, onde tudo que você tem a fazer é enviar o valor correspondente da tabela para indicar quais os eixos que deseja inverter e procurar na direção oposta.

\$23 = Usar a tabela de mascaras

## **\$24 = Homing Feed em mm/min**

Ao iniciar o ciclo de homing sua máquina irá procurar o interruptor de fim de curso em uma velocidade mais alta, depois de encontrá-los, move-se a uma velocidade mais baixa até chegar ao ponto zero da máquina. Defina essa opção para um valor que seja possível voltar ao ponto zero sempre com precisão e repetibilidade.

## **\$25 = Homing Seek em mm/min**

É a velocidade em que sua máquina irá procurar as chaves de fim de curso. Ajuste para a maior velocidade possível mas que garanta que a máquina não colida muito fortemente com os interruptores.





## **\$26 = Homing Debounce em milisegundos**

Sempre que um interruptor de fim de curso for acionado é necessário verificar se o acionamento foi de fato pela máquina ter encontrado o fim de curso ou se o acionamento foi dado através de ruído elétrico ou mecânico, ou seja, bounce é o sinal alto e baixo em milisegundos antes que ele se estabilize.

Para resolver isso você precisa estabilizar o sinal que poder ser por hardware com algum tipo de condicionador de sinal ou por software com um pequeno atraso da chegada do sinal.

Defina este valor de atraso para que o seu sistema possa obter homing repetível. Na maioria dos casos, de 5-25 milisegundos é bom.

## **\$27 = Homing Pull-Off em milímetros**

Quando sua máquina completar o homing ela poderá voltar um determinado valor em milímetros para impedir que o acionamento dos fim de cursos sejam acionados acidentalmente caso eles estejam configurados em ambos os lados do seu eixo.







## \$100, \$101 e \$102 = X,Y e Z Passos por Milímetros

O Grbl precisa saber quantos passos são necessários para mover sua ferramenta. Para calcular os passos/mm para um eixo da sua máquina você precisa saber:

- Quantos **mm** a sua máquina se moveu após uma revolução.
- Quantos passos para seu motor fazer uma revolução (normalmente 200)
- Quantos micropassos você configurou em seu controlador (tipicamente 1, 2, 4, 8 ou 16).

Dica: Utilizar valores altos de micropassos (por exemplo, 16) pode reduzir o torque do motor de passo, então use o menor que lhe dá a resolução do eixo desejado e as propriedades de funcionamento confortáveis.

**Os passos/mm pode ser calculados da seguinte forma:**

$$\text{passos\_por\_mm} = \frac{(\text{passos\_por\_revolução} * \text{micropassos})}{\text{mm\_percorridos\_revolução}}$$

Calcule este valor para cada eixo e escreva essas configurações para o Grbl.





## **\$110, \$111 e \$112 = X,Y e Z Taxa Máxima de mm/min**

Isso define a taxa máxima que cada eixo pode se mover. Sempre que o Grbl executa um movimento, ele verifica antes se o movimento a ser executado irá ultrapassar a taxa máxima. Isto significa que cada um dos eixos tem a sua própria velocidade independente, que é extremamente útil para a limitação do eixo Z tipicamente mais lentos.

A maneira mais simples para determinar esses valores é testar cada eixo um de cada vez, aumentando lentamente as configurações de taxa e movê-lo. Por exemplo, para testar o eixo X, enviar ao Grbl algo como G0 X50 com suficiente distância de deslocamento de modo a que o eixo acelera a sua velocidade máxima.

Você saberá que você bateu o limiar de taxa máxima quando seus motores pararem de funcionar. Ele vai fazer um pouco de barulho, mas não irá estragar seu motor. Introduza um ajuste de 10-20% abaixo deste valor, então você pode testar novamente até chegar a uma taxa adequada. Em seguida, repita para os outros eixos.

NOTA: Esta definição de taxa de máxima também define a velocidade do G0.





## **\$120,\$121 e \$122 = X,Y e Z Aceleração em mm/s<sup>2</sup>**

Isso define os parâmetros de aceleração dos eixos em mm/segundo ao quadrado.

Com um valor mais baixo o Grbl fará movimentos mais lentos, enquanto um valor maior produz movimentos mais rápidos. Muito parecido com o ajuste da taxa máxima, cada eixo tem o seu próprio valor de aceleração e são independentes uns dos outros. Isto significa que um movimento multi-eixo só irá acelerar levando em consideração o valor do menor eixo.

A maneira mais simples para determinar os valores para esta configuração é testar individualmente cada eixo aumentando lentamente valores até atingir o limite do motor. Em seguida, finalizar a sua configuração de aceleração com um valor de 10-20% abaixo deste valor máximo absoluto.

É altamente recomendável que você teste com alguns programas g-code pois as vezes a carga em sua máquina é diferente quando se deslocam em todos os eixos juntos.

## **\$130,\$131 e \$132 = X,Y e Z Tamanho da Área Útil em mm**

Isso define o curso máximo de ponta a ponta para cada eixo em milímetros. Isto só é útil se você tem Limites de software (e homing) ativado, pois isso só é usado pelo recurso de limites de software do Grbl para verificar se você excedeu seus limites da máquina com um comando de movimento.



## Obrigado

Espero que esse material tenha sido útil para que você aprimore seu conhecimento e faça a sua CNC funcionar da melhor forma possível.

Caso queira mais informações assista o canal Atividade Maker no youtube, lá você vai encontrar mais informações sobre o Grbl v0.9j.

## Como entrar em contato?

Você pode entrar em contato pelos seguintes meios:



**[www.atividademaker.com.br](http://www.atividademaker.com.br)**



**[www.facebook.com/atividademaker](https://www.facebook.com/atividademaker)**



**[www.youtube.com/c/AtividadeMakerOficial](https://www.youtube.com/c/AtividadeMakerOficial)**



**[contato@atividademaker.com.br](mailto:contato@atividademaker.com.br)**

