在VMware17 ubuntu 20.04环境下从0开始源码安装UHD和GNU Radio for USRP B210

本篇教程适用于想入手 USRP 搭建通信原型平台进行算法验证的朋友，从一开始出发讲解如何从 0 实现 USRP 通信的一个 demo。同时作为参照自用，方便将来在不同机器上安装进行联合通信。

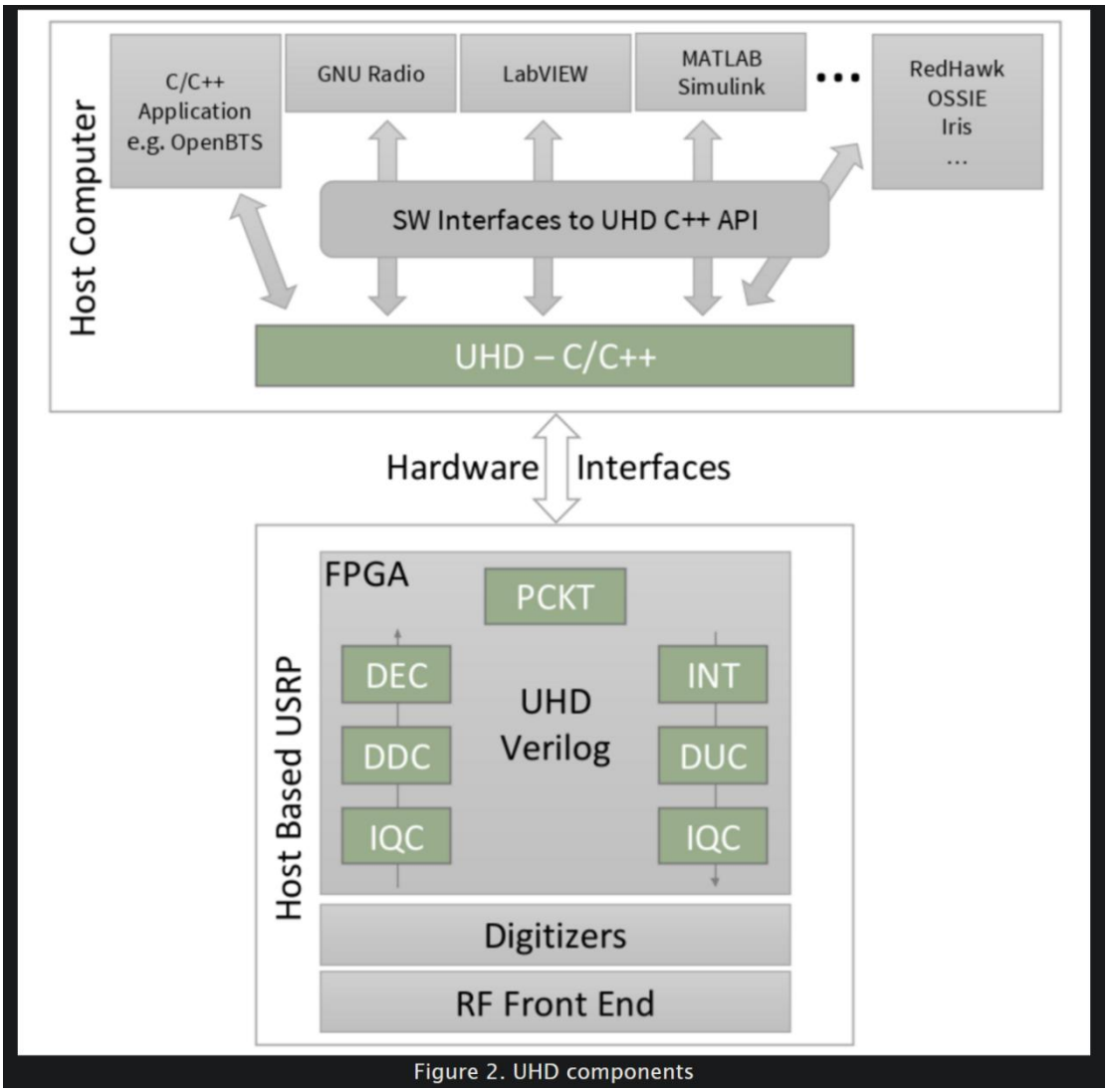


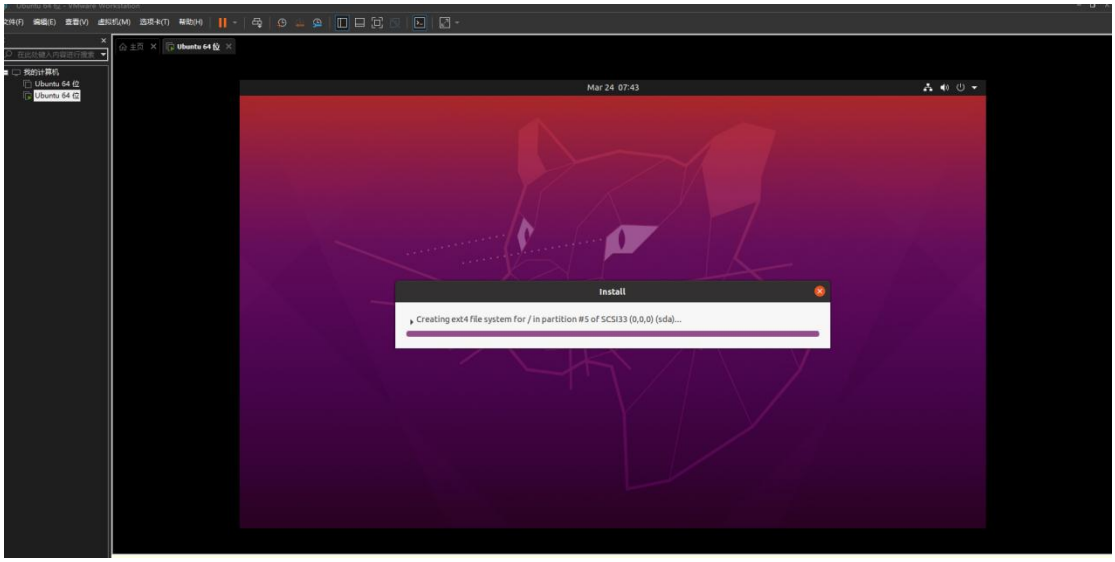
Figure 2. UHD components

- **下载并装载 VMware 虚拟机和 Ubuntu 镜像系统**

由于本人不想安装双系统且用习惯了 win 平台，故选择了虚拟机作为实验平台。首先下载虚拟机软件和 linux 系统镜像，本人采用 Ubuntu-20.04.6 系统作为虚拟机挂载镜像。镜像网站为 https://launchpad.net/ubuntu/+cdmirrors

非常重要的注意事项！！！：安装之前一定要配置虚拟机输入 USB 兼容性为 3.1！！！否则无法识别 usrp 设备！！！

开始配置 ubuntu 系统



等待系统初始化完毕后，进行一些必要的系统更新（非大版本），然后关掉系统更新，防止打扰。接着完成一些便于使用 ubuntu 的配置，这里记录一下主要的配置内容：

- 设置 vmware 双屏工作
- 更换 apt get 源
- 下载中文支持包
- 设置 root 初始密码，用于终端 su 进入 root

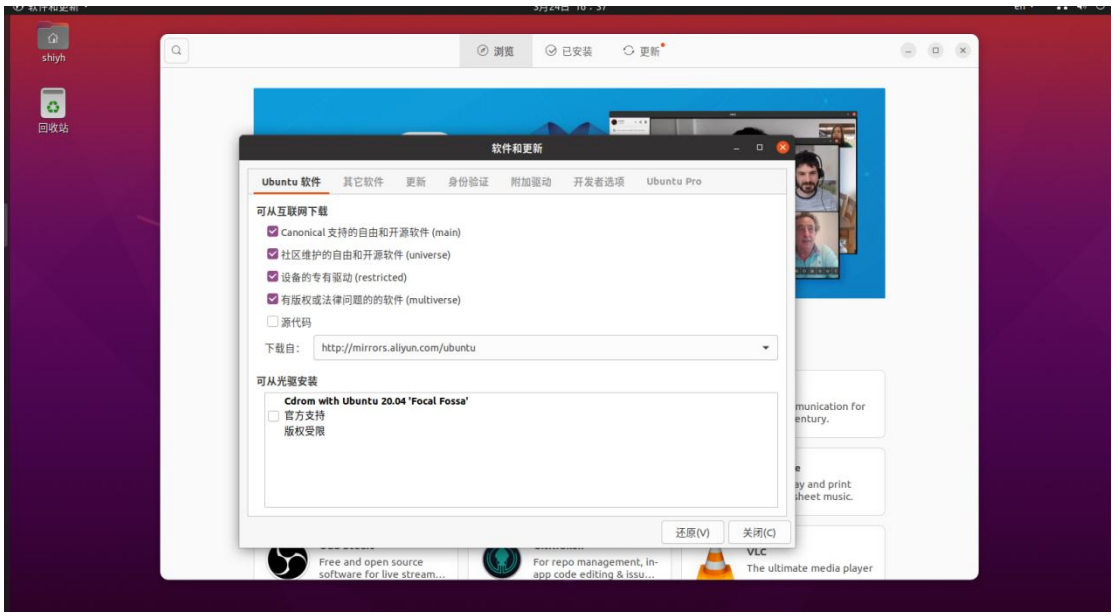
- 设置中文输入法
- 更改时区
- 下载 edge 浏览器（个人喜欢用，需要同步什么的）
- 下载 clash（用于科学上网

