Université de Poitiers

Laboratoire XLIM - Equipe RESYST

Stage de Master 2

Mise en oeuvre d'une plateforme radio logiciellle: Application à la réduction du PAPR

Tutoriel d'installation du logiciel GNU Radio et de l'UHD

Boubacar Diallo

Structure :

Laboratoire XLIM - Equipe RESYST
Université de Poitiers - France

Date: le 12 mai 2017

La plateforme GNU Radio

GNU Radio est une plateforme de développement logiciel libre et gratuit qui fournit des blocs de traitement de signal pour implémenter des radios-logiciel. Il est sous licence **GNU General Public License (GPL)**.

Qu'est ce qu'un radio logiciel ?
Un Radio-logiciel est un système radio qui effectue le traitement du signal requis dans le logiciel au lieu d'utiliser des circuits intégrés dédiés dans le matériel. L'avantage est que, puisque le logiciel peut être facilement remplacé dans le système radio, le même matériel peut être utilisé pour créer de nombreux types de radios pour de nombreuses normes de communication différentes (**Wifi 802.11a/n/g, Wimax, LTE ...**); Ainsi, un radio logiciel peut être utilisée pour une variété d'applications!

GNU Radio peut être utilisée avec un matériel RF radiofréquence externe facilement disponible à faible coût pour créer des **SDR** (*Software-Defined Radios*), ou sans matériel dans un environnement de simulation. Il est largement utilisé dans les environnements amateurs, universitaires et commerciaux pour soutenir à la fois la recherche sur les communications sans fil et les systèmes radio réels.

Figure 1: *GNU Radio logo*

GNU Radio possède plusieurs types de blocs qui se trouvent généralement dans les systèmes de traitement du signal. Il comprend en plus une méthode de connexion de ces blocs, puis gère comment les données sont transmises d'un bloc à un autre. L'extension de GNU Radio est également assez simple et si un bloc n'existe pas, on peut le créer rapidement et l'ajouter à la plateforme.
Les applications utilisé dans GNU Radio sont écrites en langage de programmation **C++** ou **Python** et permettent au développeur de mettre en œuvre des systèmes radio en temps réel à haut débit dans un environnement de développement rapide.

Universal Software Radio Peripheral (USRP)

L'USRP est une carte électronique qui communique avec l'ordinateur par une connexion USB ou Ethernet haute-vitesse pour interagir avec un logiciel de radio comme GNU Radio. Elle a été développée par une équipe dirigée par Matt Ettus.

Figure 2: *Famille d'USRP de la compagnie Matt Ettus*

La famille des produits USRP est une plate-forme matérielle peu coûteuse pour la radio logicielle et est couramment utilisée par les laboratoires de recherche, les universités et les amateurs. Elle a été conçue pour être accessible et un bon nombre des produits sont open source. Tous les produits USRP sont contrôlés par un pilote appelé UHD (*USRP Hardware Driver*) qui est un logiciel libre et gratuit.

Les USRP sont couramment utilisés avec la plateforme GNU Radio pour créer des systèmes **SDR** (*Software-Defined Radios*) complexes. Pour utilisé GNU Radio avec un **USRP**, il faut au préalable installer le pilote **UHD**.

Installation de UHD et GNU Radio

Ce tutoriel d'installation a été faite pour les distributions Ubuntu et Debian, nous vous suggérons d'utiliser l'une des versions suivantes: **Ubuntu 14.04, Ubuntu 16.04, Ubuntu 17.04 ou Debian version >= 6**

Pour vérifier sa version de distribution, vous pouvez exécuter cette commande:

```
$ lsb_release -a
```

Mise à jour et installation des dépendances

Avant d'installer l'UHD et le logiciel GNU Radio, nous devons nous assurer que toutes les dépendances sont installées pour la première fois. Toutefois, avant d'installer des dépendances, il est recommandé de mettre à jour tous les paquets déjà installés sur le système. Vous pouvez le faire par la commande:

```
$ sudo apt-get update
```

Une fois le système mise à jour, installez les dépendances:

