Building and Installing UHD and GNU Radio in an Offline Environment

Contents

AN-524

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

This application note will provide step-by-step instructions on building and installing UHD and GNU Radio in an offline environment.

- Computer with Internet connection for downloading the sources and dependencies
- Git

Note: This application note uses Ubuntu 18.x. in a Virtual Box virtual machine for downloading the sources. For newer versions of Ubuntu, the process will be the same, but the dependencies list may change.

- It is recommended to setup the VM without an internet connection during the install process. This will prevent it from automatically updating.
- During the installation process of the Virtual Machine, do not select "Update packages during install".
- Disable Automatic Updates under the "Software & Updates" section of the System Settings before providing an Internet connection to the VM

On a fresh Virtual Machine run the command:

```
$ sudo apt clean
```

Verify there is no packages in `/var/cache/apt/archives`

```
$ ls -al /var/cache/apt/archives
```

Provide internet access to the Virtual Machine and run the following command to download the require packages:

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

Create a directory to hold the dependency `deb` packages and UHD / GNU Radio source code:

```
$ mkdir -p ~/offline
$ mkdir -p ~/offline/debs
$ mkdir -p ~/offline/src
```

Move the downloaded `.deb` packages to the `~/offline/debs` folder:

```
$ cd ~/offline/debs
$ cp -v /var/cache/apt/archives/*.deb .
```

For this step, you will need to have `git` installed on your [internet connected] host:

```
$ sudo apt install git
```

Then you can clone the UHD and GNU Radio repositories:

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

Optionally, if you're using RFNoC, fetch `gr-ettus`:

```
$ git clone https://github.com/EttusResearch/gr-ettus.git
```

Next, compress the folder `~/offline` with `tar`:

```
$ cd ~/
$ tar zcvf offline.tgz offline/
```

Copy this file, `offline.tgz` to the offline environment.

The steps in this section will all be performed in the offline environment.

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*
```

```
$ tar zxvf offline.tgz
```

Note: The rest of the application note will assume the path `offline.tgz` is extracted to is `~/offline/`. If you use a different location, the paths will need to be updated.

```
$ cd ~/offline/debs
$ sudo dpkg -i *.deb
```

The command `sudo dpkg -i *.deb` will fail part way through, run it again and it will complete.

```
$ sudo dpkg -i *.deb
```

```
$ cd ~/offline/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 3.13.0.2:

```
$ git checkout v3.13.0.2
```

Update the git submodules after checking out the tagged branch:

```
$ git submodule update
```

Finally build UHD:

```
$ cd host
$ mkdir build
$ cd build
$ cmake ../
$ make -j4
$ sudo make install
$ sudo ldconfig
```

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 ~/offline/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
```

```
$ cd ~/offline/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
```

Finally build GNU Radio:

```
$ mkdir build
$ cd build
$ cmake ..
$ make -j4
$ sudo make install
$ sudo ldconfig
```

You will now need to download the corresponding FPGA images for your UHD installation.

Run the command (on your offline machine):

For UHD versions < 3.11.x.x it will print out the exact URL of the FPGA images to be downloaded in the output.

```
$ uhd_images_downloader
```

```
$ sudo uhd_images_downloader
Images destination:      /usr/local/share/uhd/images
Downloading images from: http://files.ettus.com/binaries/images/uhd-images_003.009.005-release.zip
Downloading images to:   /tmp/tmpySQA9q/uhd-images_003.009.005-release.zip
Downloader raised an unhandled exception: HTTPConnectionPool(host='files.ettus.com', port=80): Max retries exceeded with url: /binaries/image
You can run this again with the '--verbose' flag to see more information
If the problem persists, please email the output to: support@ettus.com
```

For UHD versions > 3.11.x.x, you will need to append the command line argument `-l` to list the targets:

```
$ sudo uhd_images_downloader -l
[INFO] Images destination: /usr/local/share/uhd/images
[INFO] Potential targets in manifest file:
# TARGET                          : RELATIVE_URL
b2xx_b200_fpga_default            : b2xx/fpga-494ae8bb/b2xx_b200_fpga_default-g494ae8bb.zip
b2xx_b200mini_fpga_default        : b2xx/fpga-494ae8bb/b2xx_b200mini_fpga_default-g494ae8bb.zip
b2xx_b205mini_fpga_default        : b2xx/fpga-494ae8bb/b2xx_b205mini_fpga_default-g494ae8bb.zip
b2xx_b210_fpga_default            : b2xx/fpga-494ae8bb/b2xx_b210_fpga_default-g494ae8bb.zip
b2xx_common_fw_default            : b2xx/uhd-3ff4186b/b2xx_common_fw_default-g3ff4186b.zip
e3xx_e310_fpga_default            : e3xx/fpga-494ae8bb/e3xx_e310_fpga_default-g494ae8bb.zip
e3xx_e310_fpga_rfnoc              : e3xx/fpga-d6a878b/e3xx_e310_fpga_rfnoc-gd6a878b.zip
e3xx_e320_fpga_aurora             : e3xx/fpga-494ae8bb/e3xx_e320_fpga_aurora-g494ae8bb.zip
e3xx_e320_fpga_default            : e3xx/fpga-494ae8bb/e3xx_e320_fpga_default-g494ae8bb.zip
e3xx_e320_mender_default          : e3xx/meta-ettus-v3.13.1.0/e3xx_e320_mender_default-v3.13.1.0.zip
e3xx_e320_sdimg_default           : e3xx/meta-ettus-v3.13.1.0/e3xx_e320_sdimg_default-v3.13.1.0.zip
e3xx_e320_sdk_default             : e3xx/meta-ettus-v3.13.1.0/e3xx_e320_sdk_default-v3.13.1.0.zip
n230_n230_fpga_default            : n230/fpga-494ae8bb/n230_n230_fpga_default-g494ae8bb.zip
n3xx_common_mender_default        : n3xx/meta-ettus-v3.13.1.0/n3xx_common_mender_default-v3.13.1.0.zip
n3xx_common_sdimg_default         : n3xx/meta-ettus-v3.13.1.0/n3xx_common_sdimg_default-v3.13.1.0.zip
n3xx_common_sdk_default           : n3xx/meta-ettus-v3.13.1.0/n3xx_common_sdk_default-v3.13.1.0.zip
n3xx_n300_fpga_default            : n3xx/fpga-494ae8bb/n3xx_n300_fpga_default-g494ae8bb.zip
n3xx_n310_fpga_default            : n3xx/fpga-494ae8bb/n3xx_n310_fpga_default-g494ae8bb.zip
octoclock_octoclock_fw_default    : octoclock/uhd-14000041/octoclock_octoclock_fw_default-g14000041.zip
usb_common_windrv_default         : usb/uhd-14000041/usb_common_windrv_default-g14000041.zip
usrp1_b100_fpga_default           : usrp1/fpga-6bea23d/usrp1_b100_fpga_default-g6bea23d.zip
usrp1_b100_fw_default             : usrp1/fpga-6bea23d/usrp1_b100_fw_default-g6bea23d.zip
usrp1_usrp1_fpga_default          : usrp1/fpga-6bea23d/usrp1_usrp1_fpga_default-g6bea23d.zip
usrp2_n200_fpga_default           : usrp2/fpga-6bea23d/usrp2_n200_fpga_default-g6bea23d.zip
usrp2_n200_fw_default             : usrp2/fpga-6bea23d/usrp2_n200_fw_default-g6bea23d.zip
usrp2_n210_fpga_default           : usrp2/fpga-6bea23d/usrp2_n210_fpga_default-g6bea23d.zip
usrp2_n210_fw_default             : usrp2/fpga-6bea23d/usrp2_n210_fw_default-g6bea23d.zip
usrp2_usrp2_fpga_default          : usrp2/fpga-6bea23d/usrp2_usrp2_fpga_default-g6bea23d.zip
usrp2_usrp2_fw_default            : usrp2/fpga-6bea23d/usrp2_usrp2_fw_default-g6bea23d.zip
x3xx_x300_fpga_default            : x3xx/fpga-494ae8bb/x3xx_x300_fpga_default-g494ae8bb.zip
x3xx_x310_fpga_default            : x3xx/fpga-494ae8bb/x3xx_x310_fpga_default-g494ae8bb.zip
```

Note: The example above is using UHD 3.13.0.2, the `RELATIVE_URL` value will change for different UHD versions.

Append the `RELATIVE_URL` value for your device to the URL below:

http://files.ettus.com/binaries/cache/

For example, the FPGA images for the X310 will be located at:

http://files.ettus.com/binaries/cache/x3xx/fpga-494ae8bb/x3xx_x310_fpga_default-g494ae8bb.zip

For UHD version >= 3.13.1.0, you can use the command line options `--list-targets --url-only` to print out the URLs of the FPGA packages:

```
$ uhd_images_downloader --url-only --list-targets
[INFO] Images destination: /usr/local/share/uhd/images
http://files.ettus.com/binaries/cache/usrp1/fpga-6bea23d/usrp1_b100_fw_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/x3xx/fpga-d0360f7/x3xx_x310_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_n210_fpga_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/n230/fpga-d0360f7/n230_n230_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/n3xx/fpga-494ae8bb/n3xx_n300_fpga_aurora-g494ae8bb.zip
http://files.ettus.com/binaries/cache/usrp1/fpga-6bea23d/usrp1_b100_fpga_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/e3xx/fpga-494ae8bb/e3xx_e320_fpga_aurora-g494ae8bb.zip
http://files.ettus.com/binaries/cache/e3xx/meta-ettus-v3.13.1.0/e3xx_e320_sdk_default-v3.13.1.0.zip
http://files.ettus.com/binaries/cache/b2xx/fpga-d0360f7/b2xx_b200_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_n200_fpga_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/e3xx/fpga-d0360f7/e3xx_e320_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/n3xx/fpga-d0360f7/n3xx_n310_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/b2xx/fpga-d0360f7/b2xx_b205mini_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/x3xx/fpga-d0360f7/x3xx_x300_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/octoclock/uhd-14000041/octoclock_octoclock_fw_default-g14000041.zip
http://files.ettus.com/binaries/cache/e3xx/meta-ettus-v3.13.1.0/e3xx_e320_sdimg_default-v3.13.1.0.zip
http://files.ettus.com/binaries/cache/n3xx/fpga-6bea23d/n3xx_n310_cpld_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_usrp2_fw_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_n200_fw_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_usrp2_fpga_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/b2xx/uhd-3ff4186b/b2xx_common_fw_default-g3ff4186b.zip
http://files.ettus.com/binaries/cache/n3xx/fpga-494ae8bb/n3xx_n310_fpga_aurora-g494ae8bb.zip
http://files.ettus.com/binaries/cache/b2xx/fpga-d0360f7/b2xx_b200mini_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/usrp1/fpga-6bea23d/usrp1_usrp1_fpga_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/n3xx/meta-ettus-v3.13.1.0/n3xx_common_mender_default-v3.13.1.0.zip
http://files.ettus.com/binaries/cache/e3xx/meta-ettus-v3.13.1.0/e3xx_e320_mender_default-v3.13.1.0.zip
http://files.ettus.com/binaries/cache/usb/uhd-14000041/usb_common_windrv_default-g14000041.zip
http://files.ettus.com/binaries/cache/e3xx/fpga-d6a878b/e3xx_e310_fpga_rfnoc-gd6a878b.zip
http://files.ettus.com/binaries/cache/usrp2/fpga-6bea23d/usrp2_n210_fw_default-g6bea23d.zip
http://files.ettus.com/binaries/cache/n3xx/fpga-d0360f7/n3xx_n300_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/n3xx/meta-ettus-v3.13.1.0/n3xx_common_sdk_default-v3.13.1.0.zip
http://files.ettus.com/binaries/cache/e3xx/fpga-d0360f7/e3xx_e310_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/b2xx/fpga-d0360f7/b2xx_b210_fpga_default-gd0360f7.zip
http://files.ettus.com/binaries/cache/n3xx/meta-ettus-v3.13.1.0/n3xx_common_sdimg_default-v3.13.1.0.zip
```

On your Internet connected host, fetch the FPGA images using `wget`:

Example for < UHD 3.11.x.x:

```
$ wget http://files.ettus.com/binaries/images/uhd-images_003.009.005-release.zip
```

Example for > UHD 3.11.x.x.

```
$ wget http://files.ettus.com/binaries/cache/x3xx/fpga-494ae8bb/x3xx_x310_fpga_default-g494ae8bb.zip
```

Transfer the downloaded ZIP file(s) to your offline environment.

Create a folder in the directory `~/offline` called `fpga_images`. This directory name is arbitrary, and is only used for reference within the following steps. Place the zip file containing the FPGA images into this directory, and decompress.

```
$ mkdir -p ~/offline/fpga_images
$ cd ~/offline/fpga_images
$ cp -v ~/path/to/fpga_images_package.zip .
$ unzip fpga_images_package.zip
```

Next, create the `images/` folder in your UHD installation prefix.

```
$ cd /usr/local/share/uhd/
$ mkdir -p images
$ cd images
```

Next, copy the FPGA images that were decompressed above to this location.

```
$ sudo cp -v ~/offline/fpga_images/uhd-images-xxxx/share/uhd/images/* .
```

This page summarized the step-by-step process involved in setting up an offline installation of UHD and GNU Radio. 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
- https://files.ettus.com/manual/