When time series output is selected the output data consist either of (3xBxN) or (2xBxN) words, depending on the output format, where B is the number of bins in the current range mask, and N is the number of pulses per ray. Data samples for each bin of pulse #1 are output first, followed by those for each bin of pulse #2, etc. up to pulse #N. In other words, the data are output in the same time-order that they were acquired.

In the floating point format, three words are used for each bin:

```
 15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|      Exponent      | S |              Mantissa                 |   (I)
|____________________|___|_______________________________________|
 15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|      Exponent      | S |              Mantissa                 |   (Q)
|____________________|___|_______________________________________|
 15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   0   0   0 |          Log of Power in Sample                |  (LOG)
|_______________|________________________________________________|
```

To convert these "legacy format" floating I and Q samples to voltages: First create a 12-bit signed integer in which bits zero through nine are copied from the Mantissa field, and bits ten and eleven are either 01 or 10 depending on whether S is 0 or 1. Then, multiply this number by 2**(exponent–40), where the exponent field is interpreted as an unsigned 5-bit integer. Finally multiply by the maximum voltage. The resulting value has 12-bits of precision and a dynamic range of approximately 190dB. The large dynamic range is necessary to cover the full range of data. In summary:

$$Voltage \;=\; V_{MAX} \times (Sign, Mantissa) \times 2^{\,[Exponent - 40]}$$

Note that the resulting voltage span is actually $\pm 4 \times V_{MAX}$. The extra factor of four is built into the format so that transient excursions above the full scale input voltage can still be encoded properly. These may arise for time series data that have been processed by an IIR clutter filter.

An improved "High-SNR" packed floating format is also available that offers nearly the same dynamic range but provides a 6dB improvement in SNR, i.e., a commensurate improvement in sub-clutter visibility of –78dB versus –72dB.

```
 15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|    Exponent   | S |                Mantissa                   |  High-SNR
|_______________|___|_____________________________________________|
```

The High-SNR packed format is similar to the legacy packed format except that it uses one extra mantissa bit and one fewer exponent bit. The dynamic range lost in the exponent is recovered through a formatting trick known as "soft underflow", i.e., the mantissa is allowed to become unnormalized when the exponent is zero.

To decode this format when the exponent is non-zero, first create a 13-bit signed integer in which bits zero through ten are copied from the Mantissa field, and bits eleven and twelve are either 01 or 10 depending on whether S is 0 or 1. Then, multiply this by 2**(exponent–25), where the exponent field is interpreted as an unsigned 4-bit integer.

To decode the High-SNR format when the exponent is zero simply interpret the mantissa as a 12-bit signed integer and multiply by 2**-24.

A complete analysis of the noise properties of the floating point codes would be fairly tricky. For the High-SNR format, the 12-bit mantissa with hidden normalization bit will vary from 2048 to 4095. The SNR will therefore vary from 66dB to 72dB and we can assign a mean value of 69dB. Another 9dB of useful range is contained within the code as follows:

- In a floating point encoding format, the notion of fixed additive quantization noise is not really correct. For a signal having a given power, the additive noise within each instantaneous sample will scale down according to the magnitude of that sample. The ensemble of noise terms thus contributes an RMS power that is smaller than the Peak–to–Noise ratio would imply. In the case of a sinusoidal input, this gives a 3dB boost in effective SNR.
- The format, of course, also represents negative amplitudes with the same relative precision as positive values. In a fixed–point format this would add 6dB (one more bit) to the overall dynamic range and large–signal SNR. In the floating format we really only gain 3dB (half a bit) because the RMS noises add independently on the positive and negative excursions.
- The packed format is used to encode timeseries (I,Q) pairs, and it's the SNR properties of these pairs that we're really concerned about. To a first approximation, having a pair of values roughly doubles the information content and adds another 3dB to the SNR.

The last of the three timeseries output words, the "Log of Power in Sample", is provided mainly for backwards compatibility. It can be calculated from the I and Q numbers. To convert to dBm it requires a slope and offset as follows:

$$dBm \quad = \quad P_{MAX} + Slope \times [Value - 3584]$$

Where:

P_{MAX} = +4.5dBm for 12-bit IFD, +6.0dBm for 14-bit IFD

V_{MAX} = 0.5309 Volts for 12-bit IFD, 0.6310 Volts for 14-bit IFD

Slope = "Log Power Slope" word 3 of SOPRM command. 0.03 recommended.

For backwards compatibility the RVP8 produces a 8-bit fixed point time series format. Because of the limited dynamic range available, this will only show strong signals, and is not recommended for use. The I, Q, and Log power triplets are packed into two 16-bit output words as follows:

```
 High Byte    Low Byte
|           |           |
| Q Sample  | I Sample  |   First Word
|___________|___________|
 _______________________
|           |           |
|   Zero    | Log Power |   Second Word
|___________|___________|
```