Power Amplifier Test Controller: User Manual

Ryan King, Taha Rangwala, Qihang Shan

Made in Association with Virginia Tech and Raytheon/Collin's Aerospace

ECE 4806 Spring 2021

Date of Last Revision: April 14th, 2021

Table of Contents

Introduction

The software we have created is the Universal Power Amplifier Test Controller v2.0. A Test Controller is a program that automates radio frequency tests including the Mixer Spur Test, P1dB Test, and Pin vs. Pout Test. The rest of this manual explains everything from the installation process to how to run a test.

Installation

The program was created on the Raspberry Pi 4 Model B 4GB, and it was tested on the Raspbian Buster Operating System. The code was written using Python version 3.8.8.

The following packages are required for the test controller:

- PyVISA: https://pyvisa.readthedocs.io/en/latest/ [1]
- PySimpleGUI: https://pysimplegui.readthedocs.io/en/latest/cookbook/ [2]
- MatPlotLib: https://matplotlib.org/ [3]
- Plotly: https://plotly.com/python/table/ [4]
- Numpy: https://numpy.org/ [5]

To install these packages, open up the Terminal on the Raspberry Pi and run the following commands:

1. $ sudo apt-get update
2. $ sudo apt-get install pyvisa
3. $ sudo pip3 install pyvisa-py
4. $ sudo pip3 install pysimplegui
5. $ sudo pip3 install matplotlib
6. $ sudo pip3 install plotly
7. $ sudo pip3 install numpy

The source code is available at https://github.com/TahaRangwala/CollinsTestController-v2.0. In the Raspberry Pi terminal, run the following command to download the source code:

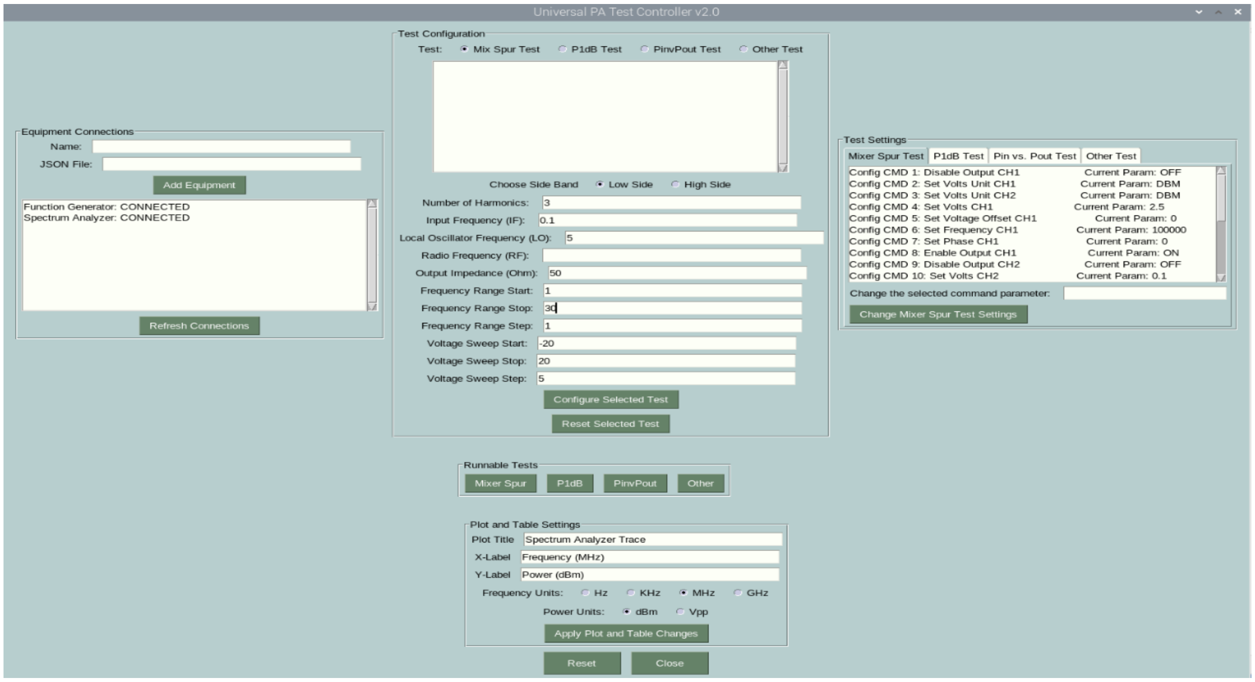
- $ git clone https://github.com/TahaRangwala/CollinsTestController-v2.0

To run the program on the Raspberry Pi, there are two options:

1. Navigate to the folder where the main.py file is and run the following command in the terminal:
   a. $ python main.py
2. Open the main.py file in the program called “Thonny” and press the run button located at the top (the play button)

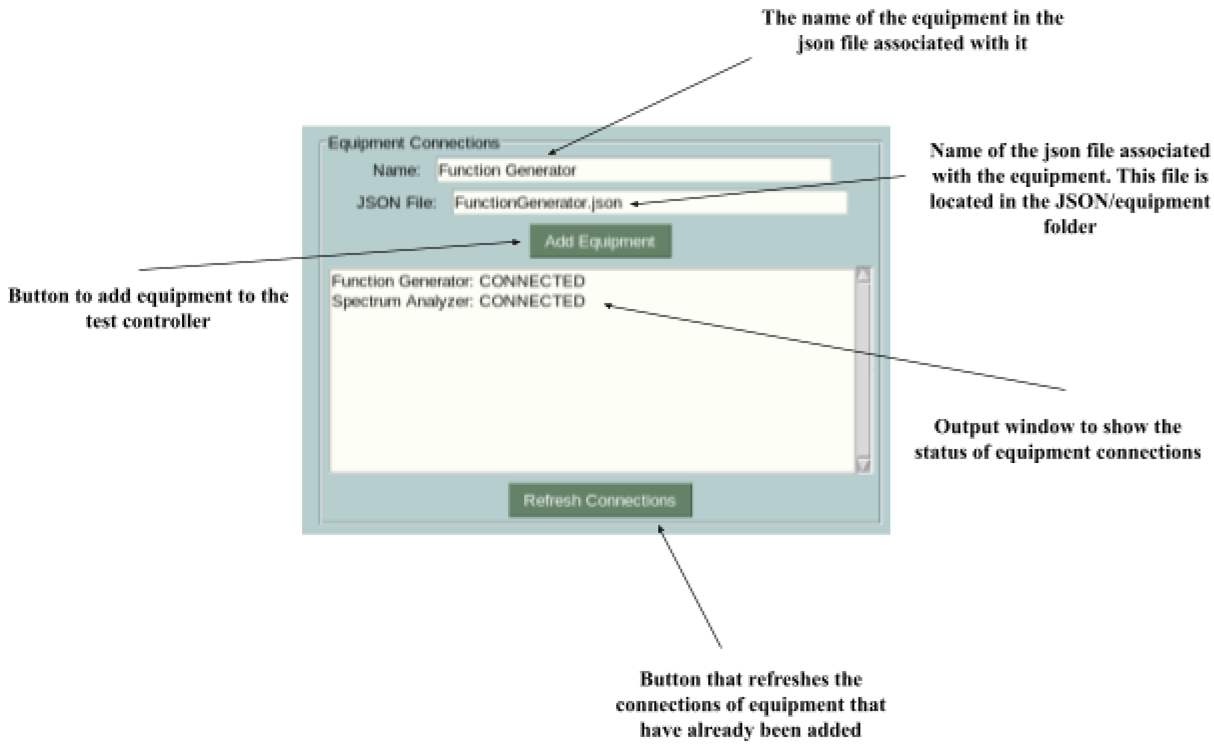
GUI Layout

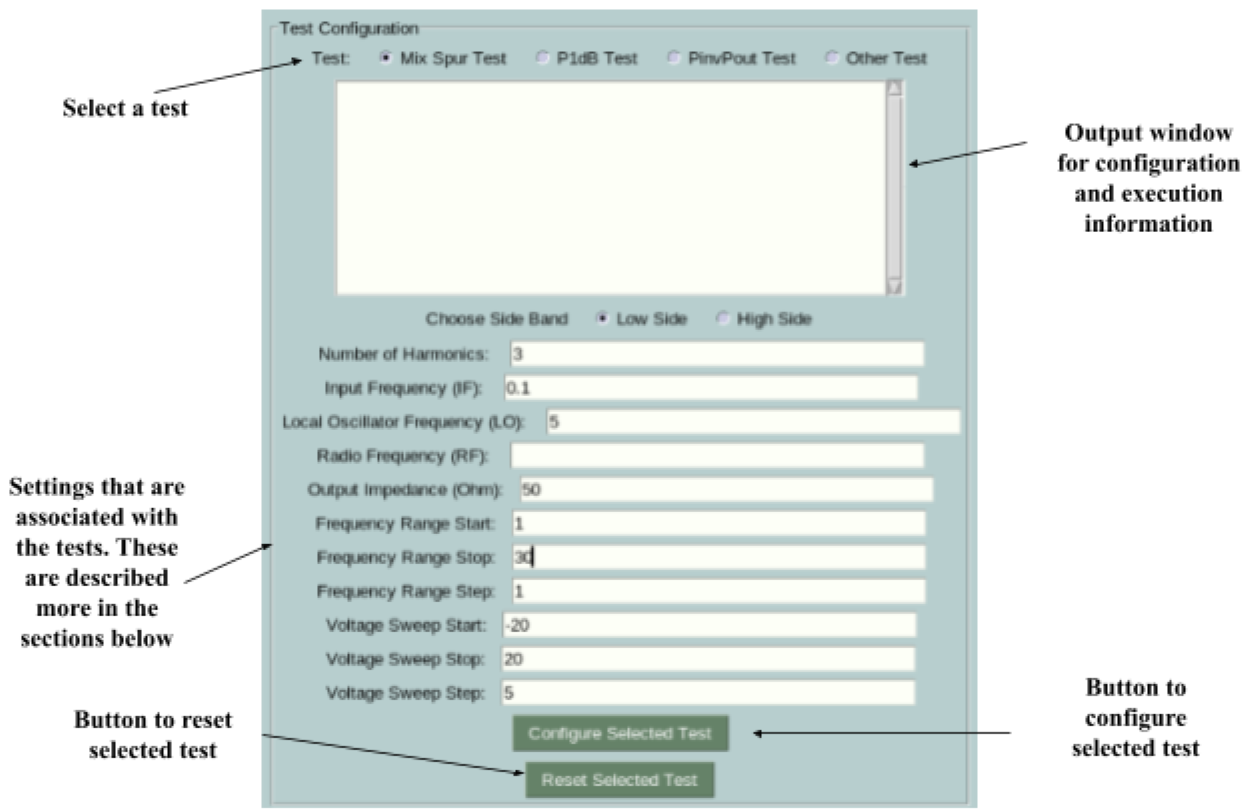
The image below shows the entire Graphical User Interface (GUI) for the test controller: [2]