设置 vmware 双屏工作：

（注意，必须在关机情况下设置）点击虚拟机，设置，显示器，指定监视器设置，选择数量为 2，确定。后面打开虚拟机后 ctrl+alt+enter 全屏，在上面的悬浮窗中点击循环使用屏幕即可。

更换 apt-get 源：

为了更快的下载速度，我们更换国内的源，这里首先点开所有应用，选择软件与更新，点击 ubuntu 软件，点击下载自，选择中国的服务器，这里我选择南大源（因为人在南大哈哈）。



Ps：

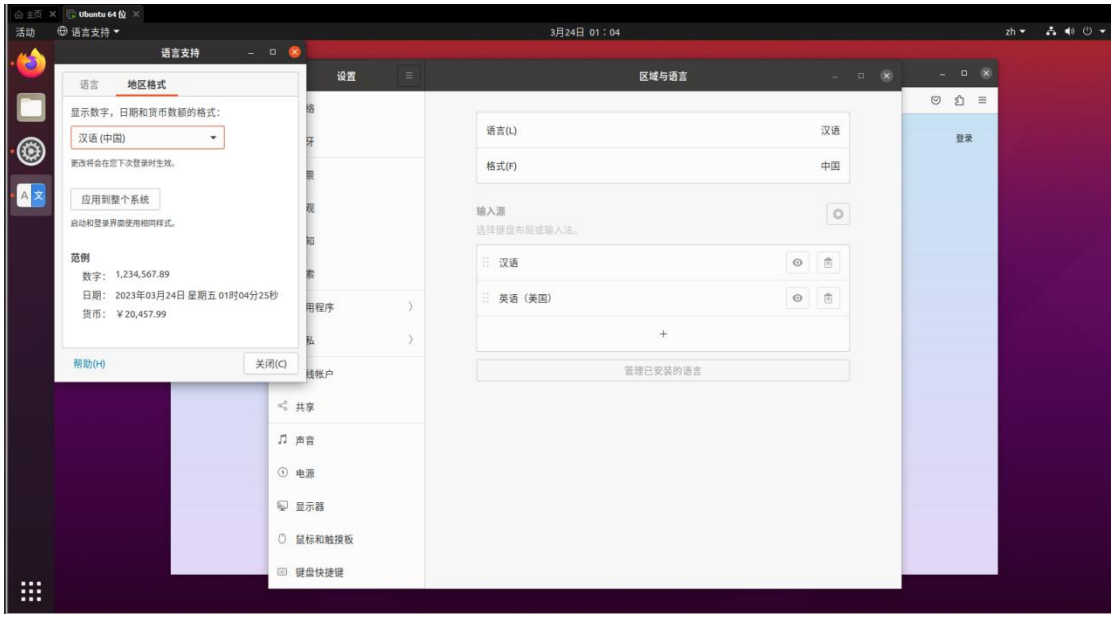
使用 ubuntu 好习惯，时刻更新（但也不要太勤快，防止遇到 bug）：

```
sudo apt-get update

sudo apt-get upgrade
```

下载中文支持包：

点击设置，点击 region&language，点击 language support，然后 install/remove language，选择中文，下载完成后在上端把汉语拖选至最优先，点击 apply system-wide，同时把地区格式选为中国，点击应用到系统，重启，即可显示中文，。



设置 root 初始密码，用于终端 su 进入 root：

ubuntu的su初始密码设置步骤:

偶尔用回到 ubuntu 系统，想切换到su，总是显示不成功，也许是初次使用，即需要设定一下:
使用sudo
$:sudo passwd
系统提示输入密码，即安装时的用户密码，然后，系统提示输入两次新密码，输入完毕后，
$:su
即可进入su，具备了相应的权限了.

设置中文输入法：

参考这篇博客 https://blog.csdn.net/a805607966/article/details/105874756

更改时区：

使用 timedatectl 命令，首先

想要列出所有可用的时区，你可以列出 `/usr/share/zoneinfo` 目录下的所有文件，或者运行 `timedatectl` 命令，加上 `list-timezones` 选项：

```
timedatectl list-timezones
...
America/Montevideo
America/Nassau
America/New_York
America/Nipigon
America/Nome
America/Noronha
...
```

这里选择 Asia/Hong_Kong，故输入

timedatectl set-timezone Asia/Hong_Kong

完成时区切换，可以重启验证是否切换成功。

下载 edge 浏览器：

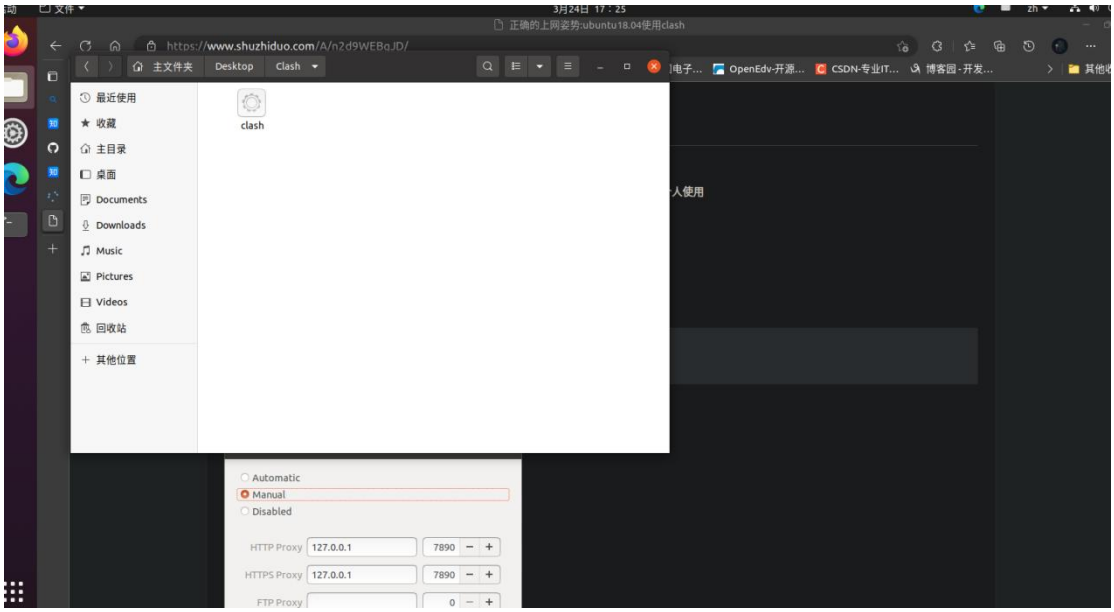
在系统自带火狐浏览器中进入下面链接 https://www.microsoft.com/en-us/edge/business/download?form=MA13DL，下载 edge linux 版本.deb 后缀文件，双击运行安装即可。

登录微软账户即可实现同步。（个人还是非常喜欢 edge 的同步功能）。

下载 clash：

在上述准备做好后，开始 clash 科学上网。下载 clash github 仓库：https://github.com/Dreamacro/clash/releases ，版本选择 clash-linux-amd64-v1.14.0.gz。

