DIGITAL CONTROL OF ELECTRONIC INSTRUMENTS OVER SCPI

Abstract

This paper presents the control of a programmable instruments such as digital signal generator, oscilloscope etc., through the interfaces like USB, LAN, Ethernet using Standard Programmable Instrument Commands (SCPI) as the standard. This can be utilized in industries where human instrument control is absolutely not possible due to their complexity, and the operation scenario must be repeated numerous times.

This study makes the suggestion that various instruments can be operated by means of a set of cross-platform software instructions. This software system is mostly utilized for instrument control, transmission with the SCPI instruction set and achieves handle instruments.

Keywords: programmable instruments; standard commands for programmable instrument

Authors

B. Sindhu
BE Studetnt
Matrusri Engineering College
Hyderabad, Telangana , India.

S. Kalyani
BE Studetnt
Matrusri Engineering College
Hyderabad, Telangana ,India.

P. Chandan
BE Studetnt
Matrusri Engineering College
Hyderabad, Telangana ,India.

M. Naresh
Assitant Professor
Department of ECE
Matrusri Engineering College
Hyderabad, Telangana ,India.

I. INTRODUCTION

With advancements in science and technology, communication technology has changed dramatically; instruments may now be controlled remotely via a computer interface rather than being controlled manually. Control and measurement of programmable instruments is performed by computer software, which is used to considerably improve operational convenience.

When manual control of instruments is not practical scenario and complex, especially whileutilizinginstrumentswithmalfunctioningbuttonsordisplaysoperatingthemmanuallycanals obe dangerous to users and in the instances where the operating scenario must be repeated numerous times, automation stands as the best option.

Standard interfaces such as GPIB, RS-232, LAN are available on most of the programmable instruments, moreover most industrial instruments are SCPI compatible. Different functions can be performed by the same complicated system using these standard interfaces with equipment from different vendors. However, hardware compatibility alone is insufficient; a programming standard that is compatible with many manufacturers and instrument functionalities is required.

The instrument and the host side are able to communicate with each other using SCPI (Standard Commands for Programmable Instruments), which is used in this study as both the host side and instrument side of the communication instructions. Due to the fact that the IEEE 488.2 standard does not specify the physical layer implementation technique, SCPI can be utilized with RS-232, RS-422, Ethernet, USB, and a variety of other hardware buses.

II. HISTORICAL REVIEW

The rapid advancement of computer technology has aided the development of measurement and control systems for more than 30 years. Its purpose was to develop faster, more automated, and integrated technologies that would give users more precise information and better control over the objects they were examining. Computer makers and software providers provided a variety of concepts, techniques, strategies, and procedures. Customized, expensive solutions were forced off the market due to severe competition. Although many of them frequently used commercial standards were translated into international standards, only a few were chosen as a foundation for measurement extensions by the instrumentation industry.

IEEE 488 was the first global standard for measurement and control systems, developed by the Institute of Electrical and Electronics Engineers (IEEE) in 1975. The Hewlett-Packard Interface Bus, a digital bus built by Hewlett-Packard, was specified (HP-IB). In 1975, an IEEE committee gave it a new number and dubbed it General Purpose Interface Bus (GPIB).

AlthoughtheIEEE488standardensuresthatmessagesarereliablycommunicatedbetween two or more devices in a system, it does not guarantee that each device will read or create all necessary responses to all messages provided to it. In 1987, the IEEE488 standard was updated to ANSI/IEEE Std. 488.1-1987 — IEEE Standard Digital Interface for

Programmable Instrumentation as ANSI/IEEE Std. 488.1-1987 — IEEE Standard Digital Interface for Programmable Instrumentation. In the same year, the IEEE 488.2 standard was released. The IEEE 488.2 standard was developed in response to a need for a standardized message format for device communication. It's an IEEE 488.1 add-on that gives the device artificial intelligence.

