Project Report
On
Automated Data Acquisition Using VISA

By

Jaison John Titus

&

Ajmal Zaheen Sirajudeen

Electrical & Electronics Engineering

College of Engineering Guindy

Chennai

Duration: 7th June 2013 - 25th June 2013

NCRA • TIFR

Table of Contents

Acknowledgements

Firstly, we would like to thank **Ajith Kumar, Group Coordinator, Digital Backend Group, GMRT** for giving us this opportunity to work at the institute. His support and guidance provided an easier course to the completion of this project.

We would like to thank our guide **Kaushal Buch** of the digital backend group at GMRT for guiding, monitoring and encouraging us throughout the project. His dedication, annotations and thought provoking discussions were the key behind the successful completion of this project.

We would like to thank the digital and analog group for providing the equipments and necessary support for carrying out the various tests for this project.

We would also like to thank the staff members of GMRT who were always supportive and friendly.

June 2013

Ajmal Zaheen Sirajudeen

Jaison John Titus

Abstract

Manual control of various instruments involved in the operation and testing of a digital signal processing system becomes difficult when the setup involves large number of instruments. Hence, an automated remote control mechanism is required to control the system. The project aims at remote control of signal generators and signal analyzers through the application of Virtual Instrument Software Architecture (VISA). The functionalities are adjusted through a C++ program involving VISA headers and Standard Commands for Programmable Instruments (SCPI).The program was built using certain VISA functions and other mathematical and string manipulation functions. The program is capable of controlling any number of instruments as per the user's requests. The various functions in the program are explained in detail with their ensuing SCPI commands. The Input/Output mechanism for the program is also elaborated.

List Of Figures

1. Radio Frequency Signal Generator

1.1 What is a Radio Frequency Signal Generator?

RF signal generators are capable of producing CW (continuous wave) tones. The output frequency can usually be tuned anywhere in their frequency range. Quite a few models offer different types of analog modulation, either as standard equipment or as an optional capability to the base unit. This could include AM, FM, ΦM (phase modulation) and pulse modulation. Another common feature is a built-in attenuator which makes it possible to vary the signal's output power. Depending on the manufacturer and model, output powers can range from -135 to +30 dBm. A wide range of output power is desirable, since different applications require different amounts of signal power.

RF signal generators are required for servicing and setting up analog radio receivers, and are used for professional RF applications.[1]

1.2 Basic Working of RF Sig Gen:

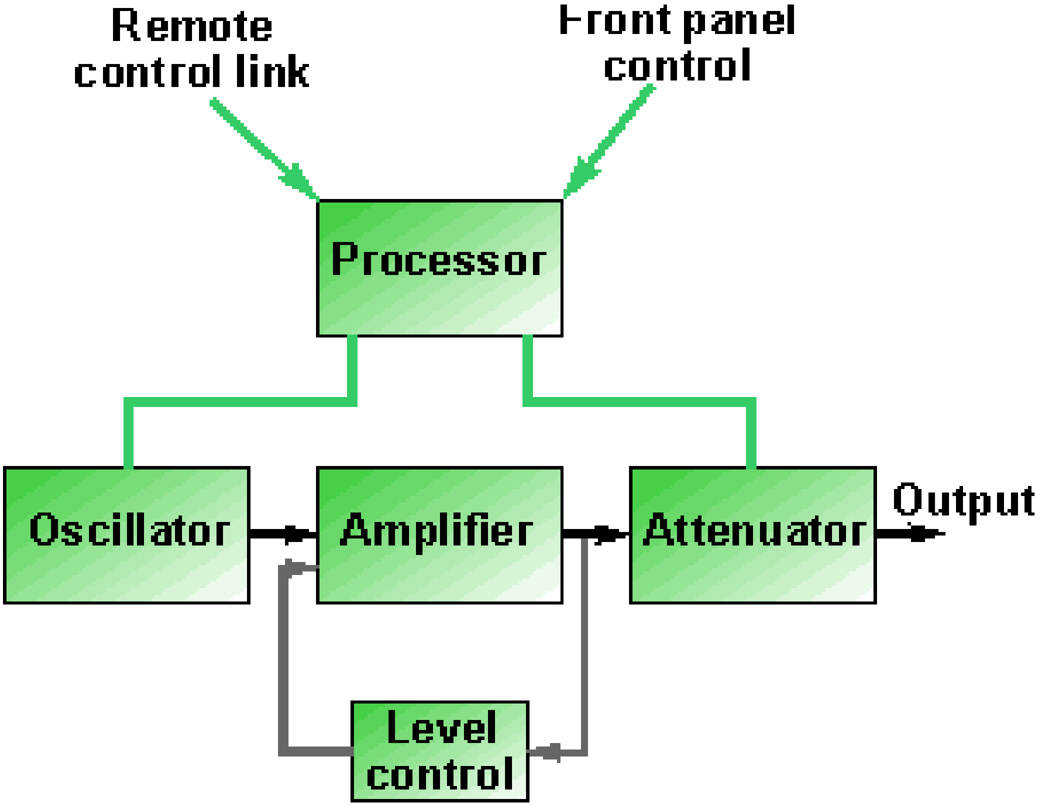


Fig 1: Block Diagram of a Signal Generator

Oscillator: The most important block within the RF signal generator is the oscillator itself. The oscillator would take commands from the controller and be set to the required frequency.

Amplifier: The output from the oscillator will need amplifying. This will be achieved using a special amplifier module. This will amplify the signal, typically to a fixed level. It would have a loop around it to maintain the output level accurately at all frequencies and temperatures.

Attenuator: An attenuator is placed on the output of the signal generator. This serves to ensure accurate source impedance is maintained as well as allowing the generator level to be adjusted very accurately. In particular the relative power levels, i.e. when changing from one level to another are very accurate and represent the accuracy of the attenuator.

Control: Advanced processors are used to ensure that the RF signal generator is easy to control and is also able to take remote control commands. The processor will control all aspects of the operation of the test equipment.

1.3 Functionality:

Frequency range:

The frequency range of the RF signal generator is of paramount importance. It must be able to cover all the frequencies that are likely to need to be generated. For example when testing a receiver in an item of equipment, be it a mobile phone or any other radio receiver, it is necessary to be able to check not only the operating frequency, but other frequencies where the issues such as image rejection, etc.

Output level:

The output range for an RF and microwave signal generator is normally controlled to a relatively high degree of accuracy. The output within the generator itself is maintained at a constant level and then passed through a high grade variable attenuator.