在桌面新建文件夹 CLash（用于与后面的 clash 重命名区分），建在桌面的目的是方便每次启动时快速翻墙。将下载压缩文件提取到此处，改名为 clash

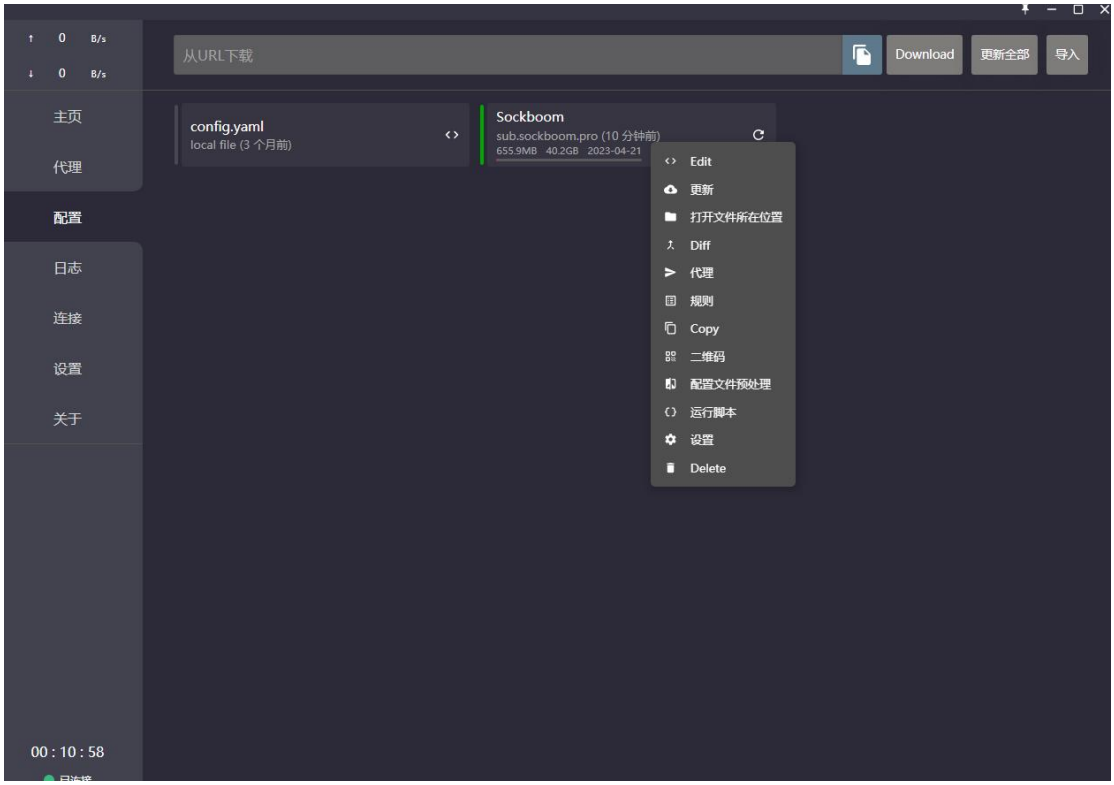


而后进入网络设置，选择更改网络代理为手动，并设置如下参数：

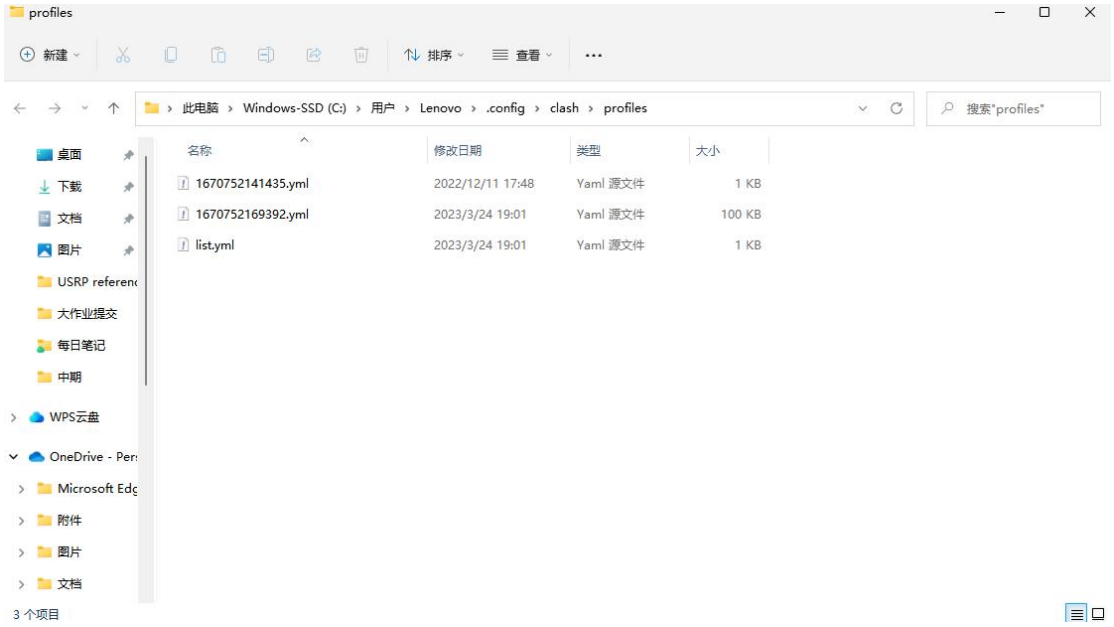
设置完毕后，运行终端并进入 Clash 文件夹下，运行

```
chmod +x clash-linux-amd64-v1.8.0
```

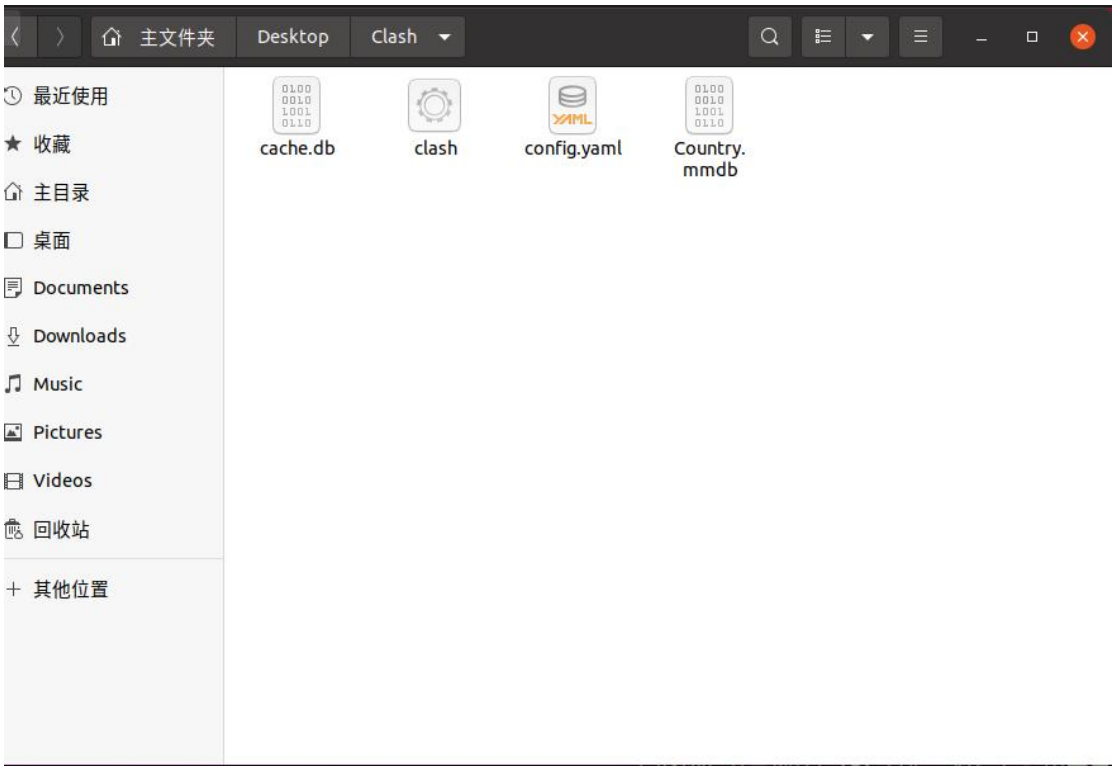
赋予执行权限。Linux 端 clash 需要自己导入配置，可在 windows 的 clash 客户端（如图所示），



选择打开文件所在位置



将默认的文件复制到 Linux 中并改名为 config.yaml，复制到 clash 目录下，再拷入 Country.mmdb



在终端输入 ./clash -d .（后面这个.一定不要忘了）,之后每次开机就输入此命令即可 fq（clash 文件夹下）

即可科学上网，可进入 http://clash.razord.top/#/settings 这个网页以 UI 方式配置 clash。

● 安装 UHD 驱动和 GNU Radio

下面使用 USRP 命令时，最好进入 root 环境，可以免去很多权限不足的问题，虽然网上有很多添加命令至管理员的博客，但是直接使用 su 进入 root 是最快捷方便的。