III. SCPI

The IEEE 488.2 standard, specifies Standard Commands for Programmable Instruments (SCPI) for communication between the programmable instrument and the controller. As SCPI messages can be sent or received as ASCII text using any programming language or development environment, including C, C++, Java, Python, Matlab, LabVIEW, they are not dependent on specific hardware and can be transmitted over any communication interface. SCPI modified IEEE 488.2 syntax, defined command sets based on a common model of measuring equipment, and determined command semantics. Despite the fact that the

SCPI Specification does not explicitly indicate that it defines command semantics; it does address the semantically meaning of messages on several occasions. SCPI is a software standard that is not dependent on the communication interface, unlike IEEE488, which is. RS232, RS482, Ethernet, GPIB, and LAN can all be used to send and receive SCPI messages. In VXI systems, SCPI has gained a lot of popularity, especially for message-based devices. SCPI has the unusual advantage of being a totally textual set of commands that can be implemented in any programming language and on any computer platform.

The commands for programming specific functionalities of a certain instrument subsystem begin with the subsystem's name. It is sufficient to write simply the four uppercase letters of the syntactic terms in a message.

IV. METHODOLOGY

Most lab equipment has a digital interface, such as USB or Serial, through which commands can be provided to control the equipment, just as if one is pushing buttons on the front panel.Consider a power supply.

One could use a command sequence to:

- set the output voltage to5V
- set the current limit to 1Amp
- Enable the output
- read-back the actual current being supplied

A command can do any action that we, the operator, can perform. Programmer can automate tasks by writing a script that sequences these commands in the correct order. A script can also coordinate various operations across numerous devices; for example, the same script can be used to control a power supply and read an oscilloscope. This article demonstrates how to automate a test using a power source and a multi-meter. We'll connect everything, adjust the voltages, take measurements, and even record some data. The commands we use are industry standard; they're the same or incredibly comparable across brands so can be used for various types of devices.

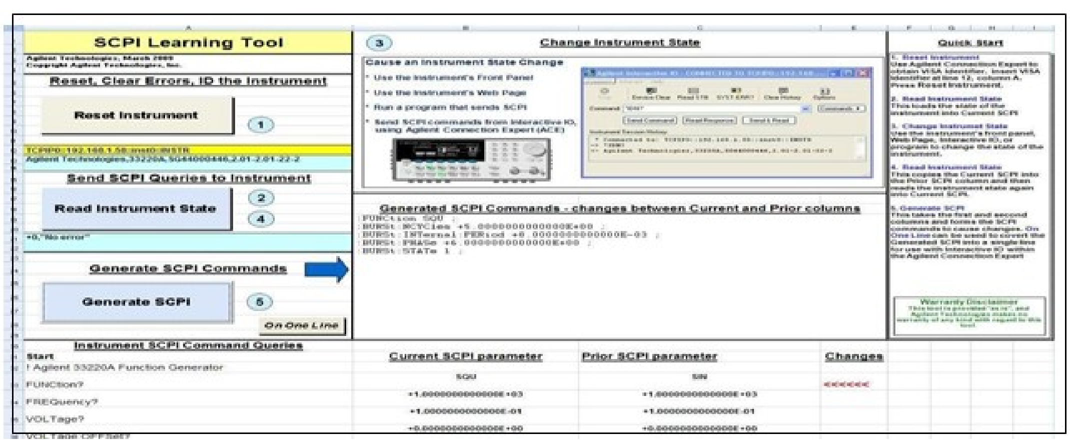
Power the equipment and connect it to computer via USB as shown in figure below. The instrument and host can be connected via a USB to USB interface or an RS232 to USB interface. Control and measure the instrument before returning to the host. In order to get control over the interface, it does parameter analysis and calculations.

V. SOFTWARE SYSTEM DESIGN

By employing the SCPI instruction set for communication between the instrument and the host, SCPI provides a standard syntax and command for managing programmable test measurement devices; the SCPI command [3-5] is an ASCII string passed in through the physical transport layer. A command is made up of a list of keywords, some of which are associated with parameters. Because its instructions are provided in an easy-to-understand format, the SCPI instruction set is also highly convenient to use.

To return the instrument to a functional state, reset or switch on the power. Send the instrument the results of each extracted query.

Modify the instrument's state, including the front panel, driver call, and Web interface.



