**Instructions for using single\_tone.py**

**Update Log**

**Updated: 03/12/2018**

* **Created file**
* **Added 4 functions**
* **Take\_noise\_set, power\_sweep, plot\_iq\_dict, plot\_power\_dict**

**Updated: 04/03/2018**

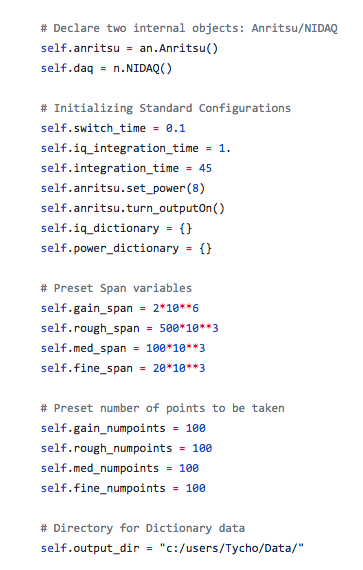
* **Added 1 function – Cleaned up file**
* **Fit\_noise\_set**

**INTRODUCTION TO SINGLE\_TONE.PY**

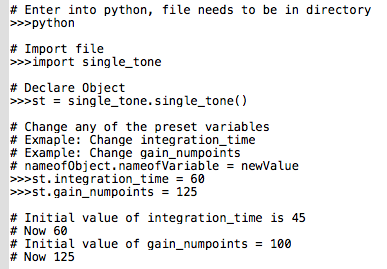
Single\_tone.py utilizes two other classes: anritsu.py and NIDAQ.py. The single\_tone.py class has two main functions, take\_noise\_set and power\_sweep to be used to take data. There are three other functions, plot\_iq\_dict, plot\_power\_dict, and fit\_noise\_set, used to plot the two data sets after it has been taken and try to fit the noise data and guess.

After the data has been taken, the function will automatically save the data into a default file location with a default name. User can change the filename to custom preference (instructions below).

**Changing Preset Variables:**



Single tone has default variables preset during the initialization of the code. To change to user preference, change the numbers directly from file or through command prompt.   
**\*\*\*\*\* To change variables in command prompt setting \*\*\*\*\***

****

**Take Noise Function**

def take\_noise\_set(center\_freq, chan3 = False, take\_noise = True, filename = “”)

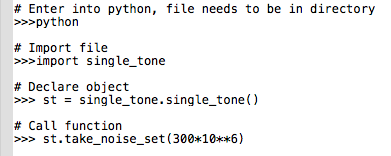
This function requires a center frequency to be passed. The other three parameters, chan3, take\_noise, and filename, are default and can be changed if a 3rd channel wants to be streamed, take\_noise is not needed, or a custom filename is wanted. To run chan3, change False to True. To not take noise, change take\_noise to False. To create custom filename, change filename “” to desired file name with quotations around it

**After the function is finished running, it will automatically save the data to the initial directory created at the preset declarations. The default file name is currentDate\_currentTime\_noiseData.**

**\*\*\*\*\* To take noise in a python file \*\*\*\*\***



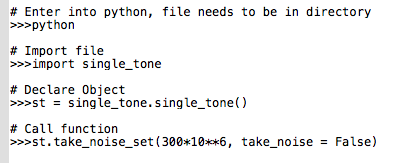
**\*\*\*\*\* To take noise in a command prompt setting, below is an example \*\*\*\*\***



**\*\*\*\*\* To do an IQ-sweep without taking noise in a python file, set take\_noise = False \*\*\*\*\***



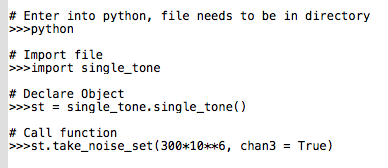
**\*\*\*\*\* To do an IQ-sweep without taking noise in command prompt \*\*\*\*\***



**\*\*\*\*\* To do a take noise with a 3rd channel in a python file, set chan3 = True \*\*\*\*\***



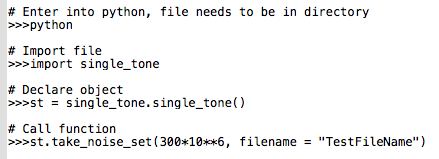
**\*\*\*\*\* To do a take noise with a 3rd channel in command prompt \*\*\*\*\***



