RadarStudio carries with it a full Matlab runtime engine which can be used to perform different kinds of analysis on the adc data with a fixed set of APIs.

What does 0dBFs mean?

The primary purpose of most of the APIs is to measure the power of a sinusoid (or the power in a certain bandwidth). This power measurement is given in dBFs.

To make sense of this number, note that the tone power of a 'full-scale' real sinusoid when processed with 'Gain Compensation' (described later) and an integration bandwidth of '0'Hz, and with sufficient oversampling gives an output of 0 dBFs. If a 'full-scale' complex sinusoid is processed with the same API and the same option, its tone power is 3 dBFs. The processing necessary for making a dBFs measurement is shown in the next section.

Note: There are different definitions for 0dBFs, and our choice (i.e. a 'full-scale' real sinusoid being 0dBFs) is arbitrary.

Raw ADC data to dBFs.

Given N complex ADC samples, with each sample having b_{adc} bits (i.e. b_{adc} bits for I and b_{adc} bits for Q), we apply a window (the elements of the window are given by $w_0, w_1,\ w_2\ \ldots\ w_{N-1}$), and then perform an FFT (note that Matlab's fft is un-normalized).

The output of the FFT is then corrected so that a 'full-scale' real sinusoid after FFT processing becomes a tone of strength 0dBFs. In other words,

$$P_{out(\,dBFs)} = P_{out\,(fft)} - correction_{factor}$$

$$correction_{factor} = 20\log_{10}\frac{2^{b_{adc}-1}}{1} + 20\log_{10}\sum_{i=0}^{N-1} w_i - 20\log_{10}\sqrt{2}$$

The first term normalizes the raw adc code-word to a number between $\pm\ 1$.

The second term compensates for windowing loss and fft gain. Normalizing by the 'sum of window coefficients' is referred to as 'Gain normalization', normalizing by the 'energy of the window coefficients' is called 'Energy normalization'. If no windowing was used (i.e. $w_i\ = 1\ \forall\, i \in [0, 1, 2, \ldots\ ,\ N-1]$), then the second term becomes $20\log_{10} N$.

The final term is a correction factor that allows a full-scale real sine-wave to be 0 dBFs.

When $b_{adc}\ = 16$, and no windowing is used, then the above equation becomes

$$correction_{factor} = 20\log_{10} 2^{15} + 20\log_{10} N - 20\log_{10}\sqrt{2}$$

Converting dBFs to dBm (at ADC input).

A power measurement in dBFs may need to be translated to a dBm (or dBV) measurement before being used. The following table can be used to convert a dBFs measurement (from radarstudio) to a dBm measurement.

Codes (ADC output)	Signal type (ADC Input)	RMS (V)	dBVrms	dBm	Radarstudio reports using API
+/-2^15 (full scale) real sinusoid	+/- 1V Real sinusoid	$1/\sqrt{2}$	-3	10	0dBFs
+/-2^15 (full scale) complex sinusoid	+/- 1V Complex sinusoid	1	0	13	3dBFs

In other words,

$$P_{out(\,dBm)} = P_{out(\,dBFs)} + 10$$