非常重要的注意事项!!!:安装之前一定要配置虚拟机输入 USB 兼容性为 3.1!!!否则无法识别 usrp 设备！！！

1、 UHD 驱动的安装采用源码编译安装方式，参考教程为 https://files.ettus.co

m/manual/page_build_guide.html，

首先新建文件夹 workspace，安装依赖包

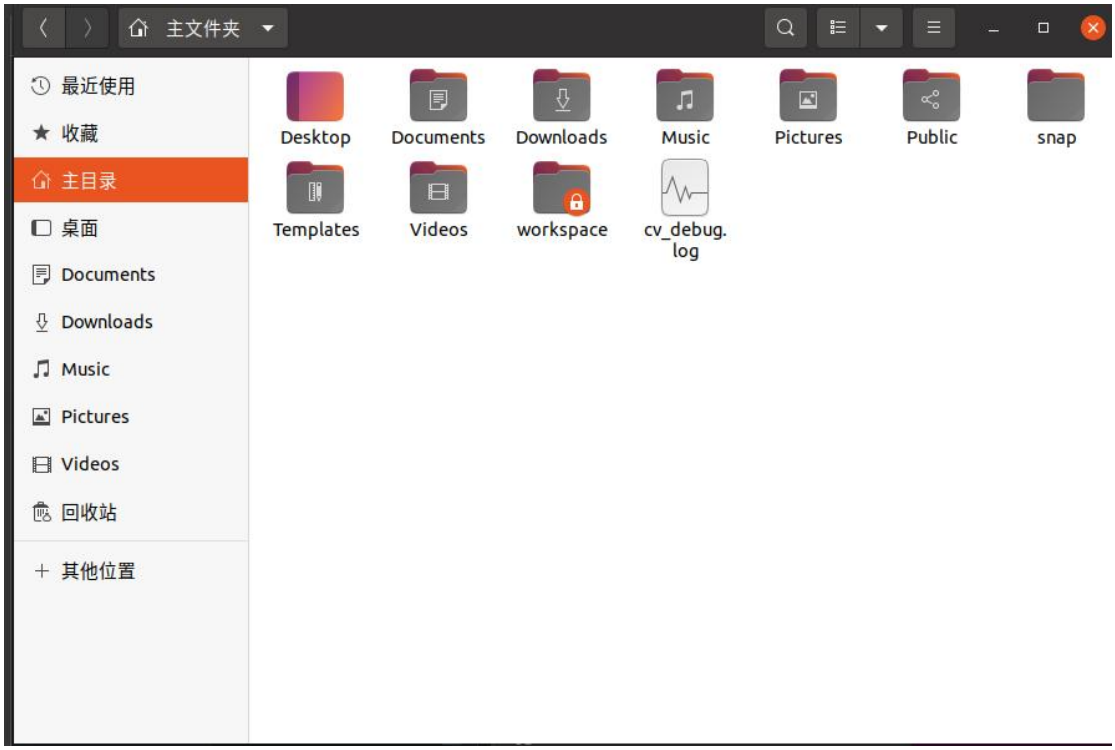
sudo apt-get install autoconf automake build-essential ccache cmake cpufr equtils doxygen ethtool \

g++ git inetutils-tools libboost-all-dev libncurses5 libncurses5-dev libusb-1.0-0 libusb-1.0-0-dev \

libusb-dev python3-dev python3-mako python3-numpy python3-requests pyt hon3-scipy python3-setuptools \

