

## 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.

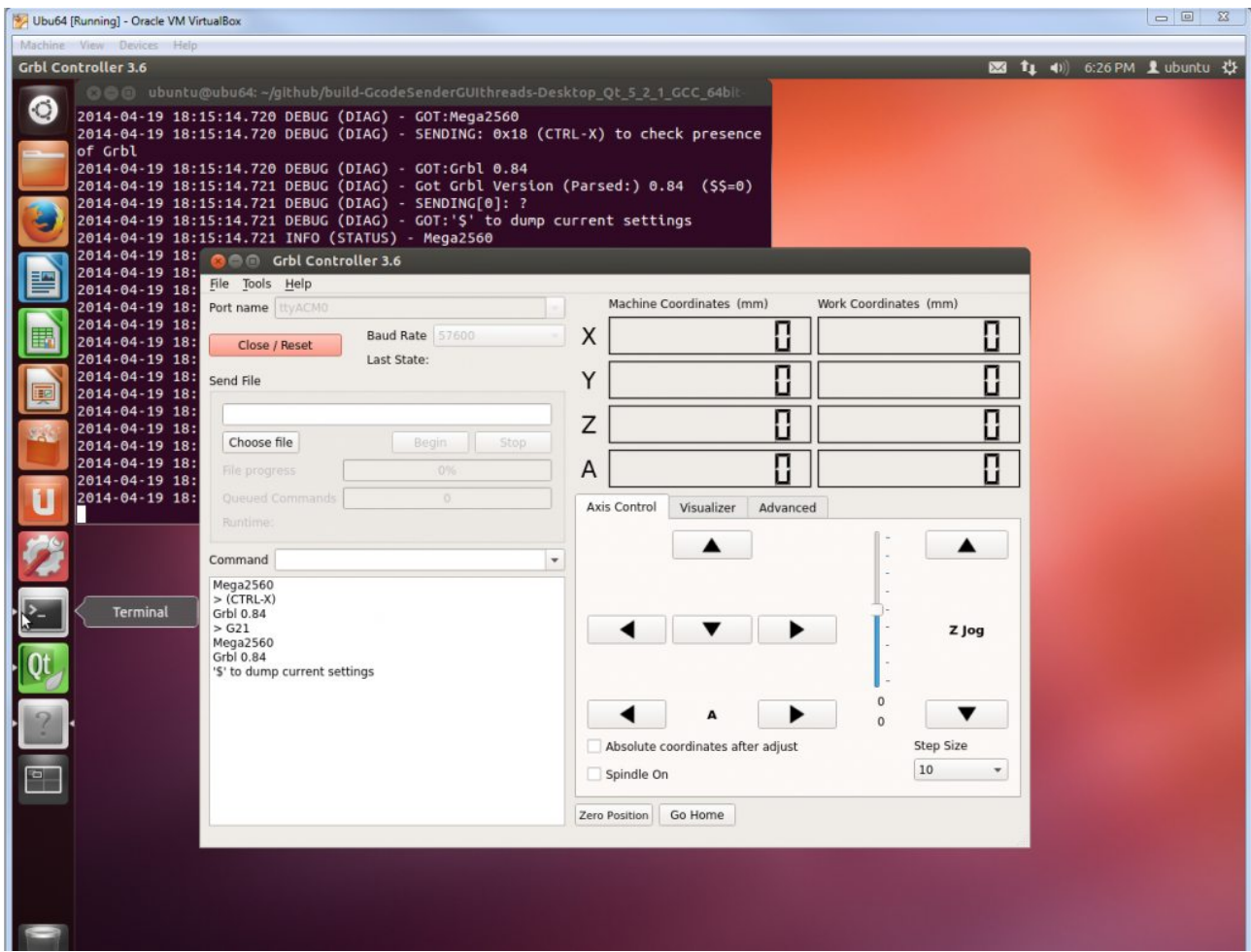
Restart the Linux virtual machine – if Windows 7 you should see a “Device Driver Installation” popup appear with “Unidentified Device” – wait about 5 minutes or so for it to finish. It should find “VirtualBox USB” and say Ready to Use”.

