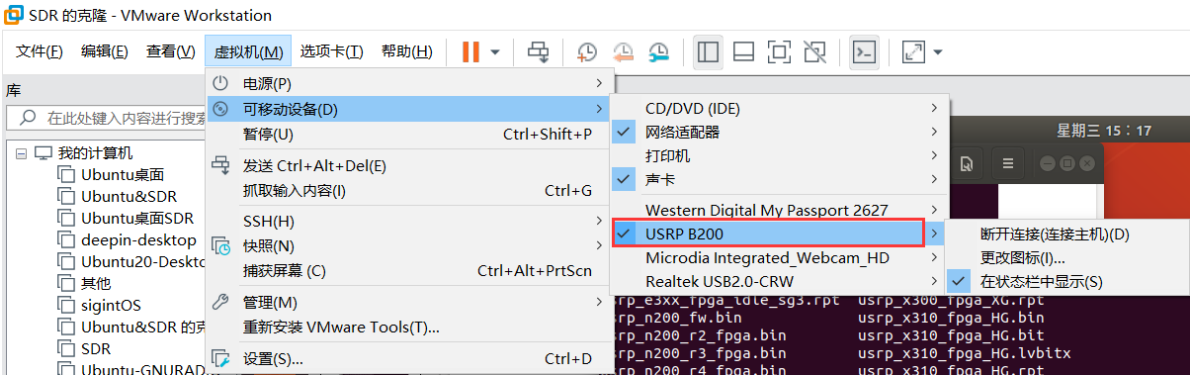
UHD驱动配置(APT方式)

1.终端执行以下命令(不报错)

```
$ sudo apt-get remove -y uhd
$ sudo apt-get remove libuhd-dev libuhd003 uhd-host -y
$ sudo apt-add-repository --remove "deb
http://files.ettus.com/binaries/uhd/repo/uhd/ubuntu/trusty trusty main"
$ sudo add-apt-repository ppa:ettusresearch/uhd -y
$ sudo apt-get update
$ sudo apt-get -y --allow-unauthenticated install python python-tk libboost-
all-dev libusb-1.0-0-dev
$ sudo apt-get -y --allow-unauthenticated install libuhd-dev libuhd003 uhd-
host
```