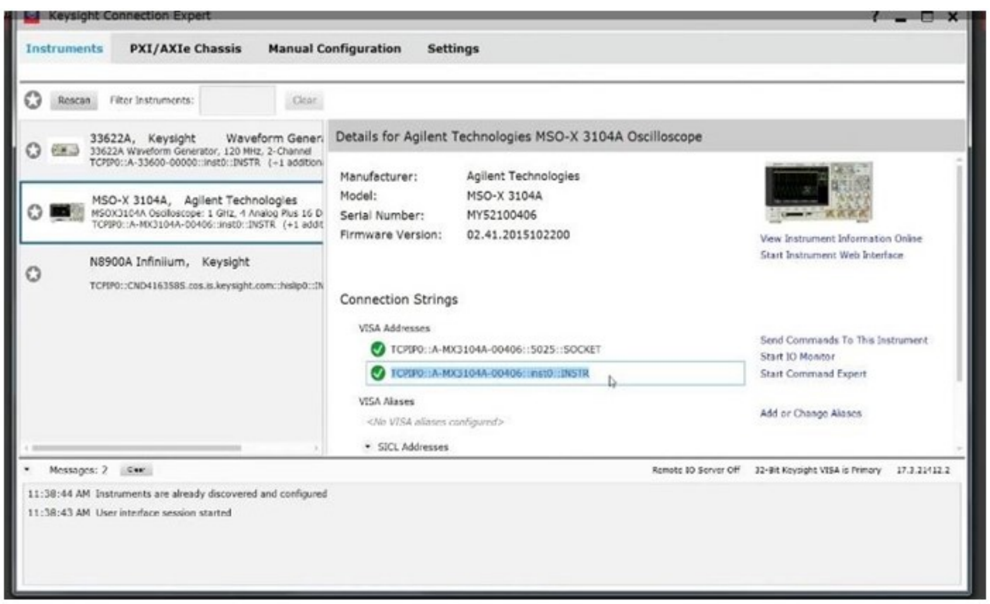
The tool's single page screen is depicted in the diagram below. It should be emphasized that the various tasks are performed using numerical sequences and buttons.

1. **Modes:** The SCPI Learning Process is followed in these sequences. All we have to do now is use the NI Measurement Automation Explorer to find the instrument's VISA string and place it in the cell with sequence. Then press Reset Instrument and Read Instrument State to return to the previous state. After making modifications to the instrument, press the Read Instrument State button again, and then the Generate SCPI button to see the SCPI commands and parameters needed to change states. After viewing how things change and all of the required changes, the SCPI Reference Handbook for the instrument can be used to further comprehend the individual commands

    In these sequences, the SCPI Learning Process is used. All that remains is for us to utilize the NI Measurement Automation Explorer to locate the instrument's VISA string and insert it into the cell with sequence. Then, to return to the former state, press Reset Instrument and Read Instrument State. After you've finished making changes to the instrument, press the Read Instrument State button again, then the Generate SCPI button to see the SCPI commands and parameters required to change states. The SCPI Reference Handbook for the instrument can be used to further understand the different commands after watching how things change and all of the required changes.

    The instrument side's query commands are located in the lower left corner. All we have to do now is fill in this column with all of the valid Configure-Query commands, and the program will take care of the rest.

2. **Results**

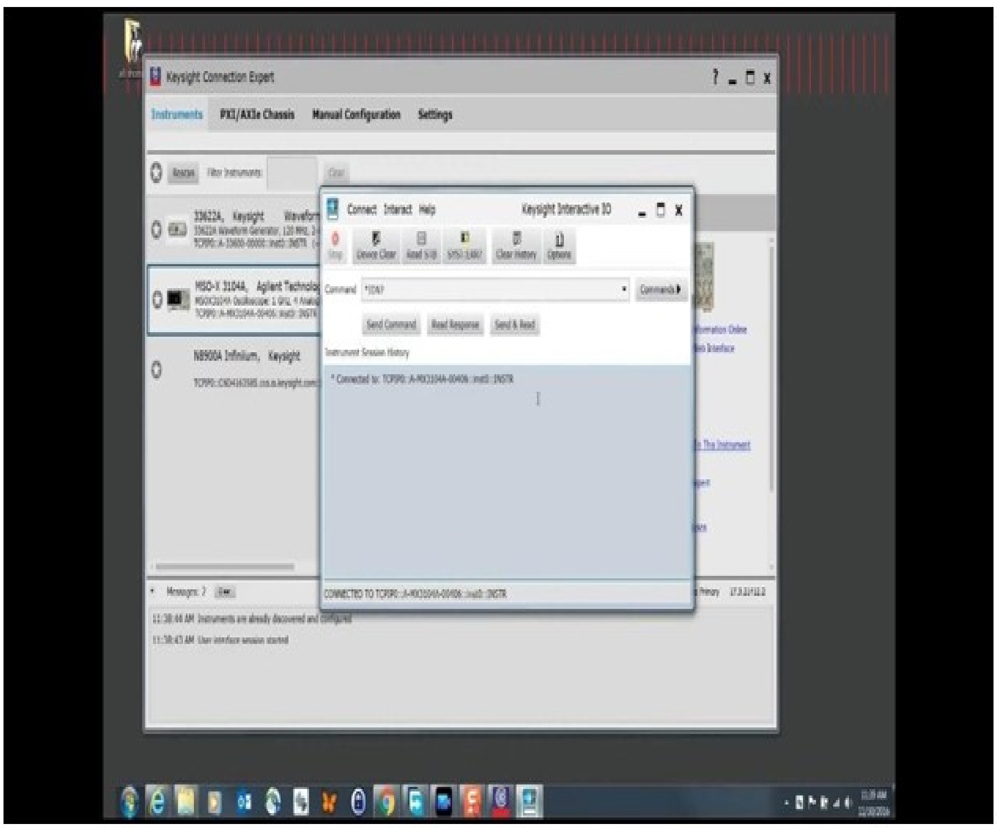


Interconnecting the Electronic Instrument (Oscilloscope, Wave Generator) and Controlling Device (Laptop)

After the connection is established then command window opens where SCPI commands can be sent. The command *IDN? is employed to allow the instrument to identify itself.

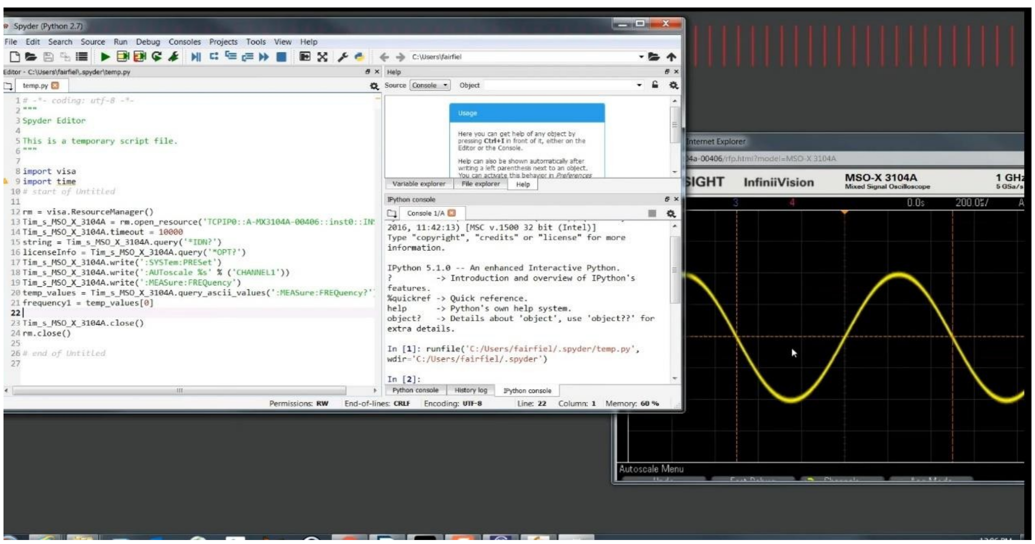
To see the working of Instrument by sending commands, Remote Front Panel of Oscilloscope is created virtually.

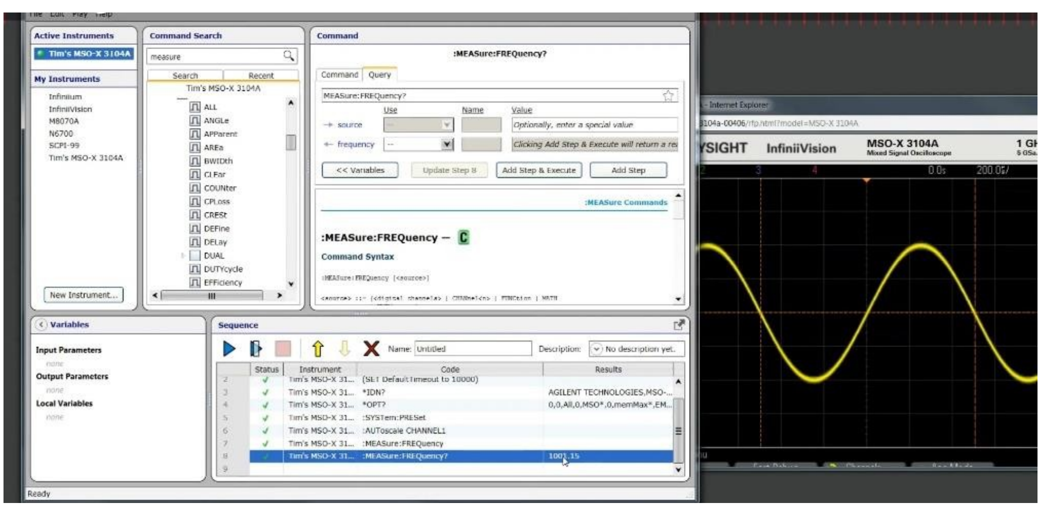
By clicking on Web Browser in preceding window we get Actual front panel where we run the commands.



To create (connect to real instrument) if real instrument is not availablewe can simulate the instrument connection.

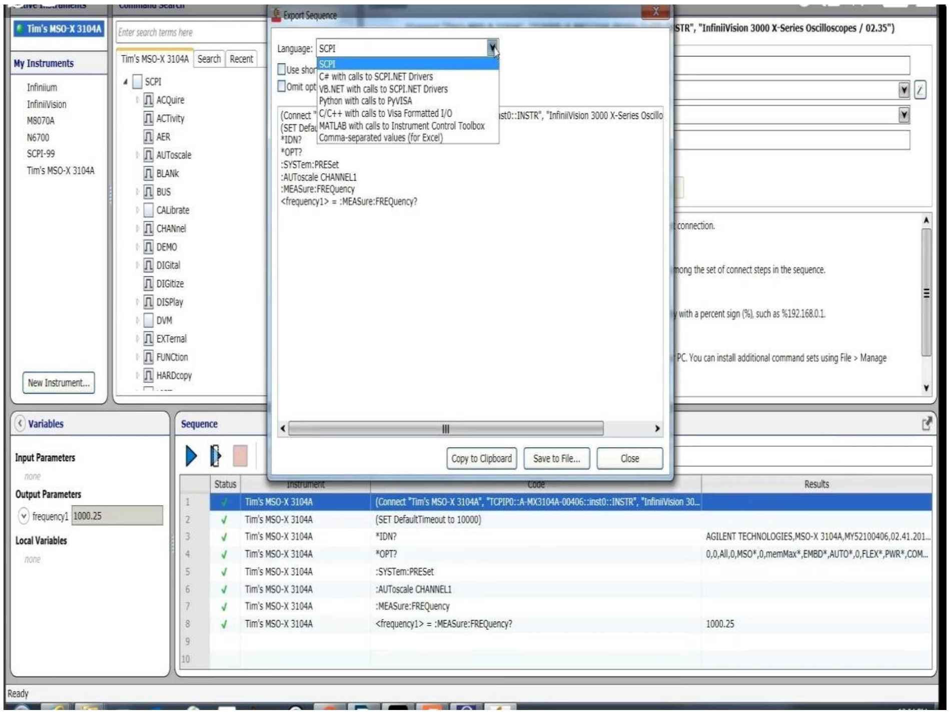
Source is selected (Here Channel 1) AUTo scale command: Sets the Scale



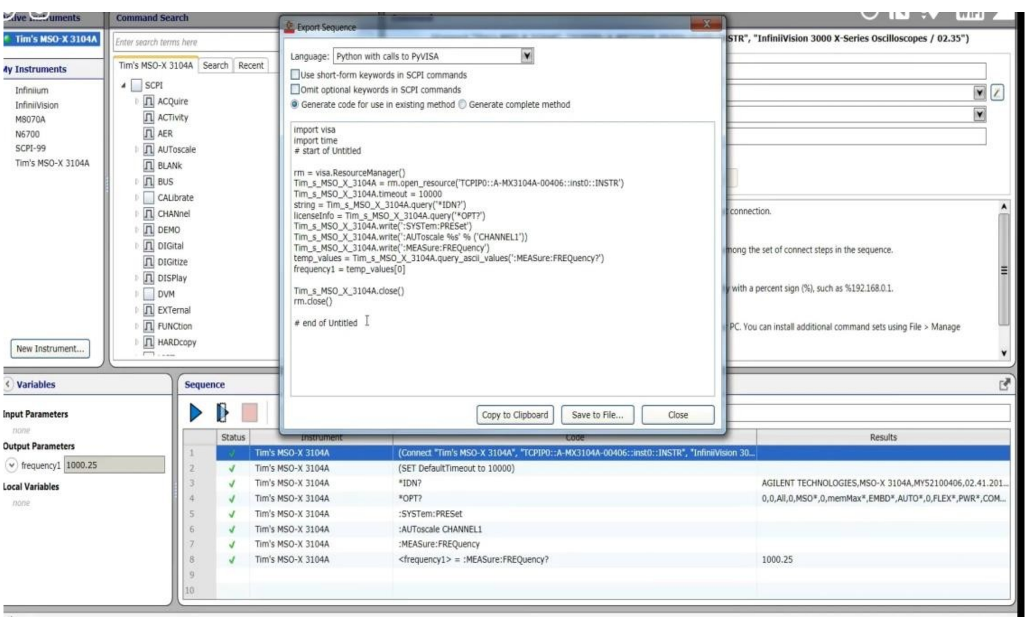


MEASure: FREQuency? Command is used to measure the frequency of the signal. FREQUENCY is outputted here.

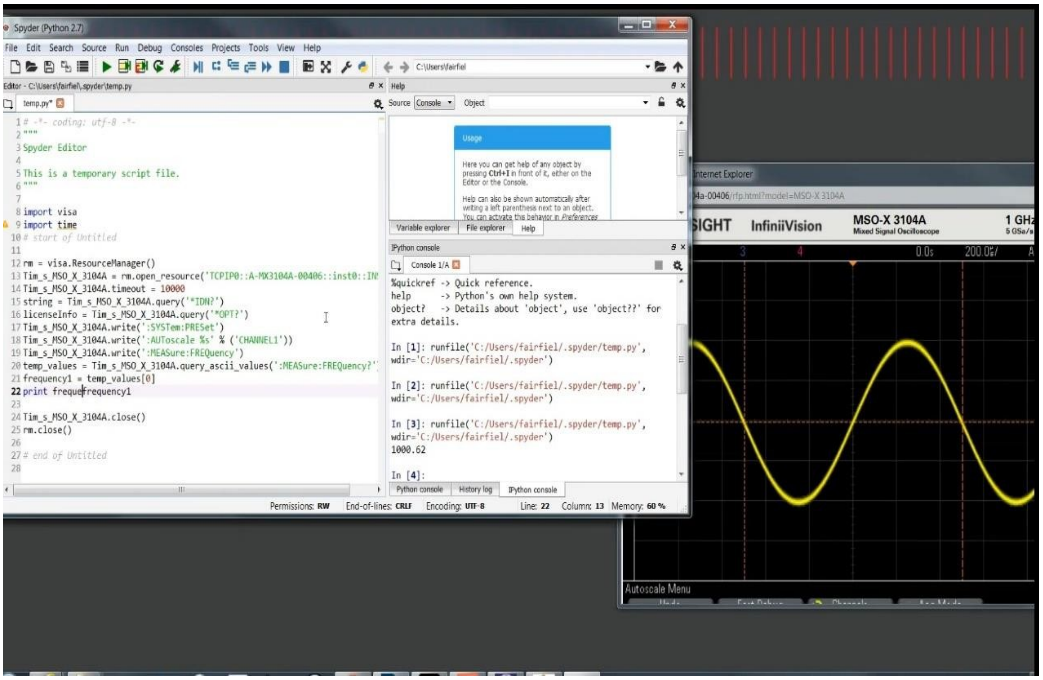
Python code to run SCPI commands using Anaconda (spyder).



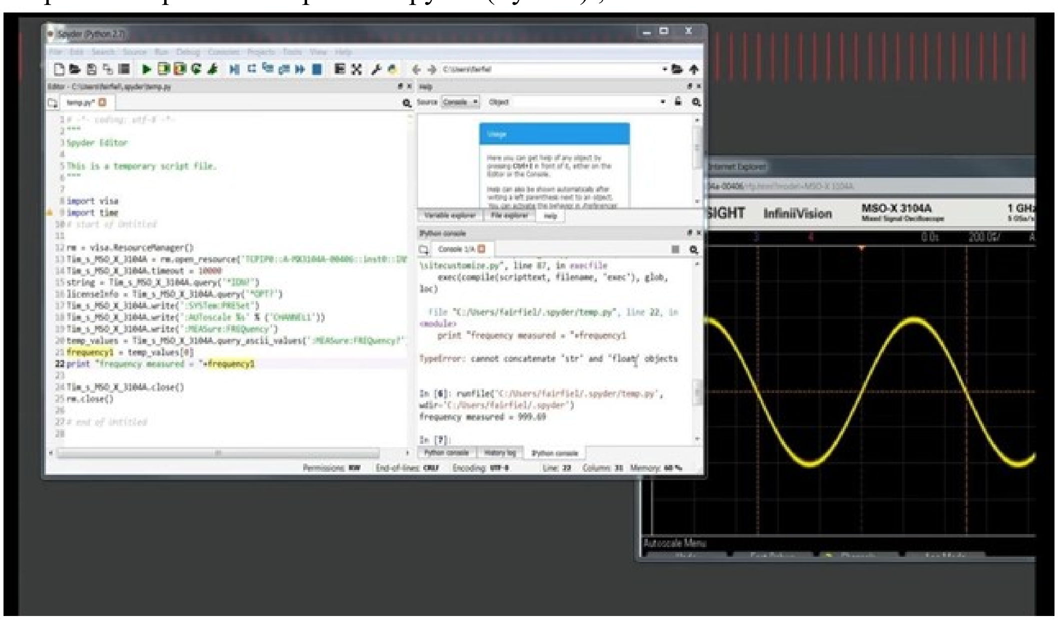
To send SCPI commands in a sequence various programming languages can be used as shown above.



Exporting the sequence of commands in python programming language.



Exported sequence is copied to Spyder (Python) , run the code and see the results.



Frequency is printed by using "MEASure:FREQency?" Command and frequency is outputted on console window by running the code.

VI. CONCLUSION

This research successfully suggested building a software interface with the SCPI command set to control two different brands of power sources. The study's usage of the software interface and SCPI command set can spare users from having to repeatedly learn new instrument interfaces and can enable a single collection of software to control measures with diverse manufactures and types, greatly decreasing development work.

REFRENCES

[1] B. A. Balaji, S. Sasikumar and K. Ramesh, "SCPI based integrated test and measurement environment using LabVIEW", IOP Conference Series: Materials Science and Engineering, vol. 1045, no. 1, pp. 012036, February2021
[2] National Instruments Virtual Instrument Software Architecture (VISA), [online] Available: https://www.ni.com/visa/.
[3] L.Guili and K.Quancun, "Design of virtual oscilloscope based on GPIB interface and SCPI", 2013 IEEE 11th International Conference on Electronic Measurement & Instruments, vol. 1, pp. 294-298, August2013.
[4] ANSI/IEEE Std. 194-1977, IEEE Standard Pulse Terms and Definitions
[5] ANSI/IEEE Std. 260-1978, An American National Standard IEEE Standard Letter Symbols for Units of Measurement (SI Units, Customary Inch-Pound Units, and Certain Other Units); ISO Std.1000-1981, SI Units and Recommendations for the Use of Their Multiples and Certain Other Units
[6] ANSI/IEEEStd.488.1-1987,IEEEStandardDigitalInterfaceforProgrammableInstrumentation
[7] Willman, J.(2021).Apress, Berkeley, CA) " Deploying PyQt Applications In Modern PyQt(pp. 283-305)"
[8] ANSI/IEEE Std. 260-1978The IEEE has written a new standard on pulse techniques and definitions to replace the withdrawn standards IEEE Std-181-1977 and Std-194-1977".