**\*\*\*\*\* To create custom filename in python file \*\*\*\*\***



**\*\*\*\*\* To create custom filename in command prompt \*\*\*\*\***



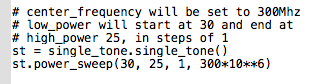
**Power Sweep Function**

def power\_sweep(low\_power, high\_power, step, center\_freq, filename = “”)

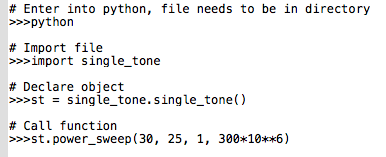
This function requires 5 parameters a **low power** to being moving in incremental **steps** to the **high power** all at a **center frequency** that is being passed. User can also set a custom file name.

**After the function is finished running, it will automatically save the data to the initial directory created at the preset declarations. The default file name is currentDate\_currentTime\_noiseData.**

**\*\*\*\*\* To take noise in a python file \*\*\*\*\***



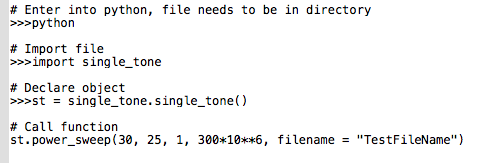
**\*\*\*\*\* To take noise in a command prompt setting, below is an example. \*\*\*\*\***



**\*\*\*\*\* To create a custom file name in python file \*\*\*\*\***



**\*\*\*\*\* TO create a custom file name in command prompt \*\*\*\*\***

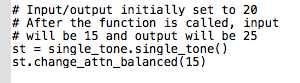


**Change Attenuators Balanced:**

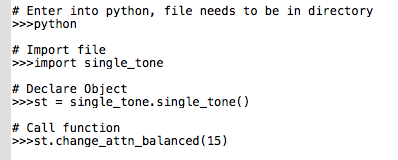
def change\_attn\_balanced(input)

This function requires 1 parameter, which is indicated as input. The function will automatically balance the input/output powers. For example, if input/output were both initially set to be 20. The change\_attn\_balanced function till change both input/put accordingly. If **input** was to be set to a new power, say 15, **output** will automatically be changed to 25. If output is not connected, the function will only change input.

**\*\*\*\*\* To change input in a python file \*\*\*\*\***

****

**\*\*\*\*\* To change input in command prompt \*\*\*\*\***

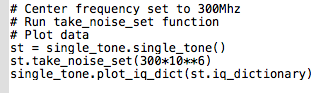
****

**Plotting IQ-Noise Data:**

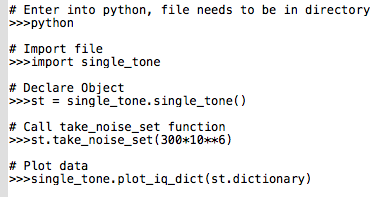
def plot\_iq\_dict(dictionary)

This function requires 1 parameter that is a dictionary to be plotted. The function will display 2 figures. The first will provide plots of I versus Q. The second will provide frequency versus magnitude.

**\*\*\*\*\* To plot IQ in python file \*\*\*\*\***

****

**\*\*\*\*\* To plot IQ in command prompt \*\*\*\*\***

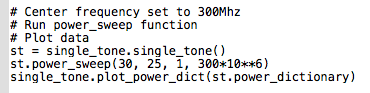
****

**Plotting Power Data:**

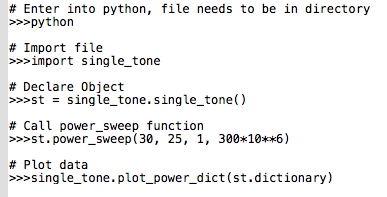
def plot\_power\_dict(dictionary)

This function requires 1 parameter that is a dictionary to be plotted. The function will display 1 figure. The figure will be an I versus Q graph at each different power level.

**\*\*\*\*\* To plot power in python file \*\*\*\*\***

****

**\*\*\*\*\* To plot power in command prompt \*\*\*\*\***

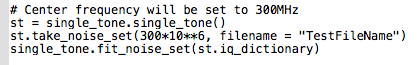
****

**Fit Noise Set:**

def fit\_noise\_set(dictionary)

This function requires 1 parameter which is a noise dictionary. The function will attempt to fit the noise set and guess.

**\*\*\*\*\* To fit noise set in python file \*\*\*\*\***



**\*\*\*\*\* To fit noise set in command prompt \*\*\*\*\***