2.下载镜像文件

- 自动下载安装

```
$ sudo sudo uhd_images_downloader
```

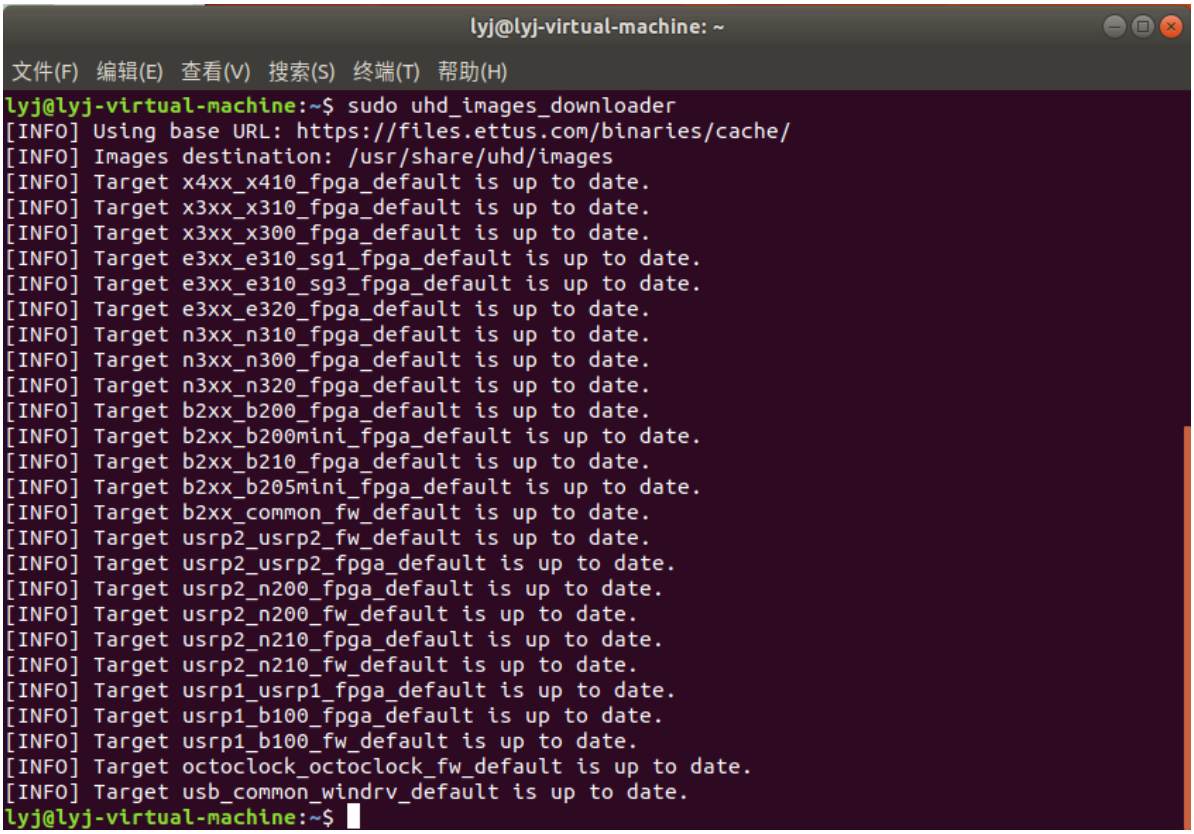
- 手动下载安装

下载地址：https://github.com/EttusResearch/uhd/releases

```
$ cd /usr/share
$ sudo mkdir -p ./uhd
$ sudo mkdir -p ./uhd/images
```

解压下载的images镜像并复制到`usr/share/uhd/images`目录，执行`sudo sudo uhd_images_downloader`

镜像下载更新完成



- 安装完成后终端输入以下命令(源码安装添加`sudo`)，已连接USRP设备应该可以正常识别

```
$ uhd_find_devices
$ uhd_usrp_probe
```

```
lyj@lyj-virtual-machine:~$ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_4.2.0.0-0ubuntu1~bionic1
--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    serial: 20BR10C
    name: 30005018600012
    product: B210
    type: b200


lyj@lyj-virtual-machine:~$ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_4.2.0.0-0ubuntu1~bionic1
[INFO] [B200] Detected Device: B210
[INFO] [B200] Loading FPGA image: /usr/share/uhd/images/usrp_b210_fpga.bin...
[INFO] [B200] Operating over USB 2.
[INFO] [B200] Detecting internal GPSDO....
[INFO] [GPS] No GPSDO found
[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: 20BR10C
|   |   name: 30005018600012
|   |   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
|   |   |     _____________________________________________________
|   |   |    /
|   |   |   |       RX Frontend: A
|   |   |   |   Name: FE-RX2
|   |   |   |   Antennas: TX/RX, RX2
|   |   |   |   Sensors: temp, rssi, lo_locked
|   |   |   |   Freq range: 50.000 to 6000.000 MHz
|   |   |   |   Gain range PGA: 0.0 to 76.0 step 1.0 dB
|   |   |   |   Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |     _____________________________________________________
|   |   |    /
|   |   |   |       RX Frontend: B
|   |   |   |   Name: FE-RX1
|   |   |   |   Antennas: TX/RX, RX2
|   |   |   |   Sensors: temp, rssi, lo_locked
|   |   |   |   Freq range: 50.000 to 6000.000 MHz
|   |   |   |   Gain range PGA: 0.0 to 76.0 step 1.0 dB
|   |   |   |   Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |     _____________________________________________________
|   |   |    /
|   |   |   |       RX Codec: A
|   |   |   |   Name: B210 RX dual ADC
```

```
|   |   |   |   Gain Elements: None
|   |     ____________________________________________________
|   |    /
|   |   |       TX DSP: 0
|   |   |
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     ____________________________________________________
|   |    /
|   |   |       TX DSP: 1
|   |   |
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     ____________________________________________________
|   |    /
|   |   |       TX Dboard: A
|   |   |     ________________________________________________
|   |   |    /
|   |   |   |       TX Frontend: A
|   |   |   |   Name: FE-TX2
|   |   |   |   Antennas: TX/RX
|   |   |   |   Sensors: temp, lo_locked
|   |   |   |   Freq range: 50.000 to 6000.000 MHz
|   |   |   |   Gain range PGA: 0.0 to 89.8 step 0.2 dB
|   |   |   |   Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |     ________________________________________________
|   |   |    /
|   |   |   |       TX Frontend: B
|   |   |   |   Name: FE-TX1
|   |   |   |   Antennas: TX/RX
|   |   |   |   Sensors: temp, lo_locked
|   |   |   |   Freq range: 50.000 to 6000.000 MHz
|   |   |   |   Gain range PGA: 0.0 to 89.8 step 0.2 dB
|   |   |   |   Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |     ________________________________________________
|   |   |    /
|   |   |   |       TX Codec: A
|   |   |   |   Name: B210 TX dual DAC
|   |   |   |   Gain Elements: None
```

安装HackRF Host

- 安装依赖

```
$ sudo apt install build-essential libusb-1.0-0-dev pkg-config libfftw3-dev
```

- 安装hackRF

```
$ git clone https://github.com/mossmann/hackrf.git
$ mkdir host/build
$ cd host/build
$ cmake ..
$ make -j4
$ sudo make install
$ sudo ldconfig
```

```
lyj@lyj-virtual-machine:~/hackrf-2021.03.1/host/build$ cmake ../
-- The C compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check if the system is big endian
-- Searching 16 bit integer
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of unsigned short
-- Check size of unsigned short - done
-- Using unsigned short
-- Check if the system is big endian - little endian
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Checking for module 'libusb-1.0'
--   Found libusb-1.0, version 1.0.21
-- Found LIBUSB: /usr/lib/x86_64-linux-gnu/libusb-1.0.so
-- Looking for include file pthread.h
-- Looking for include file pthread.h - found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Setting udev rule group to - plugdev
-- HackRF udev rules will be installed to '/etc/udev/rules.d' upon running 'make install'
-- Found FFTW: /usr/lib/x86_64-linux-gnu/libfftw3.so
-- Configuring done
-- Generating done
-- Build files have been written to: /home/lyj/hackrf-2021.03.1/host/build
```

```
lyj@lyj-virtual-machine:~/hackrf-2021.03.1/host/build$ make -j4
Scanning dependencies of target hackrf-static
Scanning dependencies of target hackrf
[  5%] Building C object libhackrf/src/CMakeFiles/hackrf-static.dir/hackrf.c.o
[ 10%] Building C object libhackrf/src/CMakeFiles/hackrf.dir/hackrf.c.o
[ 20%] Linking C static library libhackrf.a
[ 20%] Linking C shared library libhackrf.so
[ 20%] Built target hackrf
Scanning dependencies of target hackrf_cpldjtag
Scanning dependencies of target hackrf_spiflash
Scanning dependencies of target hackrf_operacake
[ 20%] Built target hackrf-static
[ 25%] Building C object hackrf-tools/src/CMakeFiles/hackrf_cpldjtag.dir/hackrf_cpldjtag.c.o
[ 30%] Building C object hackrf-tools/src/CMakeFiles/hackrf_spiflash.dir/hackrf_spiflash.c.o
[ 35%] Building C object hackrf-tools/src/CMakeFiles/hackrf_operacake.dir/hackrf_operacake.c.o
Scanning dependencies of target hackrf_sweep
[ 40%] Building C object hackrf-tools/src/CMakeFiles/hackrf_sweep.dir/hackrf_sweep.c.o
[ 45%] Linking C executable hackrf_cpldjtag
[ 50%] Linking C executable hackrf_operacake
[ 55%] Linking C executable hackrf_spiflash
[ 55%] Built target hackrf_cpldjtag
[ 55%] Built target hackrf_operacake
[ 55%] Built target hackrf_spiflash
Scanning dependencies of target hackrf_transfer
Scanning dependencies of target hackrf_info
Scanning dependencies of target hackrf_debug
[ 60%] Building C object hackrf-tools/src/CMakeFiles/hackrf_transfer.dir/hackrf_transfer.c.o
[ 65%] Building C object hackrf-tools/src/CMakeFiles/hackrf_debug.dir/hackrf_debug.c.o
[ 70%] Building C object hackrf-tools/src/CMakeFiles/hackrf_info.dir/hackrf_info.c.o
[ 75%] Linking C executable hackrf_sweep
[ 80%] Linking C executable hackrf_info
[ 85%] Linking C executable hackrf_debug
[ 90%] Linking C executable hackrf_transfer
[ 90%] Built target hackrf_sweep
[ 90%] Built target hackrf_info
Scanning dependencies of target hackrf_clock
[ 90%] Built target hackrf_debug
[ 95%] Building C object hackrf-tools/src/CMakeFiles/hackrf_clock.dir/hackrf_clock.c.o
[ 95%] Built target hackrf_transfer
[100%] Linking C executable hackrf_clock
[100%] Built target hackrf_clock
```

```
lyj@lyj-virtual-machine:~/hackrf-2021.03.1/host/build$ sudo make install
[ 10%] Built target hackrf-static
[ 20%] Built target hackrf
[ 30%] Built target hackrf_spiflash
[ 40%] Built target hackrf_operacake
[ 50%] Built target hackrf_cpldjtag
[ 60%] Built target hackrf_sweep
[ 70%] Built target hackrf_transfer
[ 80%] Built target hackrf_info
[ 90%] Built target hackrf_debug
[100%] Built target hackrf_clock
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/lib/pkgconfig/libhackrf.pc
-- Installing: /etc/udev/rules.d/53-hackrf.rules
-- Installing: /usr/local/lib/libhackrf.so.0.6.0
-- Installing: /usr/local/lib/libhackrf.so.0
-- Installing: /usr/local/lib/libhackrf.so
-- Installing: /usr/local/lib/libhackrf.a
-- Installing: /usr/local/include/libhackrf/hackrf.h
-- Installing: /usr/local/bin/hackrf_transfer
-- Set runtime path of "/usr/local/bin/hackrf_transfer" to ""
-- Installing: /usr/local/bin/hackrf_spiflash
-- Set runtime path of "/usr/local/bin/hackrf_spiflash" to ""
-- Installing: /usr/local/bin/hackrf_cpldjtag
-- Set runtime path of "/usr/local/bin/hackrf_cpldjtag" to ""
-- Installing: /usr/local/bin/hackrf_info
-- Set runtime path of "/usr/local/bin/hackrf_info" to ""
-- Installing: /usr/local/bin/hackrf_debug
-- Set runtime path of "/usr/local/bin/hackrf_debug" to ""
-- Installing: /usr/local/bin/hackrf_clock
-- Set runtime path of "/usr/local/bin/hackrf_clock" to ""
-- Installing: /usr/local/bin/hackrf_sweep
-- Set runtime path of "/usr/local/bin/hackrf_sweep" to ""
-- Installing: /usr/local/bin/hackrf_operacake
-- Set runtime path of "/usr/local/bin/hackrf_operacake" to ""
```

