Automated Frequency Response Measurement

Modified 2019-11-03 to accommodate a simplified version of the sample software.

Introduction

Many instruments can be controlled by software. This allows many test and measurement tasks to be automated.

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

Control Interfaces

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

The computer acts as the master and sends a command string 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, often 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 type of instrument. You need to 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 all instruments recognize and that you may find useful for this lab include:

***rst** sets the device to default values[1]

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

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

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

Note that the 'scope may indicate that the input signal level is out of range or that there was an error in the SCPI command by returning an impossibly large value (e.g. 9.9E37).

[1] While the instrument is being reset it may not respond to SCPI commands. Allow a few seconds before sending another command.

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) that helps you to write 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. We will use the C API.

Several test equipment manufacturers[2] provide free VISA software. For this lab we will use the Keysight (formerly Agilent) VISA software[3]. The VISA API is defined by the `visa.h` include file. The `visa64.lib` (or `visa32.lib`) library contains the functions that need to be linked into your program by the compiler. Keysight include several utilities, described below, for troubleshooting.

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

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

[3]NI VISA software supports multiple operating systems but is only free if you have purchased NI products.

Although the instrument and VISA manuals make for exciting reading, you will probably not have time to go through them from cover to cover. But you should understand where to find specific information, such as the description of a VISA function or SCPI command, and be able to quickly look it up by scanning the table of contents, the index or by searching for strings within a document.

Useful VISA Functions

You will probably find the following VISA functions useful for completing this lab:

viOpenDefault - to initialize the VISA system

ViOpen - open a communication link to an instrument

viPrintf - send a SCPI command to an instrument

viScanf - read a response and extract one or more values

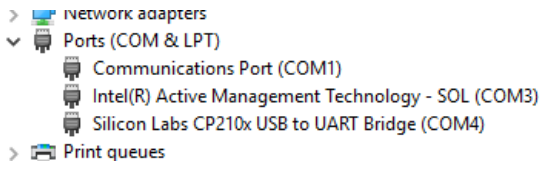
viQueryf - combines the two previous functions

viClose - close opened resources

Procedure

You may need to disable the DMM's command echo from the front panel[4]: press Shift then Esc; press → until you see `C:SYS MEU`; press ▽ then → until you see `4:RETURN`; press ▽ then → until you see `OFF`. Press Auto to save the new setting and then Shift and Esc.

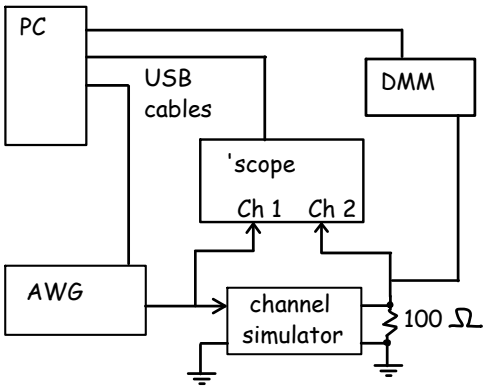
Verify that the DMM driver has been installed and the DMM is connected by running the Windows utility "Device Manager" and checking under "Ports (COM & LPT)" for the CP210x driver as shown below. Note the COM port that was assigned (e.g. COM4).



[4]If you don't change this setting the VISA software will not recognize responses from the DMM.

Connect Equipment

The following diagram shows the measurement setup:

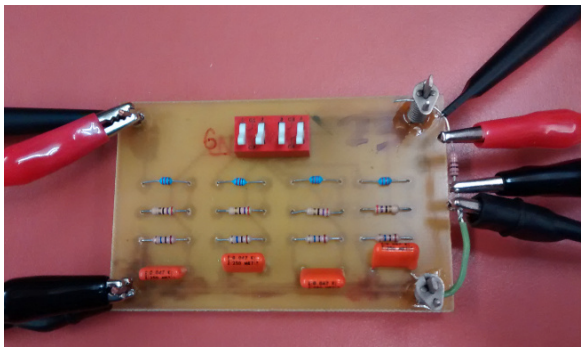


All the switches on the circuit board should in the down position[5]. Check the circuit for zero resistance between GND terminals and a finite resistance between IN and OUT.

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.

Record (one of the) number(s) written on the board.

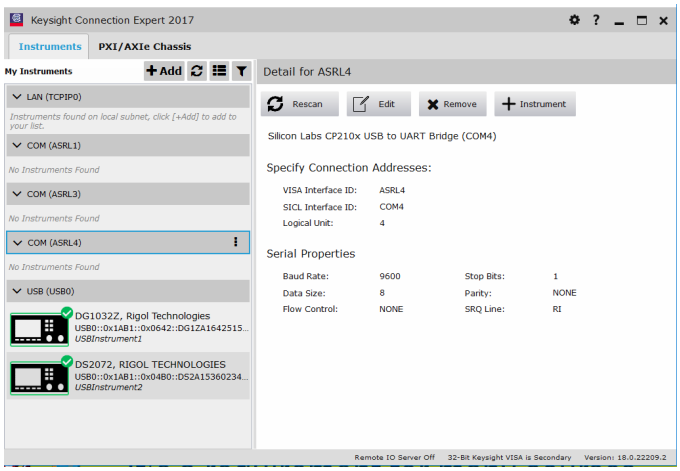
The USB ports on the back of the 'scope, AWG and DMM should already be connected to USB ports on the back of the PC.



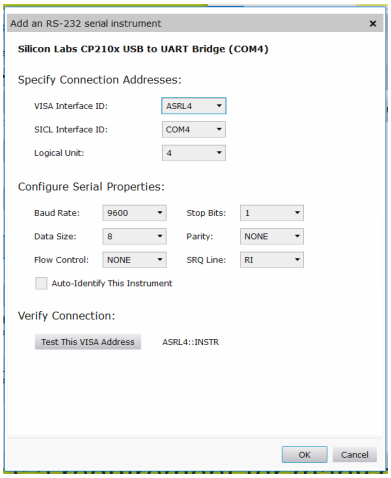
Configure DMM VISA Instrument

The DMM's USB interface emulates a serial port and serial ports do not identify the type of device so you will need to add the virtual COM port assigned to the DMM as a VISA instrument.

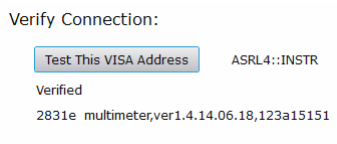
Run Keysight Connection Expert:



Find the serial port that is a USB to UART bridge (COM4 in the example above), click on the menu icon (⋮) and select "Add Instrument:"



select "Auto-Identify" and click on "Test This VISA Address." If the DMM is properly configured you will see its model and serial number:



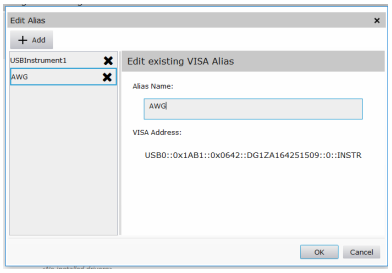
click on OK.

[5]The board orientation is with the common or "GND" terminals along the bottom of the board and the switches along the top.

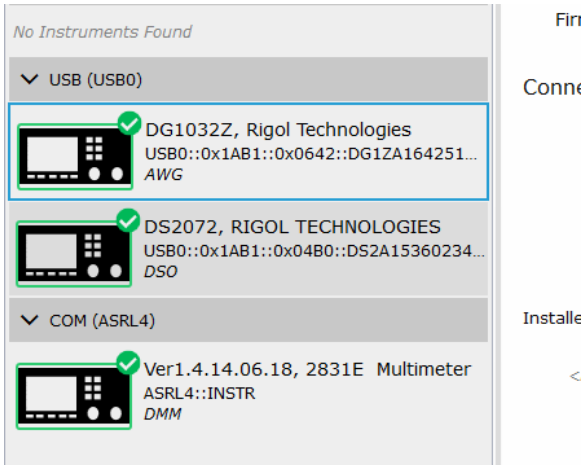
Assign VISA Aliases

Although a VISA program can select a device by model or serial number, we might want to use more than one device of the same model (e.g. two AWGs) or use the same program with equipment that has different serial numbers (e.g. multiple test setups). So VISA software is usually written to select devices using *aliases* which are labels used by your program for the different instruments in a particular test setup.

The Connection Expert utility shows you the model, serial numbers and VISA addresses of connected devices. It also allows you to assign aliases to each instrument. In the "Details" panel for each instrument you will see a "Connection Strings" section that lists different ways that each instrument can be addressed. Click on the "..." button under Aliases and change (or add) an alias that corresponds to the names used in your C program. For example, to add the alias AWG:



After you are done you should have three instruments, each with the correct alias:



Since the aliases are stored by the VISA software running in the background, you will need to repeat the VISA configuration if the computer reboots.

Write and Compile Software

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

- open the VISA default resource manager and the three instruments
- check that the correct instruments have been connected by issuing a '*IDN?' command and printing the result to `stderr`
- send commands to configure the instruments: The 'scope should be set for the probes being used (1× or 10×), AC coupling and zero vertical offsets. The Channel 1 (circuit input) vertical scaling should be set so the waveform takes 6 divisions vertically[6]. The AWG channel 1 output should be enabled. The DMM should be set to measure AC voltage.
- do the following repeatedly (in a loop) over a frequency range of 100 Hz to 100 kHz[7] in 10 steps per decade[8]:
  - set the AWG to the desired test frequency and an output level of 5 V_{pp}
  - 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
  - read the measured test circuit's input and output voltages and the phase difference between them

[6] Note 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.

[7] The DMM upper frequency limit is 100 kHz.

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

– print the frequency, output voltage and phase shift, separated by commas, to both standard output (`stdout`) and standard error (`stderr`)

You may need to insert delays (e.g. by using the `Sleep()` function) after changing instrument settings.

Use an editor such as Notepad++ to edit your C program. Open a "command prompt" or Powershell window in your working directory (shift-right-click and select "Open Powershell Here"). Download the `visabuild.bat` batch file from the course web site. This script will run the C compiler with the appropriate options (assuming the compiler and VISA libraries were installed in their default locations). Supply the base name of the C file as an argument to the script (e.g. if your C program is in the file `lab7.c` you would use the command `./visabuild lab7.c`). If there are no errors this will create an executable program called `lab7.exe` which you can run with the command `./lab7`.

Each line output by your program should have three values (frequency, ratio of output to input voltage and phase difference) separated by commas. This is called CSV (comma separated values) format and can be read by spreadsheet software.

Measurement Procedure

Start the IO Monitor utility from Keysight Connection Expert (⚙ → Tools → IO Monitor) and click on "Start Capturing Messages." VISA commands with errors will be shown in red.

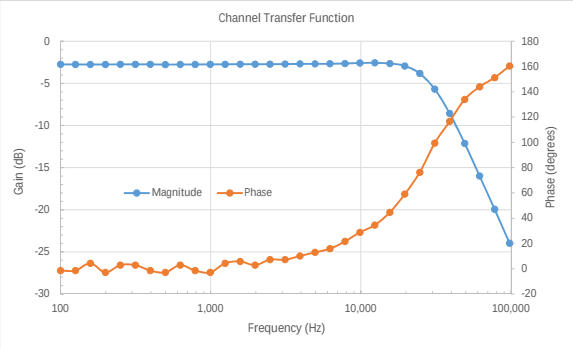
Run your program with the standard output redirected to a file. For example:

```
./lab7 >results.csv
```

Monitor the your program's operation and resolve any problems.

Data Analysis

Open the `.csv` file with a spreadsheet and *immediately save it in a spreadsheet format*[9]. 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:



Remember to save your C code and measurements on your network or flash drive.

Hints

A methodical approach will help you complete this lab on time. The following approach is recommended:

- use the **Keysight Connection Expert** utility to check that the instruments are connected and have been assigned the correct aliases.
- 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
- comment out untested portions of your program using `#ifdef`/`#endif` and test a simple version of your program. Gradually add functionality, checking for correct operation after each addition.

Pre-Lab Tasks

Write a C program that meets the requirements described above. You can start from the example given or write your own. You will not be able to test your code without the instruments but you can download the `visa.h` and `visatype.h` files and compile your program with any C compiler to check for syntax errors. For example, if the visa include files are in the current directory, the tcc command: `tcc -c -I.`

[9] Formulas and charts cannot be saved in `.csv` files, if you skip this step you will lose all your analysis!

`lab7.c` will compile your code and report any syntax errors.

Submit the answers to the following questions along with a listing of your code in a PDF file to the dropbox on the course web site before the lab:

(1) 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 $f_0 k^n$.*).

(2) What vertical gain setting in units of volts/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*).

(3) 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* $\frac{1}{f}$.

(4) 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.*

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, the number of the board you measured, a listing of your (working) C code, the graph of the circuit amplitude and phase responses formatted as described above.

2. your spreadsheet in Microsoft Excel format (.xls or equivalent) containing the captured data and your chart (graph)