python3-ruamel.yaml

sudo ldconfig

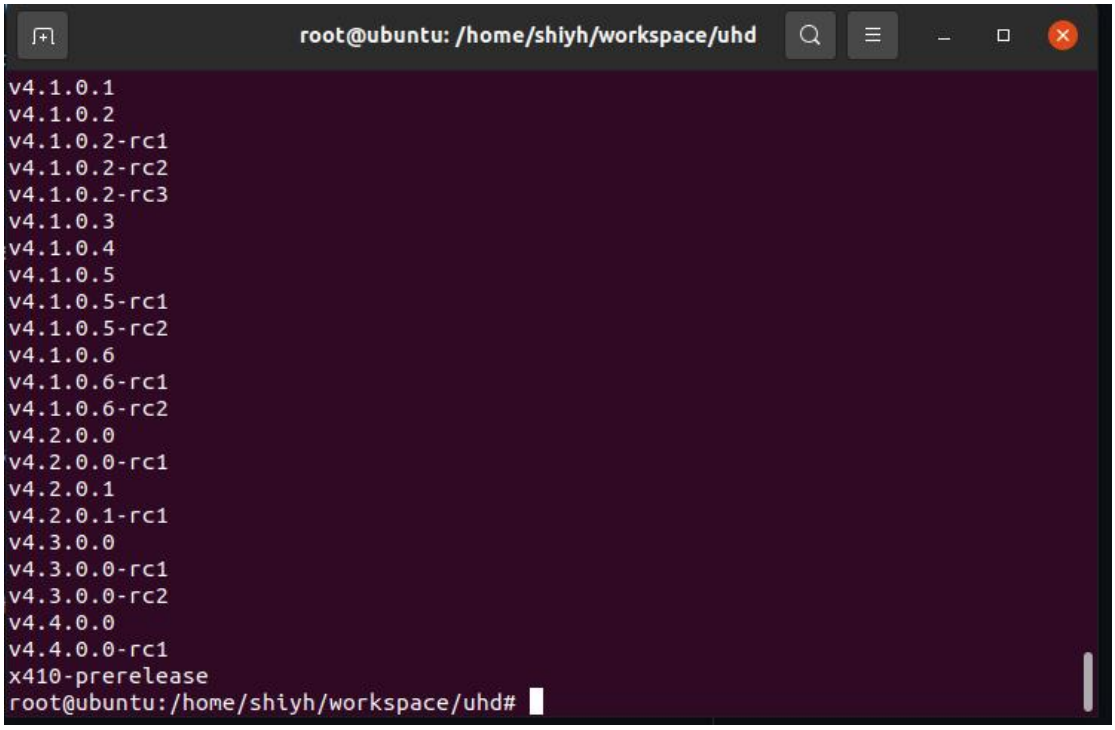


进入 workspace 后再 git 源代码

git clone https://github.com/EttusResearch/uhd.git

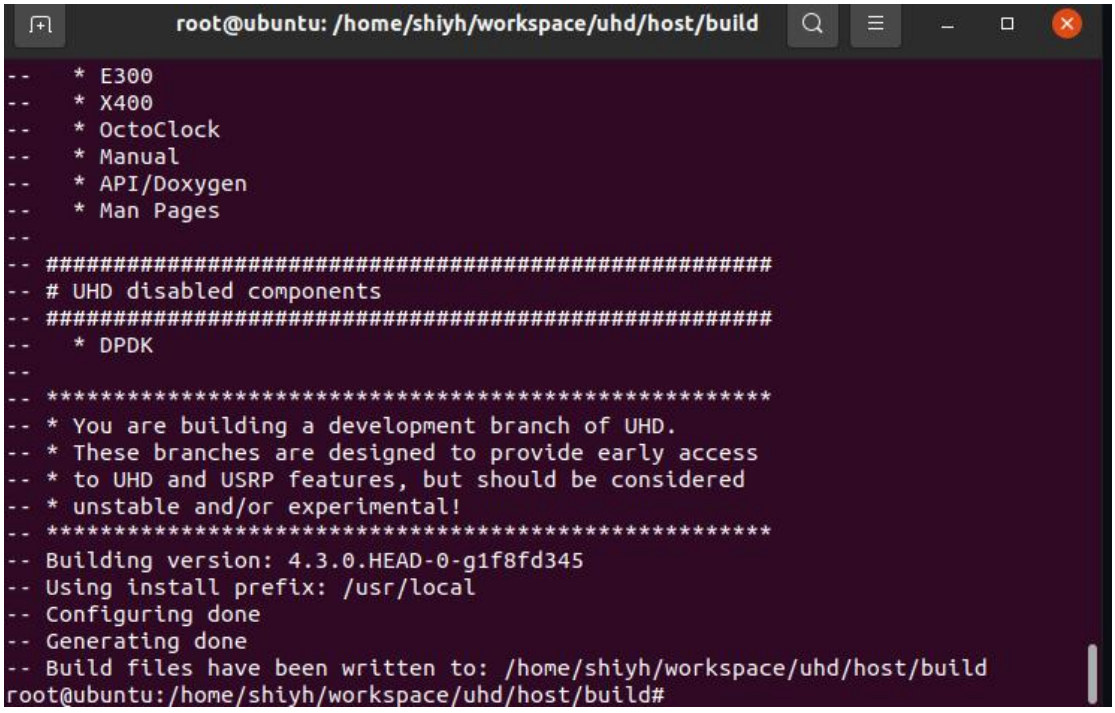
cd uhd

使用 git tag -l 查看已有 release 版本，选择一个版本，这里我选择 4.5（2023.12.13 更新）版本



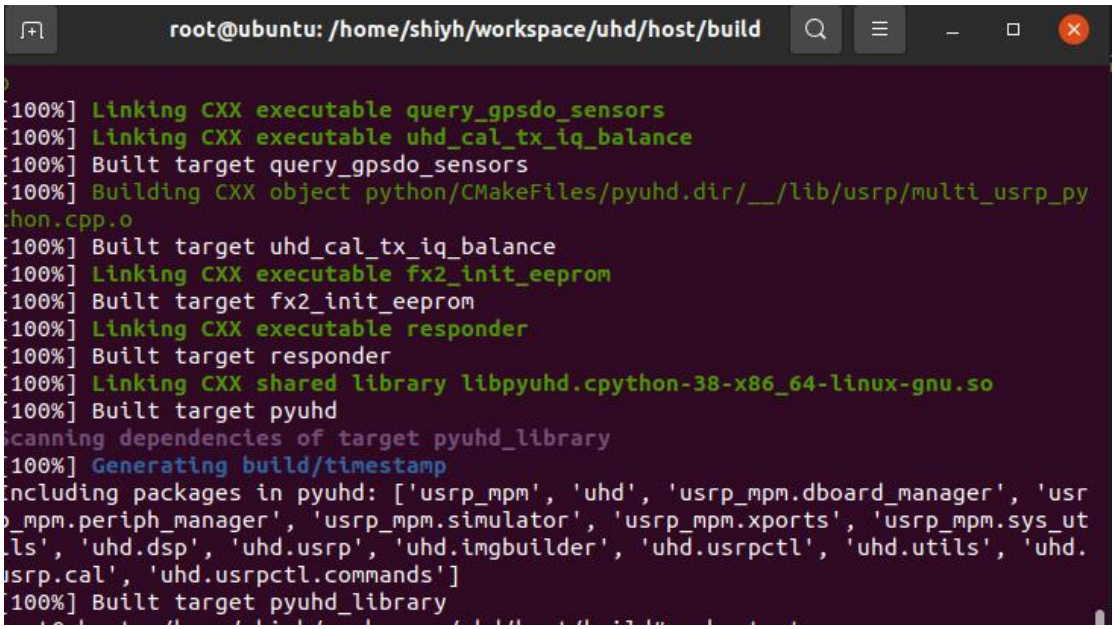
git checkout v4.5.0.0
cd host
mkdir build
cd build
cmake ../

成功后应该输出：



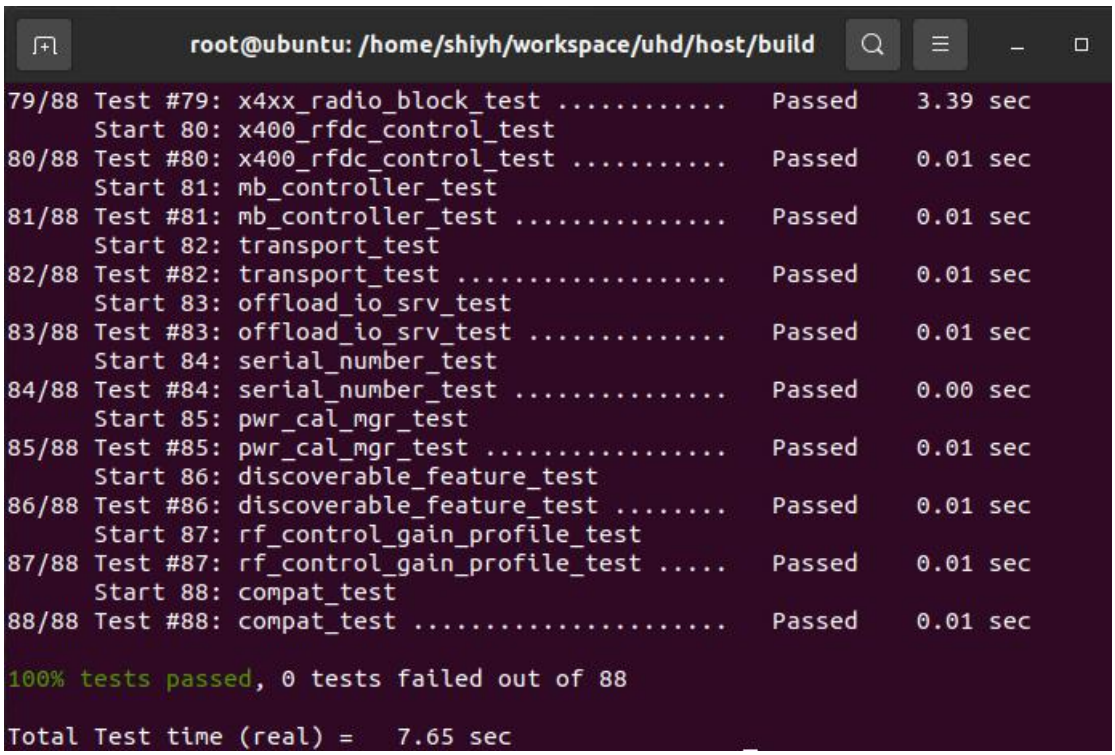
为了加快 make 编译速度，使用-j 指令使用多核编译

make -j8



等待一段时间后编译完成使用 test 测试

make test



输入

sudo make install

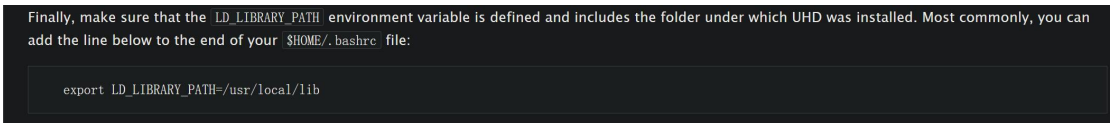
如果您运行的是 Linux，请始终记住在安装任何库后执行以下命令：

sudo ldconfig

最后，确保定义了 LD_LIBRARY_PATH 环境变量（在 usr 环境和 root 环境下）

并包含安装 UHD 的文件夹。

添加环境变量至 usr：可以将以下行添加到 $HOME/.bashrc 文件的末尾



export LD_LIBRARY_PATH=/usr/local/lib

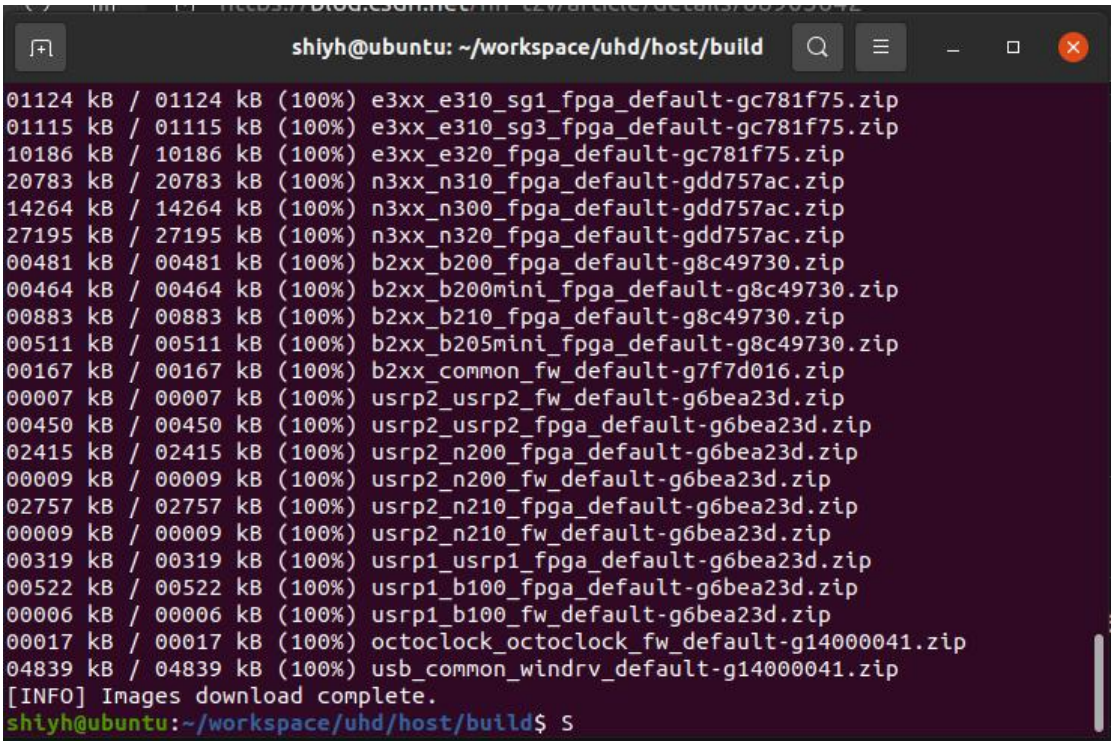
添加环境变量至 root：将上述行添加到 /root/.bashrc 文件的末尾

重启系统即可。

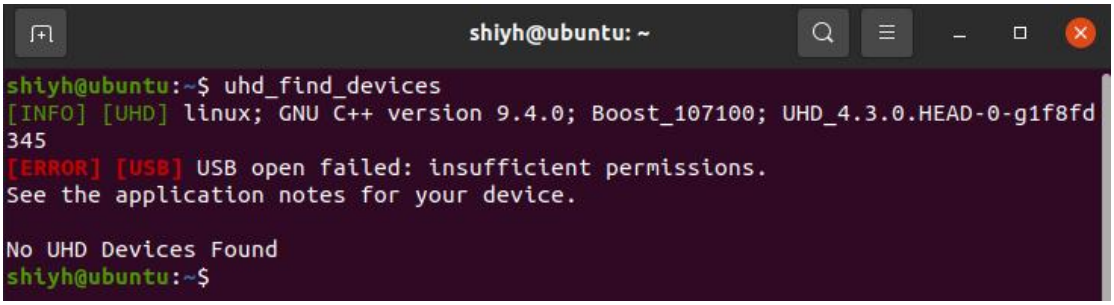
现在可以下载此安装的 UHD FPGA images。通过运行命令 uhd_images_dow

nloader 来完成

sudo uhd_images_downloader



uhd_find_devices 如果找到设备就是 uhd 驱动安装成功了，如果出现



是权限不够(安装 gnuradio 最后有解决办法)，使用 ~~sudo uhd_find_devices~~ 或~~一开始 su 进入 root 环境即可。~~