Modulation:

Some RF or microwave signal generators have inbuilt oscillators that can apply modulation to the output signal. Others also have the ability to apply modulation from an external source.

Sweep:

On some RF signal generators it is necessary to sweep the signal over a range. Some generators offer this capability.

Control:

There are many options for controlling RF signal generators these days. While they tend to have traditional front panel controls, there are also many options for remote control. Most items of laboratory bench test equipment come with GPIB fitted as standard, but options such as RS-232, and Ethernet / LXI.

2. Spectrum Analyzer

2.1 What is a Spectrum Analyzer?

A spectrum analyzer measures the magnitude of an input signal versus frequency within the full frequency range of the instrument. The primary use is to measure the power of the spectrum of known and unknown signals.

By analysing the spectra of signals, dominant frequency, power, distortion, harmonics, bandwidth, and other spectral components of a signal can be observed that are not easily detectable in time domain waveforms. These parameters are useful in the characterization of electronic devices.

2.2 Theory behind Operation:

A swept-tuned spectrum analyzer down-converts a portion of the input signal spectrum to the centre frequency of a band-pass filter by sweeping the voltage-controlled oscillator through a range of frequencies, enabling the consideration of the full frequency range of the instrument.

The bandwidth of the band-pass filter dictates the resolution bandwidth, which is related to the minimum bandwidth detectable by the instrument.[2]

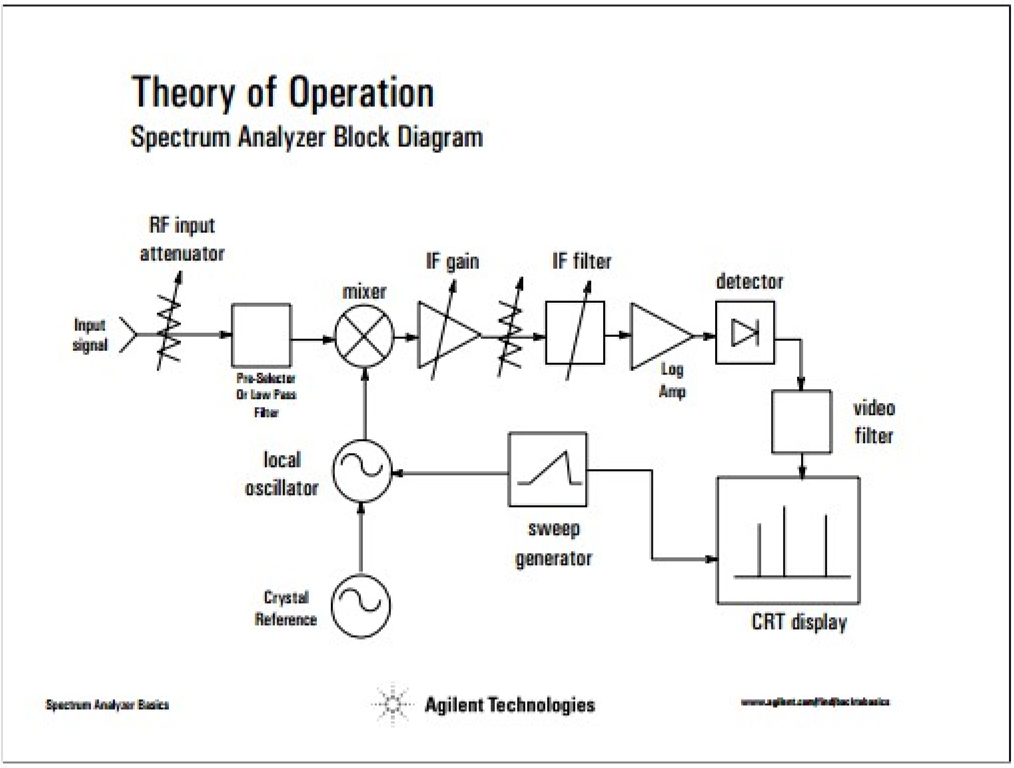


Fig 2: Spectrum Analyzer Block Diagram

2.3 Functionality:

Centre frequency and span:

In a typical spectrum analyzer there are options to set the start, stop, and center frequency. The frequency halfway between the stop and start frequencies on a spectrum analyzer display is known as the center frequency. This is the frequency that is in the middle of the display's frequency axis. Span specifies the range between the start and stop frequencies. These two parameters allow for adjustment of the display within the frequency range of the instrument to enhance visibility of the spectrum measured.

Resolution bandwidth:

The resolution bandwidth filter or RBW filter is the band pass filter in the IF path. It's the bandwidth of the RF chain before the detector (power measurement device).It determines the RF noise floor and how close two signals can be and still be resolved by the analyzer into two separate peaks. Adjusting the bandwidth of this filter allows for the discrimination of signals with closely spaced frequency components, while also changing the measured noise floor.

Video bandwidth:

The video bandwidth filter or VBW filter is the low-pass filter directly after the envelope detector. It's the bandwidth of the signal chain after the detector. Averaging or peak detection then refers to how the digital storage portion of the device records samples—it takes several samples per time step and stores only one sample, either the average of the samples or the highest one. The video bandwidth determines the capability to discriminate between two different power levels.

2.4 Radio Frequency Application:

Spectrum analyzers are widely used to measure the frequency response , noise and distortion characteristics of all kinds of radio-frequency (RF) circuitry, by comparing the input and output spectra. In telecommunications, spectrum analyzers are used to determine occupied bandwidth and track interference sources.

A spectrum analyzer interface is a device that connects to a wireless receiver or a personal computer to allow visual detection and analysis of electromagnetic signals over a defined band of frequencies.

Spectrum analyzers can also be used to assess RF shielding. RF shielding is of particular importance for the siting of a magnetic resonance imaging machine since stray RF fields would result in artefacts in an MR Image.[2]

3. Remote Instrument Control

3.1 What is Instrument Control?

Instrument control is a PC-based approach that combines programmable software and hardware connectivity for automating measurement acquisition from third-party instrumentation.

An instrument control system consists of instrumentation, connectivity hardware, and a computer with programmable software. Compared to traditional measurement systems, the combination of software, instrument drivers, and connectivity hardware is the most productive, flexible approach to automating third-party instrumentation.

