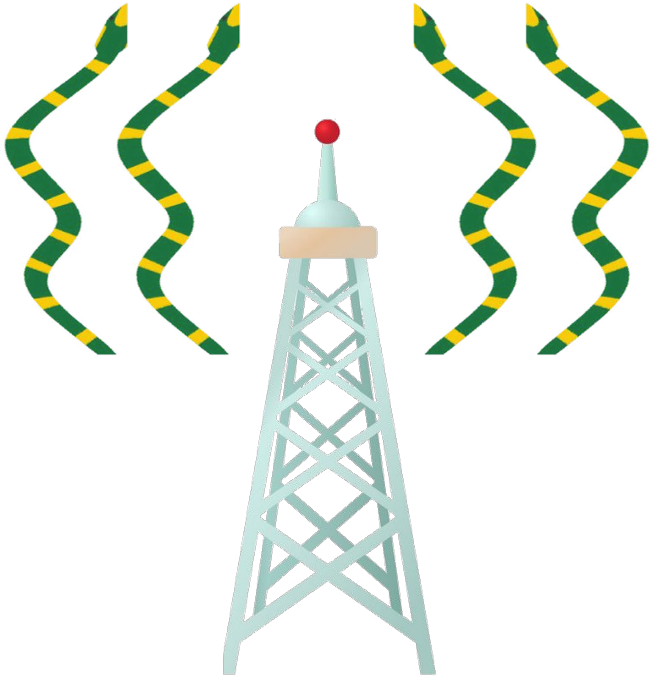
UHD for Pythonistas

Brent Stapleton
Ettus Research



Overview

- Python API Current Status
- UHD Pythons in the “wild”
  - Calibrating the Colosseum
  - Ettus CI Testing
  - Embedded UHD Python API
- Possible Applications

UHD Python API

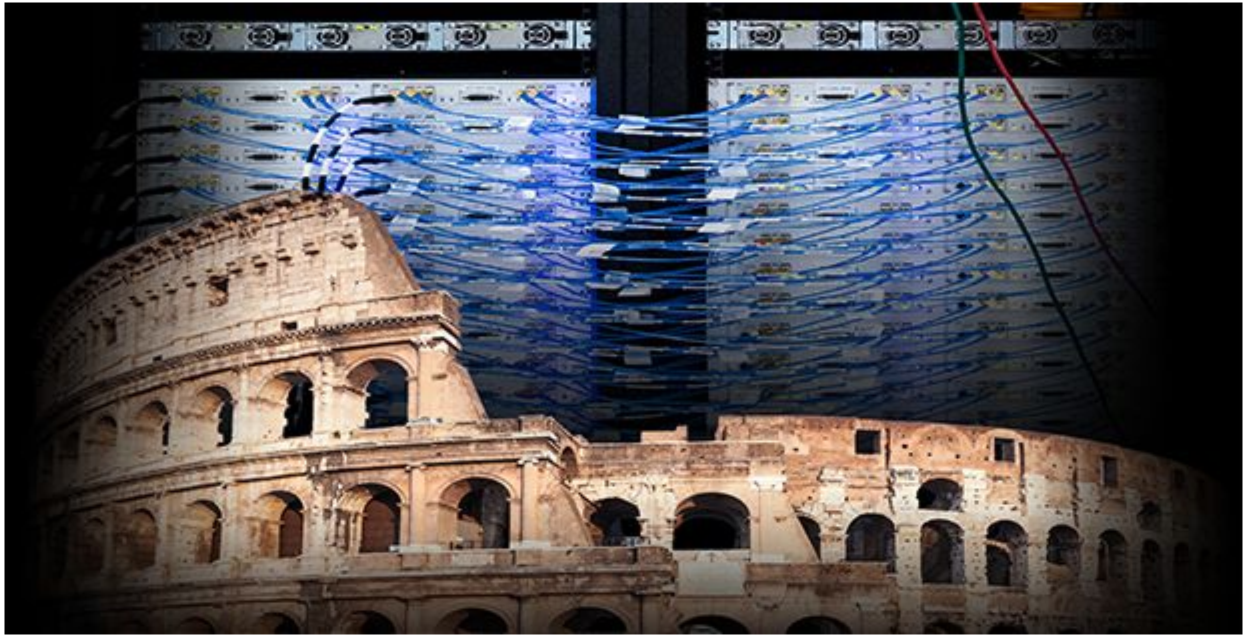
- Uses Boost Python to wrap C++ API
  - MultiUSRP API exposure through Python
- Separate API from gr-uhd
  - Very few use cases where these will be mixed