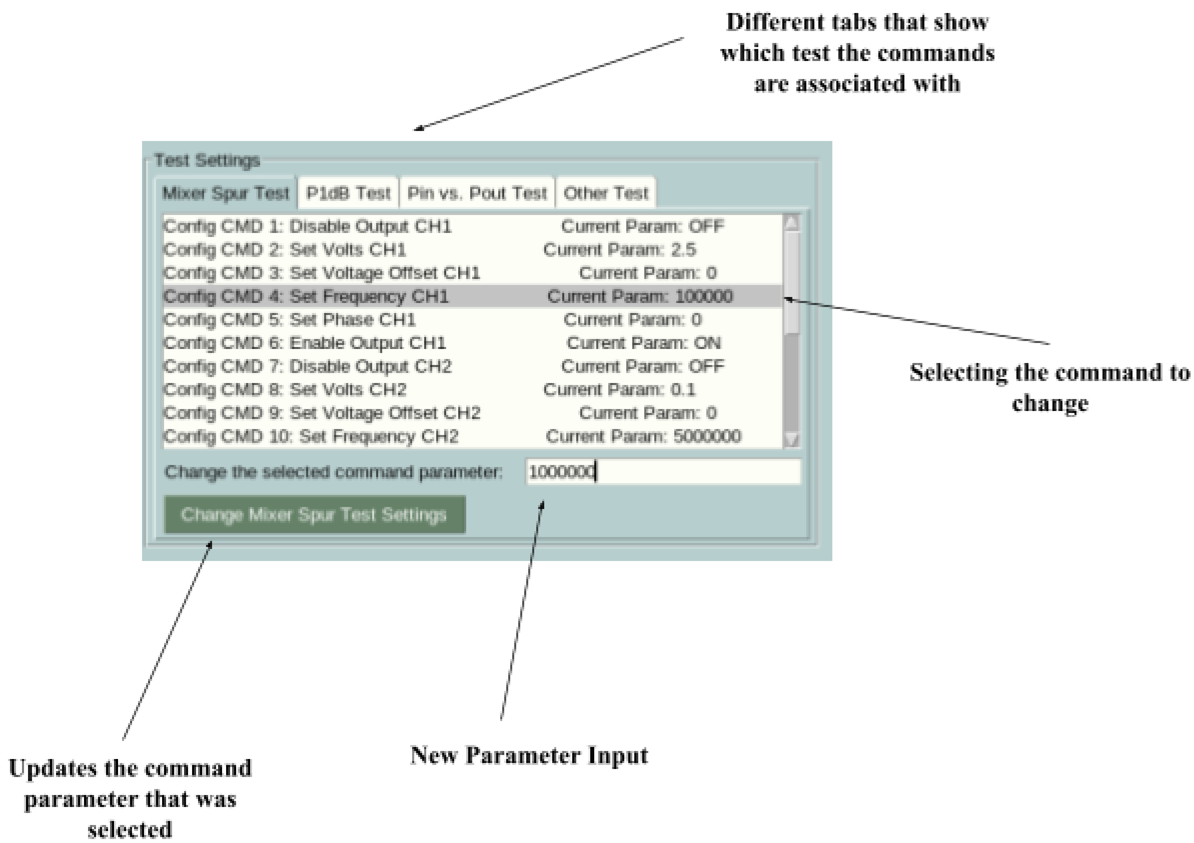


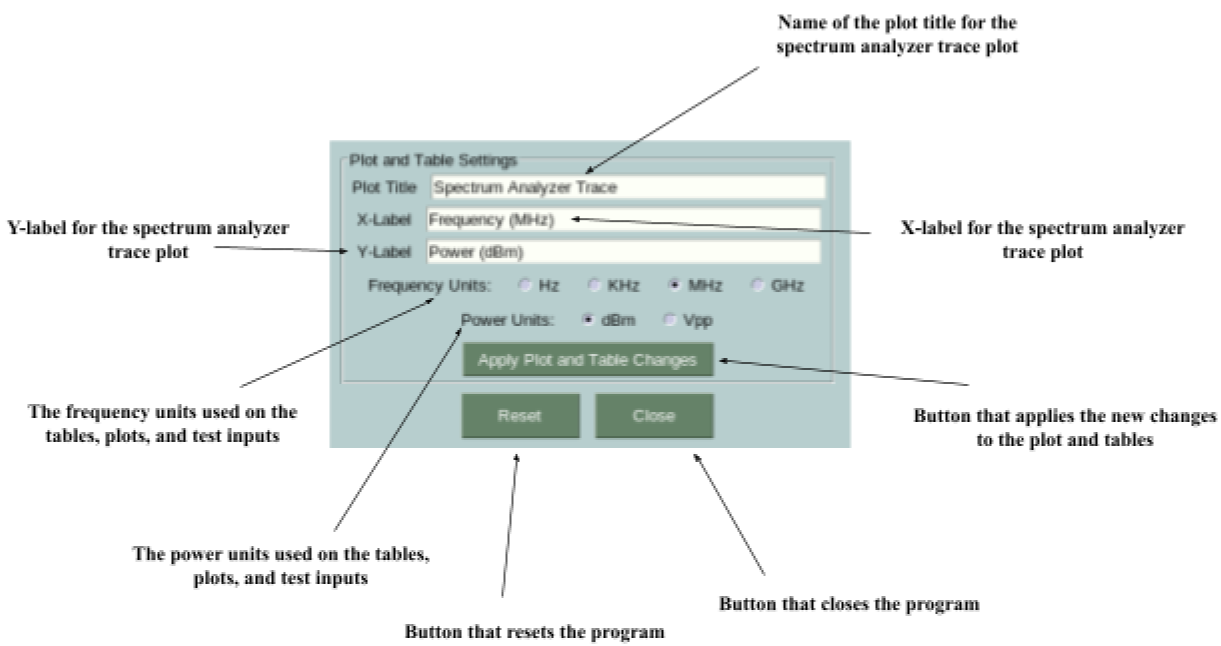
NOTE: If the GUI is too big or small for your monitor, you may need to change your screen resolution. We recommend you use the following settings: 1920x1080 60 Hz 16:9.

Pages 5 and 6 below show what each button is used for:


The name of the equipment in the json file associated with it
Equipment Connections
Name: Function Generator
JSON File: FunctionGenerator.json
Name of the json file associated with the equipment. This file is located in the JSON/equipment folder
Add Equipment
Button to add equipment to the test controller
Function Generator: CONNECTED
Spectrum Analyzer: CONNECTED
Output window to show the status of equipment connections
Refresh Connections
Button that refreshes the connections of equipment that have already been added


Test Configuration
Select a test
Test: Mix Spur Test P1dB Test PinvPout Test Other Test
Output window for configuration and execution information
Choose Side Band Low Side High Side
Number of Harmonics: 3
Input Frequency (IF): 0.1
Local Oscillator Frequency (LO): 5
Radio Frequency (RF):
Output Impedance (Ohm): 50
Frequency Range Start: 1
Frequency Range Stop: 30
Frequency Range Step: 1
Voltage Sweep Start: -20
Voltage Sweep Stop: 20
Voltage Sweep Step: 5
Settings that are associated with the tests. These are described more in the sections below
Configure Selected Test
Button to configure selected test
Reset Selected Test
Button to reset selected test


Different tabs that show which test the commands are associated with
Test Settings
Mixer Spur Test
P1dB Test
Pin vs. Pout Test
Other Test
Config CMD 1: Disable Output CH1 Current Param: OFF
Config CMD 2: Set Volts CH1 Current Param: 2.5
Config CMD 3: Set Voltage Offset CH1 Current Param: 0
Config CMD 4: Set Frequency CH1 Current Param: 100000
Config CMD 5: Set Phase CH1 Current Param: 0
Config CMD 6: Enable Output CH1 Current Param: ON
Config CMD 7: Disable Output CH2 Current Param: OFF
Config CMD 8: Set Volts CH2 Current Param: 0.1
Config CMD 9: Set Voltage Offset CH2 Current Param: 0
Config CMD 10: Set Frequency CH2 Current Param: 5000000
Change the selected command parameter:
1000000
Change Mixer Spur Test Settings
Selecting the command to change
Updates the command parameter that was selected
New Parameter Input


Name of the plot title for the spectrum analyzer trace plot
Plot and Table Settings
Plot Title
Spectrum Analyzer Trace
X-Label
Frequency (MHz)
Y-Label
Power (dBm)
Frequency Units:
Hz
KHz
MHz
GHz
Power Units:
dBm
Vpp
Apply Plot and Table Changes
Reset
Close
Y-label for the spectrum analyzer trace plot
X-label for the spectrum analyzer trace plot
The frequency units used on the tables, plots, and test inputs
Button that applies the new changes to the plot and tables
The power units used on the tables, plots, and test inputs
Button that closes the program
Button that resets the program

Connecting Test Equipment

A JSON file is required for each testing equipment that is added to the program. There are two ways to add test equipment:

1. Before running the program, testing equipment can automatically be added by modifying the allEquipment.txt file that is located in the JSON/equipment folder.

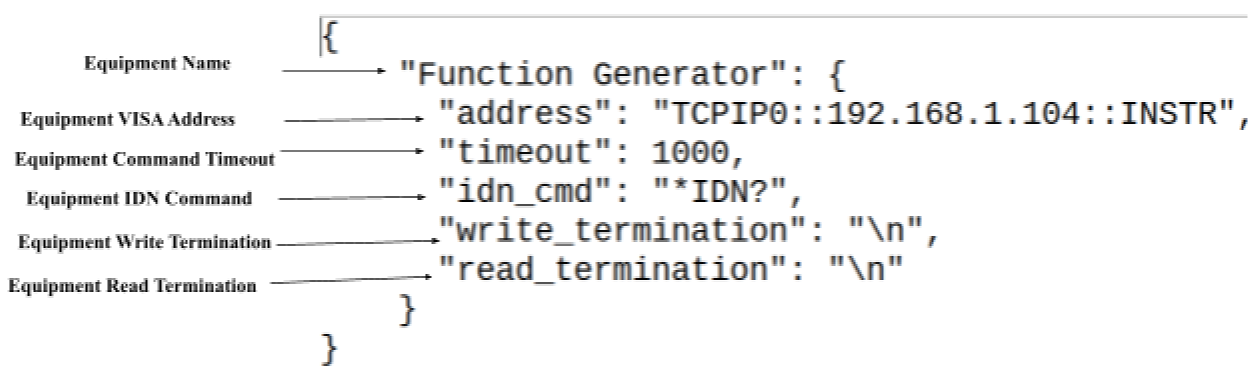
```
Function Generator, FunctionGenerator.json
Spectrum Analyzer, SpectrumAnalyzer.json
```

In the allEquipment.txt file, each line represents a single piece of testing equipment. The name of the testing equipment and the JSON file it is associated with are separated by commas, and this is shown above.