3.2 Components of Instrument Control System:

The 3 main components of the Instrument Control System are:

Instrument

Hardware (Interface)

Software (Computer)

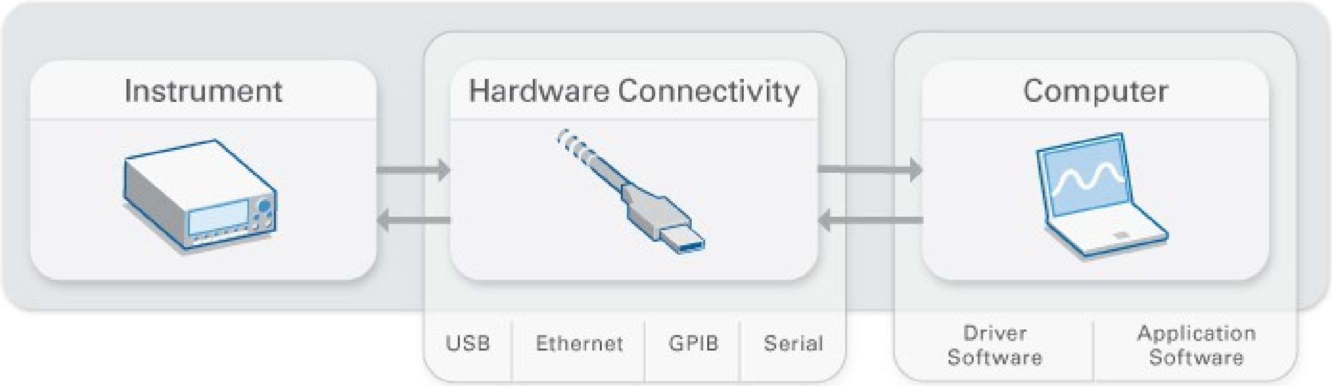


Fig 3: The components of Instrument Control System

3.2.1 Instruments:

There are two types of instruments: stand-alone and modular. Traditional instrument control applications employ stand-alone (bench top) instruments to gather specific measurement data, but as hardware technologies advance, modular instruments are becoming increasingly popular for acquiring measurement data.[3]

Stand-Alone

Although primarily designed for manual use, stand-alone instruments can also be incorporated into an automated measurement system. Using instrument control software, you can control the instrument with a PC through a communication bus such as GPIB, USB, serial, or LAN/Ethernet.

Modular

You now have the option of creating more flexible and scalable measurement systems by taking advantage of virtual instrumentation. By combining general-purpose, modular hardware with flexible, high-performance software, you can design and develop custom system solutions for your measurement needs.

3.2.2 Hardware:

Instrument control hardware connectivity is the interface that is used to connect your instrument to your PC. You rely on instrument control hardware solutions to communicate to instruments directly via stand-alone or modular buses such as GPIB, RS232, USB, Ethernet, PCI, and PXI.[3]

Stand-Alone

Stand-Alone instruments are used to communicate with rack-and-stack instruments, these include test and measurement-specific buses such as GPIB and PC-standard buses such as serial (RS232), Ethernet, and USB. You can use some stand-alone buses as a medium for other stand-alone buses such as a USB-to-GPIB converter.

Modular

Modular instruments incorporate the interface bus into the instrument itself. Modular buses include PCI, PCI Express, VXI, and PXI. You also can use these buses as a medium to add a stand-alone bus into a PC that does not have one, such as with an NI PCI-GPIB controller board.

3.2.3 Software:

A computer with programmable software controls the automation of instruments and processes, visualizes, and stores measurement data. Different types of computers are used in different types of applications. A desktop may be used in a lab for its processing power, a laptop may be used in the field for its portability, or an industrial computer may be used in a manufacturing plant for its ruggedness.[3]

Application Software

Application software facilitates the interaction between the computer and user for automating the process of acquiring, analyzing, and presenting measurement data from instruments. Instrument control software tools come in the form of either a prebuilt application with predefined functionality, or a programming environment for building applications with custom functionality. To take advantage of an application software tool that meets the needs of your current system and scales over time, choose an application development environment in which you can create custom applications.

Driver Software

Driver software provides application software the ability to interact with an instrument. You can control your instruments through direct I/O commands or by using an instrument driver. Generally, an instrument driver is the preferred approach because it simplifies communication with the instrument device by abstracting low-level programming protocols that may be specific to one instrument. Typically, instrument drivers expose an API that is used within a programming environment to build application software.

4. Virtual Instrument Software Architecture (VISA)

4.1 What is VISA?

Virtual Instrument Software Architecture, commonly known as VISA, is a widely used I/O API in the test and measurement (T&M) industry for communicating with instruments from a PC.VISA is an industry standard implemented by several T&M companies, such as Rohde & Schwarz, Agilent Technologies, Anritsu, Bustec, National Instruments, Tektronix and Kikusui.

The VISA standard includes specifications for communication with resources (usually, but not always, instruments) over T&M-specific I/O interfaces such as GPIB and VXI. There are also some specifications for T&M-specific protocols over PC-standard I/O, such as HiSLIP or VXI-11 (over TCP/IP) and USBTMC (over USB).

4.2 Advantages of VISA:

Interface Independence:

One of VISA's advantages is that it uses many of the same operations to communicate with instruments regardless of the interface type. For example, the VISA command to write an ASCII string to a message-based instrument is the same whether the instrument is Serial, GPIB, or USB. Thus, VISA provides interface independence.

Cross Platform Functionality:

VISA is also designed so that programs written using VISA function calls are easily portable from one platform to another. VISA does this by defining its own data types. This prevents problems like, for example, possible problems caused by moving from one platform to another where the size of an integer may be different.

Easy To Learn:

VISA's greatest advantage is that it is an extremely easy language to learn. VISA provides a very simple-to-use API that has bus independent functions for most of its I/O functionality. VISA provides the most commonly used functionality for instrumentation in a very compact command set, eliminating the need to learn low level communication protocols for multiple bus types.

4.3 A Typical VISA Application:

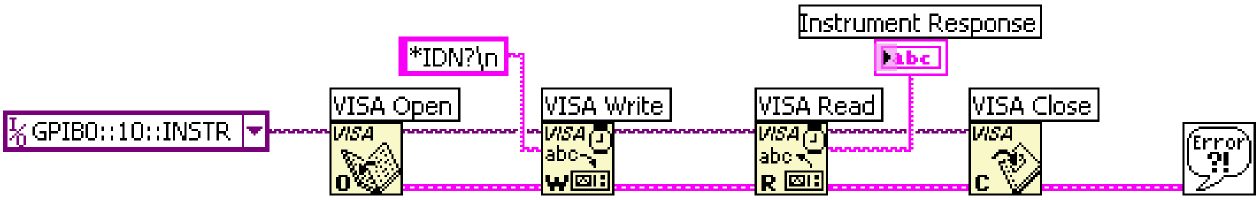


Fig 4: Control flow of a typical VISA application

A typical VISA application would go through the following steps.

