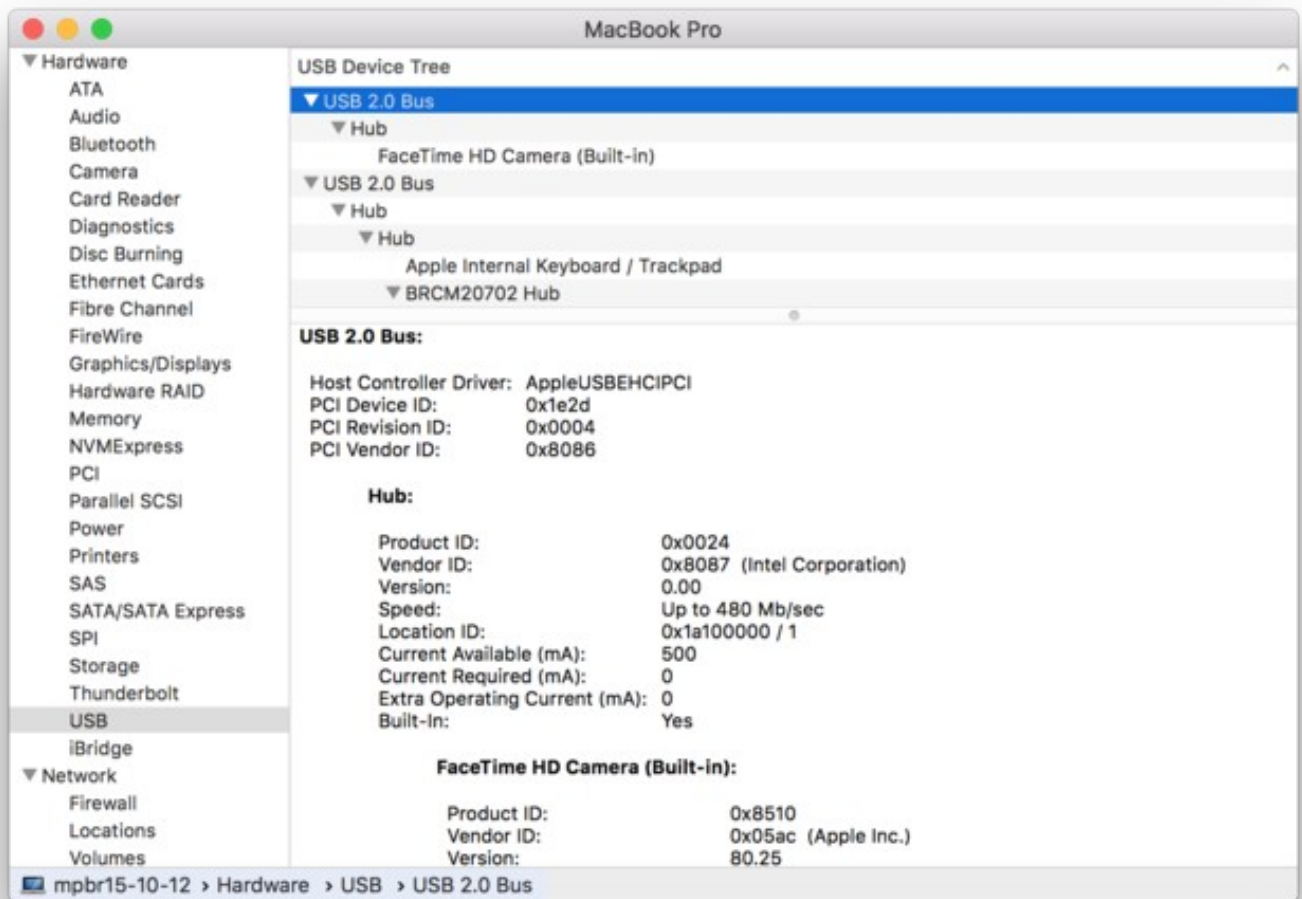
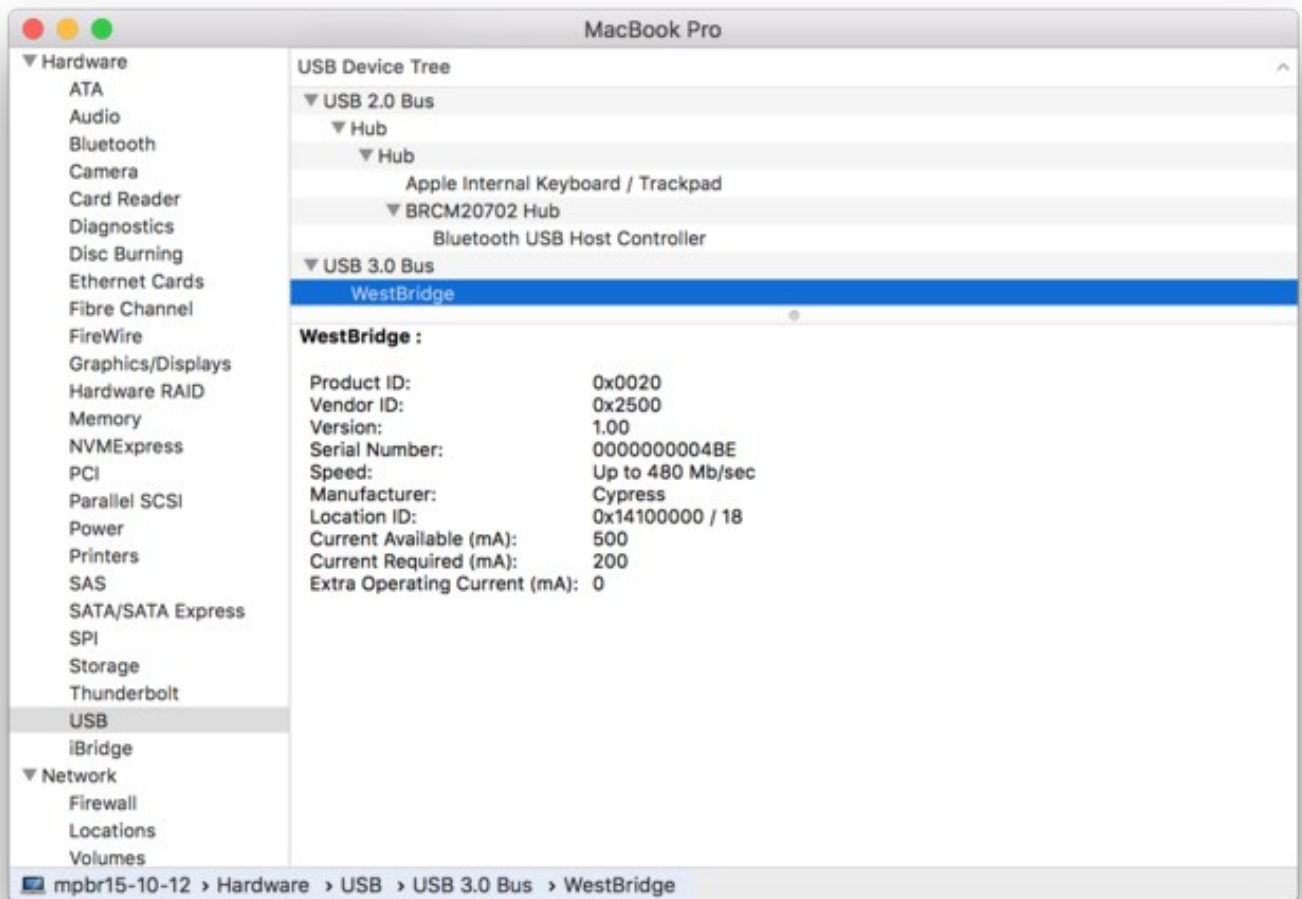