2. The second way to add testing equipment is to add tests via the Equipment Connections section in the GUI. Navigate to the GUI layout section to learn how to use this feature.

NOTE: During testing, we connected testing equipment to the Raspberry Pi via Ethernet Cables and an Ethernet Switch. Additionally the Function Generator we used was the Rigol DG4162 Waveform Generators, and the Spectrum Analyzer we used was the Rigol DSA1030A Performance Spectrum Analyzer. Different testing equipment may have different SCPI command syntax.

<u>Format of Equipment JSON Files</u>



A template equipment JSON file is also located in the JSON/equipment folder to guide users to adding their own equipment. Additionally, we have included the FunctionGenerator.json and SpectrumAnalyzer.json files we have used during our testing of the program.
<u>(IP address is fetched from the testing equipment)</u>

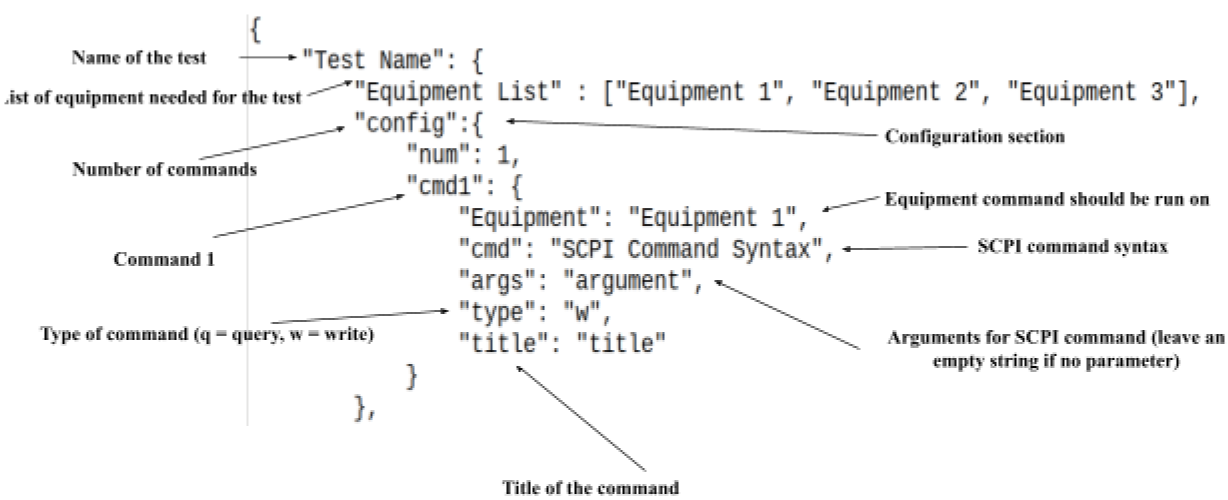
Adding Tests to the Program

A JSON file is required for each test. Before running the program, tests must be added a certain way. In the JSON/tests folder, there is a file called allTests.txt. This file contains the test name, the JSON file it is associated with, and what type of test it is.

```
MixerSpur, MixerSpur.json, MixerSpurTest
P1dB, P1dB.json, P1dBTest
PinVPout, PinVPout.json, PinvPoutTest
OtherTest, OtherTest.json, OtherTest
```

Each line in the allTests.txt file shown above corresponds to a single test. The name of the test, the JSON file, and the test it is associated with are separated by commas. The default tests included with this program are the Mixer Spur Test, the P1dB Test, and the Pin vs. Pout Test.

<u>Format of Test JSON Files</u>



```
"run":{
    "num": 1,
    "cmd1": {
        "Equipment": "Equipment 1",
        "cmd": "SCPI Command Syntax",
        "args": "argument",
        "type": "q",
        "title": "title"
    }
},
```

The run section shown above has the same format as the configuration section.

```
"reset":{
    "num": 1,
    "cmd1": {
        "Equipment": "Equipment 1",
        "cmd": "SCPI Command Syntax",
        "args": "argument",
        "type": "w",
        "title": "title"
    }
}
```

The reset section shown above has the same format as the configuration section. Example test JSON files have been included in the JSON/tests folder as well.

Default Tests

<u>Pin vs. Pout Test</u>

Introduction

The Pin vs. Pout test is one of the more simpler tests for RF devices. It allows the user to see power losses for various input frequencies and input power. Knowing these factors for RF devices helps determine the best device for the mentioned inputs.

Input

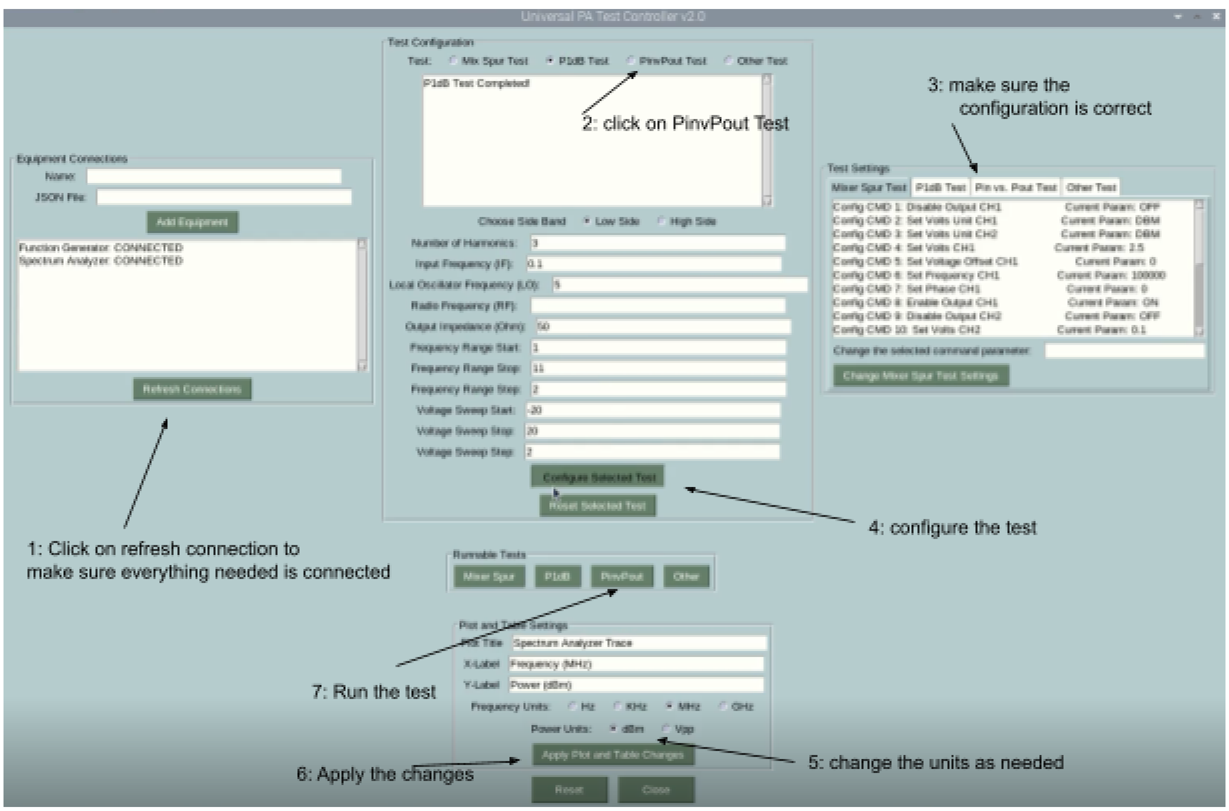
The Pin vs. Pout test requires the user to edit two (2) text files: “PowerIn.txt” and “PowerLoss.txt.” The format of “PowerIn.txt” is:

Frequency (default; MHz), Input Power (dBm)

The format of “PowerLoss.txt” is:

Power Loss (dBm)

NOTE: By default, these files are located in the JSON/tests/PinvPoutSettings folder.

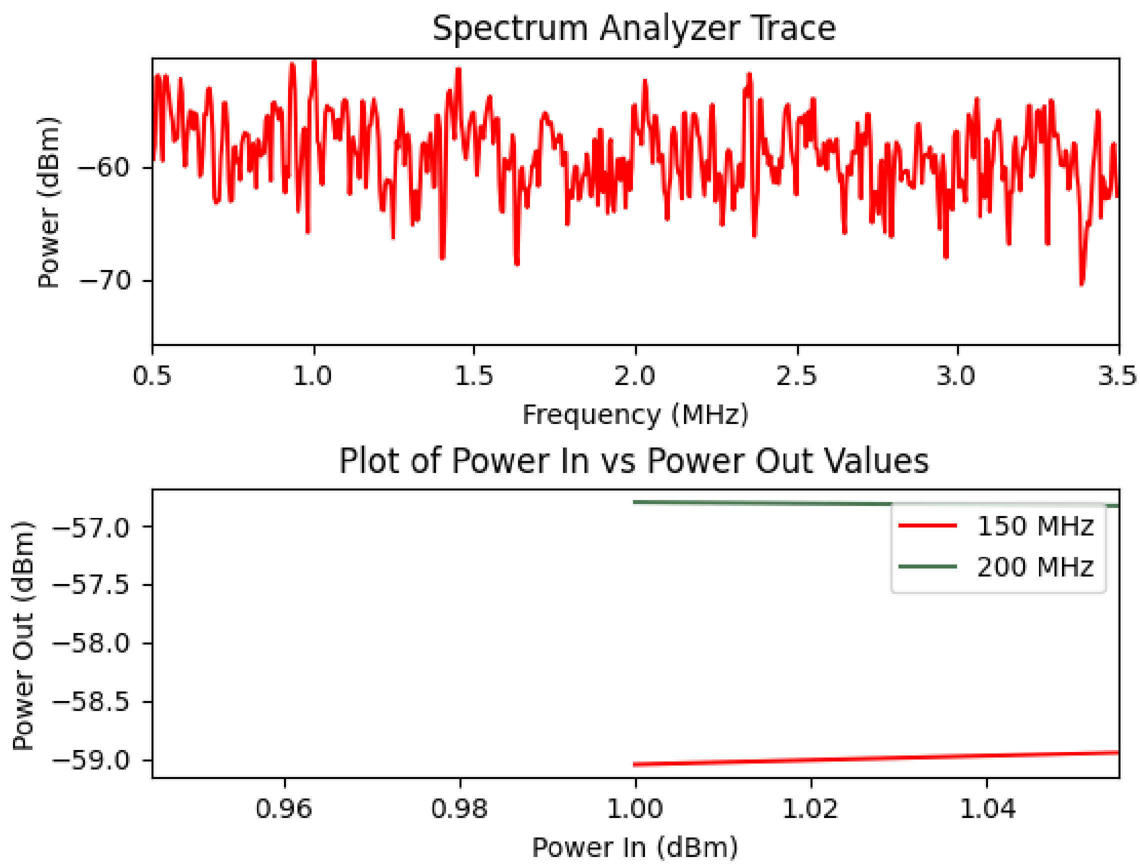


The “PowerIn.txt” file simply determines how much power the user wants for specific frequencies. The “PowerLoss.txt” file is used to model for things like cable loss, attenuation loss, etc. Note that the items in “PowerLoss.txt” correspond directly to the items in “PowerIn.txt.”

Once these text files are configured, the user may run the Pin vs. Pout test.

Output

During the test, the program will display data from the spectrum analyzer (updates every few seconds) and plot the values of output power. The plot of the output power has separate line graphs for different frequencies. An example of this display is shown below:



At any time during the test, the user can abort the test by closing out of the window of the plots. You may have to click the close button on the plot window multiple times for a test to be aborted. Once the test is finished, the program will display a table detailing the important values of the test. An example is shown below:

Peak Frequency (MHz)	Power In (dBm)	Power Measured (dBm)	Power Loss	Peak Amplitude (dBm)	Pin-Pout
150	1	-59.048	1	-59.048	59.048
150	2	-59.216	2	-57.216	59.216
150	3	-61.045	1	-60.045	63.045
150	4	-56.552	2	-54.552	58.552
150	5	-57.962	1	-56.962	61.962
150	6	-55.853	2	-53.853	59.853
150	7	-64.913	1	-63.913	70.913
150	8	-58.951	2	-56.951	64.951
200	1	-57.799	1	-56.799	57.799
200	2	-59.347	2	-57.347	59.347
200	3	-67.813	1	-66.813	69.813
200	4	-59.378	2	-57.378	61.378
200	5	-59.48	1	-58.48	63.48
200	6	-57.799	2	-55.799	61.799
200	7	-53.78	1	-52.78	59.78
200	8	-62.595	2	-60.595	68.595

The list of values are described as such:

Peak Frequency (MHz): The input frequency from PowerIn.txt file; the program evaluates power outputs from this frequency on the spectrum analyzer.

Power In (dBm): Amount of input power from PowerIn.txt file.

Power Measured (dBm): The direct power read from the spectrum analyzer

Power Loss: amount of loss the user expects from various sources (e.g. cable loss, attenuation, etc.); from PowerLoss.txt file.

