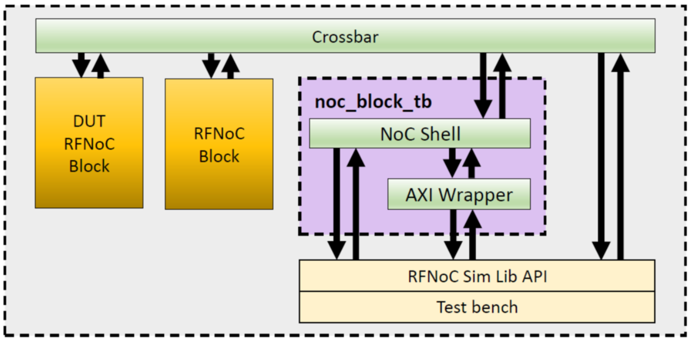


Installation of UHD (4.0), GNU Radio (3.8) & RFNOC on Ubuntu 20.04

Murthy S
Applications Engineer | FPGA & Software Defined Radio
2 articles + Suivre

7 juillet 2021

Ouvrir le lecteur immersif

The “**RF Network on Chip (RFNOC)**” FPGA development framework is supported on the 3rd Generation USRP devices and the typical installation process for the “**RFNOC 4**” includes the following software tools to be installed as prerequisites

1. Ubuntu 20.04
2. Xilinx Vivado 2019.1 (Design Edition)
3. USRP Hardware Driver (UHD - 4.0)
4. GNU Radio (GRC - 3.8)
5. gr-ettus

Before installing the necessary software tool, update the Ubuntu packages installed on the system.

Command:

```
sudo apt-get update && sudo apt-get upgrade
```

Installing the Package Dependencies

***Step 1**: Installing the necessary package dependencies required for building UHD, GNU Radio & gr-ettus*

Command:

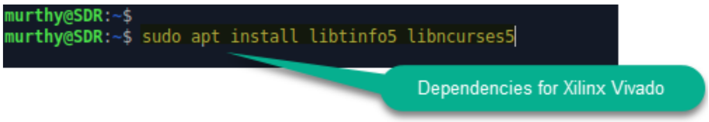
sudo apt install git cmake g++ libboost-all-dev libgmp-dev swig python3-numpy python3-mako python3-sphinx python3-lxml doxygen libfftw3-dev libsdl1.2-dev libgsl-dev libqwt-qt5-dev libqt5opengl5-dev python3-pyqt5 liblog4cpp5-dev libzmq3-dev python3-yaml python3-click python3-click-plugins python3-zmq python3-scipy python3-gi python3-gi-cairo gobject-introspection gir1.2-gtk-3.0 build-essential libusb-1.0-0-dev python3-docutils python3-setuptools python3-ruamel.yaml python-is-python3

***Step 2**: Installing the Package Dependencies for Xilinx Vivado*

Command:

sudo apt install libtinfo5 libncurses5

Example Image:



```
murthy@SDR:~$
murthy@SDR:~$ sudo apt install libtinfo5 libncurses5
```

Dependencies for Xilinx Vivado

Installing Xilinx Vivado

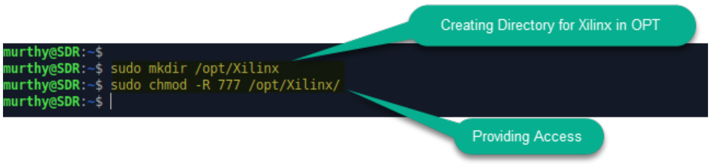
***Step 3**: RFNOC development framework looks for "**Xilinx Vivado**" in its default installation location. So, create a directory in "**/opt**" for installation.*

Command:

sudo mkdir /opt/Xilinx

sudo chmod -R 777 /opt/Xilinx

Example Image:



```
murthy@SDR:~$
murthy@SDR:~$ sudo mkdir /opt/Xilinx
murthy@SDR:~$ sudo chmod -R 777 /opt/Xilinx/
murthy@SDR:~$ |
```

Creating Directory for Xilinx in OPT

Providing Access

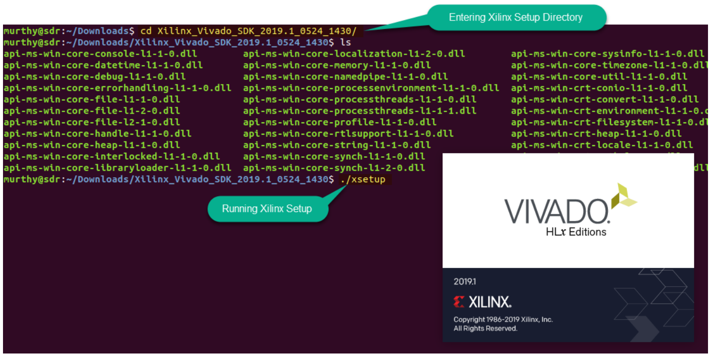
***Step 4**: Running Xilinx Setup*

Command:

cd ~/Downloads/Xilinx_Vivado_SDK_2019.1_0524_1430/

./xsetup

Example Image:

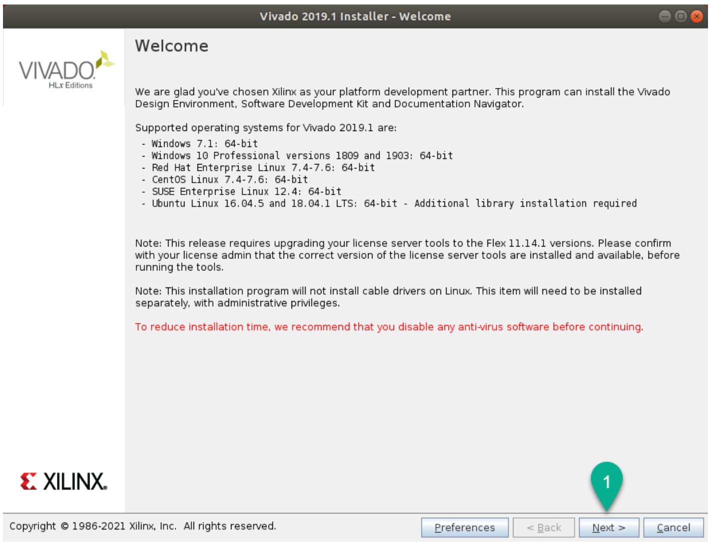


```
murthy@sdr:~/Downloads$ cd Xilinx_Vivado_SDK_2019.1_0524_1430/
murthy@sdr:~/Downloads/Xilinx_Vivado_SDK_2019.1_0524_1430$ ls
api-ms-win-core-console-l1-1-0.dll         api-ms-win-core-localization-l1-2-0.dll       api-ms-win-core-sysinfo-l1-1-0.dll
api-ms-win-core-datetime-l1-1-0.dll        api-ms-win-core-memory-l1-1-0.dll             api-ms-win-core-timezone-l1-1-0.dll
api-ms-win-core-debug-l1-1-0.dll           api-ms-win-core-namedpipe-l1-1-0.dll          api-ms-win-core-util-l1-1-0.dll
api-ms-win-core-errorhandling-l1-1-0.dll   api-ms-win-core-processenvironment-l1-1-0.dll api-ms-win-crt-conio-l1-1-0.dll
api-ms-win-core-file-l1-1-0.dll            api-ms-win-core-processthreads-l1-1-0.dll     api-ms-win-crt-convert-l1-1-0.dll
api-ms-win-core-file-l1-2-0.dll            api-ms-win-core-processthreads-l1-1-1.dll     api-ms-win-crt-environment-l1-1-0.d
api-ms-win-core-file-l2-1-0.dll            api-ms-win-core-profile-l1-1-0.dll            api-ms-win-crt-filesystem-l1-1-0.dl
api-ms-win-core-handle-l1-1-0.dll          api-ms-win-core-rtlsupport-l1-1-0.dll         api-ms-win-crt-heap-l1-1-0.dll
api-ms-win-core-heap-l1-1-0.dll            api-ms-win-core-string-l1-1-0.dll             api-ms-win-crt-locale-l1-1-0.dll
api-ms-win-core-interlocked-l1-1-0.dll     api-ms-win-core-synch-l1-1-0.dll
api-ms-win-core-libraryloader-l1-1-0.dll   api-ms-win-core-synch-l1-2-0.dll                                                dll
murthy@sdr:~/Downloads/Xilinx_Vivado_SDK_2019.1_0524_1430$ ./xsetup
```

__Step 5__: Executing the Xilinx Installation.

- From this step on-wards, the GUI interface is supported to run the Xilinx Vivado installation process.
- Point the installation location to "**/opt/Xilinx**", and proceed the installation.

Example Image:



Welcome

We are glad you've chosen Xilinx as your platform development partner. This program can install the Vivado Design Environment, Software Development Kit and Documentation Navigator.