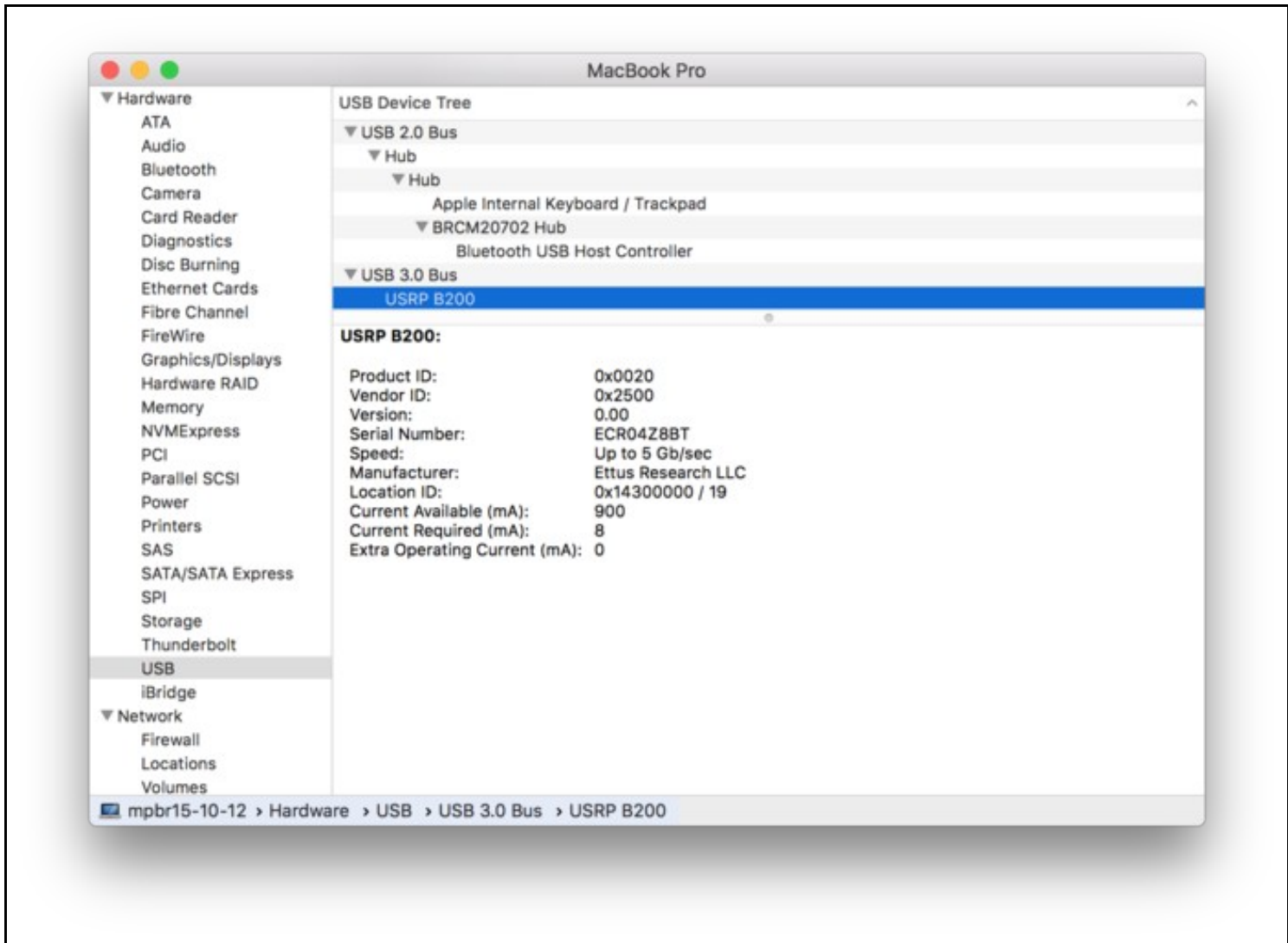
Clicking on the USB option shows the USB device tree in the upper pane and information for the first instantiated bus in the lower pane.



In the upper pane, scroll down until you find an entry for either *Westbridge* or *USRP B2**. The following screenshot shows the hardware information of a *USRP B210* immediately after it is plugged in and powered up.



Once the USRP firmware is loaded -- more on this in the next section -- the device with "morph" on the bus, which will change the device name and capabilities. The following screenshot shows the hardware information of a USRP B210 after the firmware was successfully loaded.



When a USRP B-series device is plugged into a USB port and powers up, both the firmware and FPGA images need to be loaded for it to work with UHD. These binaries will be loaded when UHD is started and pointed to the B-series USRP, regardless of the application, host computer, or OS.

Executing `uhd_find_devices` will load up just the firmware. Typical terminal information would look like the following, which uses UHD 4.0.0.0 release as installed by MacPorts on macOS 11 Big Sur; this information will be roughly the same regardless of the version of UHD or OSX / macOS. Note that the actual serial number for your USRP will be printed; we have X'd it out here. Also note that in this case UHD succeeded in loading the firmware and macOS also successfully reenumerated USB. The `name` entry is likely to be different for each USRP, as it is user-settable.

```
% uhd_find_devices
[INFO] [UHD] Mac OS; Clang version 12.0.0 (clang-1200.0.32.28); Boost_107100; UHD_4.0.0.0-MacPorts-20201218-3b9ced8f
[INFO] [B200] Loading firmware image: /opt/local/share/uhd/images/usrp_b200_fw.hex...
-----
-- UHD Device 0
-----
Device Address:
  serial: XXXXXXXX
  name: MyB210
  product: B210
  type: b200
```

For older revision B-series USRPs such as the white B200/B210, the USB morphing / reenumeration fails around 50% of the time upon initial USRP power up. When this event occurs, any UHD-based application will fail to find the USRP once the firmware is loaded; here is an example with the same USRP B210 as just above.

```
% uhd_find_devices
[INFO] [UHD] Mac OS; Clang version 9.0.0 (clang-900.0.39.2); Boost_107100; UHD_3.15.0.0-MacPorts-Release
[INFO] [B200] Loading firmware image: /opt/local/share/uhd/images/usrp_b200_fw.hex...
No UHD Devices Found
```

When USB and the USRP fail in this way, the USRP must be *reset* for it to work. There are 3 primary options for resetting that seem to work reliably on OSX / macOS.

1. Unplug the USB cable from the host computer and plug it back into the exact same USB port. This technique works most of the time, and is applicable to all B-series USRPs.
2. B200/B210 : Use an external power supply, and then unplug / replug the USB connection; note that you can use any USB port in this option. The USRP will reset USB, but will not power cycle since it is powered throughout the procedure.
3. B200/B210 : Leave the USRP plugged into the host computer via USB, and press the USB reset button (S700) on the USRP. The following images show the switch location on two revisions of the B200/B210.



Sdd

10MHz



S102

Once the B2* USRP has the firmware loaded properly, `uhd_find_devices` will find it properly. Next up, execute `uhd_usrp_probe` to load the FPGA image and return basic information about this USRP. For the B210, the following is typical information printout for UHD 3.15:

```
% uhd_usrp_probe
[INFO] [UHD] Mac OS; Clang version 9.0.0 (clang-900.0.39.2); Boost_107100; UHD_3.15.0.0-MacPorts-Release
[INFO] [B200] Detected Device: B210
[INFO] [B200] Loading FPGA image: /opt/local/share/uhd/images/usrp_b210_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Detecting internal GPSDO....
[INFO] [GPS] No GPSDO found
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
```

```
Device: B-Series Device
```

```
Mboard: B210
revision: 4
product: 2
serial: XXXXXXX
FW Version: 8.0
FPGA Version: 16.0
```

```
Time sources: none, internal, external, gpsdo
Clock sources: internal, external, gpsdo
Sensors: ref_locked
```

```
RX DSP: 0
```

```
Freq range: -8.000 to 8.000 MHz
```

```
RX DSP: 1
```

```
Freq range: -8.000 to 8.000 MHz
```

```
RX Dboard: A
```

```
RX Frontend: A
```

```
Name: FE-RX2
Antennas: TX/RX, RX2
Sensors: temp, rssi, lo_locked
Freq range: 50.000 to 6000.000 MHz
Gain range PGA: 0.0 to 76.0 step 1.0 dB
Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
Connection Type: IQ
Uses LO offset: No
```

```
RX Frontend: B
```

```
Name: FE-RX1
Antennas: TX/RX, RX2
Sensors: temp, rssi, lo_locked
Freq range: 50.000 to 6000.000 MHz
Gain range PGA: 0.0 to 76.0 step 1.0 dB
Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
Connection Type: IQ
Uses LO offset: No
```

```
RX Codec: A
```

```
Name: B210 RX dual ADC
Gain Elements: None
```

```
TX DSP: 0
```

```
Freq range: -8.000 to 8.000 MHz
```

```
TX DSP: 1
```

```
Freq range: -8.000 to 8.000 MHz
```

```
TX Dboard: A
```

```
TX Frontend: A
```

```
Name: FE-TX2
Antennas: TX/RX
Sensors: temp, lo_locked
Freq range: 50.000 to 6000.000 MHz
Gain range PGA: 0.0 to 89.8 step 0.2 dB
Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
Connection Type: IQ
Uses LO offset: No
```

```
TX Frontend: B
```

```
Name: FE-TX1
Antennas: TX/RX
Sensors: temp, lo_locked
Freq range: 50.000 to 6000.000 MHz
Gain range PGA: 0.0 to 89.8 step 0.2 dB
Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
Connection Type: IQ
Uses LO offset: No
```



```
TX Codec: A
Name: B210 TX dual DAC
Gain Elements: None
```

