GNSS Firehose quick start

Thank you for purchasing GNSS Firehose!

As a quick start, you can use a small ceramic L1 GNSS antenna (not supplied) connected to the RF1 input. The Firehose has an internal bias tee on each RF input supplying 3 V at 50 mA maximum. Any GNSS antenna covering L1 and supporting this power profile will work—check your antenna's data sheet.

Connect the Ethernet cable from the Firehose to a desktop or laptop computer with a gigabit Ethernet (1000baseT) port. You should see the link LEDs light up on both ends and a blinking activity light on the Firehose and most likely on the other endpoint as well.

Waveform capture

To capture the Ethernet packets broadcast by the Firehose to a file, use the following command (a Linux computer is assumed):

```
tcpdump -nn -i eth0 -c 100000 -B 100000 -w test.cap ether proto 0x88b5
```

This will capture about 1.3 seconds of waveform to the file test.cap. All three RF channels will be stored in this file, but in this configuration only the first channel from RF1 will contain any signal—the other two channels from RF ports 2 and 3 will contain only noise. By way of explanation, "-i eth0" selects the Ethernet interface (it may be different on your machine); be sure the Ethernet interface is active with "ifconfig eth0 up" or similar. "-c 100000" sets the packet count (the Firehose emits about 73000 packets per second in its default mode); "ether proto 0x88b5" selects only packets with the given Ethernet protocol, to avoid recording miscellaneous packets that may be emitted by the host. Consult the tcpdump documentation for more details.

Acquisition

Once the test.cap file is available (it should have length 147800024 bytes), issue the following command to acquire any GPS L1 C/A signals in it:

```
<test.cap packet2wav_3ch 1 | acquire-gps-l1.py --doppler-search=-7000,7000,100
   --prn 1-32 /dev/stdin 69984000 -9334875
```

This command consists of two parts with a pipe between them. The first part, `packet2wav_3ch`, outputs a raw stream of I/Q samples from the input packet stream after selecting the given channel (here channel 1). The second part, `acquire-gps-l1.py`, reads samples from the standard input and searches for L1 C/A signals with the given PRNs and doppler bins. The final two arguments are the nominal sample rate and nominal L1 carrier offset, both in Hz.

A typical output:

```
prn   1 doppler -4400.0 metric  1.56 code_offset  927.3
prn   2 doppler -2200.0 metric  3.99 code_offset  647.4
prn   3 doppler   400.0 metric  1.54 code_offset  791.5
prn   4 doppler -1500.0 metric  1.59 code_offset  985.3
prn   5 doppler  -300.0 metric  11.36 code_offset  293.2
prn   6 doppler -3600.0 metric  5.44 code_offset  389.4
prn   7 doppler  4000.0 metric  1.56 code_offset  252.3
prn   8 doppler  1500.0 metric  1.56 code_offset  936.3
prn   9 doppler  6900.0 metric  1.57 code_offset  217.3
prn  10 doppler  4200.0 metric  1.60 code_offset   48.5
prn  11 doppler -3000.0 metric  2.65 code_offset  799.7
prn  12 doppler -4300.0 metric  4.99 code_offset  724.5
prn  13 doppler  2500.0 metric  5.38 code_offset  424.3
prn  14 doppler -2700.0 metric  1.58 code_offset  662.4
prn  15 doppler  3100.0 metric  2.08 code_offset  194.6
prn  16 doppler -1900.0 metric  1.52 code_offset  791.2
prn  17 doppler  6400.0 metric  1.58 code_offset  895.4
prn  18 doppler  4800.0 metric  1.49 code_offset  443.3
prn  19 doppler  -300.0 metric  1.60 code_offset  916.9
prn  20 doppler  6300.0 metric  1.56 code_offset  120.4
prn  21 doppler -2100.0 metric  1.56 code_offset  227.3
prn  22 doppler  5100.0 metric  1.63 code_offset  571.2
prn  23 doppler -2500.0 metric  1.64 code_offset  217.3
prn  24 doppler  4000.0 metric  1.55 code_offset  419.3
prn  25 doppler -3500.0 metric  4.63 code_offset  689.1
prn  26 doppler   900.0 metric  1.50 code_offset  359.1
prn  27 doppler -6800.0 metric  1.62 code_offset  346.4
prn  28 doppler -4600.0 metric  1.54 code_offset  719.3
prn  29 doppler -2200.0 metric  1.55 code_offset  458.3
prn  30 doppler  5200.0 metric  1.57 code_offset 1003.3
prn  31 doppler -2100.0 metric  1.55 code_offset  107.4
prn  32 doppler   800.0 metric  1.50 code_offset  404.1
```