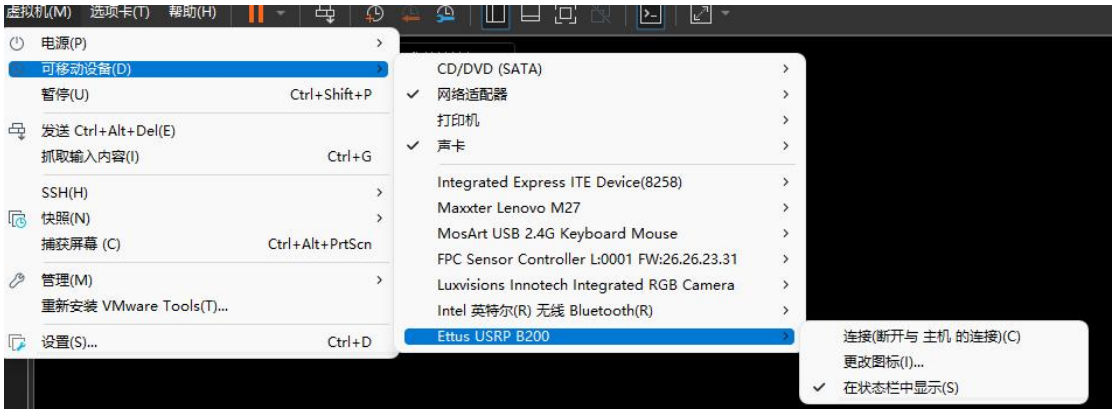
若无论如何都是 No UHD Devices Found，以下为解决步骤

首先说明本人的硬件设备情况，使用 USRP B210，win 11 主机，VMware® Workstation 17 Pro 虚拟机，ubuntu 20.04.6 LTS 系统，具体硬件可以有差别，思路是一样的。

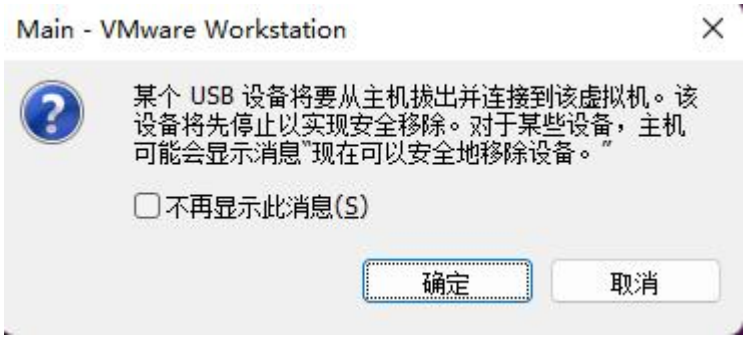
首先插入 usrp 设备，在 win 系统下查看设备管理器，检查是否识别该设备，正确识别的结果为

其余的一切情况，如感叹号，其他设备等出现，没有 USRPs 等皆为电脑未识别 USRP 设备，需要安装 windows 下的驱动（其实理论上来讲，这种行为是在虚拟机中操控 B210 的妥协，因为本质上 USRP 并不会考虑 win 的驱动是否有用，连接时直接选择 Ubuntu 系统，只要 linux 中成功安装了 uhd 即可识别。但 vmware 虚拟机是建立在 win 主机的前提下的，所以 win 的识别与否会直接影响 vmware 中虚拟机系统能否成功接入 usrp 设备，本人之前就是没管 win 系统的驱动结果始终无法正确识别。） windows 的 b210 驱动在网上搜索即可，在电脑文件夹 D:\usrp_b210_driver 中有我的驱动。安装好驱动重启重插设备即可。

安装好驱动后，设备管理器应该正常如上图显示。这时候进入虚拟机测试，看能否识别。如果还不行，检查 vmware 顶部菜单栏 虚拟机=>可移动设备



点击“连接(断开与主机的连接）”，这个时候会弹出



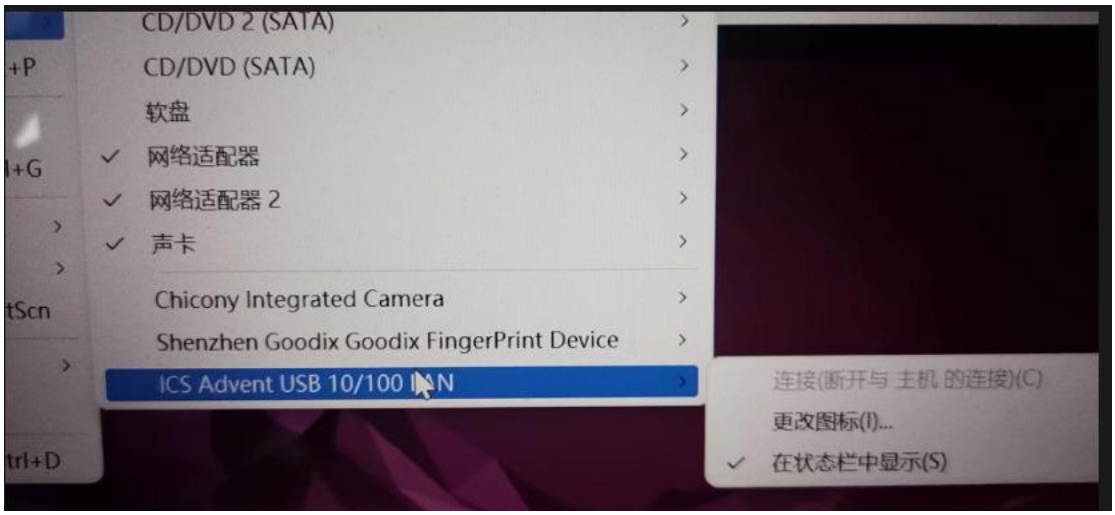
点击确定后，应该能听到系统提示音滴答响了，说明设备已经切换到 ubuntu 系统中，再次 uhd_find_devices 测试即可输出

```
root@ubuntu:/home/shiyh# uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.4.0; Boost_107100; UHD_4.3.0.HEAD-0-g1f8fd345
--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    serial: 31C9260
    name: MyB210_1
    product: B210
    type: b200
```

使用 uhd_usrp_probe 检测 uhd 版本与 usrp 设备固件版本是否兼容，输出如下，可以看到各种 dsp、adc、dac 发送接收端硬件信息等

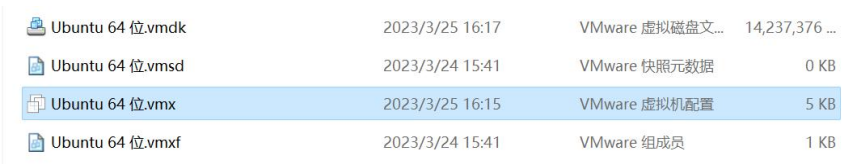
```
root@ubuntu:/home/shiyh# uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.4.0; Boost_107100; UHD_4.3.0.HEAD-0-g1f8fd345
No UHD Devices Found
root@ubuntu:/home/shiyh# uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 9.4.0; Boost_107100; UHD_4.3.0.HEAD-0-g1f8fd345
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
  _____________________________________________________
 /
|       Device: B-Series Device
|     _____________________________________________________
|    /
|   |       Mboard: B210
|   |   serial: 31C9260
|   |   name: MyB210_1
|   |   product: 2
|   |   revision: 4
|   |   FW Version: 8.0
|   |   FPGA Version: 16.0
|   |
|   |   Time sources:  none, internal, external, gpsdo
|   |   Clock sources: internal, external, gpsdo
|   |   Sensors: ref_locked
|   |     _____________________________________________________
|   |    /
|   |   |       RX DSP: 0
|   |   |
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     _____________________________________________________
|   |    /
|   |   |       RX DSP: 1
|   |   |
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     _____________________________________________________
|   |    /
|   |   |       RX Dboard: A
```

注意：如果开始时虚拟机可移动设备里的连接是灰色的，



原因可能是 VMware 17 本身的 bug，还是说在安装虚拟机时不小心哪儿配置出问题了，在虚拟机的配置文件里有 usb.restrictions.defaultAllow = "FALSE"这么一个配置项，使得虚拟机无法使用 USB 设备。

