Building and Installing UHD and GNU Radio to a Custom Prefix

Contents

AN-525

Date	Author	Details
2018-12-12	Nate Temple	Initial creation

This application note provides step-by-step instructions on building and installing UHD and GNU Radio to a local directory.

This application note will cover the details of installing UHD and GNU Radio from source, into a local directory on Linux systems. This is often useful if you desire to have multiple UHD and GNU Radio installations on the same system.

This application note uses Ubuntu 18.x for an example. Other versions of Ubuntu and Linux will be similar in process.

Switch your default shell on the host computer from `Dash` to `Bash`. In some Linux distributions (e.g. Ubuntu) `Dash` is set as default shell, which may cause some issues. It is recommended to set the shell to `Bash` by running the following commands in the terminal. Choose `No` when prompted by the first command and the second command will validate the that `Bash` will be used.

```
$ sudo dpkg-reconfigure dash
```

Verify Bash is the default shell.

```
$ ll /bin/sh
```

Expected Output:

```
lrwxrwxrwx 1 root root 4 Apr  2 22:00 /bin/sh -> bash*
```

```
sudo apt update
sudo apt upgrade -y
```

```
sudo apt-get -y install git swig cmake doxygen build-essential libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurs
```

Dependencies for other operating systems including Ubuntu 14.x to 17.x and Fedora can be found here:
https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux#Update_and_Install_dependencies

In this step we will create a `workarea` directory. This will be used to clone the sources and build UHD and GNU Radio.

```
$ mkdir -p ~/workarea
$ mkdir -p ~/workarea/src
```

We will target an arbitary directory `~/workarea/installs` for the installation prefix.

```
$ mkdir -p ~/workarea/installs
```

First clone the UHD sources:

```
$ cd ~/workarea/src
$ git clone --recursive https://github.com/EttusResearch/uhd

$ cd ~/workarea/src/uhd
```

Checkout your desired version of UHD:

To identify git tags, either look at github.com/ettusresearch/uhd or run

```
$ git tag -l
```

Then checkout a tagged release:

Example for UHD 3.9.5:

```
$ git checkout release_003_009_005
```

or example for UHD 3.13.1.0:

```
$ git checkout v3.13.1.0
```

Update the git submodules after checking out the tagged branch:

```
$ git submodule update
```

Finally build UHD:

```
$ cd host
$ mkdir build
$ cd build
```

Note: The CMake parameter `CMAKE_INSTALL_PREFIX` is added in the configuration step, which points to our desired installation prefix of `~/workarea/installs`.

```
$ cmake -DCMAKE_INSTALL_PREFIX=~/workarea/installs ../
$ make -j4
```

Note: Since this installation is not being installed to a system level directory (e.g. `/usr/local`), the `make install` command does not require `sudo` privileges.

```
$ make install
```

On Linux, udev handles USB plug and unplug events. The following commands install a udev rule so that non-root users may access the device. This step is only necessary for devices that use USB to connect to the host computer, such as the B200, B210, and B200mini. This setting should take effect immediately and does not require a reboot or logout/login. Be sure that no USRP device is connected via USB when running these commands.

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

When UHD spawns a new thread, it may try to boost the thread's scheduling priority. If setting the new priority fails, the UHD software prints a warning to the console, as shown below. This warning is harmless; it simply means that the thread will retain a normal or default scheduling priority.

```
UHD Warning:
    Unable to set the thread priority. Performance may be negatively affected.
    Please see the general application notes in the manual for instructions.
    EnvironmentError: OSError: error in pthread_setschedparam
```

To address this issue, non-privileged (non-root) users need to be given special permission to change the scheduling priority.

To enable this, first create a Linux group `usrp`

```
sudo groupadd usrp
```

Add your user to this group:

```
sudo usermod -aG usrp $USER
```

Append the following line to end of /etc/security/limits.conf file.

```
@usrp - rtprio  99
```

This can be performed with the command as shown below:

```
sudo sh -c "echo '@usrp\t-\trtprio\t99' >> /etc/security/limits.conf"
```

You must log out and log back in for this setting to take effect.

```
$ cd ~/workarea/src
$ git clone --recursive https://github.com/gnuradio/gnuradio
$ cd ~/workarea/src/gnuradio
```

Checkout your desired version of GNU Radio:

To identify git tags, either look at github.com/gnuradio/gnuradio or run

```
$ git tag -l
```

Then checkout a tagged release:

```
$ git checkout v3.7.10.2
```

or

```
$ git checkout v3.7.13.4
```

Update the submodules:

```
$ git submodule update
```

Create a directory to build GNU Radio:

```
$ mkdir build
$ cd build
```

Next, configure the build to use the custom installation directory prefix and previously installed version of UHD, by providing the CMake parameters:

- `CMAKE_INSTALL_PREFIX`

- UHD_DIR
- UHD_INCLUDE_DIRS
- UHD_LIBRARIES

```
$ cmake -DCMAKE_INSTALL_PREFIX=~/workarea/installs -DUHD_DIR=~/workarea/installs/lib/cmake/uhd/ -DUHD_INCLUDE_DIRS=~/workarea/installs/inc

$ make -j4
$ make install
```

Since this installation is in a custom directory, we must setup a environment file to tell the operating system where to look for various files.

In the installation directory, create a file `setup.env`:

```
$ cd ~/workarea/installs
$ touch setup.env
```

Add the content below to the `setup.env` file:

```
LOCALPREFIX=~/workarea/installs
export PATH=$LOCALPREFIX/bin:$PATH
export LD_LOAD_LIBRARY=$LOCALPREFIX/lib:$LD_LOAD_LIBRARY
export LD_LIBRARY_PATH=$LOCALPREFIX/lib:$LD_LIBRARY_PATH
export PYTHONPATH=$LOCALPREFIX/lib/python2.7/site-packages:$PYTHONPATH
export PYTHONPATH=$LOCALPREFIX/lib/python2.7/dist-packages:$PYTHONPATH
export PKG_CONFIG_PATH=$LOCALPREFIX/lib/pkgconfig:$PKG_CONFIG_PATH
export UHD_RFNOC_DIR=$LOCALPREFIX/share/uhd/rfnoc/
export UHD_IMAGES_DIR=$LOCALPREFIX/share/uhd/images
```

Next `source` this environment setup file to configure your system to use this new installation. This step must be done anytime you open a new terminal window/shell.

```
$ source setup.env
```

After sourcing the `setup.env` file, you can verify that you're using this local installation with the `which` command.

```
$ which uhd_usrp_probe
```

Expected Output:

```
$ which uhd_usrp_probe
/home/user/workarea/installs/bin/uhd_usrp_probe
```

You can now download the UHD FPGA Images for this installation. This can be done by running the command `uhd_images_downloader`.

```
$ uhd_images_downloader
```

Note: Since this installation is not being installed to a system level directory (e.g. `/usr/local`), the `uhd_images_downloader` command does not require `sudo` privileges.

Example ouput for UHD 3.13.3.0:

```
$ uhd_images_downloader
Images destination:      /home/user/workarea/installs/share/uhd/images
Downloading images from: http://files.ettus.com/binaries/images/uhd-images_003.010.003.000-release.zip
Downloading images to:   /tmp/tmpm46JDg/uhd-images_003.010.003.000-release.zip
57009 kB / 57009 kB (100%)

Images successfully installed to: /home/user/workarea/installs/share/uhd/images
```

Example output for UHD 3.13:

```
$ uhd_images_downloader
[INFO] Images destination: /home/user/workarea/installs/share/uhd/images
[INFO] No inventory file found at /home/user/workarea/installs/share/uhd/images/inventory.json. Creating an empty one.
00006 kB / 00006 kB (100%) usrp1_b100_fw_default-g6bea23d.zip
19484 kB / 19484 kB (100%) x3xx_x310_fpga_default-g494ae8bb.zip
02757 kB / 02757 kB (100%) usrp2_n210_fpga_default-g6bea23d.zip
02109 kB / 02109 kB (100%) n230_n230_fpga_default-g494ae8bb.zip
00522 kB / 00522 kB (100%) usrp1_b100_fpga_default-g6bea23d.zip
00474 kB / 00474 kB (100%) b2xx_b200_fpga_default-g494ae8bb.zip
02415 kB / 02415 kB (100%) usrp2_n200_fpga_default-g6bea23d.zip
05920 kB / 05920 kB (100%) e3xx_e320_fpga_default-g494ae8bb.zip
15883 kB / 15883 kB (100%) n3xx_n310_fpga_default-g494ae8bb.zip
00506 kB / 00506 kB (100%) b2xx_b205mini_fpga_default-g494ae8bb.zip
18676 kB / 18676 kB (100%) x3xx_x300_fpga_default-g494ae8bb.zip
00017 kB / 00017 kB (100%) octoclock_octoclock_fw_default-g14000041.zip
04839 kB / 04839 kB (100%) usb_common_windrv_default-g14000041.zip
00007 kB / 00007 kB (100%) usrp2_usrp2_fw_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n200_fw_default-g6bea23d.zip
00450 kB / 00450 kB (100%) usrp2_usrp2_fpga_default-g6bea23d.zip
00142 kB / 00142 kB (100%) b2xx_common_fw_default-g3ff4186b.zip
00460 kB / 00460 kB (100%) b2xx_b200mini_fpga_default-g494ae8bb.zip
00319 kB / 00319 kB (100%) usrp1_usrp1_fpga_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n210_fw_default-g6bea23d.zip
11537 kB / 11537 kB (100%) n3xx_n300_fpga_default-g494ae8bb.zip
05349 kB / 05349 kB (100%) e3xx_e310_fpga_default-g494ae8bb.zip
00866 kB / 00866 kB (100%) b2xx_b210_fpga_default-g494ae8bb.zip
[INFO] Images download complete.
```

To build a GNU Radio Out Of Tree module (OOT) against this custom installation, you must provide a few additional CMake parameters during the configuration setup as shown below.

```
$ git clone <URL_OF_OOT>
$ cd gr-oot/
```

```
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX=~/workarea/installs -DUHD_DIR=~/workarea/installs/lib/cmake/uhd/ -DUHD_INCLUDE_DIRS=~/workarea/installs/inc
$ make
$ make install
```

This page summarized the step-by-step process involved in building and installing UHD and GNU Radio to a custom prefix. Any questions or feedback should be sent to support@ettus.com.

- https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux
- http://files.ettus.com/manual/