Pour la version Ubuntu 17.04, exécutez la commande suivante:

```
$ sudo apt-get -y --ignore-missing  install git swig cmake doxygen build-essential
libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-dev libfftw3-bin
libfftw3-dev libfftw3-doc libcppunit-1.13-0v5 libcppunit-dev libcppunit-doc
ncurses-bin cpufrequtils python-numpy python-numpy-doc python-numpy-dbg
python-scipy python-docutils qt4-bin-dbg qt4-default qt4-doc libqt4-dev
libqt4-dev-bin python-qt4 python-qt4-dbg python-qt4-dev python-qt4-doc doxygen
python-qt4-doc libqwt6abi1 libfftw3-bin libfftw3-dev libfftw3-doc ncurses-bin
libncurses5 libncurses5-dev libncurses5-dbg libfontconfig1-dev libxrender-dev
autoconf libtool python-dev libfftw3-dev libcppunit-dev libboost-all-dev
libusb-1.0-0-dev fort77 libsdl1.2-dev python-wxgtk3.0 git-core libqt4-dev
ccache python-opengl libgsl-dev python-cheetah python-mako python-lxml
qt4-dev-tools libusb-1.0-0-dev libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools
python-qwt5-qt4 cmake git-core wget libxi-dev gtk2-engines-pixbuf r-base-dev
liborc-0.4-0 liborc-0.4-dev libasound2-dev python-gtk2 libzmq3-dev libzmq5
python-sphinx libcomedi-dev python-zmq libpulse-dev swig g++ automake
qt4-default libusb-dev python-tk python-requests python-numpy libboost-all-dev
```

Pour la version Ubuntu 16.04, exécutez la commande suivante:

```
$ sudo apt-get -y --ignore-missing install git swig cmake doxygen build-essential
libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-dev libfftw3-bin
libfftw3-dev libfftw3-doc libcppunit-1.13-0v5 libcppunit-dev libcppunit-doc
ncurses-bin cpufrequtils python-numpy python-numpy-doc python-numpy-dbg python-scipy
python-docutils qt4-bin-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-bin python-qt4
python-qt4-dbg python-qt4-dev python-qt4-doc python-qt4-doc libqwt6abi1 libfftw3-bin
libfftw3-dev libfftw3-doc ncurses-bin libncurses5 libncurses5-dev libncurses5-dbg
libfontconfig1-dev libxrender-dev libpulse-dev swig g++ automake autoconf libtool
python-dev libfftw3-dev libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev
fort77 libsdl1.2-dev python-wxgtk3.0 git-core libqt4-dev python-numpy ccache
python-opengl libgsl-dev python-cheetah python-mako python-lxml doxygen qt4-default
qt4-dev-tools libusb-1.0-0-dev libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools
python-qwt5-qt4 cmake git-core wget libxi-dev gtk2-engines-pixbuf r-base-dev
liborc-0.4-0 liborc-0.4-dev libasound2-dev python-gtk2 libzmq-dev libzmq1 python-tk
python-sphinx libcomedi-dev python-zmq python-requests libboost-all-dev
```

Pour la version Ubuntu 14.04 et 14.10, exécutez la commande suivante:

```
$ sudo apt-get -y --ignore-missing install git swig cmake doxygen build-essential
libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-dev libfftw3-bin
libfftw3-dev libfftw3-doc libcppunit-1.13-0 libcppunit-dev libcppunit-doc ncurses-bin
cpufrequtils python-numpy python-numpy-doc python-numpy-dbg python-scipy
python-docutils libxrender-dev git-core libcppunit-dev libboost-all-dev
```

```
qt4-bin-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg
python-qt4-dev python-qt4-doc python-qt4-doc libfftw3-bin libfftw3-dev libfftw3-doc
ncurses-bin libncurses5 libncurses5-dev libncurses5-dbg   libfontconfig1-dev
libpulse-dev swig g++ automake autoconf libtool python-dev libfftw3-dev
libboost-all-dev libusb-dev libusb-1.0-0-dev fort77 libsdl1.2-dev python-wxgtk2.8
libqt4-dev python-numpy ccache python-opengl libgsl0-dev python-cheetah python-mako
python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev libqwt5-qt4-dev
libqwtplot3d-qt4-dev pyqt4-dev-tools python-qwt5-qt4 cmake git-core wget libxi-dev
gtk2-engines-pixbuf r-base-dev python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev
python-gtk2 libzmq1 libzmq-dev python-requests python-sphinx libcomedi-dev
```

Pour la distribution Debian version >= 6, exécutez la commande suivante:

```
$ sudo apt-get -y --ignore-missing install wget libfontconfig1-dev libxrender-dev
libpulse-dev swig g++ automake autoconf libtool python-dev libfftw3-dev doxygen
libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev fort77 libsdl1.2-dev
python-wxgtk2.8 git-core libqt4-dev python-numpy ccache python-opengl libgsl0-dev
python-cheetah python-lxml qt4-dev-tools libusb-1.0-0-dev libqwt5-qt4-dev
libqwtplot3d-qt4-dev pyqt4-dev-tools python-qwt5-qt4 cmake git-core libxi-dev
python-docutils gtk2-engines-pixbuf r-base-dev python-tk liborc-0.4-0 liborc-0.4-dev
libasound2-dev python-gtk2 libportaudio2 portaudio19-dev ca-certificates
```

Après le téléchargement de tous les dépendances du système, vous pouvez récupérer les codes sources.