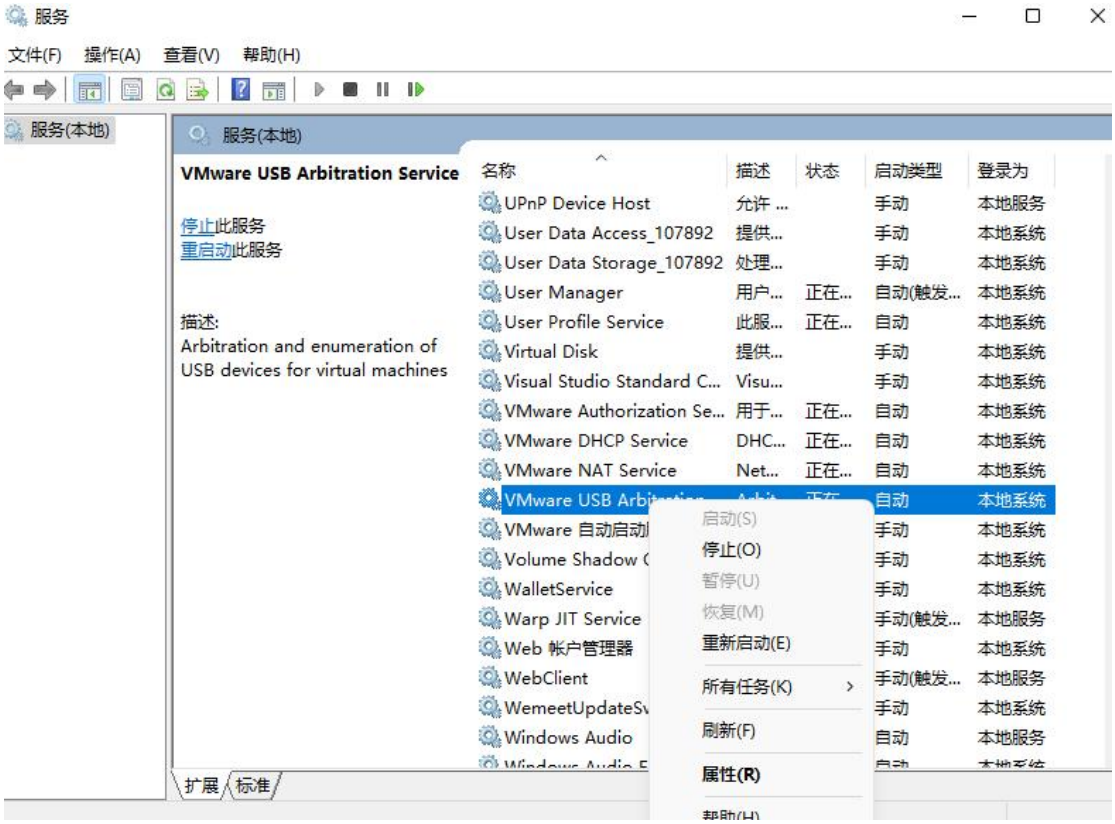
解决方法是关闭运行的虚拟机，打开虚拟机的安装目录，而不是 vmware 的，找到



使用 vscode 打开，ctrl+f 查找 usb.restrictions.defaultAllow，将其值改为 TRUE，

如果没有就自己加一行，保存后重启虚拟机，可以发现连接按钮亮了。再按上述步骤来就行。

如果没有找到 usb 设备，右键桌面上的计算机 此电脑-》管理-》服务，检查主机系统中 VMware USB Arbitration Service 能否正常启动。找到 VMware USB Arbitration Service 看看是否是正在运行，一般默认是手动启动。那么右键选择启动，直接重启虚拟机就能看到上文的 USB 设备了。重新启动也可以。



至此，算是完成虚拟机中 linux 系统的 uhd 安装了，并且成功测试了与 usrp 设备的连接和固件匹配情况，接下来安装 gnuradio 或其他“app”

2、 安装 gnuradio，参考 gnu radio 官方 wiki 文档 https://wiki.gnuradio.org/index.php?title=LinuxInstall

~~本文安装的 gnuradio 版本为 Release v3.10.5.0 发布时间为 2022 年 12 19，而 uhd 版本为 4.3 发布时间是 2022 9 16，~~（2023.12.13 更新）本文安装的 gnuradio

版本为 Release v3.10.8.0 发布时间为 2023 年 10-20，而 uhd 版本为 4.5 发布时间是 2023-9-13，这里 gnuradio 版本的发布时间最好比 uhd 晚，因为 gnuradio 为上层应用，需要 uhd 驱动的底层支持，新版本的 gnuradio 会兼容早发布的 uhd 但反过来不一定，所以为了避免可能出现的 bug，按此规则安装。

首先安装依赖包：

sudo apt install git cmake g++ libboost-all-dev libgmp-dev swig python3-numpy python3-mako python3-sphinx python3-lxml doxygen libfftw3-dev libsdl1.2-dev libgsl-dev libqwt-qt5-dev libqt5opengl5-dev python3-pyqt5 liblog4cpp5-dev libzmq3-dev python3-yaml python3-click python3-click-plugins python3-zmq python3-scipy python3-gi python3-gi-cairo gir1.2-gtk-3.0 libcodec2-dev libgsm1-dev

而后

sudo apt install pybind11-dev python3-matplotlib libsndfile1-dev python3-pip libsoapysdr-dev soapysdr-tools

而后

pip install pygccxml
pip install pyqtgraph

而后

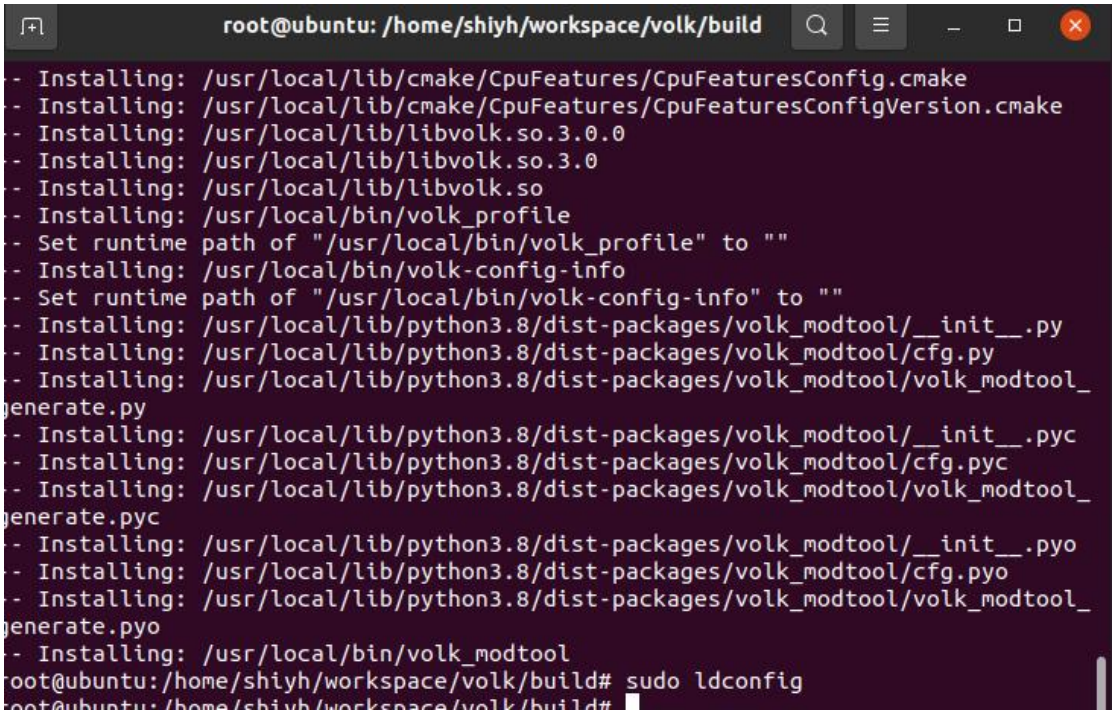
sudo apt install libiio-dev libad9361-dev libspdlog-dev python3-packaging python3-jsonschema
sudo ldconfig

依赖安装完毕

然后安装 volk，官方给出的解释为

> Since Volk is no longer considered as a submodule of GNU Radio (GNU Radio commit #80c04479da962d048d41165081b026aafdaa0316), you **MUST FIRST** install Volk, and then install GNU Radio.

cd workspace
git clone --recursive https://github.com/gnuradio/volk.git
cd volk
mkdir build
cd build
cmake ../
make -j8
make test
sudo make install