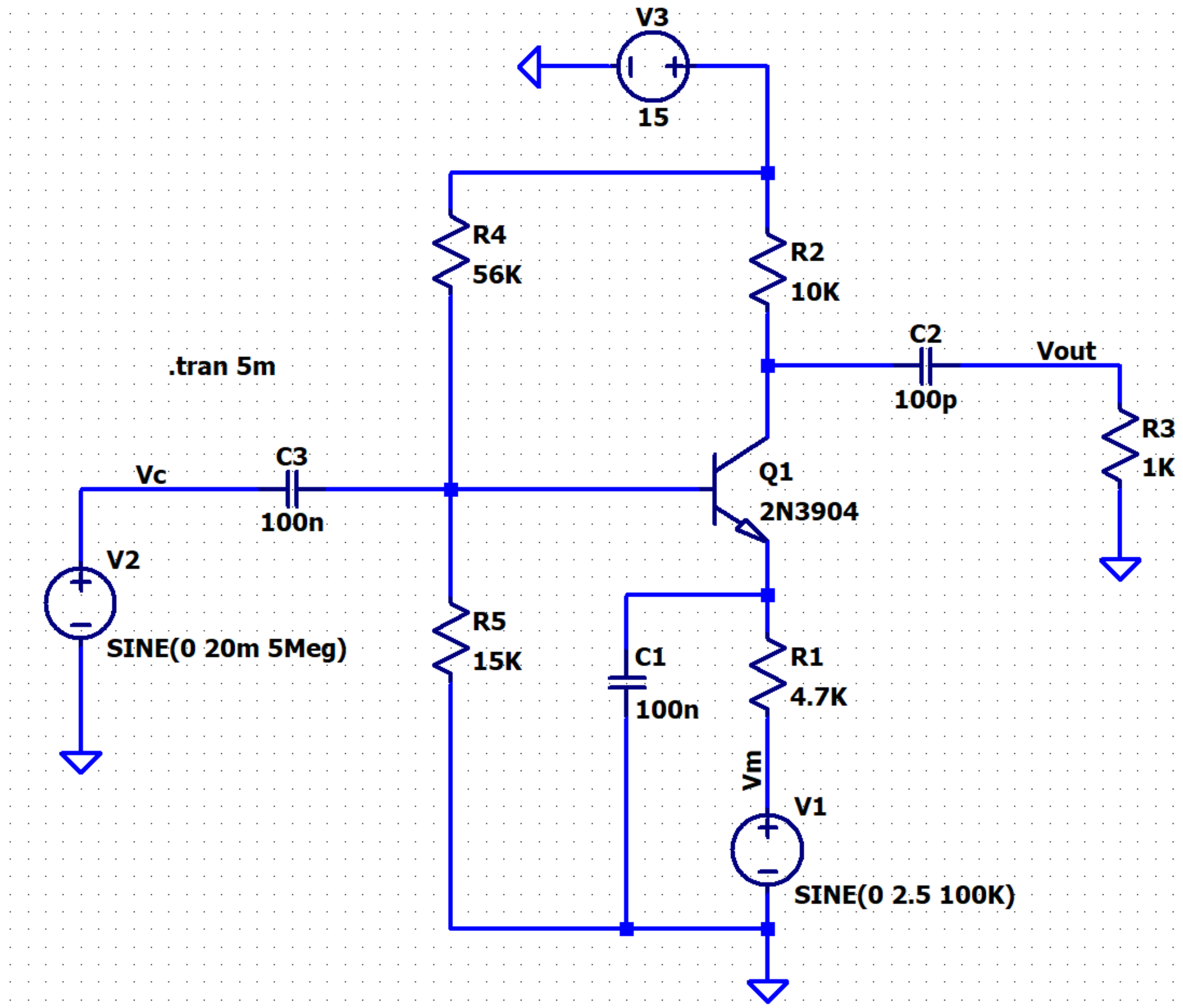
Peak Amplitude (dBm): Computed as the sum of “Power Measured (dBm)” and the absolute value of the difference between “Power In (dBm)” and “Power Loss”.

Pin-Pout: Computed as the sum of the absolute value of “Power Measured (dBm)” and the absolute value of the difference between “Power In (dBm)” and “Power Loss”.

<u>Mixer Spur Test</u>

Introduction

The mixer spur test is used to determine the ability of an RF device to filter out noise at harmonic-frequency combinations of input signals. This test is necessary in determining the correct RF device for a specific set of input signals. To illustrate the execution of this test, an NPN-transistor-based circuit was used to create an AM wave. The diagram of the circuit is shown below:



The circuit is based on a couple of references found online. [6][7] The circuit was built on a protoboard and connected as necessary to the proper test equipment. From the circuit, the following is important:

Carrier Signal: 20 mVp, 5 MHz

Message Signal: 2.5 Vp, 100 kHz

Input

For the mixer spur test to operate, all of the inputs must be configured. The following inputs are needed:

- Number of Harmonics
- 2 of the following 3:
    - RF Frequency
    - IF Frequency

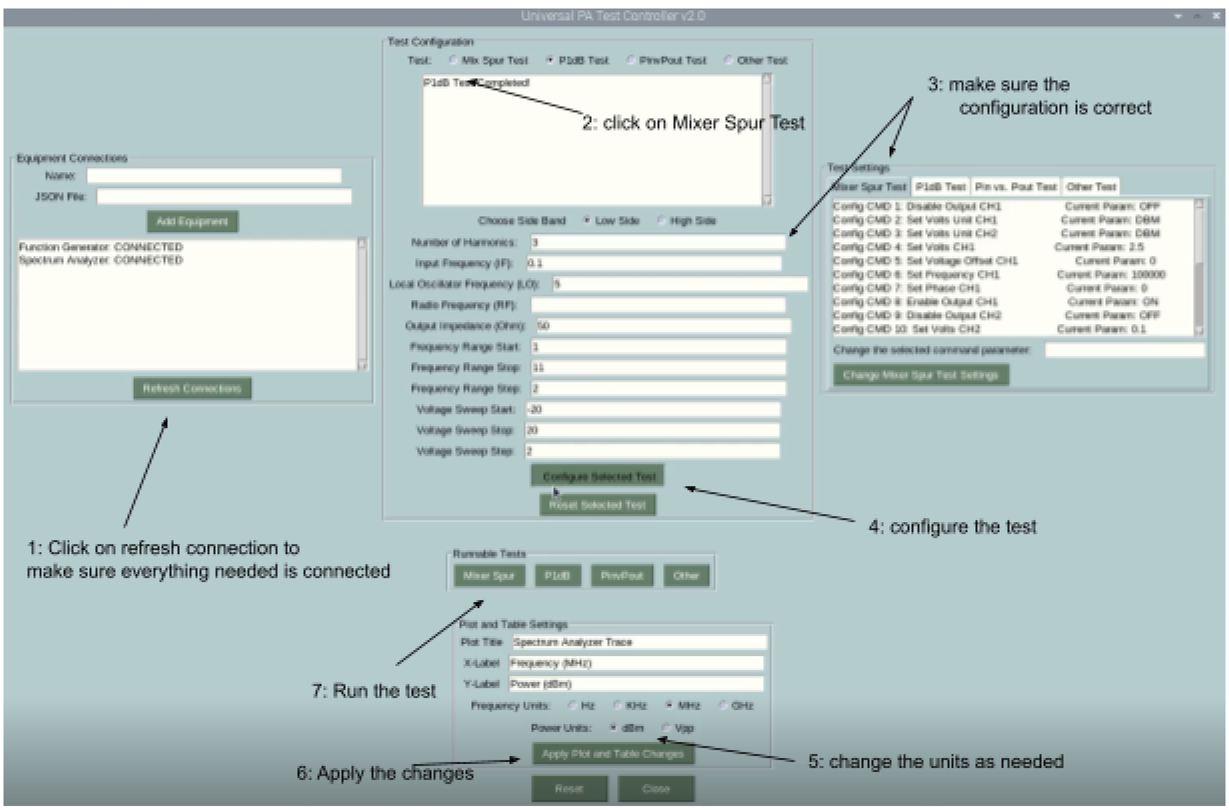
  - LO Frequency
- Frequency Range (Start and Stop)
- Low side or high side

The number of harmonics will determine the size of the mixer spur table (described in subsequent "Output" section). The user must only choose two of the three listed input frequencies for the table to base measurements off of. The frequency range is self-explanatory, however, the user should be aware of the frequency limits of their equipment. Additionally, the user must specify whether the power outputs map to the high side or the low side of the signal. They following the following formulas:

High Side Frequency Output = | m*(IF or RF or LO) + n*(IF or RF or LO) |

Low Side Frequency Output = | m*(IF or RF or LO) - n*(IF or RF or LO) |

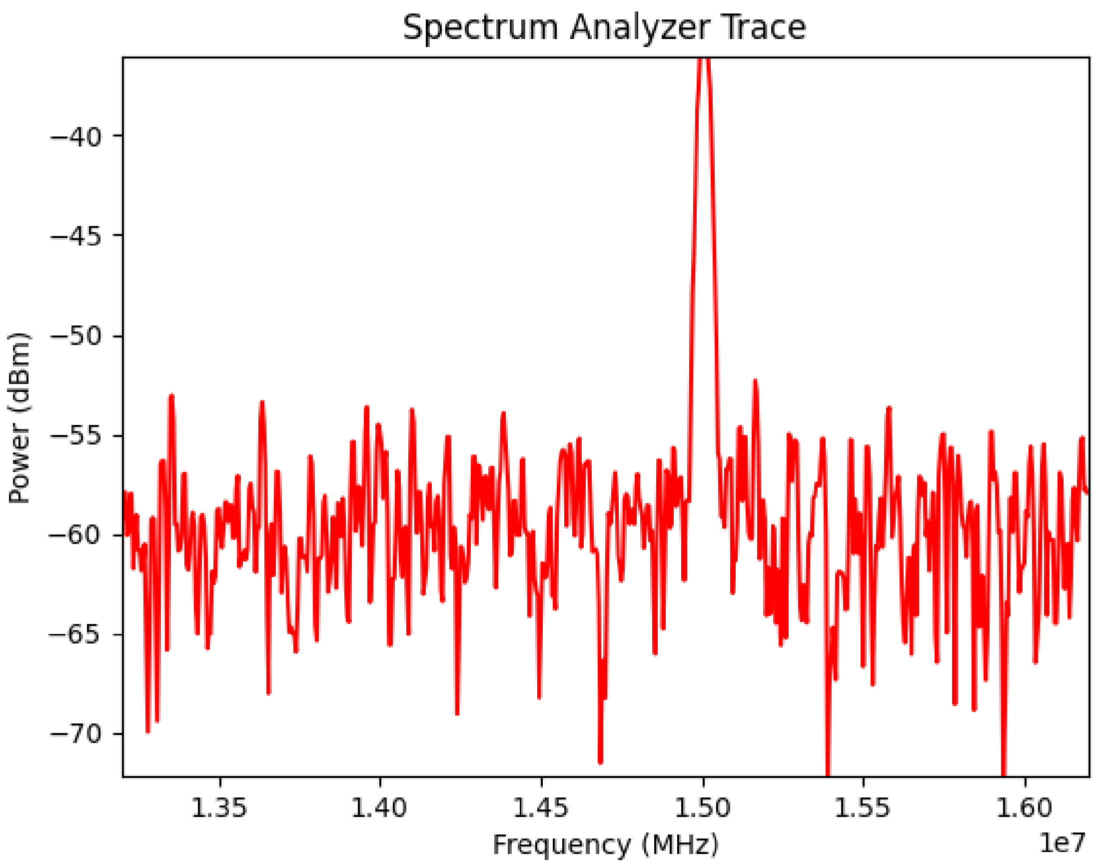
For the AM-circuit example, an AM wave is being created. This means that the output signal is the RF wave and our inputs are the IF and LO waves. As shown in the circuit diagram from before, the IF is set to 100 kHz and the LO is set to 5 MHz (units in MHz). For simplicity, the number of harmonics wanted is 3 and the frequency range is from 1-30 MHz. The inputs are placed in the GUI as shown below:



Note: Ignore the "Impedance" and "Voltage Sweep" inputs, as they are not used for the mixer spur test.

Output

Once the proper inputs and test equipment are configured, the user can run the test. During the test, the program will output an update from the spectrum analyzer. An example of such is shown below:



At any point during the test the user can exit out of the trace window to abort the test. You may have to click the close button on the trace window multiple times for a test to be aborted. However, once the test is finished, the program will output a spur table. For the example with the AM circuit, the output is shown below:

	0x0.1MHz	1x0.1MHz	2x0.1MHz	3x0.1MHz
0x5MHz	Out of Range	Out of Range	Out of Range	Out of Range
1x5MHz	-19.62392 dBm	-44.80186 dBm	-60.12447 dBm	-59.55498 dBm
2x5MHz	-29.87074 dBm	-54.38129 dBm	-61.03436 dBm	-60.97209 dBm
3x5MHz	-44.84071 dBm	-61.52055 dBm	-56.06066 dBm	-60.69922 dBm

Note that the table includes zero combinations of the harmonics. Also note that if the frequency combination is outside the range the user set, the table will output "Out of Range."

For this specific data, the test was executed on the mentioned AM circuit with the following parameters:

Number of Harmonics:	3
2 of the following 3:	
RF Frequency:	N/A
IF Frequency:	0.1 (MHz)
LO Frequency:	5 (MHz)
Frequency Range (Start and Stop):	1, 30 (MHz)
Low side or high side:	Low Side

P1 dB Test

Introduction

The P1dB test is used to find the input power at which point the output power is attenuated by 1dB. All powers are measured in dBm. This test might take a long time to run. It is recommended to run the test twice because the spectrum analyzer value fluctuates too much at low power.

Note: If the compression point is NOT FOUND or too low, it means the result is affected by the low power fluctuation. If the voltage stop is set to a small value, the gain might be attenuated by less than 1dB instead of 1dB results in the incorrect output. Also, we manually changed the output impedance of the signal generator to 50 Ohm instead of highZ

Input

Required equipment are signal generator and spectrum analyzer.