Supported operating systems for Vivado 2019.1 are:

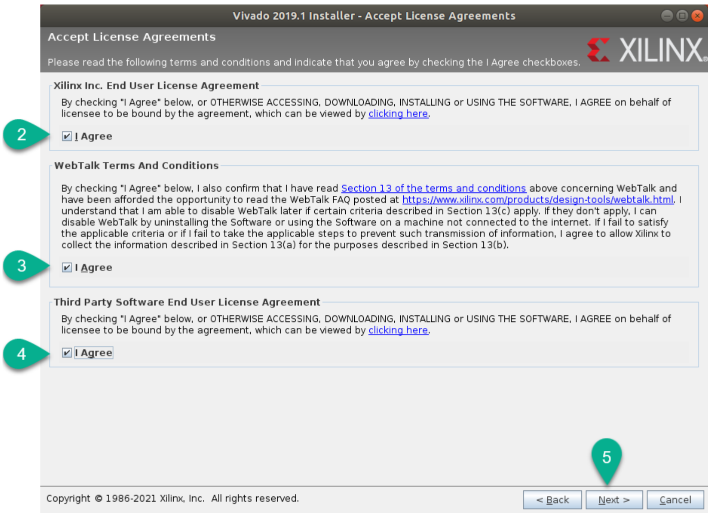
```
 - Windows 7.1: 64-bit
 - Windows 10 Professional versions 1809 and 1903: 64-bit
 - Red Hat Enterprise Linux 7.4-7.6: 64-bit
 - CentOS Linux 7.4-7.6: 64-bit
 - SUSE Enterprise Linux 12.4: 64-bit
 - Ubuntu Linux 16.04.5 and 18.04.1 LTS: 64-bit - Additional library installation required
```

Note: This release requires upgrading your license server tools to the Flex 11.14.1 versions. Please confirm with your license admin that the correct version of the license server tools are installed and available, before running the tools.

Note: This installation program will not install cable drivers on Linux. This item will need to be installed separately, with administrative privileges.

To reduce installation time, we recommend that you disable any anti-virus software before continuing.

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.



Accept License Agreements

Please read the following terms and conditions and indicate that you agree by checking the I Agree checkboxes.

Xilinx Inc. End User License Agreement

By checking "I Agree" below, or OTHERWISE ACCESSING, DOWNLOADING, INSTALLING or USING THE SOFTWARE, I AGREE on behalf of licensee to be bound by the agreement, which can be viewed by clicking here.

☑ I Agree

WebTalk Terms And Conditions

By checking "I Agree" below, I also confirm that I have read Section 13 of the terms and conditions above concerning WebTalk and have been afforded the opportunity to read the WebTalk FAQ posted at https://www.xilinx.com/products/design-tools/webtalk.html. I understand that I am able to disable WebTalk later if certain criteria described in Section 13(c) apply. If they don't apply, I can disable WebTalk by uninstalling the Software or using the Software on a machine not connected to the internet. If I fail to satisfy the applicable criteria or if I fail to take the applicable steps to prevent such transmission of information, I agree to allow Xilinx to collect the information described in Section 13(a) for the purposes described in Section 13(b).

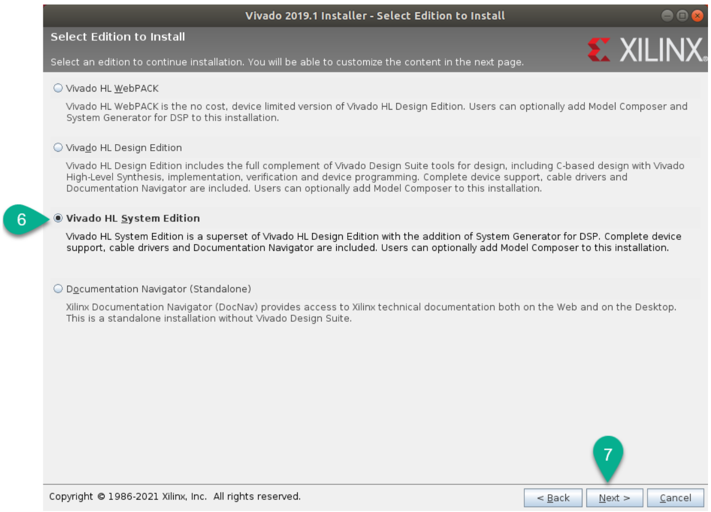
☑ I Agree

Third Party Software End User License Agreement

By checking "I Agree" below, or OTHERWISE ACCESSING, DOWNLOADING, INSTALLING or USING THE SOFTWARE, I AGREE on behalf of licensee to be bound by the agreement, which can be viewed by clicking here.

☑ I Agree

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.



Vivado 2019.1 Installer - Select Edition to Install

Select Edition to Install

Select an edition to continue installation. You will be able to customize the content in the next page.

○ Vivado HL WebPACK

Vivado HL WebPACK is the no cost, device limited version of Vivado HL Design Edition. Users can optionally add Model Composer and System Generator for DSP to this installation.

○ Vivado HL Design Edition

Vivado HL Design Edition includes the full complement of Vivado Design Suite tools for design, including C-based design with Vivado High-Level Synthesis, implementation, verification and device programming. Complete device support, cable drivers and Documentation Navigator are included. Users can optionally add Model Composer to this installation.

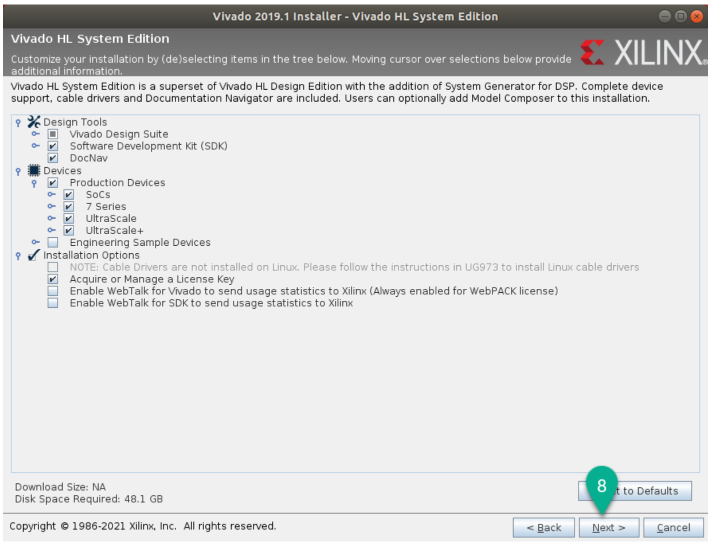
● **Vivado HL System Edition**

Vivado HL System Edition is a superset of Vivado HL Design Edition with the addition of System Generator for DSP. Complete device support, cable drivers and Documentation Navigator are included. Users can optionally add Model Composer to this installation.

○ Documentation Navigator (Standalone)

Xilinx Documentation Navigator (DocNav) provides access to Xilinx technical documentation both on the Web and on the Desktop. This is a standalone installation without Vivado Design Suite.

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.



Vivado 2019.1 Installer - Vivado HL System Edition

Vivado HL System Edition

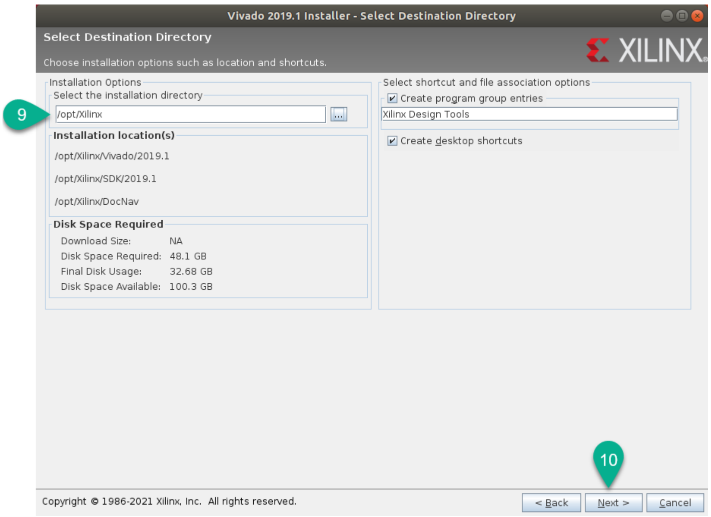
Customize your installation by (de)selecting items in the tree below. Moving cursor over selections below provide additional information.

Vivado HL System Edition is a superset of Vivado HL Design Edition with the addition of System Generator for DSP. Complete device support, cable drivers and Documentation Navigator are included. Users can optionally add Model Composer to this installation.

- Design Tools
  - ▣ Vivado Design Suite
  - ☑ Software Development Kit (SDK)
  - ☑ DocNav
- Devices
  - ☑ Production Devices
    - ☑ SoCs
    - ☑ 7 Series
    - ☑ UltraScale
    - ☑ UltraScale+
  - □ Engineering Sample Devices
- Installation Options
  - □ NOTE: Cable Drivers are not installed on Linux. Please follow the instructions in UG973 to install Linux cable drivers
  - ☑ Acquire or Manage a License Key
  - □ Enable WebTalk for Vivado to send usage statistics to Xilinx (Always enabled for WebPACK license)
  - □ Enable WebTalk for SDK to send usage statistics to Xilinx

Download Size: NA
Disk Space Required: 48.1 GB

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.



Vivado 2019.1 Installer - Select Destination Directory

Select Destination Directory

Choose installation options such as location and shortcuts.

Installation Options

Select the installation directory

/opt/Xilinx

Installation location(s)

/opt/Xilinx/Vivado/2019.1

/opt/Xilinx/SDK/2019.1

/opt/Xilinx/DocNav

Disk Space Required

Download Size:	NA
Disk Space Required:	48.1 GB
Final Disk Usage:	32.68 GB
Disk Space Available:	100.3 GB

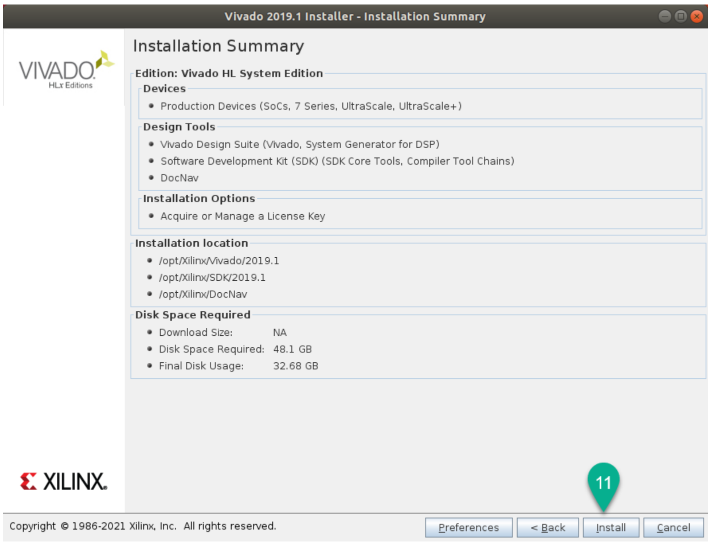
Select shortcut and file association options

☑ Create program group entries

Xilinx Design Tools

☑ Create desktop shortcuts

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.



Vivado 2019.1 Installer - Installation Summary

Installation Summary

Edition: Vivado HL System Edition

Devices
- Production Devices (SoCs, 7 Series, UltraScale, UltraScale+)

Design Tools
- Vivado Design Suite (Vivado, System Generator for DSP)
- Software Development Kit (SDK) (SDK Core Tools, Compiler Tool Chains)
- DocNav

Installation Options
- Acquire or Manage a License Key

Installation location
- /opt/Xilinx/Vivado/2019.1
- /opt/Xilinx/SDK/2019.1
- /opt/Xilinx/DocNav

Disk Space Required
- Download Size: NA
- Disk Space Required: 48.1 GB
- Final Disk Usage: 32.68 GB

Copyright © 1986-2021 Xilinx, Inc. All rights reserved.

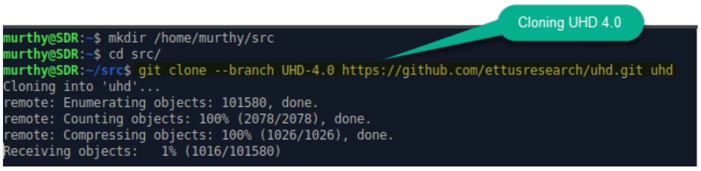
Installing USRP Hardware Driver (UHD 4.0)

Step 6: *Cloning the USRP Hardware Driver (UHD 4.0 branch) from Ettus Research GitHub repository.*

Command:

git clone - -branch UHD-4.0 https://github.com/ettusresearch/uhd.git uhd

Example Image:



```
murthy@SDR:~$ mkdir /home/murthy/src
murthy@SDR:~$ cd src/
murthy@SDR:~/src$ git clone --branch UHD-4.0 https://github.com/ettusresearch/uhd.git uhd
Cloning into 'uhd'...
remote: Enumerating objects: 101580, done.
remote: Counting objects: 100% (2078/2078), done.
remote: Compressing objects: 100% (1026/1026), done.
Receiving objects:   1% (1016/101580)
```

Step 7: *Creating build directory for executing the build process and running cmake to create the make files required for the build process.*

Command:

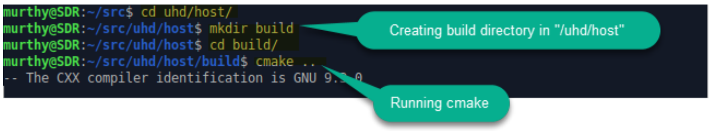
cd /uhd/host/

mkdir build

cd build

cmake ..

Example Image:



```
murthy@SDR:~/src$ cd uhd/host/
murthy@SDR:~/src/uhd/host$ mkdir build
murthy@SDR:~/src/uhd/host$ cd build/
murthy@SDR:~/src/uhd/host/build$ cmake ..
-- The CXX compiler identification is GNU 9.3.0
```

Step 8: *Building & Installing the UHD to the default prefix "**/usr/local/lib**", and updating the shared library cache.*

<u>Command</u>:

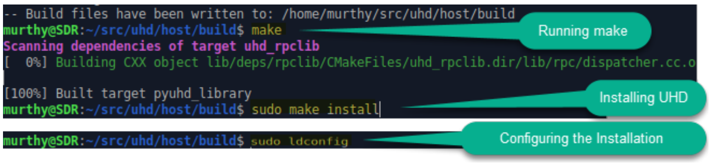
make

make test

sudo make install

sudo ldconfig

<u>Example Image</u>:



```
-- Build files have been written to: /home/murthy/src/uhd/host/build
murthy@SDR:~/src/uhd/host/build$ make
Scanning dependencies of target uhd_rpclib
[  0%] Building CXX object lib/deps/rpclib/CMakeFiles/uhd_rpclib.dir/lib/rpc/dispatcher.cc.o

[100%] Built target pyuhd_library
murthy@SDR:~/src/uhd/host/build$ sudo make install
```

```
murthy@SDR:~/src/uhd/host/build$ sudo ldconfig
```

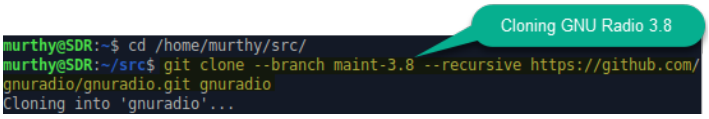
Installing GNU Radio

Step 9*: Cloning the GNU Radio (branch – 3.8) from GNU Radio GitHub repository.*

<u>Command</u>:

git clone - -branch maint-3.8 - -recursive https://github.com/gnuradio/gnuradio.git gnuradio

<u>Example Image</u>:



```
murthy@SDR:~$ cd /home/murthy/src/
murthy@SDR:~/src$ git clone --branch maint-3.8 --recursive https://github.com/
gnuradio/gnuradio.git gnuradio
Cloning into 'gnuradio'...
```

Step 10*: Creating build directory for executing the build process and running cmake to create the make files required for build process.*

<u>Command</u>:

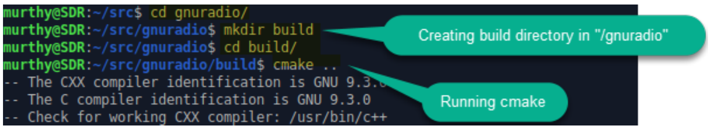
cd /gnuradio/

mkdir build

cd build

cmake ..

<u>Example Image</u>:



```
murthy@SDR:~/src$ cd gnuradio/
murthy@SDR:~/src/gnuradio$ mkdir build
murthy@SDR:~/src/gnuradio$ cd build/
murthy@SDR:~/src/gnuradio/build$ cmake ..
-- The CXX compiler identification is GNU 9.3.0
-- The C compiler identification is GNU 9.3.0
-- Check for working CXX compiler: /usr/bin/c++
```

Step 11*: Building and Installing the GNU Radio to the default prefix "****/usr/local/lib****", and updating the shared library cache.*

<u>Command</u>:

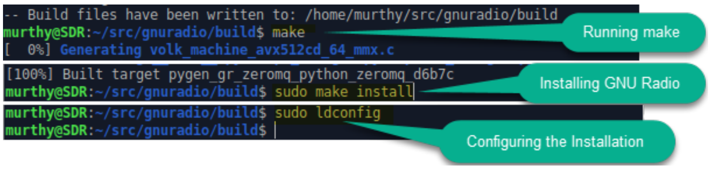
make

make test

sudo make install

sudo ldconfig

Example Image:



```
-- Build files have been written to: /home/murthy/src/gnuradio/build
murthy@SDR:~/src/gnuradio/build$ make
[  0%] Generating volk_machine_avx512cd_64_mmx.c
[100%] Built target pygen_gr_zeromq_python_zeromq_d6b7c
murthy@SDR:~/src/gnuradio/build$ sudo make install
murthy@SDR:~/src/gnuradio/build$ sudo ldconfig
murthy@SDR:~/src/gnuradio/build$ 
```

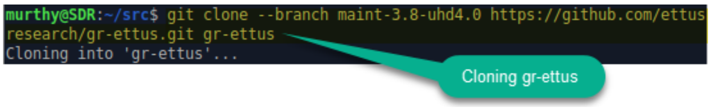
Installing gr-ettus

Step 12: *Cloning the gr-ettus from Ettus Research GitHub repository.*

Command:

git clone - -branch maint-3.8-uhd4.0 https://github.com/ettusresearch/gr-ettus.git gr-ettus

Example Image:



```
murthy@SDR:~/src$ git clone --branch maint-3.8-uhd4.0 https://github.com/ettus
research/gr-ettus.git gr-ettus
Cloning into 'gr-ettus'...
```

Step 13: *Creating build directory for executing the build process and running cmake to create the make files required for build process.*

Command:

cd /gr-ettus/

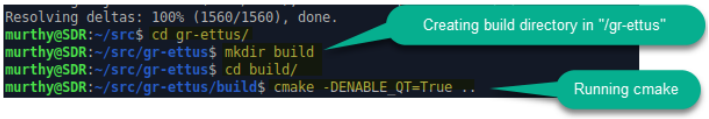
mkdir build

cd build

cmake -DENABLE_QT=True ..

Note: "**-DENABLE_QT=True**" in cmake process will enable the "**gr-fosphor**" installation for RFNOC.

Example Image:



```
Resolving deltas: 100% (1560/1560), done.
murthy@SDR:~/src$ cd gr-ettus/
murthy@SDR:~/src/gr-ettus$ mkdir build
murthy@SDR:~/src/gr-ettus$ cd build/
murthy@SDR:~/src/gr-ettus/build$ cmake -DENABLE_QT=True ..
```

Step 14: *Building and Installing the gr-ettus to the default prefix "**/usr/local/lib**", and updating the shared library cache.*

Command:

make

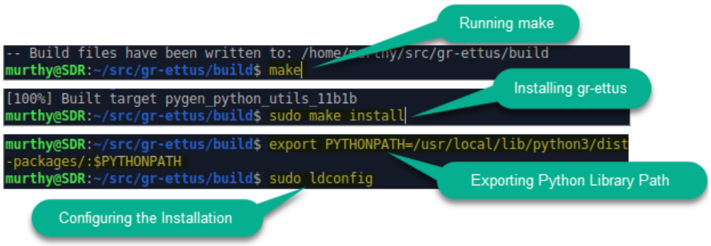
make test

sudo make install

export PYTHONPATH=/usr/local/lib/python3/dist-packages/:$PYTHONPATH

sudo ldconfig

Example Image:



```
-- Build files have been written to: /home/murthy/src/gr-ettus/build
murthy@SDR:~/src/gr-ettus/build$ make
```

```
[100%] Built target pygen_python_utils_11b1b
murthy@SDR:~/src/gr-ettus/build$ sudo make install
```

```
murthy@SDR:~/src/gr-ettus/build$ export PYTHONPATH=/usr/local/lib/python3/dist
-packages/:$PYTHONPATH
murthy@SDR:~/src/gr-ettus/build$ sudo ldconfig
```

Validating the Installations

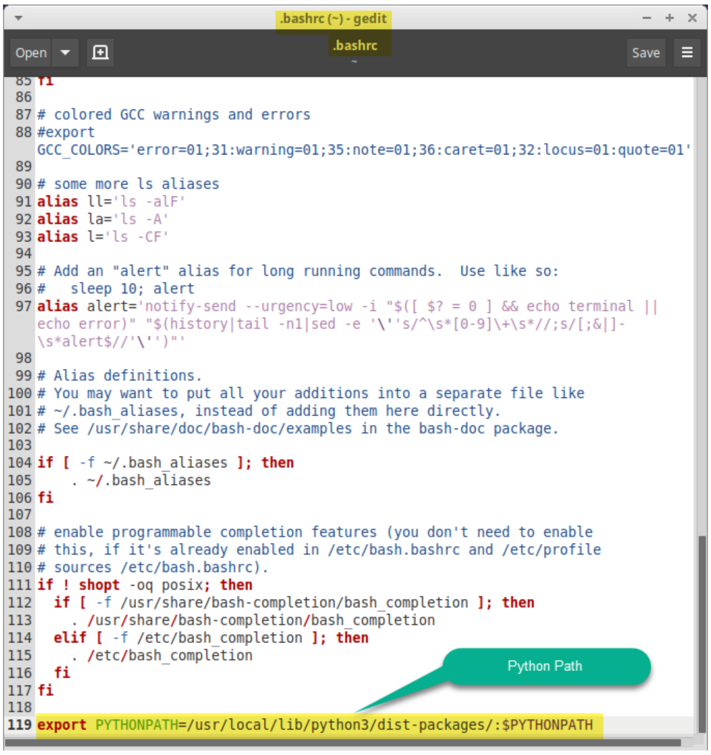
Step 15*: Set the default "****Python Path****" for UHD, GNU Radio & gr-ettus in the "****.bashrc****".*

- The location of the "**.bashrc**" file is "**/home/*user_name*/**.bashrc".
- Open the "**.bashrc**" file and add the below python path at the end of the file and save.

Python Path:

export PYTHONPATH=/usr/local/lib/python3/dist-packages/:$PYTHONPATH

Example Image:



```
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]-\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PYTHONPATH=/usr/local/lib/python3/dist-packages/:$PYTHONPATH
```

__Step 16__: Validating the UHD Installation.

The Installation can be validated without the USRP connected to the system by running the following commands.

Command:

uhd_config_info - -version

Command Response:

UHD 4.0.0.0-133-g7ec04886

(or)

Command:

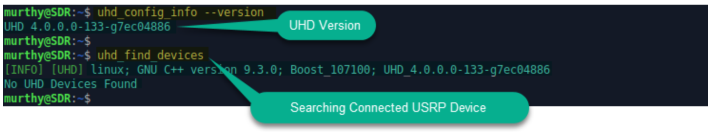
uhd_find_devices

Command Response:

[info] [UHD] Linux; GNU C++ version 9.3.0; Boost_107100; UHD_4.0.0.0-133-g7ec04886

No UHD Devices Found

Example Image:



```
murthy@SDR:~$ uhd_config_info --version
UHD 4.0.0.0-133-g7ec04886
murthy@SDR:~$
murthy@SDR:~$ uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.3.0; Boost_107100; UHD_4.0.0.0-133-g7ec04886
No UHD Devices Found
murthy@SDR:~$
```

__Step 17__: Validating the GNU Radio Installation.

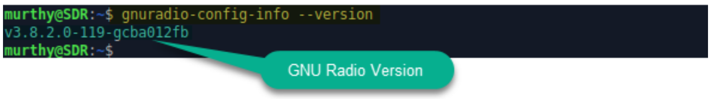
<u>Command</u>:

gnuradio_config_info - -version

<u>Command Response</u>:

V3.8.2.0-199-gcba012fb

<u>Example Image</u>:



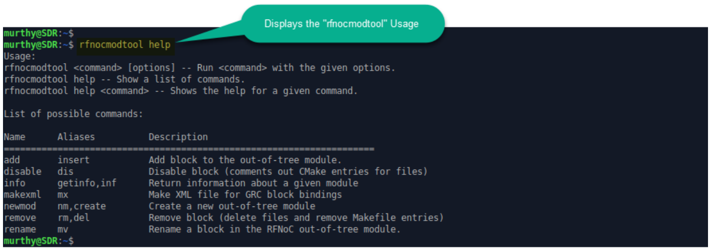
```
murthy@SDR:~$ gnuradio-config-info --version
v3.8.2.0-119-gcba012fb
murthy@SDR:~$
```

***Step 18**: Validating the gr-ettus Installation.*

<u>Command</u>:

rfnocmodtool help

<u>Example Image</u>:



```
murthy@SDR:~$
murthy@SDR:~$ rfnocmodtool help
Usage:
rfnocmodtool <command> [options] -- Run <command> with the given options.
rfnocmodtool help -- Show a list of commands.
rfnocmodtool help <command> -- Shows the help for a given command.

List of possible commands:

Name      Aliases          Description
=====================================================================
add       insert           Add block to the out-of-tree module.
disable   dis              Disable block (comments out CMake entries for files)
info      getinfo,inf      Return information about a given module
makexml   mx               Make XML file for GRC block bindings
newmod    nm,create        Create a new out-of-tree module
remove    rm,del           Remove block (delete files and remove Makefile entries)
rename    mv               Rename a block in the RFNoC out-of-tree module.
murthy@SDR:~$
```

For any information contact :-

iam.gnanamurthy@gmail.com

iam.murthysrikanth@gmail.com

Signaler ceci

Publié par

Murthy S
Applications Engineer | FPGA & Software Defined Radio
Publié • 2 ans

2 articles

+ Suivre

#Wireless #SoftwareDefinedRadio #EttusResearch #GNURadio #RFNOC #USRP #Xilinx #NationalInstruments

J'aime Commenter Partager

15 1 commentaire

Réactions

+3

1 commentaire

Les plus pertinents ▾

Ajouter un commentaire…

Chinmaya Bhat • + que 3e

2 ans •••

Vision ML/DL |Computer vision|Python|low power Edge App development and deployment(Jetson nano,Renesas,Brainchip)|Docker|kubernetes

👍

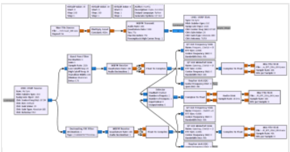
J'aime | Répondre

Murthy S

Applications Engineer | FPGA & Software Defined Radio

+ Suivre

Plus de Murthy S



Frequency Modulation Using Gnu Radio

Murthy S sur LinkedIn