Current Status

- Fully merged into UHD master branch
  - CMake option `-DENABLE_PYTHON_API=ON`
- Easiest to install on Linux
  - Windows installers in the works

Colosseum

- DARPA Spectrum Challenge was conducted in the Colosseum Environment Emulator



Colosseum - Calibration

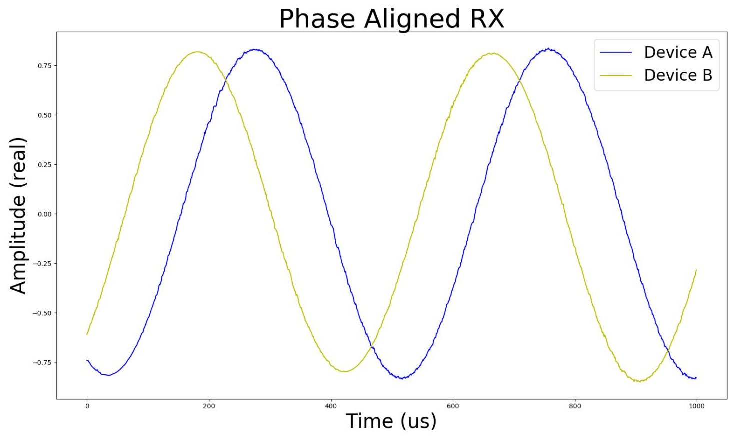
- Fairness for competitors was a high priority
- Calibration was done using UHD Python API
  - Pairs of USRPs take turns transmitting/receiving
  - Error Vector Magnitude (EVM) for a given waveform was computed for each pair

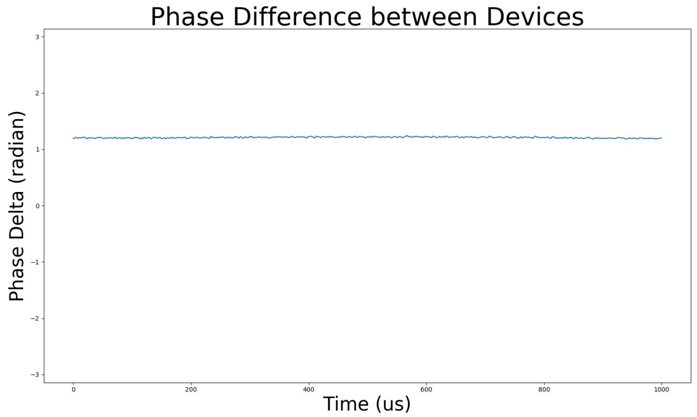
Ettus CI - Signal Processing

- Phase alignment tests for devices moving to Python API
  - GNU Radio works well, but is bulky
  - Phase alignment algorithm: s1 * conj(s2)
    - Super simple
    - All necessary function in NumPy

```
alignment = np.angle(np.conj(samps[0]) * samps[1])
```

Ettus CI - Signal Processing





Setup

- 2x USRP X310’s with UBX-40 dboard
  - Shared PPS and 10MHz reference clock provided by an Octoclock
- Signal Generator (USRP B200, in this case)

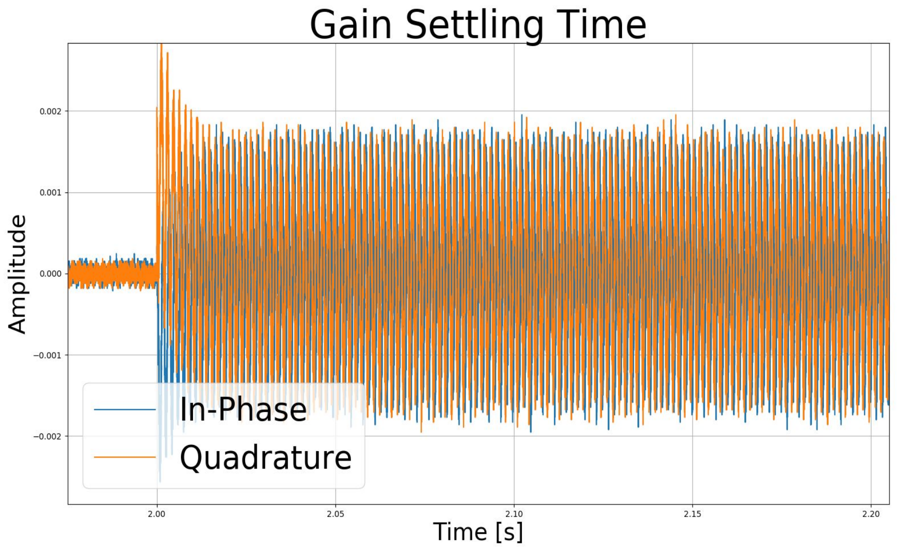
Single Run Results

- A few seconds of RX
- Constant phase difference throughout the test, with a standard deviation <1 degree between 2 signals

