

Lab 3 - Automated Frequency Response Measurement

Introduction

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

In this lab you will write a program that uses the lab's digital oscilloscope to measure the transfer function of a simulated telephone channel.

Interfaces

You will connect the lab PC to the 'scope through its USB port. We could also have used an Ethernet interface. Older instruments often have an instrumentation-specific parallel bus variously known as IEEE-488, GPIB or HPIB

The computer acts as the interface master and sends commands to the instrument. The instrument acts on each command and sends back a response, if necessary, after the command has been carried out.

SCPI

SCPI (Standard Commands for Programmable Instruments, or "skippy") is a standard format 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 commands. Each instrument's manual describes the specific commands that are supported. The programmer's guide for the Rigol DS2072 oscilloscope that we will use is available on the course web site – but do not print it out, it is over 600 pages long!

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 values¹

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

***idn?** retrieves a device's identification string to check if you've connected to the right device

Some of the 'scope's 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)

: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 there is an error in the SCPI command by returning an impossibly large value (e.g. 9.9E37).

VISA

VISA (Virtual Instrument System Architecture) is an application programming interface (API) that allows 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 various operating systems (Windows, Unix and OS-X) and

interfaces (USB, Ethernet, GPIB, and others). VISA defines slightly different APIs for different programming languages. We will use the C API.

Several test equipment manufacturers² provide free VISA software packages. For this lab we will use the Agilent VISA software³. This software package includes the `visa.h` and `visatypes.h` include files that define the API functions and constants, a `visa64.lib` library with the API functions that need to be linked into your program by the compiler and several utilities, described below, that can be used for troubleshooting.

The VISA API function arguments and return values are defined 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.

Although the SCPI 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, 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

viScanf - read a response and extract one or more values

viQueryf - combines the two previous functions

viClose - close opened resources

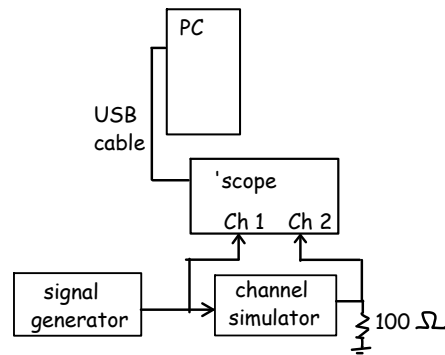
²Suppliers of the most popular VISA software are NI - National Instruments and Keysight (formerly Agilent, formerly HP).

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

Procedure

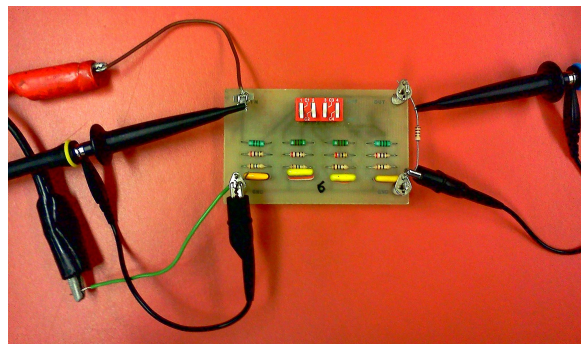
Equipment

The following diagram shows the measurement setup:



As shown in the photo below, the signal generator and the Channel 1 probes should be connected to the channel simulator⁴ terminals marked IN and GND and the 100 Ω resistor and the Channel 2 probe to the terminals marked OUT and GND. All the switches on the channel simulator should be in the down position. The USB port on the back of the scope should already be connected to a USB port on the back of the PC.

You will need to bring two short pieces of wire to the lab to make the connections.



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 will do the following:

⁴The simulator is a four-section lumped-element approximation of the distributed RGLC model of a transmission line.

- open the VISA default resource manager and the oscilloscope instrument
- check that the correct instrument has been connected by issuing a `*IDN?` command and printing the result to `stderr`
- send commands to configure the scope for the probes being used, AC coupling, set the vertical and horizontal scales to reasonable values and both channels' vertical offsets to zero
- repeatedly (in a loop):
 - read the channel 1 frequency, the two channel's RMS voltages and the phase difference between the channels
 - print these four values, separated by commas, to both standard output (`stdout`) and standard error (`stderr`)
 - configure the horizontal scaling and vertical scaling on Channel 2 so that four periods of the most recently measured waveform takes 10 divisions horizontally and the waveform takes 6 divisions vertically (peak-to-peak). 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.
 - wait 1000ms (1s) (you can use the `Sleep()` function)
 - exit the loop if the measured frequency exceeds 200 kHz
- close the instrument and resource manager (`viClose`)

Use an editor such as Notepad++ to edit your C program. Open a “command prompt” window and change to your working directory. Download the `build.bat` batch file from the course web site. This script will run the Pelles C compiler⁵ and linker with the appropriate options⁶. Supply the base name of the C file as an argument to the script (i.e. use the command `build lab3`).

⁵ `tcc` can't link to the VISA libraries.

⁶ Assuming the compiler and VISA libraries were installed in their default locations

Each line output by your program should have four values (frequency, input and output voltages and phase difference) separated by commas. This is called CSV (comma separated values) format and can be read by all commonly-used spreadsheets.

Measurement Procedure

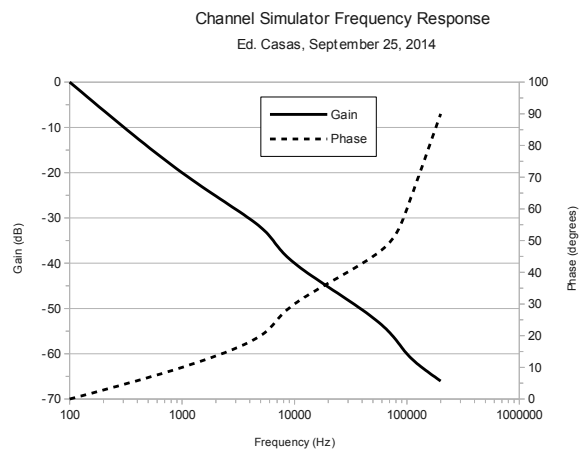
Set the signal generator to output a sine wave of about 4 V peak-to-peak at a frequency of 100 Hz. Run your program with the standard output redirected to a file. For example:

```
lab3 >results1.csv
```

Increase (“sweep”) the generator frequency until it reaches the upper frequency limit and your program terminates.

Data Analysis

Open the `.csv` file with a spreadsheet and immediately save it in spreadsheet format (formulas and charts cannot be saved in `.csv` files). Sort your results by increasing frequency. Add a column that computes the gain of the channel in dB. Add a chart (graph) that shows both the gain and the phase of channel as a function of frequency. Use an XY graph showing smoothed lines with logarithmic axis for frequency. Add two axes (gain on left, phase on the right). Label the graph and the axes. Your graph (but not the name, date or data) should look as follows:



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

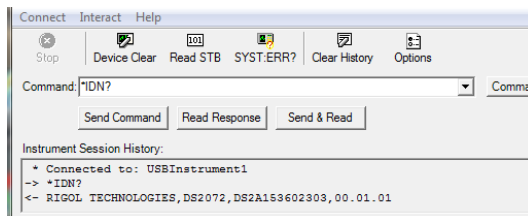
Hints

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

- use the **Agilent Connection Expert** utility ('IO' status bar icon > Agilent Connection Expert) to check that the instrument is connected and find its default VISA Alias (probably "USBInstrument1").



- record the steps required to set up the instrument manually to obtain the required measurements and check that the expected values are displayed
- use the **Interactive IO** utility ('IO' status bar icon > Utilities > Interactive IO) to issue the corresponding SCPI commands manually and ensure the instrument responds with the expected values



- comment out untested portions of your program using `#ifdef/#endif` and use the **IO Monitor** utility ('IO' status bar icon > Utilities > IO Monitor) to 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 an instrument 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. lab3.c` will compile your code and report any syntax errors.

Submit a listing of your code as a PDF file to the dropbox on the course web site before the lab.

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) C code, the graph of the channel amplitude and phase responses formatted as described above, and the answers to the following questions:
 - (a) is this a low-pass, high-pass or bandpass channel?
 - (b) what are the -3, -6 and -10 dB bandwidths?
 - (c) how quickly does the channel "roll-off" with frequency, in units of dB per octave?
2. your spreadsheet in Microsoft Excel format (.xls or equivalent) containing the captured data and your chart (graph)