Récupération des sources du UHD et GNU Radio

Tout d'abord, choisir le dossier où vous voulez stocker les sources puis récupérez les dépôts.

Code source de GNU Radio:

Téléchargez le code source par ici ou clonez le référentiel GitHub par la commande:

```
$ git clone --progress --recursive https://github.com/gnuradio/gnuradio.git
```

Code sources du pilote UHD (USRP Hardware Driver):

Téléchargez le code source par ici ou clonez le référentiel GitHub par la commande:

```
$ git clone --progress https://github.com/EttusResearch/uhd
```

NB: Il est fortement recommandé d'installer d'abord le pilote UHD avant GNU Radio pour que ce dernier puisse prendre en compte la version du UHD.

Afin d'accélérer l'installation en tenant compte du nombre de processeur de l'ordinateur lors de l'installation, nous allons récupérer cette valeur et l'enregistrer dans une variable appélée $JFLAG:

```
$ cnt=`grep 'processor.*:' /proc/cpuinfo|wc -l`
$ cnt=`expr $cnt - 1`
$ JFLAG=-j$cnt
```

Installation du pilote UHD:

Pour installer le pilote UHD, il faudra tout d'abord aller dans le repertoire ***uhd/host*** et créer un dossier d'exécution nommé ***build***:

```
    $ cd uhd/host
    $ mkdir build
    $ cd build
    $ cmake ../
    $ make $JFLAG
    $ sudo make $JFLAG install
    $ sudo ldconfig
```

- **cmake:** permet de créer les Makefiles.
- **make:** pour l'exécution du paquet.
- **make test:** permet d'exécuter éventuellement certains tests de base pour vérifier que le processus de construction a été correctement exécuté.
- **sudo make install:** permet d'installer le paquet dans le dossier /usr/local/lib. Vous devez exécuter cette commande en tant que **root** en raison des autorisations sur ce dossier.
- **sudo ldconfig:** met à jour le cache de la bibliothèque partagée du système.

On termine par le téléchargement du microprogramme vers ***/usr/local/share/uhd/images***.

```
    $ sudo /usr/local/lib/uhd/utils/uhd_images_downloader.py
```

À ce stade, le pilote UHD est installé et prêt à l'emploi. Vous pouvez tester rapidement cela sans connecter un périphérique USRP, en exécutant la commande *uhd_find_devices*, vous obtenez le résultat suivant:

```
    $ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 4.8.2; Boost_105400; UHD_3.11.0.git-162-g2790b51f
No UHD Devices Found
```

Cela nous indique qu'aucun USRP n'est connecté à l'ordinateur, ce qui est tout à fait normale.

Installation de GNU Radio:

Il faudra tout d'abord aller dans le répertoire ***gnuradio/*** et créer un dossier d'exécution ***build***, puis faire la même démarche d'installation que précédemment:

```
$ cd gnuradio/
$ mkdir build
$ cd build
$ cmake ../
$ make $JFLAG
$ sudo make $JFLAG install
$ sudo ldconfig
```

À ce stade, GNU Radio est installé et prêt à l'emploi. Vous pouvez tester rapidement cela, sans périphérique USRP, en exécutant les tests rapides suivants :

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

La première commande vous donne la version de GNU Radio installée, la suivante le répertoire d'installation et la dernière vous donne les différentes composants qui sont installés.

Vous pouvez à présent lancer l'outil GNU Radio Companion (GRC) qui est un outil visuel pour construire et exécuter des flux de données GNU Radio.

```
$ gnuradio-companion
```

Nous obtenons la fenêtre suivante:

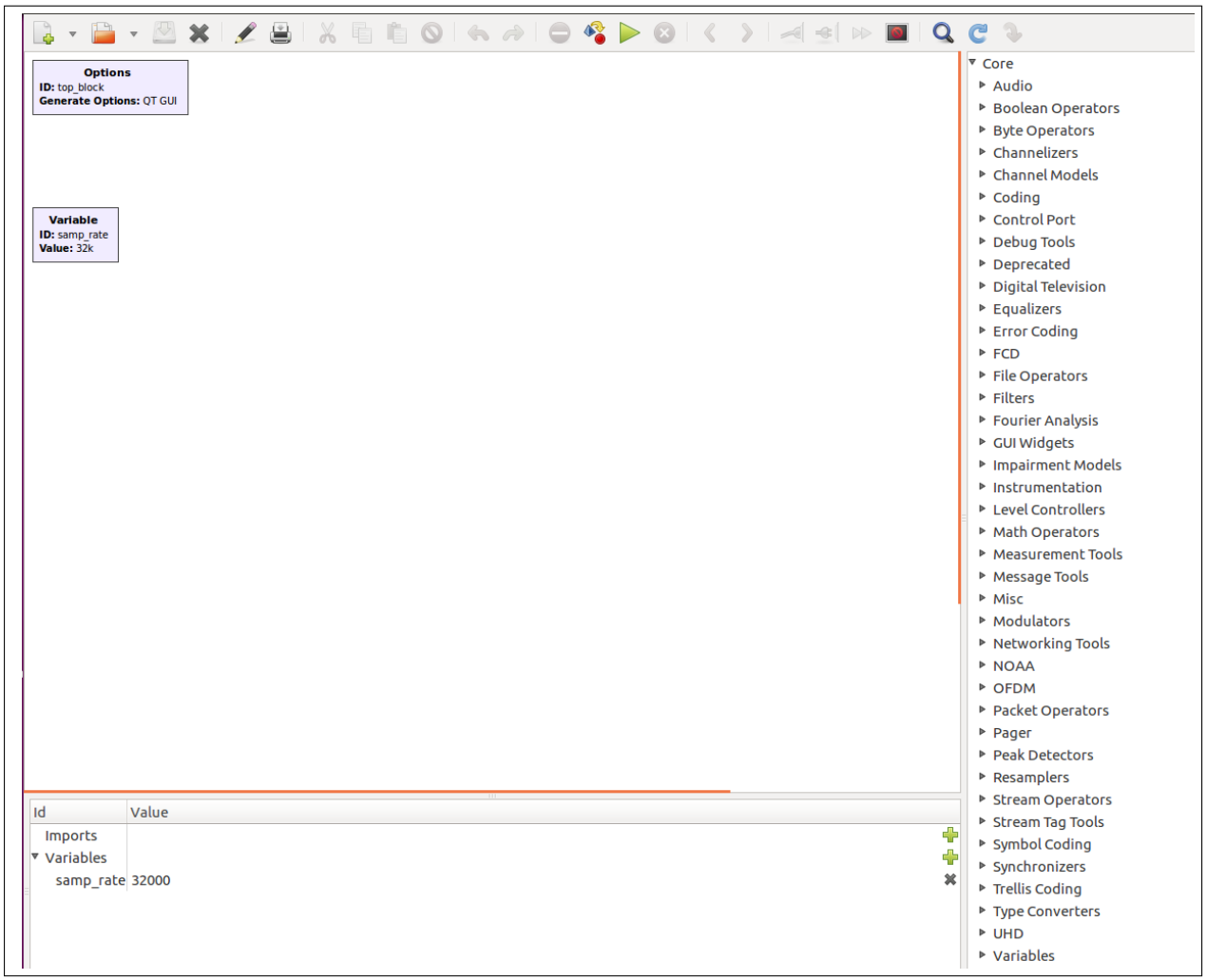


Figure 3: *Plateforme GNU Radio Companion*

Il existe un flux graphique simple que vous pouvez exécuter ne nécessitant aucun matériel USRP. Ce flux est appelé "*dial_tone.grc*", il produit une tonalité de numérotation sur l'ordinateur. Son exécution vérifie que toutes les bibliothèques peuvent être trouvées et que le temps de fonctionnement de GNU Radio est acceptable. Veillez parcourir le chemin suivant: ***gnuradio/gr-audio/examples/grc/dial_tone.grc***
L'exécution sous GRC donne une tonalité avec une interface où vous pouvez jouer sur le volume et l'amplitude du bruit:

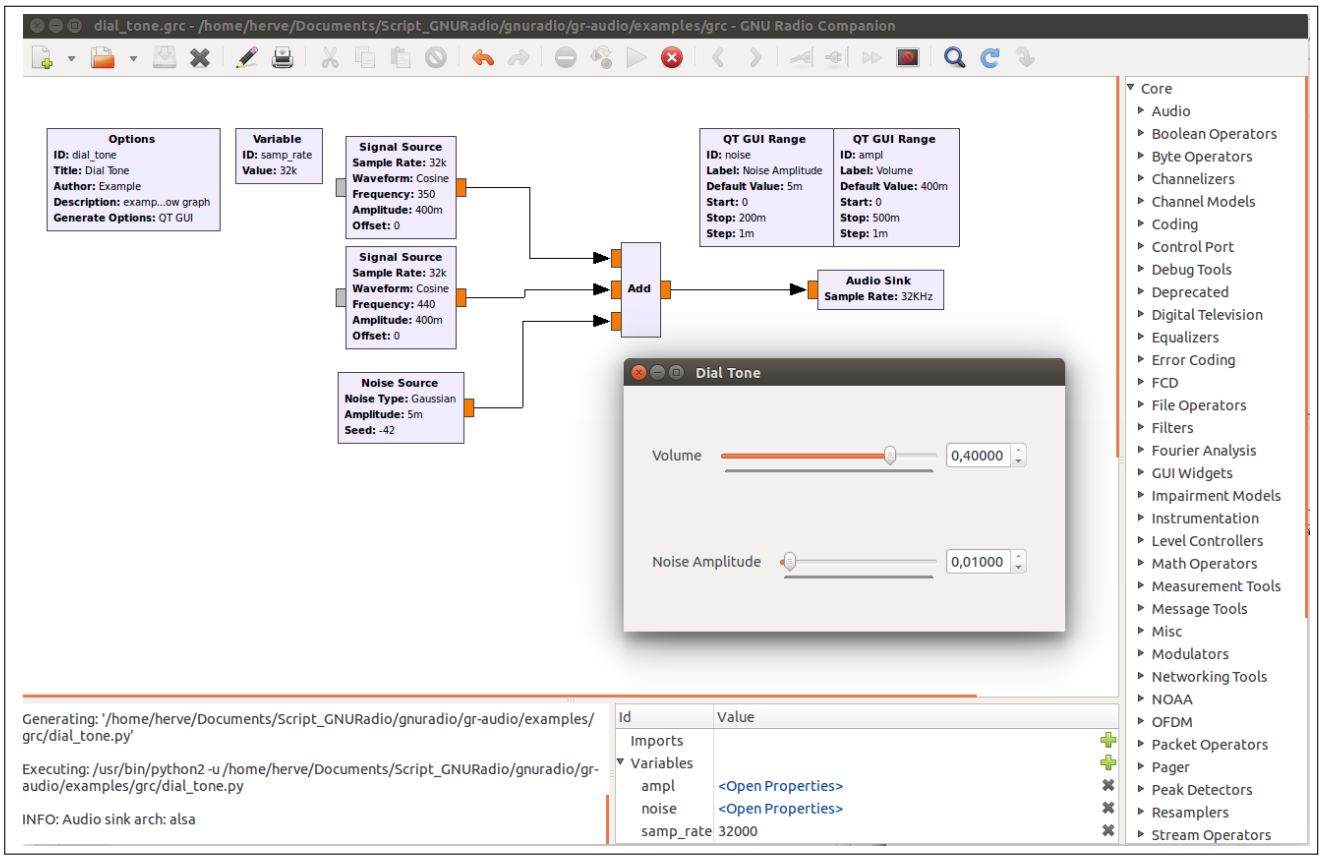


Figure 4: *Exemple dial_tone.grc*

Sur GNU Radio Companion (GRC), l'exécution d'un flux de données génère un script python. Ce script peut aussi être utilisé pour ouvrir le flux. la démarche est la suivante:

```
$ python /chemin_du_dossier/gnuradio/gr-audio/examples/grc/dial_tone.py
```

Configuration de l'USRP avec l'ordinateur

L'installation de UHD et de GNU Radio devrait maintenant être complétée. À ce stade, connectez l'USRP à l'ordinateur hôte. Il existe 2 types de connexion entre l'USRP et l'ordinateur, une connexion par port USB ou par port Ethernet.

Si les périphériques de l'USRP utilisent un port Ethernet pour se connecter à l'ordinateur hôte, tels que les familles d'USRP **N200, N210, X300, X310**; dans ce cas définissez l'adresse IP statique **192.168.10.1** pour votre système, avec un masque de réseau de **255.255.255.0** et une passerelle de **192.168.10.1**. L'adresse IP par défaut de l'USRP est **192.168.10.2**, avec un masque de réseau de **255.255.255.0**.
Sous Ubuntu et Debian, l'adresse IP peut être définit à l'aide du Gestionnaire de réseau graphique ou à partir de la ligne de commande avec ***ifconfig***. Après avoir définit l'adresse IP, essayez de faire un ping sur l'USRP avec "**ping 192.168.10.2**". L'USRP devrait répondre aux demandes de ping.

```
    $ ifconfig
eth0      Link encap:Ethernet  HWaddr 38:63:bb:8a:4d:e6
          inet adr:192.168.10.1  Bcast:192.168.10.255  Masque:255.255.255.0
          adr inet6: fe80::3a63:bbff:fe8a:4de6/64 Scope:Lien
          ......................
```

```
    $ ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=32 time=1.81 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=32 time=1.09 ms
......................
```

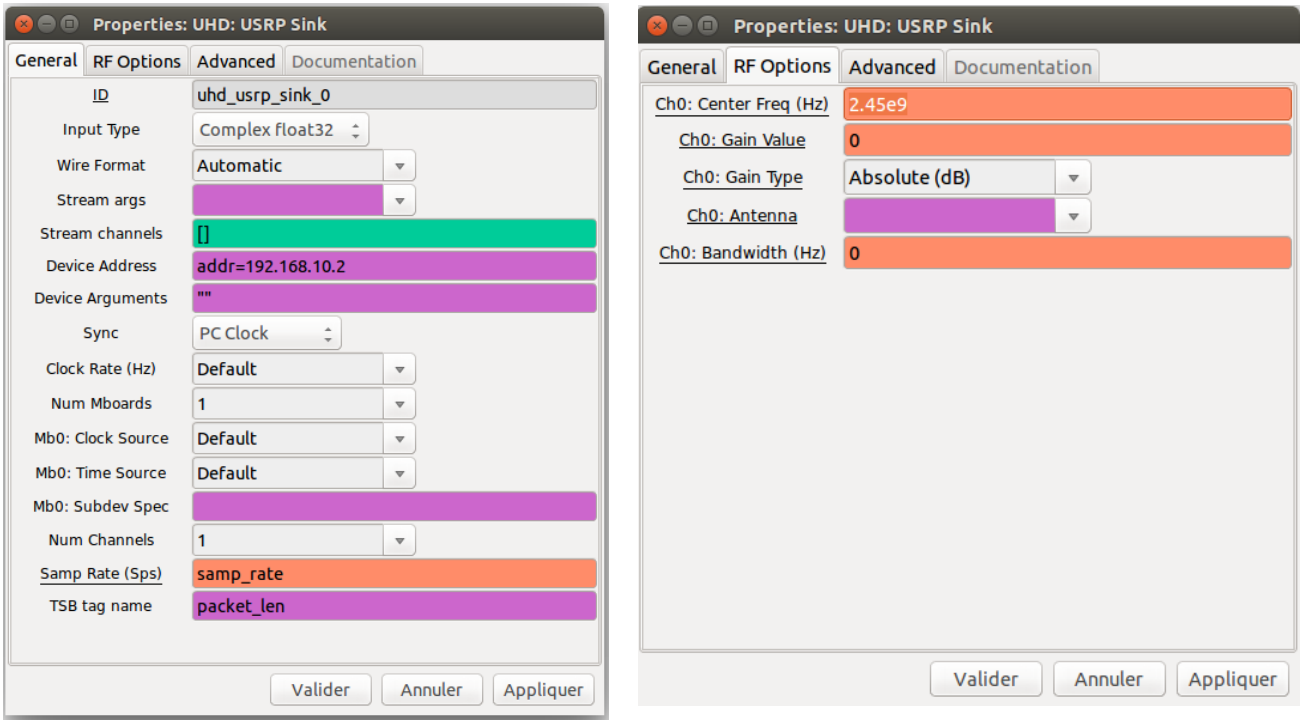
Essayez également d'exécuter les commandes "*uhd_find_devices*" et "*uhd_usrp_probe*".
uhd_find_devices vérifie si un équipement USRP est installé ou pas et donne quelques informations sur celui-ci.
uhd_usrp_probe donne toutes les caractéristiques de l'USRP dont on peut citer: la version de l'image installée, l'adresse IP et Mac, le nombre et type d'antenne, les bandes de fréquences, le gain etc.

```
    $ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 4.8.2; Boost_105400; UHD_3.11.0.git-162-g2790b51f
--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    serial: E3R13WFUP
    addr: 192.168.10.2
    name:
    type: usrp2
```

```
    $ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 4.8.2; Boost_105400; UHD_3.11.0.git-162-g2790b51f
[INFO] [USRP2] Opening a USRP2/N-Series device...
[INFO] [USRP2] Current recv frame size: 1472 bytes
[INFO] [USRP2] Current send frame size: 1472 bytes
[INFO] [USRP2] Detecting internal GPSDO....
[INFO] [GPS] Found an internal GPSDO: Jackson-Labs, FireFly , Firmware Rev 0.923
[INFO] [USRP2] Setting references to the internal GPSDO
  ----------------------------------------------------
 /
|       Device: USRP2 / N-Series Device
|     ------------------------------------------------
|    /
|   |   Mboard: N210
|   |   hardware: 2561
|   |   mac-addr: a0:36:fa:25:33:3f
```

```
|   |   ip-addr: 192.168.10.2
|   |   subnet: 255.255.255.0
|   |   gateway: 192.168.10.1
|   |   gpsdo: internal
|   |   serial: E3R13WFUP
|   |   FW Version: 12.4
|   |   FPGA Version: 11.1
|   |
|   |   Time sources:  none, external, _external_, mimo, gpsdo
|   |   Clock sources: internal, external, mimo, gpsdo
|   |   Sensors: gps_gpgga, gps_gprmc, gps_time, gps_locked, gps_servo, mimo_loc
|   |     ____________________________________________________
|   |    /
|   |   |       RX DSP: 0
|   |   |
|   |   |   Freq range: -50.000 to 50.000 MHz
|   |     ____________________________________________________
|   |    /
|   |   |       RX DSP: 1
|   |   |
|   |   |   Freq range: -50.000 to 50.000 MHz
|   |     ____________________________________________________
|   |    /
|   |   |       RX Dboard: A
|   |   |   ID: XCVR2450, XCVR2450 - r2.1 (0x0061)
|   |   |     ________________________________________________
|   |   |    /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: XCVR2450 RX
|   |   |   |   Antennas: J1, J2
|   |   |   |   Sensors: lo_locked, rssi
|   |   |   |   Freq range: 2400.000 to 6000.000 MHz
|   |   |   |   Gain range LNA: 0.0 to 30.5 step 15.0 dB
|   |   |   |   Gain range VGA: 0.0 to 62.0 step 2.0 dB
|   |   |   |   Bandwidth range: 13500000.0 to 39600000.0 step 600000.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |   |
```