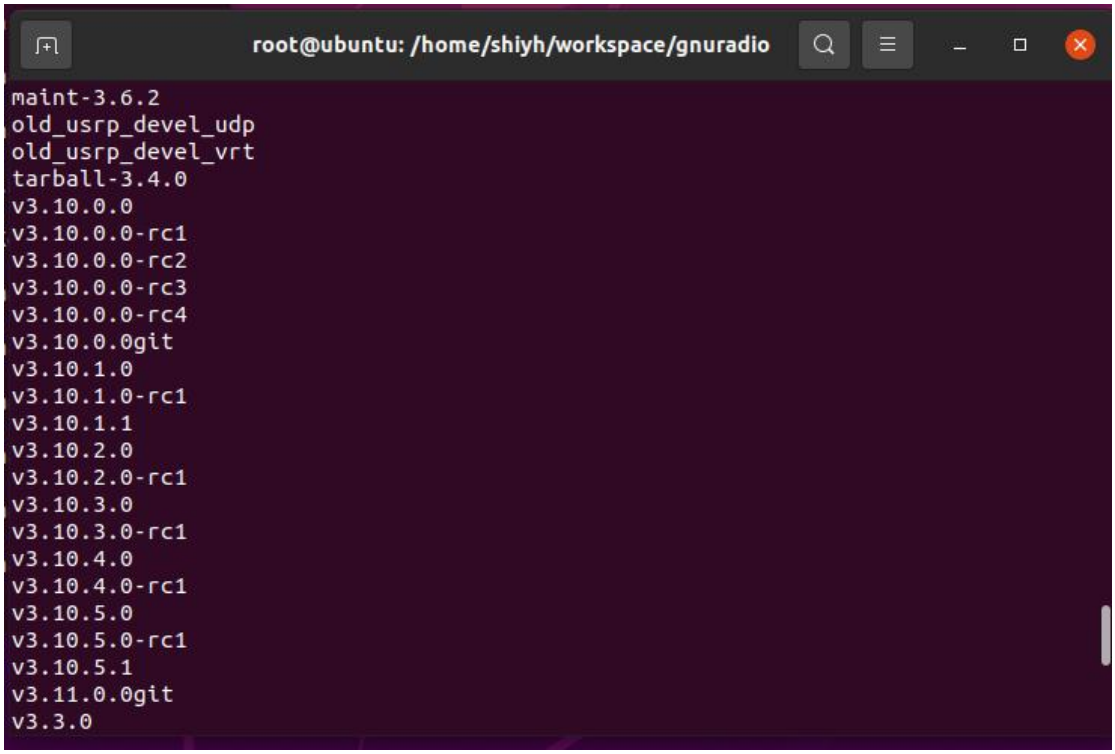
如果您运行的是 Linux，请始终记住在安装任何库后执行以下命令：

sudo ldconfig

开始正式安装 gnuradio

cd workspace
git clone https://github.com/gnuradio/gnuradio.git
cd gnuradio

使用 git tag -l 查看已有 release 版本，选择一个版本，这里我选择 v3.10.5.0 版本




```
git checkout v3.10.5.0
mkdir build
cd build
cmake ../
make -j8
```

make test (不一定全通过)

```
sudo make install
sudo ldconfig
```

此时，GNU Radio 应该已安装并准备使用。您可以通过运行以下快速测试，在不连接 USRP 设备的情况下快速测试这一点。

```
gnuradio-config-info --version
gnuradio-config-info --prefix
gnuradio-config-info --enabled-components
```

您可以运行一个简单的流程图，它不需要任何 USRP 硬件。它称为拨号音测试，它在计算机的扬声器上产生 PSTN 拨号音。运行它会验证是否可以找到所有库，以及 GNU Radio 运行时是否正常工作。

```
python3 workarea/gnuradio/gr-audio/examples/python/dial_tone.py
```

您可以尝试启动 GNU Radio Companion （GRC） 工具，这是一个用于构建和运行 GNU Radio 流程图的可视化工具。

gnuradio-companion

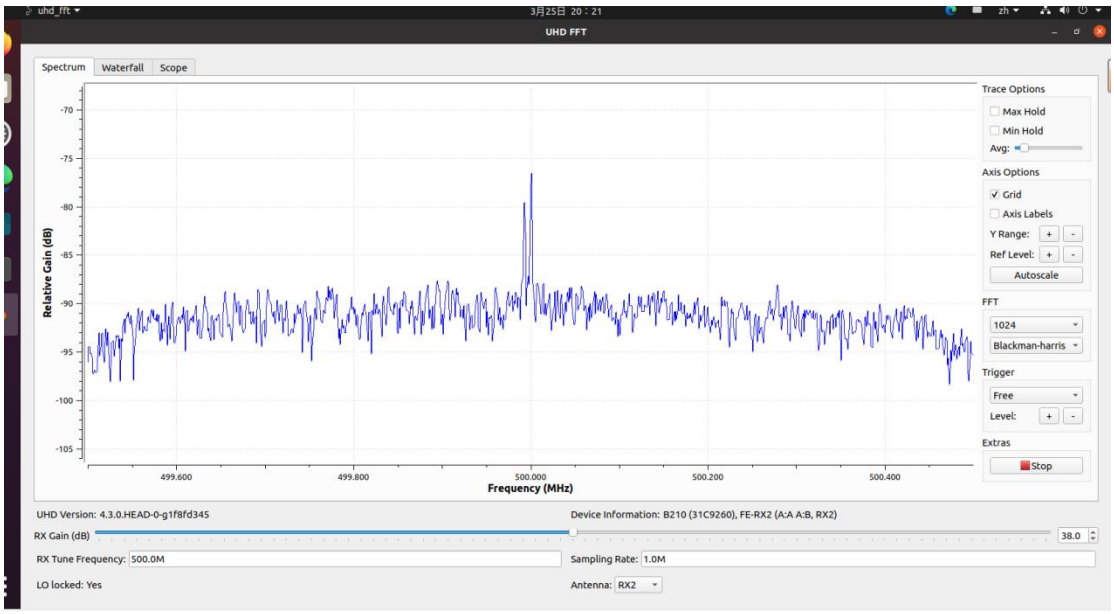
在 Linux 上，udev 处理 USB 插头和拔出事件。以下命令安装 udev 规则，以便非 root 用户可以访问设备。此步骤仅对使用 USB 连接到主机的设备（如 B200、B210 和 B200mini）是必需的。此设置应立即生效，不需要重新启动或注销/登录。运行这些命令时，请确保没有 USRP 设备通过 USB 连接。

```
cd $HOME/workarea/uhd/host/utils
sudo cp uhd-usrp.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
```

之后便可以在非 root 命令下直接使用 uhd 与 gnuradio 命令访问设备，如 uhd_find_devices

最后，我们进行连接 usrp 的实际测试，参考宗熙师兄的博客 https://blog.csdn.net/weixin_44863193/article/details/122176754?spm=1001.2014.3001.5502，只需要一行代码即可验证：

uhd_fft -f $freq （这里 freq 需自己选，我选了 2.4G，结果如下）



最终，经历了千辛万苦，我们实现了 usrp 开发的第一步，也就是配好了 linux 环境，包括 uhd 驱动和 gnuradio 的安装，算是正式的步入软件无线电的大门啦！

附录：可能出现的问题：

linux 查看、修改文件读写权限方法(切记！！！这个方法不要乱该系统文件夹权限，如/usr,/etc 等，否则会导致很严重的问题，如 root 无法正常进入，如果出现此情况，参考博客 https://blog.csdn.net/qq_19734597/article/details/103401394 及其评论区的一位的方法）

1、使用 ls -la 命令可查看文件夹下权限情况

解析“drwxrwxrwx"，这个权限说明一共 10 位。

第一位代表文件类型，有两个数值：“d”和“-”，“d”代表目录，“-”代表非目录。

后面 9 位可以拆分为 3 组来看，分别对应不同用户，2-4 位代表所有者 user 的权限说明，5-7 位代表组群 group 的权限说明，8-10 位代表其他人 other 的权限说明。

r 代表可读权限，w 代表可写权限，x 代表可执行权限。

"drwxrwxrwx”表示所有用户都对这个目录有可读可写可执行权限。

2、修改权限

r、w、x 也有对应的数字：

r—4
w—2
x—1

```
sudo chmod -R 777 /var/www
```

这行命令就是给“/var/www”这个目录赋予所有人可读可写可执行权限，4+2+1=7。

对应的：

5=4 + 1,表示拥有可读可执行权限，但是没有写权限

0 代表没有任何权限

-rw------- (600) 只有所有者才有读和写的权限

-rw-r–r-- (644) 只有所有者才有读和写的权限，组群和其他人只有读的权限

-rwx------ (700) 只有所有者才有读，写，执行的权限

-rwxr-xr-x (755) 只有所有者才有读，写，执行的权限，组群和其他人只有读和执行的权限

-rwx–x--x (711) 只有所有者才有读，写，执行的权限，组群和其他人只有执行的权限

-rw-rw-rw- (666) 每个人都有读写的权限

无法正常运行 gnuradio python 脚本

绝大多数情况下是权限不够，使用 sudo 命令可以解决。