For UHD 4.0.0.0 release there is a Clang-specific bug that prevents `uhd_usrp_probe` from functioning as desired; this bug is already fixed and will be part of the next UHD 4 release. If you have more than one B-series USRP attached to this host computer, you will need to specify `--args` that include the type and serial number of the specific USRP of interest; for example with the USRP here, we would use `uhd_usrp_probe --args type=b200,serial=XXXXXXX` but using the actual serial number as returned by `uhd_find_devices`. The following is typical for the information printout for this version of UHD:

```
% uhd_usrp_probe
[INFO] [UHD] Mac OS; Clang version 12.0.0 (clang-1200.0.32.28); Boost_107100; UHD_4.0.0.0-MacPorts-20201218-3b9ced8f
[INFO] [B200] Detected Device: B210
[INFO] [B200] Loading FPGA image: /opt/local/share/uhd/images/usrp_b210_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Detecting internal GPSDO...
[INFO] [GPS] Found an internal GPSDO: GPSTCXO, Firmware Rev 0.929b
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
[ERROR] [UHD] Device discovery error: send_to: No route to host
[ERROR] [USRP2] USRP2 Network discovery error send_to: Host is down
[ERROR] [X300] X300 Network discovery error send_to: Host is down
[ERROR] [OCTOCLOCK] OctoClock network discovery error - send_to: Host is down
Error: RuntimeError: Cannot access! Property types do not match at: /mboards/0/eeprom
```

Note that most UHD-based applications will still work properly even with this specific bug in `uhd_usrp_probe`.

With the USRP information as provided by `uhd_usrp_probe`, we can do some throughput testing via the `benchmark_rate` example. MacPorts installs this example into `/opt/local/share/uhd/examples/`; the default install location as provided by CMake is `/usr/local/lib/uhd/examples/`. Here is an example execution from within the `examples` directory, showing 2x2 at 1e6 Samples/second (often written as "1 MS/s" or "1 MSPs") per channel for an aggregate throughput of 4 MS/s.

```
%. /benchmark_rate --args type=b200 --rx_rate 1e6 --tx_rate 1e6 --channels 0,1
[INFO] [UHD] Mac OS; Clang version 12.0.0 (clang-1200.0.32.28); Boost_107100; UHD_4.0.0.0-MacPorts-20201218-3b9ced8f
[00:00:00.019563] Creating the usrp device with: type=b200...
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Detecting internal GPSDO...
[INFO] [GPS] Found an internal GPSDO: GPSTCXO, Firmware Rev 0.929b
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
Using Device: Single USRP:
  Device: B-Series Device
  Mboard 0: B210
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: FE-RX2
  RX Channel: 1
    RX DSP: 1
    RX Dboard: A
    RX Subdev: FE-RX1
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: FE-TX2
  TX Channel: 1
    TX DSP: 1
    TX Dboard: A
    TX Subdev: FE-TX1
[00:00:02.482435284] Setting device timestamp to 0...
[INFO] [MULTI_USRP] 1) catch time transition at pps edge
[INFO] [MULTI_USRP] 2) set times next pps (synchronously)
[INFO] [B200] Asking for clock rate 32.000000 MHz...
[INFO] [B200] Actually got clock rate 32.000000 MHz.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
[00:00:05.221018875] Testing receive rate 1.000000 Msps on 2 channels
Setting TX spp to 2040
[00:00:05.247918999] Testing transmit rate 1.000000 Msps on 2 channels
[00:00:15.537108088] Benchmark complete.

Benchmark rate summary:
  Num received samples: 20461952
  Num dropped samples: 0
  Num overruns detected: 0
  Num transmitted samples: 20077680
  Num sequence errors (Tx): 0
  Num sequence errors (Rx): 0
  Num underruns detected: 0
  Num late commands: 0
  Num timeouts (Tx): 0
  Num timeouts (Rx): 0
```

Done!

If your UHD-based application results in USB errors such as `ERROR_CODE_TIMEOUT`, underruns (u), overflows (o), sequence errors, or the like, then your USRP might need to be reset as noted above in `uhd_find_devices`. You will also want to find ways to reduce the aggregate TX (if underrun issues) and/or RX (if overflow issues) throughput so that your application consumes and/or produces data in real time for what the USRP expects.

OSX / macOS does not provide tuneable OS, thread, or USB options beyond buffer sizes within UHD. These buffers are generally set via the number of samples per frame, and the number of frames, which are provided as `--args` as `num_recv_frames`, `num_send_frames`, `recv_frame_size`, and `send_frame_size`. Defaults for these variables with the B-series USRPs are as follows: the number of frames is 16; the frame size (in bytes) is 512 for when using USB 2 and 1024 when using USB 3. One can easily and safely increase these to gain more throughput, though larger frames will also increase the latency between USRP antenna to data on the host computer (and vice versa) as more data must be collected before transmission between the host and USRP.

Some errors such as `USBError -4: usb rx8 submit failed: LIBUSB_ERROR_NO_DEVICE` generally require the USRP to be reset before it can be used again, though for some USB errors the USB controller on the USRP will properly reset itself.