Ettus CI - Signal Processing

Ettus Research™ A National Instruments Company

- Opportunities for other simple RF tests
  - Gain settling time
  - Spur detection
  - Anything else that NumPy+SciPy can process

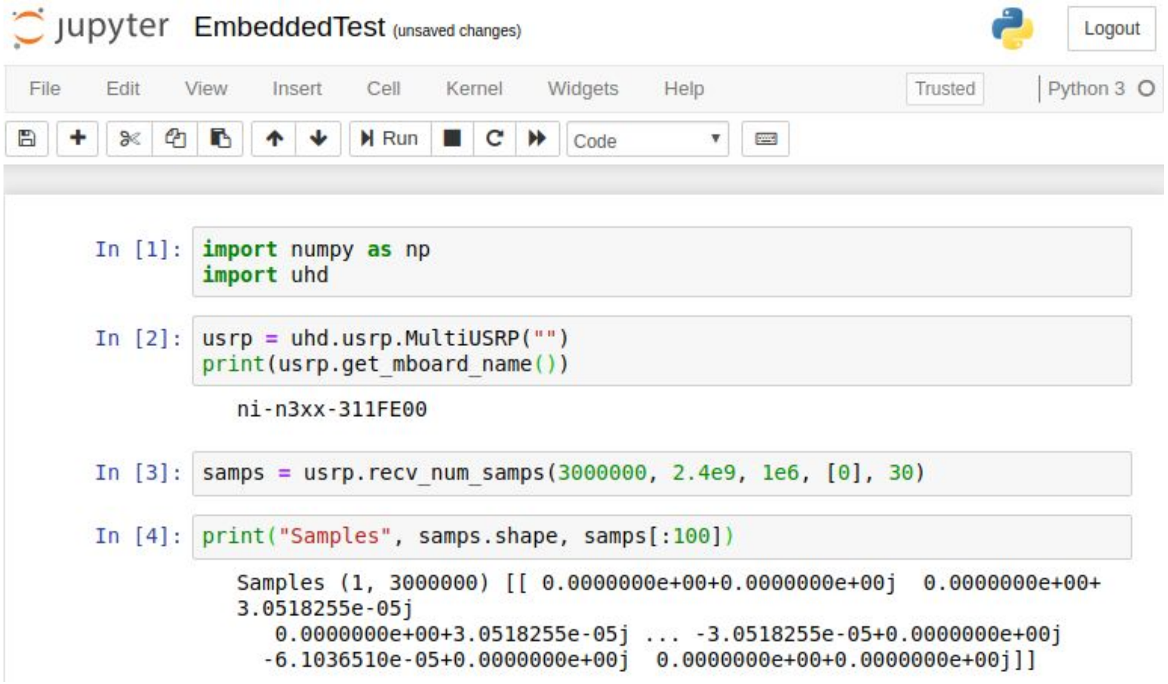


Embedded Python API

- Python API not built by default on MPM-enabled devices… But it does work*!
  - *With some finagling
  - Performance is slightly worse than C++ applications

Embedded Python API

```
root@ni-n3xx-311FE00: # jupyter-notebook --no-browser \
                          --port=4037 --allow-root
```



```
In [1]: import numpy as np
        import uhd

In [2]: usrp = uhd.usrp.MultiUSRP("")
        print(usrp.get_mboard_name())

          ni-n3xx-311FE00

In [3]: samps = usrp.recv_num_samps(3000000, 2.4e9, 1e6, [0], 30)

In [4]: print("Samples", samps.shape, samps[:100])

          Samples (1, 3000000) [[ 0.0000000e+00+0.0000000e+00j   0.0000000e+00+
          3.0518255e-05j
             0.0000000e+00+3.0518255e-05j ... -3.0518255e-05+0.0000000e+00j
            -6.1036510e-05+0.0000000e+00j   0.0000000e+00+0.0000000e+00j]]
```

Other Applications

Ettus Research™

- We're hoping to see more applications use the UHD Python API
- Trillions* of Python modules available

NumPy

TensorFlow

theano

SciPy

jupyter

bokeh

*approximately

Summary

- UHD Python API available since UHD 3.13, but needs to be enabled through compiler flags
- Not a complete replacement for gr-uhd or GNURadio in general, but has clear benefits for simple DSP and non-streaming applications especially
- Usage questions to the USRP mailing list (usrp-users@lists.ettus.com)
- Bug reports to the UHD Github Issue tracker (https://github.com/EttusResearch/uhd/issues)