

Report: Migrating a UHD-Fairwaves Ethernet-Based Circuit from Linux to Windows using Python Wrappers

Overview

This report outlines the requirements, components, and steps involved in migrating a C-based circuit system from Linux to Windows. The circuit communicates over an Ethernet interface and uses the UHD driver provided by Fairwaves. UHD (USRP Hardware Driver) is a user-space C++ library that enables communication between host systems and software-defined radio (SDR) hardware. The migration process also involves the creation of Python bindings or wrappers to interface the existing C components on a Windows platform.

1. System Description

- **Circuit:** An SDR-based circuit with an Ethernet interface for communication.
- **Existing Environment:** The system currently runs on Linux.
- **Programming Language:** The control and communication logic is written in **C**, using UHD libraries.
- **Dependencies:**
 - UHD Fairwaves Driver
 - Python Wrappers/Bindings
 - Python Tools (e.g., SWIG, pybind11) for interfacing C/C++ with Python

2. Linux Setup

To make the circuit work on Linux, the following components must be installed:

2.1 Dependencies

1. UHD Fairwaves Driver

A modified version of the official UHD driver tailored for Fairwaves hardware.

Repository: <https://github.com/fairwaves/UHD-Fairwaves>

2. Python Wrapper Tools

These allow Python code to call functions from C/C++ code.

Common tools:

SWIG (Simplified Wrapper and Interface Generator)

pybind11 (C++11 headers for binding)

Cython (generates C code from Python)

3. Python Packages

numpy, scipy, matplotlib (if plotting or signal processing is involved)

setuptools, wheel, pip for package management

2.2 How the UHD Driver Works

- The UHD Fairwaves driver acts as a user-space C++ library.
- It facilitates the transfer of control commands and data between the host system (Linux) and the SDR circuit (via Ethernet).
- UHD uses socket-based communication (UDP/TCP) to talk to the circuit over Ethernet.

2.3 How the Python Wrapper Works

- A Python wrapper allows Python scripts to interact with the C/C++ UHD driver functions.
- It uses bindings (via tools like SWIG or pybind11) to expose C classes/functions to Python.

3. Windows Migration Requirements

The goal is to make the circuit work on Windows using the same Ethernet interface and UHD Fairwaves driver, with a Python interface for control and testing.

3.1 General Strategy

- Install and configure UHD Fairwaves on Windows.
- Build the driver from source code.
- Create Python wrappers for Windows using appropriate tools.

3.2 Tools to Install on Windows

1. CMake

Required to build UHD from source.

2. Visual Studio (with C++ Toolset)

Needed to compile C++ libraries and the driver.

3. **Python (3.x)**

With pip, setuptools, virtualenv, etc.

4. **Wrapper Tools:**

SWIG

If starting from scratch, SWIG is easier for C code.

3.3 Steps to Migrate

1. **Clone the UHD-Fairwaves Repository:**

2. `git clone https://github.com/fairwaves/UHD-Fairwaves.git`

3. **Build UHD on Windows:**

Use CMake GUI or CLI to configure the build.

Generate a Visual Studio solution.

Build the project from Visual Studio.

4. **Install UHD Utilities:**

Ensure tools like `uhd_usrp_probe`, `uhd_rx_cfile`, etc., are working.

5. **Create Python Wrapper:**

Identify the core functions in the UHD C++ API needed by your application.

Write interface files if using SWIG, or module code if using pybind11.

Compile the wrapper into a Python extension (.pyd file).

6. **Test Python Communication:**

Write Python scripts that use the wrapper to call the driver functions.

Verify end-to-end connectivity with the circuit via Ethernet.