- 测试HackRF设备连接

```
lyj@lyj-virtual-machine: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
lyj@lyj-virtual-machine:~$ hackrf_info
hackrf_info version: 2021.03.1
libhackrf version: 2021.03.1 (0.6)
Found HackRF
Index: 0
Serial number: 0000000000000000681861dc3524a357
Board ID Number: 2 (HackRF One)
Firmware Version: 2018.01.1 (API:1.02)
Part ID Number: 0xa000cb3c 0x0066475e
```

安装GNU Radio(APT方式)

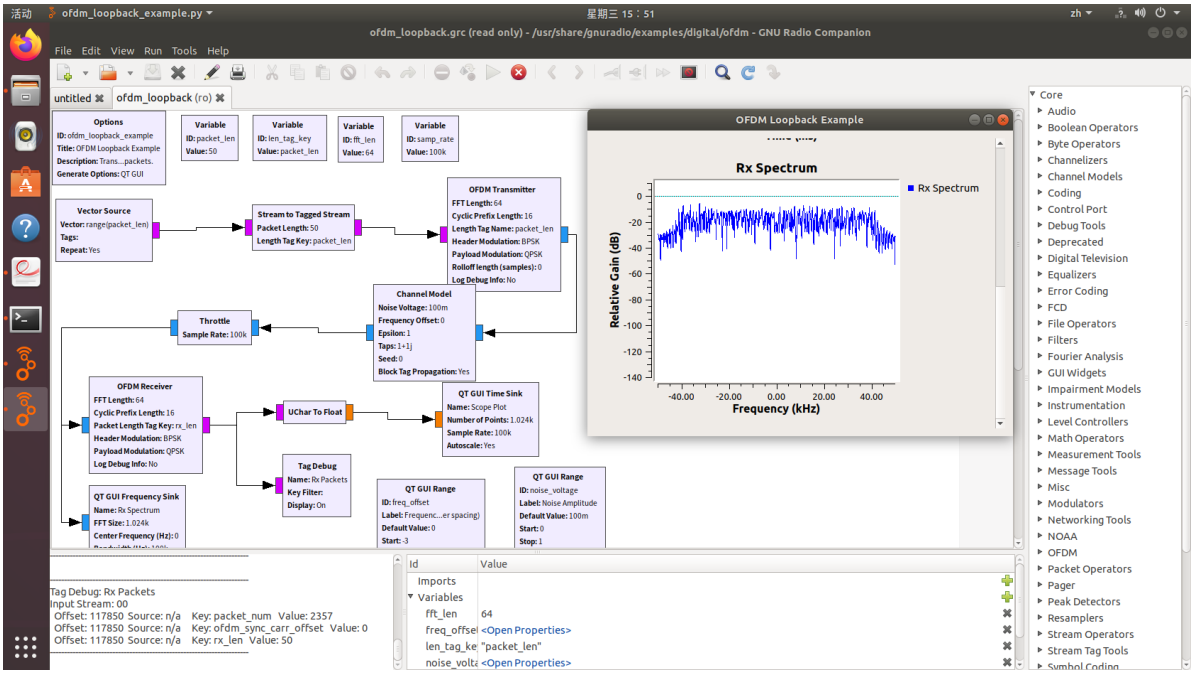
```
sudo apt install cmake git g++ libboost-all-dev python-dev
sudo apt install python-mako python-numpypython-wxgtk3.0
sudo apt install python-sphinx python-cheetah swig
sudo apt install libzmq3-dev libfftw3-dev libgsl-dev
sudo apt install libcppunit-dev doxygen libcomedi-dev
sudo apt install libqt4-opengl-dev python-qt4 libqwt-dev
sudo apt install libsdl1.2-dev libusb-1.0-0-dev python-gtk2
sudo apt install python-lxml pkg-config python-sip-dev
```