There are signals on GPS PRNs 2, 5, 6, 11, 12, 13, 15, and 25. The tracking script `track-gps-l1.py` can now be run—given an initial code phase and doppler, it will track the carrier and output post-correlation baseband samples. Similarly, acquisition and tracking can be performed for the other GNSS systems (GLONASS, Galileo, and BeiDou) using the supplied programs in the GNSS-DSP-tools software suite.

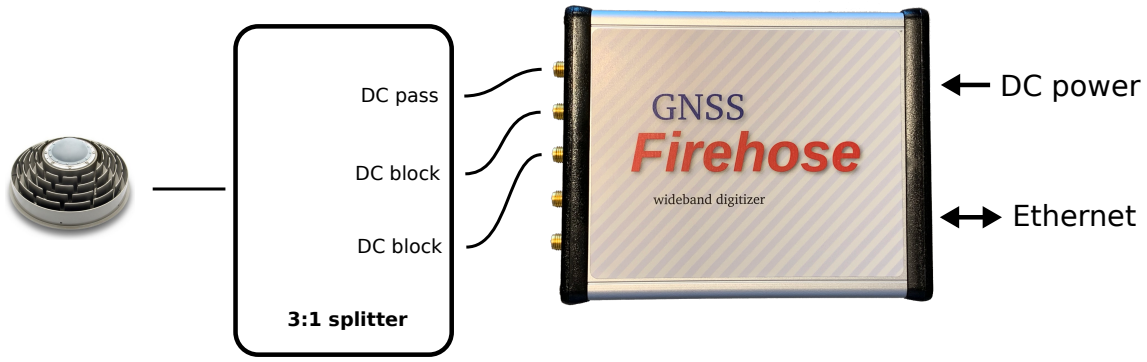
Next steps

At this point, feel free to feed the waveforms to your favorite GNSS software receiver. The command

```
<test.cap packet2wav_3ch 1 >test-L1.iq
```

will create a suitable raw waveform file containing alternating 8-bit I and Q samples at the 69.984 Ms/s rate.

Finally, you can use all three of the Firehose's RF inputs in an arrangement like this:



By default, the Firehose configures the tuning of RF1 to L1, RF2 to L2, and RF3 to L5/E5ab. Each channel can be independently tuned to any frequency between 0.95 GHz and 2.15 GHz. For example, multiple channels tuned to the same frequency may be desired, or a channel tuned to E6 may be desired.

When using multiple channels, waveform samples may be obtained as follows:

```
<test2.cap packet2wav_3ch 1 >test2-L1.iq
<test2.cap packet2wav_3ch 2 >test2-L2.iq
<test2.cap packet2wav_3ch 3 >test2-L5.iq
```

The samples are synchronized, that is, each sample across the various channels was sampled at the same time (within a few ns).

Documentation

The Firehose hardware (schematic and PCB layout), FPGA, and firmware design files and source code are in this github repository:

`https://github.com/pmonta/GNSS_Firehose`

Also included are supporting software tools such as `packet2wav_3ch` for extracting waveform samples from the Ethernet packet captures.

Software-receiver tools for acquisition and tracking can be found here:

`https://github.com/pmonta/GNSS-DSP-tools`

This contains the `acquire-gps-l1.py` program mentioned above as well as acquisition and tracking for the other GNSS systems and bands/signals.

Finally, some discussion of results, experiments, and a GNSS waveform archive can be found at my blog:

`http://pmonta.com/blog/`