

Automated Frequency Response Measurement

Introduction

Modern test instruments can be controlled over USB or network interfaces. This enables automated measurements.

In this lab you will write a script that uses the lab's AWG, oscilloscope and DMM to measure the frequency response of a circuit.

Control Interfaces

The lab PC is connected to the three instruments through their USB ports. The AWG and 'scope also have Ethernet network interfaces. Older instruments may have an instrumentation-specific parallel bus known by the names IEEE-488, GPIB or HPIB.

The computer acts as the master and sends command strings to the "slave" instrument. The instrument acts on each command and, when necessary, sends back a response string after the command has been carried out.

SCPI

SCPI (Standard Commands for Programmable Instruments, pronounced "skippy") is a standard for these instrument commands. The SCPI standard defines different commands for each class of instrument (meters, oscilloscopes, power supplies, etc).

However, a particular model may not support all of the SCPI commands for that instrument. You must check each instrument's programmer's manual for the commands that it supports. The programmer's guides for the lab AWG, 'scope and DMM are available on the course web site – but don't print them out, they total nearly a thousand pages!

Useful SCPI Commands

SCPI commands that most instruments recognize and that you may find useful for this lab include:

***rst** sets the device to default values¹

¹While the instrument is being reset it may not respond to

***idn?** retrieves a device's identification string to check that you're "talking" to the right device

***opc?** returns 1 when all previous commands have completed. This is useful for synchronising operations across several instruments. The DMM does not support this command.

***esr?** returns a non-zero value of the "error status register" if a previous command has encountered an error and then resets this register. The DMM does not support this command.

Rigol DS2074 DSO

The lab 'scope is a Rigol DS2074 Digital Storage Oscilloscope (DSO). Some of its SCPI commands that you may find useful for this lab include:

:channel<n>:probe 10 - tells the instrument that channel <n> is using a 10x probe where <n> is 1 or 2 (omit the angle brackets)

:channel<n>:coupling AC - sets channel <n> to AC-coupled input

:channel<n>:offset 0 - sets the vertical display offset for channel <n> to zero

:channel<n>:scale <scale> - sets the vertical scale on channel <n> to <scale> volts/division

:timebase:scale: <scale> - sets the horizontal sweep to <scale> seconds/division

:measure - to enable and return measurement results (frequency, phase, RMS voltage). For example:

:meas:freq? chan1 - frequency of Channel 1

:meas:vrms? chan1 - rms voltage of Channel 1

:meas:vrms? chan2 - rms voltage of Channel 2

SCPI commands. Allow a few seconds before sending another command.

:meas:ph? chan1,chan2 - phase difference (in degrees) between Channels 1 and 2

Note that the Rigol 'scope indicates measurement errors by returning an impossibly large value (e.g. 9.9E37) rather than by setting the error status register.

Rigol DG1000Z AWG

Some of the AWG's SCPI commands that you may find useful for this lab include:

:output<n> on sets output <n> (default 1) on.

:source<n>:apply:sin <f>,<a> configure source <n> (default 1) to generate a sine wave of frequency <f> (Hz) and amplitude <a> (Vpp).

B&K 2831E DMM

Chapter 5 of the [DMM manual](#) describes the available commands. Some of the DMM's SCPI commands that you may find useful for this lab include:

:function voltage:ac sets the DMM to measure AC voltage

:fetch? reports the most recent measured value

VISA

VISA (Virtual Instrument System Architecture) is an application programming interface (API) for writing programs to control instruments. VISA is a cross-platform multi-interface API that allows the same program to be used on computers with different operating systems (Windows, Unix and OS-X) and different interfaces (USB, Ethernet, GPIB, and others). VISA defines different APIs for different programming languages.

Several test equipment manufacturers² provide free VISA software. The lab PCs have libraries and utilities from both National Instruments and Keysight installed³.

The VISA API function arguments and return values are described in the VISA API Reference. The

²Suppliers of the most popular VISA software are NI - National Instruments and Keysight (formerly Agilent, formerly HP).

³NI VISA software is only free if you have purchased NI products.

NI-VISA and Agilent VISA User's Manuals explain in detail how to use these functions. All three are available on the course web site.

pyvisa

Python is a popular computer language that is well suited for writing test automation scripts. [pyvisa](#) is a Python library that provides an interface to the VISA library.

You do not need to learn the Python language for this lab. You will be provided with a set of Python commands that you can edit and arrange into a script to make automated frequency response measurements.

Useful pyvisa Methods

The following pyvisa methods (object functions) suffice for completing this lab:

pyvisa.ResourceManager() - to initialize the VISA system

rm.list_resources() and **rm.open_resource()** - find an instrument and create a connection to it

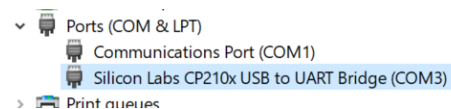
inst.write() - send a SCPI command to an instrument

inst.query() - send a SCPI command to an instrument and read a response

In the above, **rm** is an object returned from **ResourceManager()** and **inst** is an object returned from **rm.open_resource()**.

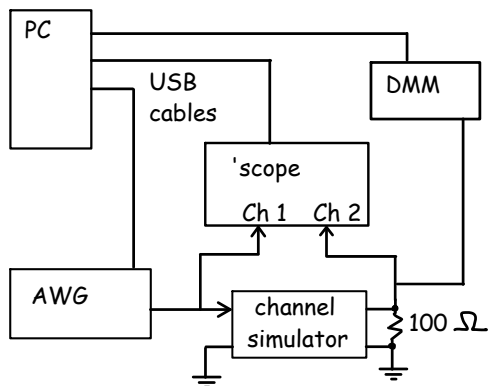
Procedure

Verify that the DMM driver has been installed and the DMM is connected by running the Windows utility "Device Manager" and checking under "Ports (COM & LPT)" for the CP210x driver as shown below. Note the COM port that was assigned (e.g. COM3 in this case).



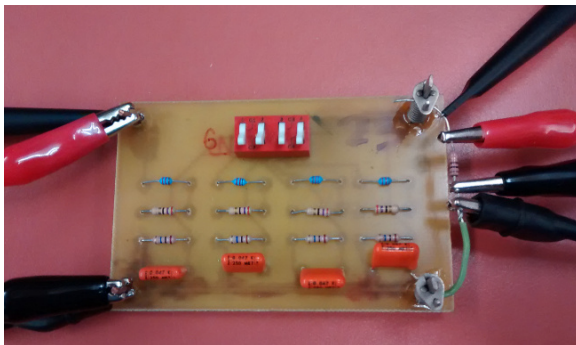
Connect Equipment

You will be testing your script by measuring the frequency response of a baseband channel simulator built on a printed circuit board. The following diagram shows the measurement setup:



As shown in the photo below, the AWG and the Channel 1 probes should be connected to the circuit terminals marked IN and GND and the 100 Ω resistor, the Channel 2 probe and the DMM probes to the terminals marked OUT and GND.

The USB ports on the back of the 'scope, AWG and DMM should be connected to USB ports on the back of the PC.

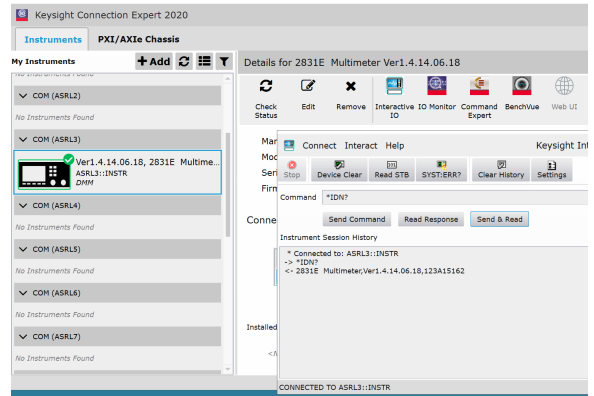


Check Instrument Status

The Keysight Connection Expert utility shows you the model, serial numbers and VISA addresses of connected devices. It also allows you to issue SCPI commands and queries to each instrument.

The interface marks the available instruments in green. Selecting an instrument and clicking on the “Interactive IO” button brings up a window you can use to “Send Command” or “Send & Read” a query to a particular instrument. The `*idn?` command can

be used to check that an instrument is connected and identify it.



Edit the Python Script

A sample Python script is provided on the course web site to get you started with this lab. The script includes some, but not all, of the functionality required. You will need to enhance it so that your script does the following:

- create the VISA default resource manager (**rm**) and the three instrument objects (**dmm**, **awg** and **dso**).
- check that the correct instruments have been connected by issuing a `*IDN?` query to each and printing the result
- send the appropriate SCPI commands to configure the instruments: Both 'scope channels should be set for the probes being used (10 \times), AC coupling and zero vertical offsets. The Channel 1 (channel input) vertical scaling should be set so the waveform takes 6 divisions vertically⁴. The AWG channel 1 output should be enabled. The DMM should be set to measure AC voltage.
- do the following repeatedly (in a `while`-loop) over a frequency range of 1 kHz to about 200 kHz in 10 steps per decade⁵:

- set the AWG to the desired test frequency and an output level of 6 V_{pp}

⁴Note that the scale values can be set to arbitrary values – they don't need to be usual powers of 10 times 1, 2 or 5.

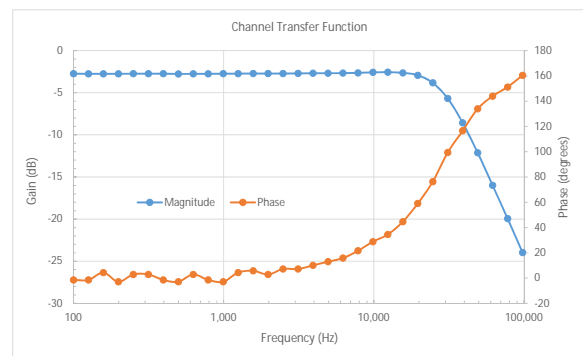
⁵A decade is a range of frequencies from f to $10f$; an octave is from f to $2f$.

- read the DMM AC voltage (this is the voltage at the output of the test circuit and at the input of Channel 2 of the 'scope)
- use the measured DMM voltage to compute and set the 'scope's Channel 2 voltage scale so that the Channel 2 signal takes 6 divisions vertically.
- based on the test frequency, compute and set the 'scope's horizontal scale to display two cycles of the sine wave
- use the 'scope to measure the test circuit's input and output voltages and the phase difference between input and output
- print the frequency, input voltage, output voltage and phase shift, separated by commas

```
PS Z:\> python lab5.py
RIGOL TECHNOLOGIES,DS2072,DS2A153702419,00.01.01
Rigol Technologies,DG1032Z,DG1ZA164251577,00.01.06
2831E Multimeter,Ver1.4.14.06.18,123A15162
1000.0, 2.31, 1.0925e+00, 1.1212e+00, 2.3
1250.0, 1.58, 1.0971e+00, 1.1214e+00, 3.1
1562.5, 1.59, 1.0971e+00, 1.1214e+00, 3.1
1953.1, 1.59, 1.0963e+00, 1.1219e+00, 4.6
2441.4, 1.59, 1.0951e+00, 1.1208e+00, 5.4
3051.8, 1.58, 1.0929e+00, 1.1193e+00, 6.2
3814.7, 1.58, 1.0896e+00, 1.1174e+00, 8.5
4768.4, 1.57, 1.0844e+00, 1.1123e+00, 11.6
5960.5, 1.56, 1.0766e+00, 1.1057e+00, 13.1
7450.6, 1.54, 1.0649e+00, 1.0976e+00, 15.5
9313.2, 1.50, 1.0470e+00, 1.0805e+00, 19.3
11641.5, 1.45, 1.0204e+00, 1.0545e+00, 25.5
14551.9, 1.39, 9.8106e-01, 1.0155e+00, 31.6
18189.9, 1.30, 9.2492e-01, 9.5844e-01, 41.7
22737.4, 1.21, 8.4825e-01, 8.8025e-01, 54.8
28421.7, 1.16, 7.5023e-01, 7.8030e-01, 74.2
35527.1, 1.18, 6.3269e-01, 6.6148e-01, 93.5
44408.9, 1.29, 5.0907e-01, 5.3194e-01, 113.6
55511.2, 1.47, 3.9241e-01, 4.0434e-01, 128.7
69388.9, 1.65, 2.9116e-01, 2.9410e-01, 143.4
86736.2, 1.82, 2.1211e-01, 2.0691e-01, 149.9
108420.2, 1.99, 1.3347e-01, 1.4415e-01, 156.5
135525.3, 2.09, 9.0544e-02, 9.6106e-02, 161.1
169406.6, 2.15, 6.1928e-02, 6.3493e-02, 164.2
PS Z:\>
```

Data Analysis

Copy and paste the data on the screen into a blank spreadsheet (select the script output, right-click on the window title bar and select Edit>Copy). Use Data > Text to Columns to parse a column of text into individual spreadsheet columns. Add a column that computes the gain of the circuit in dB. Add a chart (graph) that shows both the gain and the phase shift of circuit as a function of frequency. Use an XY graph showing smoothed lines with logarithmic axis for frequency. Use two axes (gain on left, phase on the right). Label the graph and the axes. Your graph (but not necessarily the data) should look as follows:



You should end a sequence of commands with a `*opc?` query and add a delay (e.g. by using the `time.sleep(<seconds>)` function after changing the frequency to allow time for the commands and the measurements to be completed.

Use an editor such as Notepad++ to edit your Python script. Open a “command prompt” or Powershell window in your working directory (shift-right-click and select “Open Powershell Here”). Run the command `python lab5.py` to run your script.

Each line output by your script should have five values (frequency, DMM voltage, test circuit input and output voltages and phase shift) separated by commas. This CSV (comma separated values) format can be parsed by a spreadsheet:

Remember to save your script and measurements for your report.

Hints

A methodical approach will help you complete this lab. The following hints might help:

- use the **Keysight Connection Expert** utility to check that the instruments are connected.
- use the **Interactive IO** utility (select an instrument and click on the “Interactive IO” button) to issue SCPI commands manually and make sure the instrument responds as expected
- if you run the UltraScope utility you must leave it in **STOP** mode so it does not interfere with your script. Since the display will not update you must right-click on the screen and choose Print Instrument Screen to see what’s on the ’scope screen.
- test a simple version of your script. Gradually add functionality, checking for correct operation after each addition.

Report

Submit the following two files to the dropbox on the course web site:

1. a report in PDF format containing the usual identification information, a listing of your (working) script, the graph of the circuit amplitude and phase responses formatted as described above, and the answers to the following questions:
 - (a) You wish cover a frequency range of four decades with an equal *ratio* between successive frequencies. What multiplicative factor should you use when computing one frequency from the previous one if you wish to cover this range with 20 test frequencies? (*Hint: if f_0 is the start frequency then after n multiplications by k the frequency f_n is $f_0 k^n$.*).
 - (b) What vertical gain setting in units of volt/division would you need to use so that a signal with a peak-to-peak voltage range of V volts extended over a range of ± 3 divisions? (*Hint: check your answer by analyzing the units and using a simple example*).

- (c) What sweep rate, in seconds per division, is required to display four cycles of a sine wave of frequency f ? *Hints: The ’scope has 14 divisions horizontally. The duration of one cycle is $\frac{1}{f}$.*
 - (d) Is the DMM’s AC voltage reading an RMS or average measurement? How would you scale this reading to obtain the peak-to-peak voltage? *Hint: See the DMM manual’s Feature Overview or Specifications section under 2831E, AC voltage.*
2. your spreadsheet in Microsoft Excel format (.xls or equivalent) containing the captured data and your chart (graph)