- 然后就能安装成功GNU Radio 3.7.11

测试安装成功

- 命令行输入`gnuradio-companion`打开并能运行例程即表示安装成功

> 若报错Gtk-Message: 17:07:20.373: Failed to load module "canberra-gtk-module"则执行如下命令即可解决：
>
> sudo apt install libcanberra-gtk-module

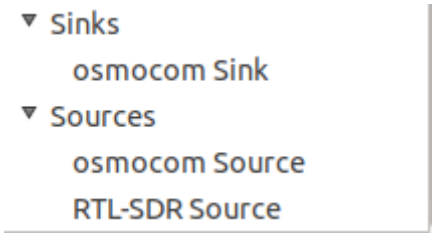


安装gr-osmosdr

```
1 sudo apt install gr-osmosdr
```

```
lyj@lyj-virtual-machine: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
lyj@lyj-virtual-machine:~$ sudo apt install gr-osmosdr
[sudo] lyj 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
  gr-fcdproplus gr-fosphor gr-iqbal libairspy0 libairspyhf0 libbladerf1
  libfreesrp0 libglfw3 libgnuradio-fcdproplus3.7.11 libgnuradio-fosphor3.7.11
  libgnuradio-iqbalance3.7.11 libgnuradio-osmosdr0.1.4 libhackrf0 libhamlib2
  libhidapi-libusb0 liblimesuite17.12-1 libmirisdr0 libosmosdr0 librtaudio6
  libsoapysdr0.6 limesuite-udev soapyosmo-common0.6 soapysdr0.6-module-airspy
  soapysdr0.6-module-all soapysdr0.6-module-audio soapysdr0.6-module-bladerf
  soapysdr0.6-module-hackrf soapysdr0.6-module-lms7 soapysdr0.6-module-osmosdr
  soapysdr0.6-module-redpitaya soapysdr0.6-module-remote
  soapysdr0.6-module-rtlsdr soapysdr0.6-module-uhd
建议安装：
  bladerf bladerf-firmware bladerf-fpga libvulkan1
下列【新】软件包将被安装：
  gr-fcdproplus gr-fosphor gr-iqbal gr-osmosdr libairspy0 libairspyhf0
  libbladerf1 libfreesrp0 libglfw3 libgnuradio-fcdproplus3.7.11
  libgnuradio-fosphor3.7.11 libgnuradio-iqbalance3.7.11
  libgnuradio-osmosdr0.1.4 libhackrf0 libhamlib2 libhidapi-libusb0
  liblimesuite17.12-1 libmirisdr0 libosmosdr0 librtaudio6 libsoapysdr0.6
  limesuite-udev soapyosmo-common0.6 soapysdr0.6-module-airspy
  soapysdr0.6-module-all soapysdr0.6-module-audio soapysdr0.6-module-bladerf
  soapysdr0.6-module-hackrf soapysdr0.6-module-lms7 soapysdr0.6-module-osmosdr
  soapysdr0.6-module-redpitaya soapysdr0.6-module-remote
  soapysdr0.6-module-rtlsdr soapysdr0.6-module-uhd
升级了 0 个软件包，新安装了 34 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
```

- 安装完成后在GNU Radio中的模块列表中可以找到相关的osmocom收发机模块，用以提供GNU Radio和HackRF One等硬件的接口驱动



GNU Radio使用HackRF One实现FM收音机：