Après avoir configurer l'USRP, vous pouvez le tester par un exemple dans GNU Radio. Dans cet exemple (***gnuradio/gr-uhd/examples/grc/uhd_tx_dpsk.grc***), vous devez d'abord configurer l'USRP en renseignant par exemple l'adresse IP, la fréquence centrale mais aussi le gain.



Dans cette exemple, l'adresse IP de l'USRP est le "**192.168.10.2**" avec une fréquence centrale égale à **2.45 GHz** et un Gain égale à **0 dB**. En lançant la simulation, nous obtenons:

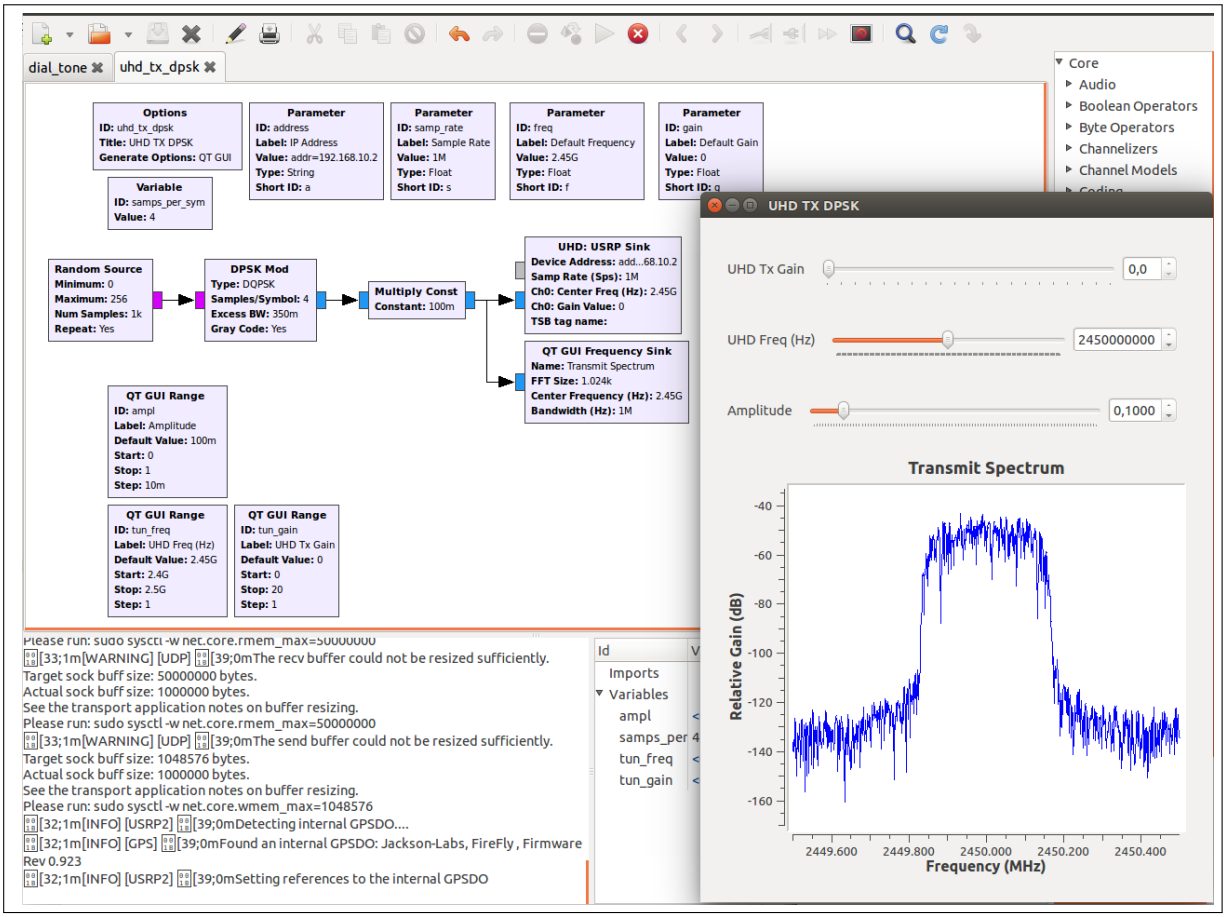


Figure 5: *Exemple uhd_tx_dpsk.grc*

Installation à partir d'un script bash

Le script bash ***build-gnuradio*** est un script d'installation pour les systèmes récents d'**Ubuntu, Debian** et de **Fedora** fournis par **Marcus Leech**. Il comporte plusieurs options que vous pouvez utiliser pour installer différentes versions de GNU Radio.

Pour installation, ouvrez un terminal, aller dans votre répertoire de travail que vous souhaitez que les fichiers sources soient stockés et exécutez cette commande:

```
$ wget http://www.sbrac.org/files/build-gnuradio
$ chmod a+x build-gnuradio
$ ./build-gnuradio -ja
```

L'option ***-ja*** permet de prendre en compte le nombre de processeur de votre système ce qui accélère l'installation qui doit durer environ moins d'une demi-heure avec un Core i7.

La commande télécharge le script d'installation (build-gnuradio) et le rend exécutable. D'abord il télécharge et installe toutes les dépendances selon la version de votre système, ensuite télécharge les deux sources UHD et GNU Radio par clonage Git signifiant qu'il va installer automatiquement la dernière version existante de la branche 'master' et enfin exécute le processus de création et l'installe sur votre système.

NB: Beaucoup de chose sont fait en silence, alors s'il y a beaucoup à faire pour le script, ne soyez pas surpris si ça ne dit rien pendant un certain temps.

L'exécution du script fera tout ce qu'il faut pour obtenir un système GNU Radio et UHD exécuté à partir de la source. De plus, vous disposez de tout le code source sur votre disque dur (répertoire où a été exécuté le script) et donc disponible pour les modifications futures. Ce script combine la flexibilité de l'installation à partir de la source avec la facilité d'utilisation et est recommandé pour la plupart des utilisateurs d'**Ubuntu**, de **Debian** et de **Fedora**.

Références

GNU Radio

`https://wiki.gnuradio.org/index.php/BuildGuide`

`https://gnuradio.org/doc/doxygen/build_guide.html`

`https://wiki.gnuradio.org/index.php/InstallingGRFromSource`

USRP Universal Software Radio Peripheral

`https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux`

`https://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral`