

Building and Installing the USRP Open Source Toolchain (UHD and GNU Radio) on Windows

Contents

- 1 Application Note Number
- 2 Revision History
- 3 Abstract
- 4 UHD on Windows
- 5 Setup the Environment
- 6 Installing the dependencies
 - ◆ 6.1 Microsoft Visual Studio
 - ◆ 6.2 CMake
 - ◆ 6.3 Boost
 - ◆ 6.4 LibUSB
 - ◆ 6.5 Python
 - ◆ 6.6 Mako
 - ◆ 6.7 Doxygen
 - ◆ 6.8 NSIS
- 7 Obtain the UHD source code
 - ◆ 7.1 From a release source archive
 - ◆ 7.2 From the Git repository
- 8 Building
 - ◆ 8.1 Configuring the Building
 - ◆ 8.2 Compiling UHD
- 9 Installing UHD
 - ◆ 9.1 Compiling a binary installer
- 10 Running UHD
- 11 Installing the USB driver
- 12 Note on Dependencies for Current and Previous Versions
- 13 GNU Radio and UHD

AN-611

Date	Author	Details
2016-01-19	Derek Kozel	Initial creation
2017-03-07	Nate Temple	Update GNU Radio instructions
2018-12-14	Sam Reiter	Update UHD instructions

This Application Note provides a step-by-step guide for building, installing, and updating the open-source toolchain, specifically UHD and GNU Radio, for the USRP from source code on Windows.

UHD is fully supported on Windows 7, 8, 8.1, and 10 and can be compiled using Visual Studio 2013, 2015, or 2017. The future target is to support the latest three OS and Visual Studio releases.

The following dependencies are required for a regular build.

- Microsoft Windows (7, 8, 8.1, 10)
- Microsoft Visual Studio (2013, 2015, 2017)
- CMake (2.8.0 or later)
- Boost (1.53 or later)
- LibUSB (1.0 or later)
- Python (2.7.x)
- Mako (0.5.0 or later)
- Doxygen (1.8 or later, optional)
- NSIS (2.50 or later, optional)

Optional Tools:

- 7zip (<http://www.7-zip.org/download.html>)
- msysGit (<https://gitforwindows.org/>)

The free version of Visual Studio is sufficient for building UHD. This guide was tested using Windows 10 x64 and Microsoft Visual Studio Community 2017 (v15.9).

Users will need to install the "Desktop Development for C++" Workload for Visual Studio. This can be found in:

Tools >> Get Tools and Features... >> Workloads

CMake is a cross-platform build system used to configure and generate the files necessary to compile and test UHD for a particular computer environment.

During installation select the option to add CMake to the PATH for the current user.

CMake 3.13.2 (win64-x64) was used for the guide.

<https://cmake.org/download/>

Boost is a set of C++ libraries providing useful algorithms and data structures.

The Boost binary installer must be selected to match the version of MSVC being used to compile UHD and architecture of Windows being run. Watch out as Microsoft has done the version numbering of MSVC such that the year and version number do not match.

VS2017 uses the MSVC Toolset 14.1 so boost_1_68_0-msvc-14.1-64 was selected.

<http://sourceforge.net/projects/boost/files/boost-binaries/>

LibUSB is a cross-platform library providing access to USB devices.

LibUSB releases are distributed in 7zip archives, the free 7zip program needed to open them is linked in the ?Setting Up the Environment? Section. After installing 7zip the LibUSB release archive can be extracted by right clicking on the downloaded file and selecting 7zip >> Extract files.

LibUSB 1.0.22 was used for the guide and extracted to C:\Users\username\libusb-1.0.22.

<http://sourceforge.net/projects/libusb/files/libusb-1.0/>

Note: The directory to which you extract libusb must not contain spaces. This is to say that C:\Users\user name\libusb-1.0.22 will cause compile issues moving forward.

Python is a widely used general-purpose, high-level programming language. UHD includes several utilities written in Python and has several scripts which are part of the build process.

The latest Python 2 release, 2.7.15 was used.

<https://www.python.org/downloads/>

Mako is a python template library used to generate source files and is distributed using a Python package management system.

Open a command line (Ctrl-X, select Run, type cmd.exe and click OK)

```
cd C:\Python27\Scripts
pip.exe install mako
```

This installed version 1.0.7 at the time of writing this guide.

Doxygen is a documentation generator which creates the HTML manual from text in the source code. It optional for building UHD.

Version 1.8.14 was used for this guide.

<https://sourceforge.net/projects/doxygen/files/rel-1.8.14/>

NSIS is a toolkit for creating Windows installers. NSIS is used for creating binary packages of UHD enabling easy distribution and installation of UHD, associated utilities, and examples.

Version 3.03 was used for this guide.

<https://sourceforge.net/projects/nsis/files/NSIS%203/>

Archives of the source code for each stable release can be downloaded from the Ettus website.

<http://files.ettus.com/binaries/uhd/>

7zip can be used to extract the tar.gz archive to a location of your choosing.

This guide used the Git repository to obtain the newest release of UHD, read on for details.

The latest development code, as well as tagged releases, is available from the git repository hosted on GitHub

<https://www.github.com/EttusResearch/uhd>

For step-by-step instructions using a git client, see section [Building and installing UHD from source code](#) in the UHD Linux Installation Guide. Return to this document after you have successfully checked out your desired release of UHD, and note that some directory names moving forward may differ slightly.

For the purposes of this guide, uhd-master.zip (v3.14.0.0) was downloaded from Github, and extracted C:\Users\username\uhd-master\.

All prerequisites have now been installed and downloaded. You are encouraged to restart your machine before continuing.

- Open the Cmake GUI
- Select source code directory
C:\Users\username\uhd-master\host\
- Select binary build directory (this may require creating the folder \build)
C:\Users\username\uhd-master\host\build\
- Check the Advanced checkbox
- Click Configure
 - ◆ Set Visual Studio 15 2017 Win64 as the compiler
 - ◆ Click ?Finish? and allow CMake to Generate
- Change or add the following entries
 - Boost_INCLUDE_DIR**
C:\local\boost_1_68_0\

- Add the following entries with type PATH
 - Boost_LIBRARY_DIRS**
C:\local\boost_1_68_0\lib64-msvc-14.1\
 - LIBUSB_INCLUDE_DIRS**
C:\Users\username\libusb-1.0.22\include\libusb-1.0\
- Add the following entry with type FILEPATH
 - LIBUSB_LIBRARIES**
C:\Users\username\libusb-1.0.22\MS64\dll\libusb-1.0.lib
- Click Generate

Open Visual Studio 2017 and open the UHD project file generated by CMake.

```
File > Open Project
C:\Users\username\uhd-master\host\build\UHD.sln
```

Change the build type from Debug to Release. The **ALL_BUILD** project should be selected in the Solution Explorer, select it if this is not the case. Run the build, `Build > Build ALL_BUILD`.

Select the **INSTALL** project in the Solution Explorer and run the build, `Build > Build INSTALL`.

Visual Studio must be run as Administrator for this to succeed as it needs write permission for the `C:\Program Files` directory.

Building the **PACKAGE** project will produce a binary installer if NSIS is installed. This installer will be for either 64 bit or 32 bit as chosen during the CMake step. Select the **PACKAGE** project in the Solution Explorer and run the build, `Build > Package`.

Running programs using UHD requires the Visual Studio C++ Runtime Redistributable to be installed. The version of the C++ Runtime Redistributable must match the version of Visual Studio used to compile the program.

- Visual Studio 2017
- Visual Studio 2015
- Visual Studio 2012 and 2013

If a USB connected USRP is used then the USB drivers must be installed. The drivers are located at http://files.ettus.com/binaries/misc/erlrc_uhd_winusb_driver.zip.

There is a known issue with Windows 10 where an error message is shown at the end of driver installation. However resetting or power cycling the USRP enables full functionality.

It should be noted that this guide is intended to serve as an example; the software versions and dependencies listed above are not the only possible combinations. Be sure to review the [Build Dependencies](#) in the [USRP Hardware Driver and USRP Manual](#) for information regarding the minimum required driver versions for your build of UHD. To preserve the software stack from the previous iteration of this Application Note, here is an example driver stack for Windows 7 + VS 2013:

- Microsoft Windows 7 x64
- Microsoft Visual Studio 2013
- CMake 3.4.1 win32-x86
- Boost 1.58.0 msvc-12.0 x64
- LibUSB 1.0.20
- Python 2.7.11
- Mako 1.0.3
- Doxygen 1.8
- NSIS 2.50

Building GNU Radio from source on Windows is still an involved process due to the large number of dependencies. A set of scripts have been developed to automate the process by Geof Nieboer. The links below will detail the process to building GNU Radio + UHD.

Note: The linked instructions below will build GNU Radio along with UHD, which is separate from the instructions above within this Application Note. The scripts linked below are not maintained by Ettus Research, and are considered third-party binary packages, and are not directly supported by Ettus Research.

https://github.com/gnieboer/gnuradio_windows_build_scripts <http://www.gcndevelopment.com/gnuradio/>