

# **Progress Report: Migrating a UHD-Fairwaves Ethernet-Based Circuit from Linux to Windows Using Python Wrappers**

**Date:** July 22, 2025

## **Objective**

The primary objective of this project is to migrate a Linux-based SDR circuit system that utilizes the UHD-Fairwaves driver to a Windows environment. This includes adapting the build system and source code for compatibility with Windows and creating Python wrappers to enable Python-based communication with the circuit over Ethernet. The goal is to maintain the full functionality of the original system while ensuring cross-platform operability and ease of use through Python.

## **Summary of Work Completed**

### **1. Environment Preparation**

Installed all necessary development tools and libraries using Homebrew, including CMake, Boost, and UHD. Also ensured that the Python environment is correctly configured for any future scripting or testing.

### **2. Cloning the UHD-Fairwaves Repository**

Successfully cloned the Fairwaves fork of UHD to the local machine and set up the working directory on the desktop.

### **3. Build Directory Setup**

Created a dedicated build directory and navigated into it to keep the build files organized and separate from the source.

### **4. Initial CMake Configuration**

Faced a compatibility issue during the first attempt at running CMake, which was resolved by adjusting the configuration to match minimum policy and standard requirements.

### **5. Build Attempt**

Initiated the build process. The process started successfully but encountered build errors related to deprecated Boost libraries, which are no longer supported in newer Boost versions.

### **6. Ongoing Debugging**

Currently addressing the build errors by updating deprecated code in the source files. Once these updates are complete, I will re-attempt the build and proceed with further testing.

## **Next Steps**

- Continue identifying and resolving build errors related to deprecated or unsupported C++/Boost components.
- Rebuild the project after applying the necessary fixes.
- Choose and implement a Python wrapping method for interfacing with the compiled C/C++ driver on Windows.