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 FRANÇAIS

## 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.

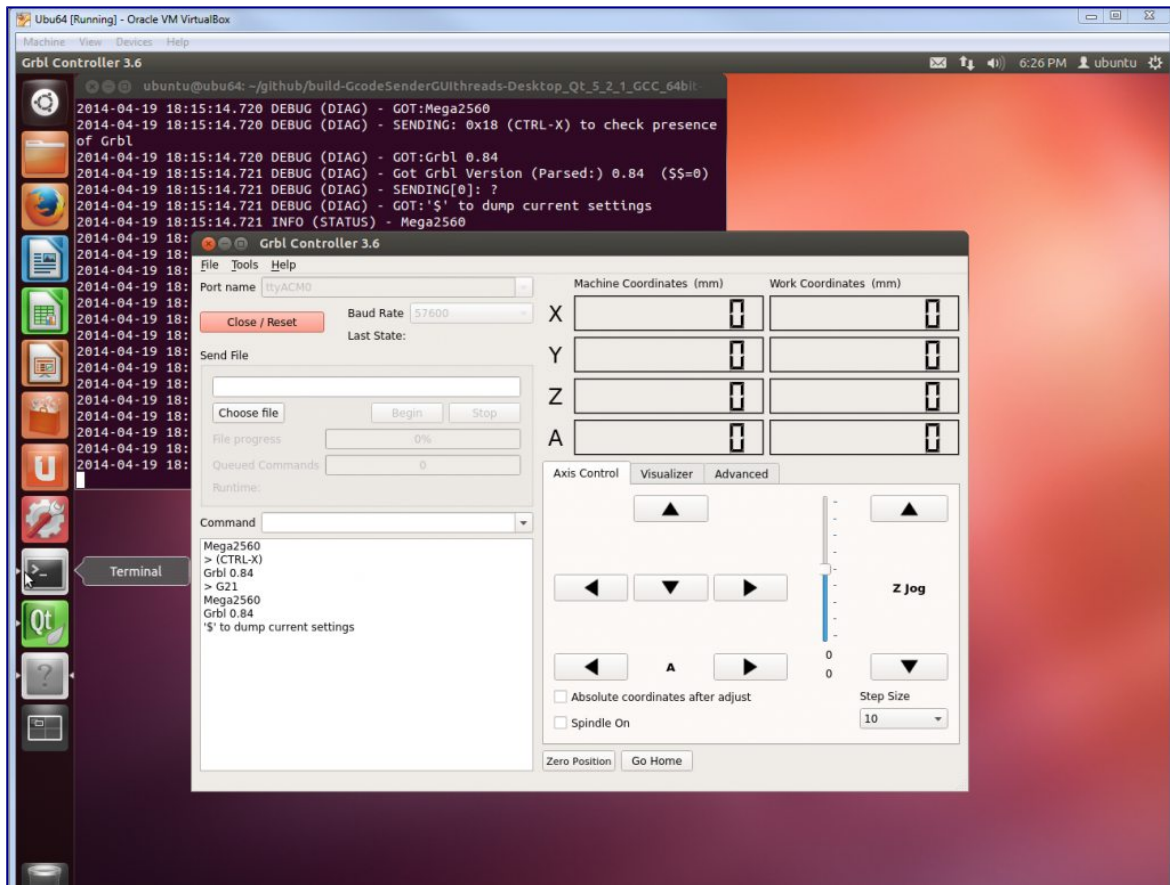
Après l'exécution, vous devrez choisir le bon port série USB. Dans la liste déroulante des ports disponibles, il peut y avoir un port USB dans le nom ou il peut s'agir de ttyS0.

### 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 version -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