The required input parameters are:

Output Impedance of the Channel
Frequency range
Frequency Step
Voltage Range
Voltage Step

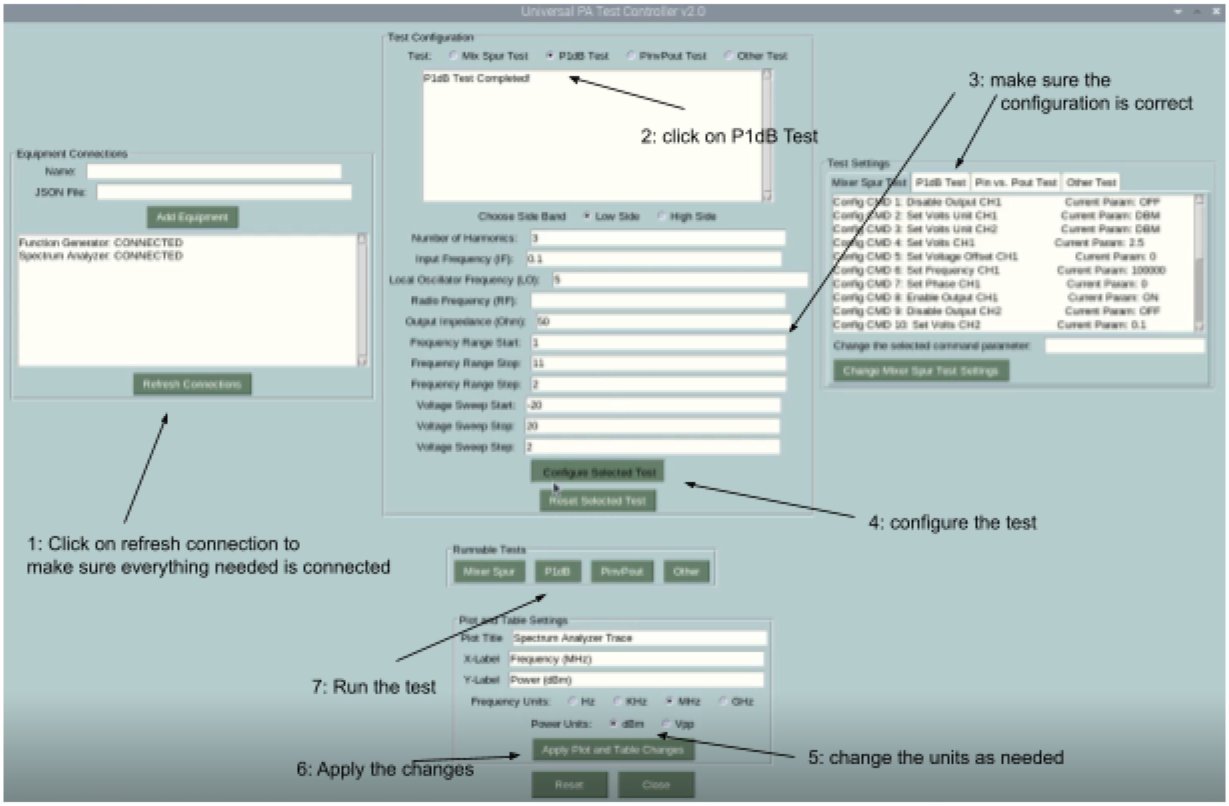
For the circuit and the settings we use:

We used a Class A BJT Common Emitter for P1dB testing

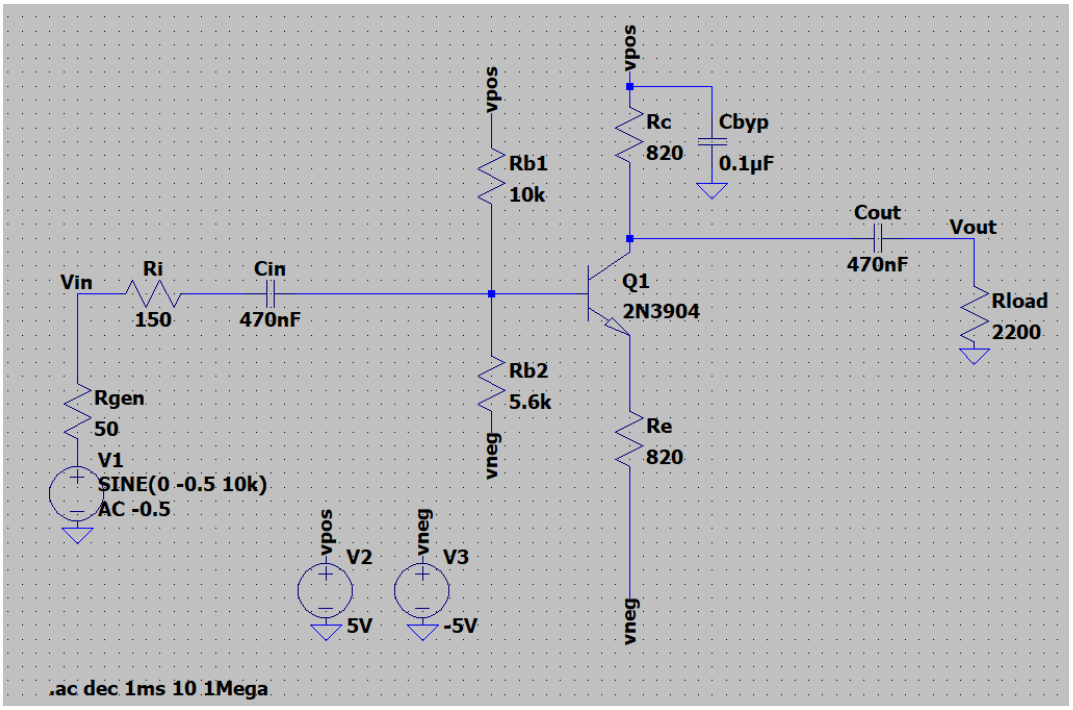
Output Impedance of the Sig Gen:	50 Ohm
Frequency range:	1M - 11M Hz

Frequency Step:	2MHz
Voltage Range:	-20dBm - 20dBm
Voltage Step:	2dBm

(User can choose the unit of the frequency and voltage, in this case, MHz and dBm are chosen)



(please ignore other input parameters for which are for other tests)



[8] CE Amplifier

Output

The program will slowly sweep through each frequency to find its corresponding input power 1dB compression point. The speed of running the test varies based on the hardware.

A table of input power 1dB compression point and its corresponding frequency is generated

Frequency (MHz)	Input Power 1dB Compression Point
1.0	-16.0
3.0	16.0
5.0	14.0
7.0	Not Found
9.0	20.0
11.0	20.0

Result output

References

[1] “Control your instruments with Python,” *PyVISA*. [Online]. Available: https://pyvisa.readthedocs.io/en/latest/. [Accessed: 22-Mar-2021].

[2] “The PySimpleGUI Cookbook,” *PySimpleGUI*. [Online]. Available: https://pysimplegui.readthedocs.io/en/latest/cookbook/. [Accessed: 22-Mar-2021].

[3] “Visualization with Python,” *Matplotlib*.[Online]. Available: https://matplotlib.org/. [Accessed: 22-Mar-2021].

[4] “Tables,” *Plotly*. [Online]. Available: https://plotly.com/python/table/. [Accessed: 22-Mar-2021].

[5] *NumPy*. [Online]. Available: https://numpy.org/. [Accessed: 22-Mar-2021].

[6] Rik99, “AM Modulator with BJT 2N3904,” Electrical Engineering Stack Exchange, 23-Feb-2018. [Online]. Available: https://electronics.stackexchange.com/questions/357942/am-modulator-with-bjt-2n3904. [Accessed: 20-Mar-2021].

[7] J. P, AMPLITUDE MODULATION, 04-Jan-2018. [Online]. Available: http://www.ktulabs.com/2018/01/amplitude-modulation.html. [Accessed: 20-Mar-2021].

[8] Virginia Tech, Common Emitter Amplifier Project. [Online]. Available: https://www.courses.ece.vt.edu/ece3274/common_emitter.pdf. [Accessed: 25-Mar-2021].