Steps to follow to get access to source code

1. Install virtual box and then ubuntu 20.4 inside virtualbox
2. Type this command in the Ubuntu terminal:
   2.1. `ssh-keygen -t ed25519 -C "your_email@example.com"`
3. Then follow the steps in this link: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#adding-your-ssh-key-to-the-ssh-agent
4. Then follow these 8 steps: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account
5. Send mail to admin@omnirom.net requesting access to omnirom repositories.
6. Next, follow these steps:
   6.1. `sudo apt install curl`
   6.2. `sudo apt-get install python-is-python3`
   6.3. `mkdir ~/bin`
   6.4. `curl https://storage.googleapis.com/git-repo-downloads/repo-1 > ~/bin/repo`
   6.5. `chmod a+x ~/bin/repo`
   6.6. `git config --global user.email "example@gmail.com”`
   6.7. `git config --global user.name  "your username"`
   6.8. `mkdir omni-rom`
   6.9. `cd omni-rom`
   6.10. `python3 ~/bin/repo init -u git://github.com/omnirom/android.git -b android-11`
   6.11. `python3 ~/bin/repo sync -j4`
   6.12. `sudo apt-get install git-core gnupg flex bison build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev libgl1-mesa-dev libxml2-utils xsltproc unzip fontconfig kpartx sudo dosfstools rsync libncurses5 libssl-dev python3-mako unzip`
   6.13. `git clone https://github.com/omnirom/android_device_brcm_rpi4.git -b android-11 device/brcm/rpi4/`
   6.14. `git clone https://github.com/omnirom/proprietary_vendor_brcm.git -b android-11 vendor/brcm/`
   6.15. `export ROM_BUILDTYPE=WEEKLY`
   6.16. `export TEMPORARY_DISABLE_PATH_RESTRICTIONS=true`
   6.17. `export PRODUCT_EXCLUDE_EXTRA_PACKAGES=true`
   6.18. `source build/envsetup.sh`
   6.19. `breakfast rpi4 userdebug`
   6.20. `ls ~/.repo/local_manifests/`
   6.21. If the above directory is not present: `mkdir .repo/local_manifests/`
   6.22. If it contains `roomservice.xml`, continue further or else manually add it to the folder.
      6.22.1. Manually copy `roomservice.xml` to `.repo/local_manifests/`

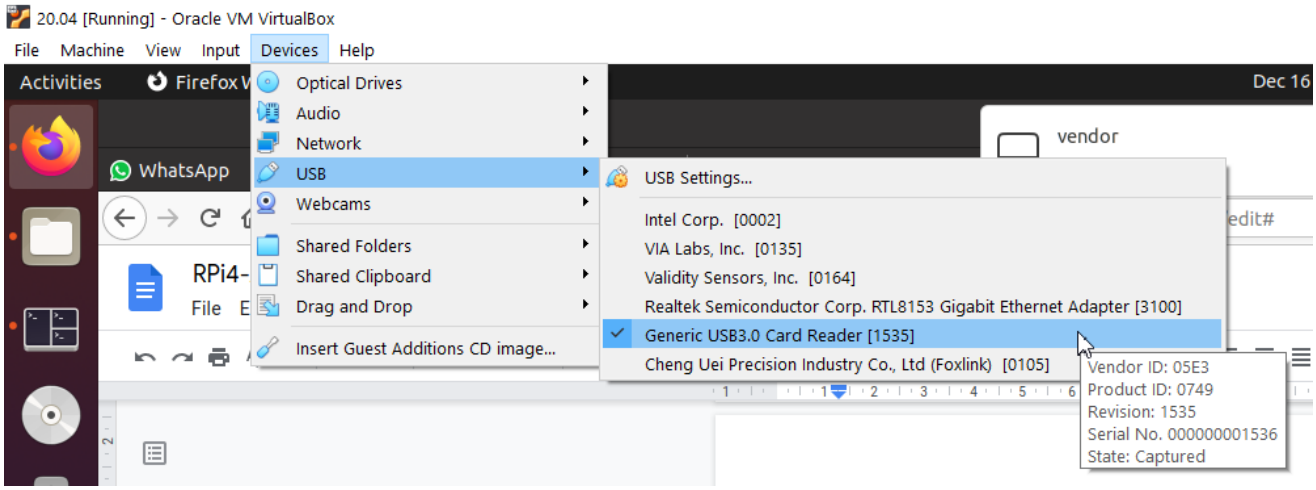
6.22.2. `python3 ~/bin/repo sync --force-sync -j4`

*ctrl + H to show hidden files (useful in finding .repo)
*ctrl + R to search for previous commands easily

6.23. `m -j4`

#build should be completed in 5-6 hours

7. Create an Image by running the script (`mk-raspi64-large-image-isana.sh`). This is slightly technical, reach out to Isana incase of doubts.
   7.1. Variables `IN_IMAGE_DIR, IN_BOOT_FILES` and `OUT_IMAGE_FILE` need to be set to proper paths
   7.2. The loop device used must be numbered correctly. Run `lsblk` to find existing loop devices. The scripts in the directory point to `loop13` and `loop14`.
8. Insert the SD card in the card reader and connect it to the machine (also make sure the sdcard is transferred to the Virtual Machine).



Now you flash the image created in the above step into the sdcard
In this command carefully choose the sd card or else the main hard disk will be wiped out; never choose sda as that is the main hard disk in the system...

This is how your system looks without the sd card inserted

```
pi@pi-vm ~
❯ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0    7:0    0     4K  1 loop /snap/bare/5
loop1    7:1    0 144.6M  1 loop /snap/chromium/1810
loop2    7:2    0 147.8M  1 loop /snap/chromium/1827
loop3    7:3    0  55.5M  1 loop /snap/core18/2246
loop4    7:4    0   219M  1 loop /snap/gnome-3-34-1804/72
loop5    7:5    0  55.5M  1 loop /snap/core18/2253
loop6    7:6    0  65.2M  1 loop /snap/gtk-common-themes/1519
loop7    7:7    0  49.8M  1 loop /snap/snap-store/433
loop8    7:8    0    51M  1 loop /snap/snap-store/547
loop9    7:9    0 164.8M  1 loop /snap/gnome-3-28-1804/161
loop10   7:10   0  62.1M  1 loop /snap/gtk-common-themes/1506
loop11   7:11   0  27.1M  1 loop /snap/snapd/7264
loop12   7:12   0  32.5M  1 loop /snap/snapd/13640
loop13   7:13   0 240.8M  1 loop /snap/gnome-3-34-1804/24
sda      8:0    0   320G  0 disk
├─sda1   8:1    0   512M  0 part /boot/efi
├─sda2   8:2    0     1K  0 part
└─sda5   8:5    0 319.5G  0 part /
sdb      8:16   0   500G  0 disk /home/pi/coder2
sr0     11:0    1  58.3M  0 rom  /media/pi/VBox_GAs_6.1.26
sr1     11:1    1  1024M  0 rom
pi@pi-vm ~
❯
```

Insert the sdcard and your sd card will be named as sdb/sdc/sdd etc.,

```
pi@pi-vm ~
❯ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0    7:0    0     4K  1 loop /snap/bare/5
loop1    7:1    0 144.6M  1 loop /snap/chromium/1810
loop2    7:2    0 147.8M  1 loop /snap/chromium/1827
loop3    7:3    0  55.5M  1 loop /snap/core18/2246
loop4    7:4    0   219M  1 loop /snap/gnome-3-34-1804/72
loop5    7:5    0  55.5M  1 loop /snap/core18/2253
loop6    7:6    0  65.2M  1 loop /snap/gtk-common-themes/1519
loop7    7:7    0  49.8M  1 loop /snap/snap-store/433
loop8    7:8    0    51M  1 loop /snap/snap-store/547
loop9    7:9    0 164.8M  1 loop /snap/gnome-3-28-1804/161
loop10   7:10   0  62.1M  1 loop /snap/gtk-common-themes/1506
loop11   7:11   0  27.1M  1 loop /snap/snapd/7264
loop12   7:12   0  32.5M  1 loop /snap/snapd/13640
loop13   7:13   0 240.8M  1 loop /snap/gnome-3-34-1804/24
sda      8:0    0   320G  0 disk
├─sda1   8:1    0   512M  0 part /boot/efi
├─sda2   8:2    0     1K  0 part
└─sda5   8:5    0 319.5G  0 part /
sdb      8:16   0   500G  0 disk /home/pi/coder2
sdd      8:48   1  14.9G  0 disk
├─sdd1   8:49   1   128M  0 part /media/pi/boot
├─sdd2   8:50   1     2G  0 part /media/pi/_
├─sdd3   8:51   1   256M  0 part /media/pi/vendor
└─sdd4   8:52   1   5.6G  0 part /media/pi/userdata
sr0     11:0    1  58.3M  0 rom  /media/pi/VBox_GAs_6.1.26
sr1     11:1    1  1024M  0 rom
pi@pi-vm ~
❯
```

8.1. Then do this command

```
sudo dd if=~/omni-WEEKLY-13.12.2021-2120.img of=/dev/sdb bs=1M
```