1) Open a Session to a given Resource.
2) Do any configuration on the given resource (setting baud rates, termination character, etc...).
3) Perform writes and reads to the device.
4) Close the Session to the Resource.
5) Handle any errors that may have occurred.[4]

5. SCPI

5.1 What is SCPI?

The Standard Commands for Programmable Instruments (SCPI) defines a standard for syntax and commands to use in controlling programmable test and measurement devices.

The standard specifies a common syntax, command structure, and data formats, to be used with all instruments. It introduced generic commands (such as `CONFigure` and `MEASure`), which could be used with any instrument. These commands are grouped into subsystems. SCPI also defines several classes of instruments. Instrument classes specify which subsystems they implement, as well as any instrument-specific features.

The physical communications link is not defined by SCPI. While originally created for IEEE-488 (GPIB), it can also be used with RS-232, Ethernet, USB, VXIbus, etc.[5]

5.2 Command Structure:

SCPI commands to an instrument may either perform a set operation (e.g. switching a power supply on) or a query operation (e.g. reading a voltage). Queries are issued to an instrument by appending a question-mark to the end of a command. Some commands can be used for both setting and querying an instrument.

For example, the data-acquisition mode of an instrument could be set by using the `ACQuire:MODe` command or it could be queried by using the `ACQuire:MODe?` command.

Some commands can both set and query an instrument at once.

For example, the `*CAL?` command runs a self-calibration routine on some equipment, and then returns the results of the calibration.

SCPI commands follow a tree-like structure. Related commands are grouped in the branches of the tree. Each keyword in the command is called a node. The first node is called the root node.

Commands are formed by traversing the tree downwards, placing colon (:) delimiters between nodes, e.g. the following are all legal SCPI commands from the tree shown:

```
MEASure:VOLTage:DC?

MEASure:VOLTage:DC:RATio?

MEASure:CURRent:AC?
```

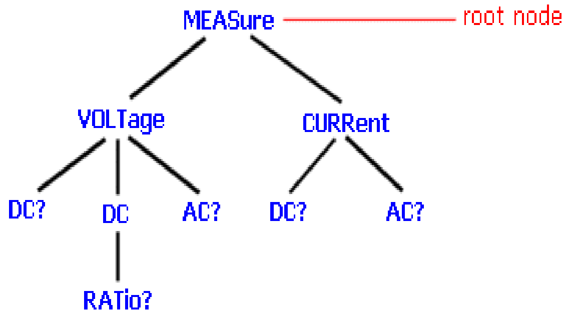


Fig 5: Command Tree

6. Program Flow

The program was compiled using Visual C++ 2010 compiler and incorporates the required VISA libraries and header files. The libraries used during the course of the project were from the Agilent I/O Library Suites available at the Agilent home site.

Basic Setup:

The basic set up involved an **Agilent N9310A Signal Generator** and **Rohde & Schwarz FSV Signal Analyzer**, with the signal generated passed to the analyser through a probe connector.

The interface used with the Generator was **USB**, while the analyser was controlled using an **Ethernet cable**.

The integrated program allows control of both signal generator as well as signal analyser. The program is divided into various subroutines, each pertaining to specific functions.

6.1 device_connect()

This function is used to **establish a connection between the software and the instrument** which takes the **port address of the instrument** as argument.
This is done using the VISA command **viOpen()** which opens a session for the instrument using its port address.
The function will return **-1** if the connection fails.

6.2 class signalgenerator

This class involves the complete operation of the signal generator. It combines the different functions and variables needed for the various modes of operation of the generator.
The various subroutines are:

6.2.1 gencw()

This function is used for the generation of a **carrier wave**.

It takes the **Frequency, Amplitude and RF Output State** from the user and generates the appropriate wave.

The mode of entry for this function can be **through keyboard or through data file** which is specified in the configuration file pre-process.

SCPI Commands used:

FREQ:CW	- To set the frequency
AMPL:CW	- To set the amplitude
MOD:STAT	- To set the RF Output state to ON or OFF

6.2.2 rf_gen()

This function is used to generate a **RF step sweep signal at constant power level**.

It takes the **Power level(Amplitude),Frequency Range(Start/Stop),number of sweep points, Dwell time and the type of sweep(Single/Continuous)** from the user and generates the sweep signal.

The mode of entry for this function can be **through keyboard or through data files** which is specified in the configuration file preprocess.

SCPI Commands used:

AMPL:CW	-	To set the Amplitude
SWE:RF:STAR	-	To set the start frequency for the sweep
SWE:RF:STOP	-	To set the stop frequency for the sweep
SWE:STEP:POIN	-	To set the number of sweep points
SWE:STEP:DWEL	-	To set the dwell time
SWE:REP	-	To set the type of sweep (SING|CONT)
SWE:RF:STAT ON	-	To Set RF output ON

6.2.3 amp_gen()

This function is used to generate **an Amplitude swept signal at constant frequency**.

It takes the **Frequency, Amplitude Range(Start/Stop),number of sweep points, Dwell time and the type of sweep(Single/Continuous)** from the user and generates the sweep signal.

The mode of entry for this function can be **through keyboard or through data files** which is specified in the configuration file preprocess.

SCPI Commands used:

FREQ:CW - To set the Frequency

SWE:AMPL:STAR -To set the start power level for the sweep

SWE:AMPL:STOP - To set the stop power level for the sweep

SWE:STEP:POIN - To set the number of sweep points

SWE:STEP:DWEL -To set the dwell time

SWE:REP - To set the type of sweep (SING-single|CONT-continuous)

SWE:AMPL:STAT ON - To Set RF output ON

6.2.4 lf_gen()

This function is used to generate **an LF swept signal at default amplitude**.

It takes the **Frequency Range(Start/Stop),number of sweep points, Dwell time and the type of sweep(Single/Continuous)** from the user and generates the sweep signal.

The mode of entry for this function can be **through keyboard or through data files** which is specified in the configuration file preprocess.

SCPI Commands used:

SWE:LF:STAR - To set the start frequency for the LF sweep

SWE:LF:STOP - To set the stop frequency for the LF sweep

SWE:STEP:POIN - To set the number of sweep points

SWE:STEP:DWEL - To set the dwell time

SWE:REP - To set the type of sweep (SING-single|CONT-continuous)

SWE:LF:STAT ON - To Set LF output ON

6.2.5 signal_generator()

This is an interface function whereupon the user can choose **the type of signal to be generated** and also the **type of external or internal signal required.**

6.3 class spectrumanalyser

This class involves the complete operation of the spectrum analyser which receives the signal from the generator. It combines the different functions and variables needed for the various modes of operation of the analyzer.

The spectrum analyser also involves returning the measured values and writing them onto a data file.

The various subroutines are:

6.3.1 spectrum_analyser_defaultsettings()

This function is used to set the **default settings** of the signal analyser.

The default settings include **Center frequency, Frequency span, Resolution bandwidth, Video bandwidth, Sweep time, Type of Sweep(Continuous/Single).**

SCPI Commands Used:

FREQUENCY:CENTER - To set the Center Frequency

FREQ:SPAN - To set the Frequency Span

BAND:RES - To set the Resolution bandwidth

BAND:VID - To set the Video bandwidth

SWE:TIME - To set the Sweep time

INIT:CONT - To set type of Sweep(ON-continuous/OFF-single)

6.3.2 poweratfreq()

This function is used to **measure the power level at a specific frequency** of the generated signal.

It takes the required **Frequency at which power is to be measured** from the user. The function uses a **marker** to measure the power.

Multiple markers can also be used.

The mode of entry for this function can be **through keyboard or through data file** which is specified in the configuration file preprocess.

The measured power level is written onto an **output file**.

SCPI Commands Used:

CALC:MARK<n>:X - To place the marker at the required Frequency

CALC:MARK<n>:Y? - To query the Power level at marker

6.3.3 channelpower()

This function is used to **measure the integrated power over a specified channel** of the generated signal.

It takes the **Bandwidth of the channel to be measured** from the user.

The mode of entry for this function can be **through keyboard or through data file** which is specified in the configuration file preprocess.

The measured integrated power level is written onto an **output file**.

SCPI Commands Used:

POW:ACH:BWID:CHAN - To set the Channel bandwidth

POW:HSP OFF - To set Integrated Bandwidth Method

CALC:MARK:FUNC:POW:SEL CPOW - To measure the Power over Channel

CALC:MARK:FUNC:POW:RES? CPOW - To query the measure Power

6.3.4 tracepoints()

This function is used to get **the trace points** of the generated signal.

It does not take any inputs from the user and returns **a list of the frequency and power level co-ordinates at all the sweep points present**.

The list is written onto a separate output file.

The return format has to be further manipulated using algebraic and string functions to get the trace points in ASCII format.

SCPI Commands used:

:INIT:CONT 0- Set to single measurement mode

:FORM REAL,32 - Set the output format to binary 32 bit format

:TRAC:DATA? TRACE1 - Query trace points

6.3.5 spectrum_analyser()

This is an interface function whereupon the **spectrum_analyser_defaultsettings()** is called and the user can enter the required default settings. The user can also choose **the operation mode** of the analyser for the desired results.

The mode of entry for this function can be **through keyboard or through data file** which is specified in the configuration file preprocess.

6.4 err()

This function is the generalized error function for both the instruments used. It is therefore present under the generator and analyser sub-routines.

It is a query function which returns the **SCPI error** projected by the instrument.

SCPI commands used:

SYST:ERR? - Queries the SCPI error

6.5 closesession()

This functions **closes the current VISA session** using the **viClose()**.

7. Control Flowchart

7.1 Configuration:

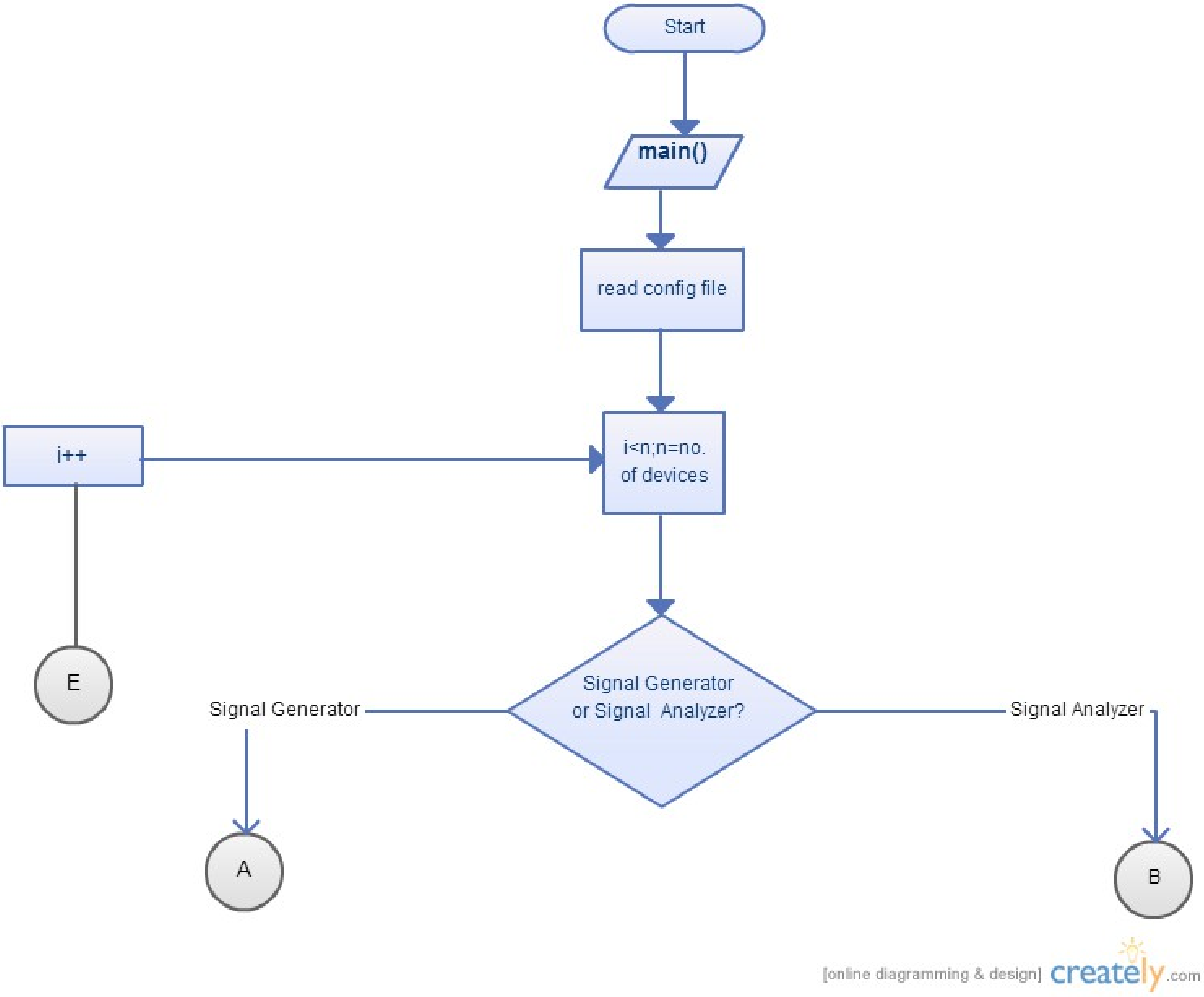


Fig 6: Configuration of Program

7.2 Signal Generator:

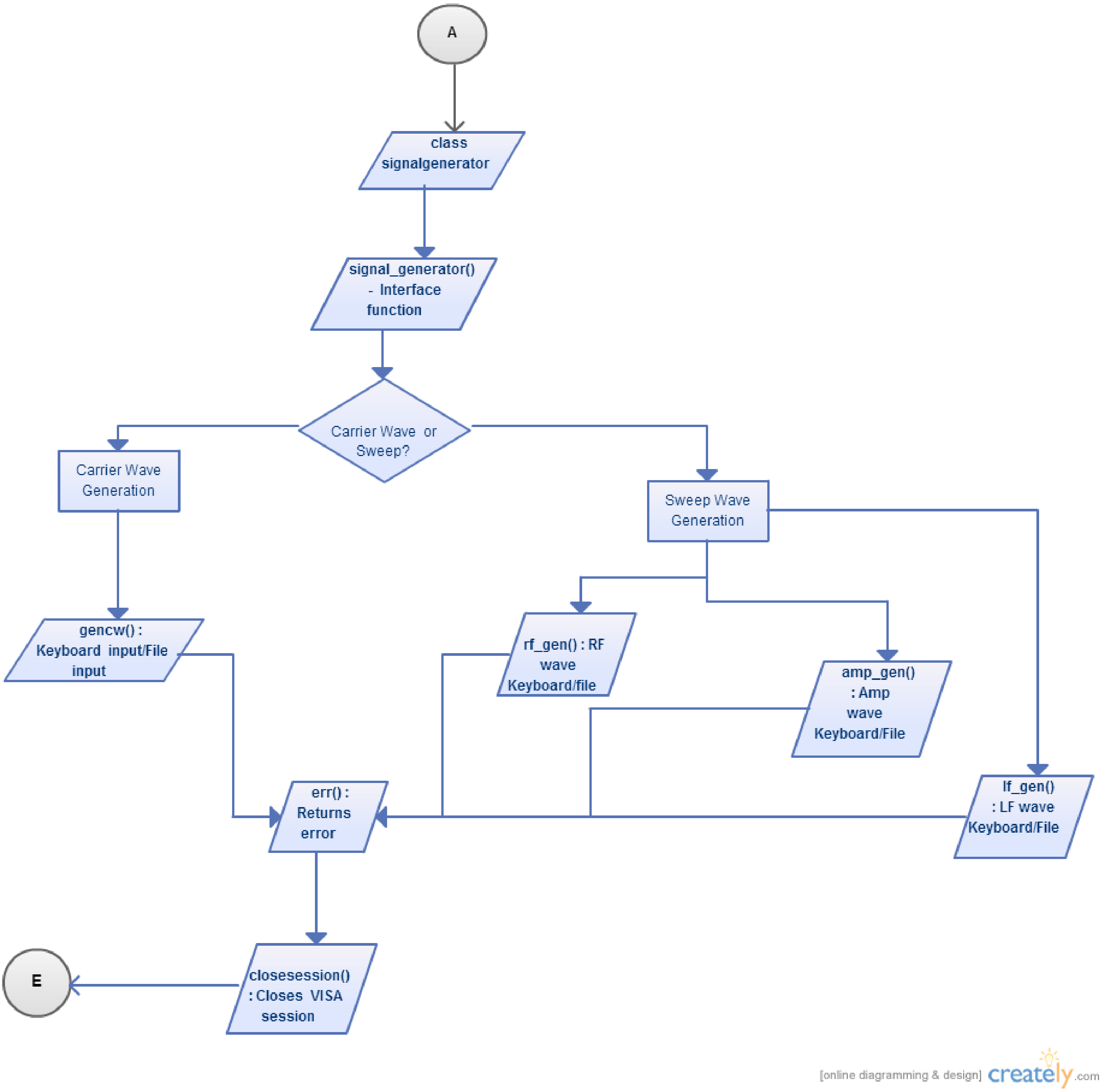


Fig 7: Signal Generator subroutine flowchart

7.3 Signal Analyzer

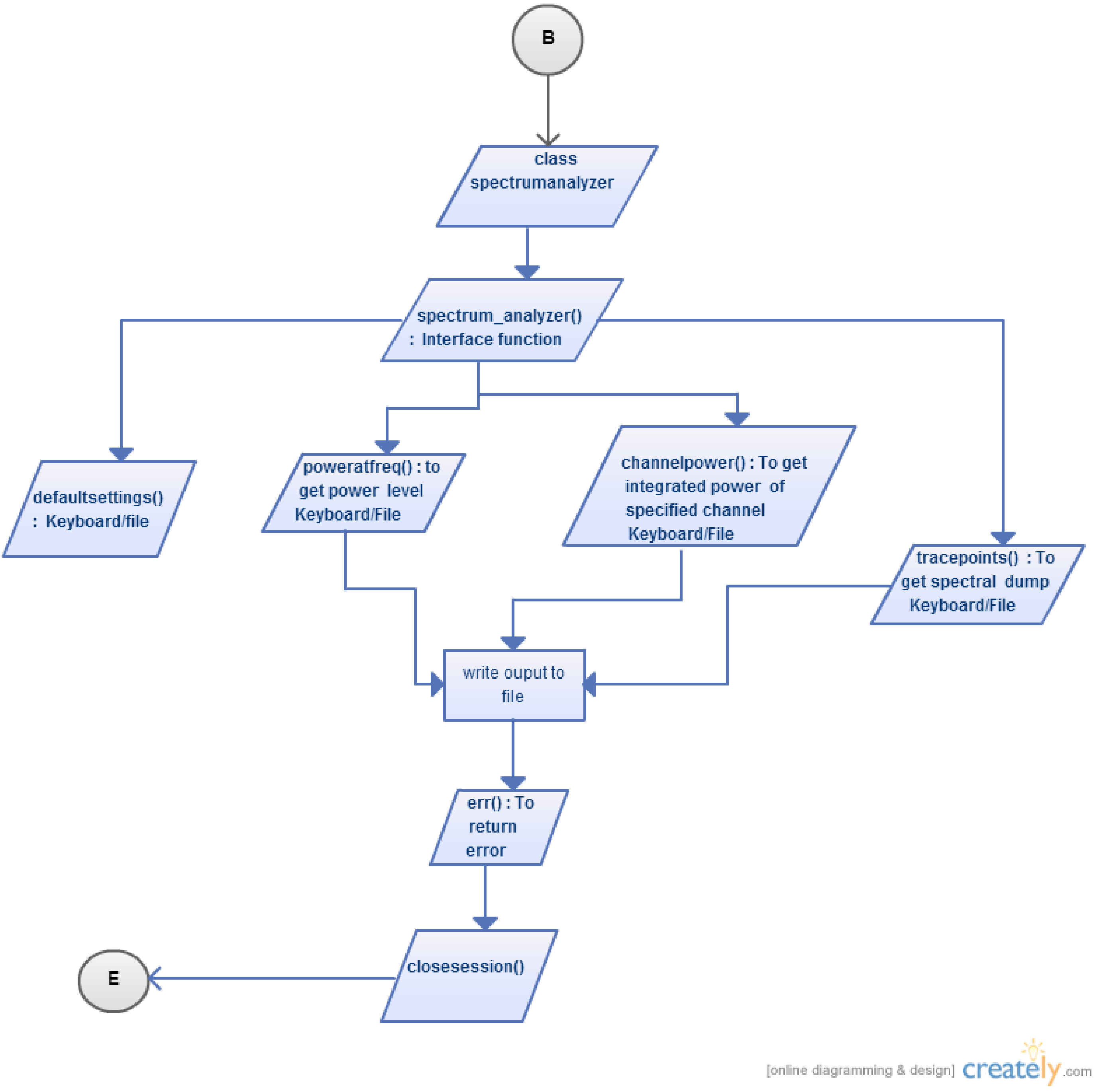


Fig 8: Signal Analyzer subroutine flowchart

8. Input/Output Mechanism

8.1 Input

The **user input** for the program to initialize the signal characteristics in the generator as well as setting the default settings in the analyser is done through 2 methods:

a) Through keyboard entry

b) Through data files

Keyboard entry functions provide a direct method of entry. They different keyboard entry functions in the program are:

keyboard_gencw() , keyboard_rf_gen(), keyboard_amp_gen(), keyboard_lf_gen()

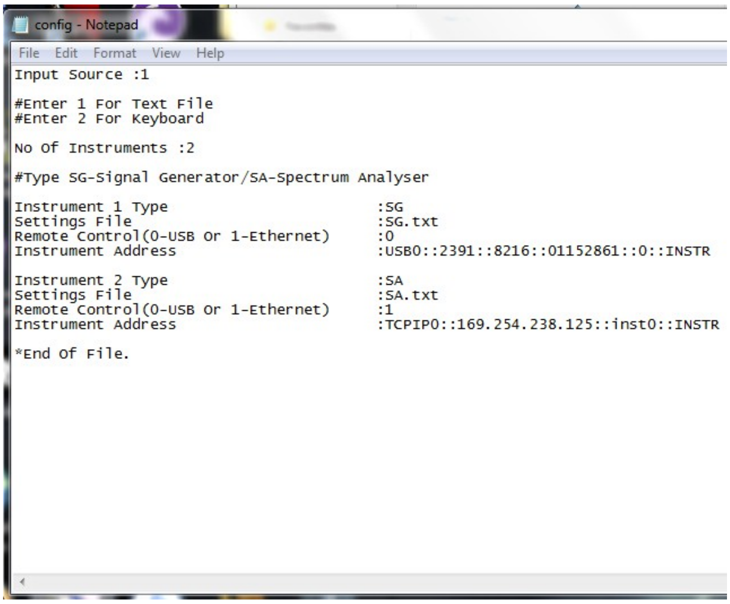
keyboard_poweratfreq(), keyboard_channelpower()

Data file entry provides an organized method of entry and can be used for **automated control** of both the instruments. The data files used is in a set format where the values of the entry variable only need be changed.

The device and program establishes its initial connection through **the configuration file**, which contains complete information for the instruments involved including their **port addresses**. It also specifies **the type of entry(keyboard/file)** which also can be altered according to users need.

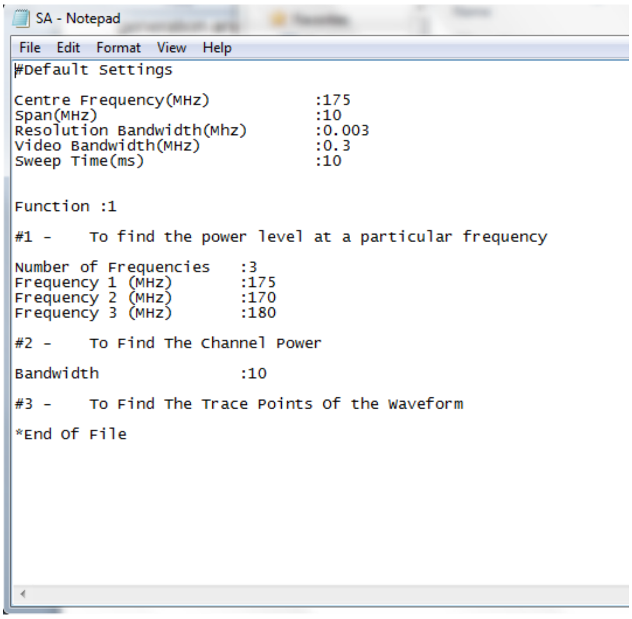
The other data files contain information regarding the entry values for the required signal generation and the default settings as well as the operational characteristics of the spectrum analyser.

The function used for reading these data files is **getfiledata()**.



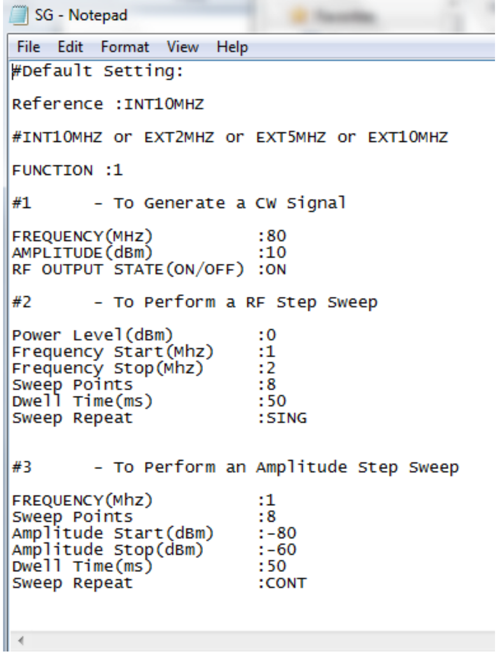
```
Input Source :1

#Enter 1 For Text File
#Enter 2 For Keyboard

No Of Instruments :2

#Type SG-Signal Generator/SA-Spectrum Analyser

Instrument 1 Type                         :SG
Settings File                             :SG.txt
Remote Control(0-USB Or 1-Ethernet)       :0
Instrument Address                        :USB0::2391::8216::01152861::0::INSTR

Instrument 2 Type                         :SA
Settings File                             :SA.txt
Remote Control(0-USB Or 1-Ethernet)       :1
Instrument Address                        :TCPIP0::169.254.238.125::inst0::INSTR

*End Of File.
```

Fig 9: Configuration file



```
#Default Settings

Centre Frequency(MHz)           :175
Span(MHz)                       :10
Resolution Bandwidth(Mhz)       :0.003
Video Bandwidth(MHz)            :0.3
Sweep Time(ms)                  :10


Function :1

#1 -    To find the power level at a particular frequency

Number of Frequencies   :3
Frequency 1 (MHz)       :175
Frequency 2 (MHz)       :170
Frequency 3 (MHz)       :180

#2 -    To Find The Channel Power

Bandwidth               :10

#3 -    To Find The Trace Points Of the Waveform

*End Of File
```

Fig 10: Analyzer input file



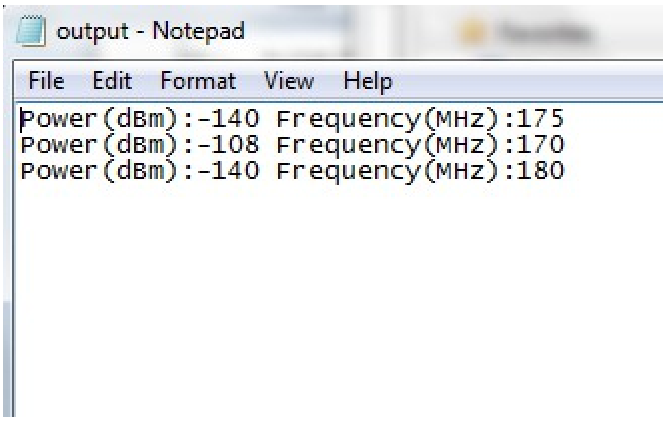
```
#Default Setting:

Reference :INT10MHZ

#INT10MHZ or EXT2MHZ or EXT5MHZ or EXT10MHZ

FUNCTION :1

#1      - To Generate a CW Signal

FREQUENCY(MHz)           :80
AMPLITUDE(dBm)           :10
RF OUTPUT STATE(ON/OFF) :ON

#2      - To Perform a RF Step Sweep

Power Level(dBm)         :0
Frequency Start(Mhz)     :1
Frequency Stop(Mhz)      :2
Sweep Points             :8
Dwell Time(ms)           :50
Sweep Repeat             :SING


#3      - To Perform an Amplitude Step Sweep

FREQUENCY(Mhz)           :1
Sweep Points             :8
Amplitude Start(dBm)     :-80
Amplitude Stop(dBm)      :-60
Dwell Time(ms)           :50
Sweep Repeat             :CONT
```

Fig 11: Generator input file

8.2 Output

The output mechanism in the program is designed **only for the signal analyser**. The measured values in the various operation modes of the analyser are written onto to **an output file**.

The function used to write the output file is **writefiledata()**.



```
Power(dBm):-140 Frequency(MHz):175
Power(dBm):-108 Frequency(MHz):170
Power(dBm):-140 Frequency(MHz):180
```

Fig 12: Output for poweratfreq()

9. Conclusion

The program can effectively control signal analyzers and generators of different make. During the course of the project, the control program was tested on **Agilent N9310A Signal Generator, R&S FSV Signal Analyzer, R&S FSL Signal Analyzer, Agilent N9320B Signal analyser**.

Furthermore, the program was successfully able to generate signals of different characteristics through the generators. The measured values of the analyzers were found to be accurate and documentation of the return values was systematically done. The cross platform adaptability of the program was therefore verified.

The use of VISA and SCPI for the remote control of RF generators and analyzers was studied in detail and appropriately put into application.

The program was built and compiled on the Visual C++ 2010 compiler, though the program can be run through any other compiler. Modifications to the program will involve separate functions, such that it doesn’t interfere with the current operation. Hence the program presents efficiency and easy accessibility.

The only discernible limitation of the program lies in the **unavailability of VISA drivers and library files for LINUX operating system**, hence inhibiting the mass use of the program.

The other limitation is the **class system of the SCPI commands** for instruments of different make, which therefore requires different commands to be sent for instrument of different make for the same operation

References

[1] “RF Radio Frequency Signal Generator” Web.22 June 2013. http://www.radio-electronics.com/info/t_and_m/generators/radio-frequency-rf-signal-generator.php

[2] “Theory Of Operation/Spectrum Analyzer” Web.22 June 2013. http://en.wikipedia.org/wiki/Spectrum_analyzer

[3] “What is Instrument Control?” Web.22 June 2013.http://www.ni.com/instrument-control/what-is/

[4] “VISA Tutorial” Web.22 June 2013. http://www.ni.com/support/visa/vintro.pdf

[5] “SCPI” Web.22 June 2013. http://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments