

Chapter 2

LabVIEW Programming

This chapter provides the basic knowledge about the LabVIEW and its programming for instrument control. LabVIEW, i.e., Laboratory Virtual Instrument Engineering Workbench, is a graphical programming language used by test and measurement researchers and engineers for building customized software solution to their experiments [1–3]. It has an easy approach and can be used with ease. The chapter provides sufficient information to the reader for starting basic programming in LabVIEW for instrument control. This chapter can be used as a guide to connect the instrument to the computer system and basic input/output instructions. The programming in LabVIEW environment is based *data-flow* method and is mostly graphical and easy to understand.

The chapter will provide the brief introduction to LabVIEW, LabVIEW programming and its instrument control capabilities. It will cover the basic features to give an understanding of the LabVIEW programming language and its environment. Examples for controlling the RF instruments explained in this chapter will help to develop calibration suite for attenuation given in Chap. 4.

2.1 Basic Concepts

In this section, we will discuss the basic concepts about LabVIEW and its programming technique. The section is self-contained and is sufficient to understand the basics of instrument control using LabVIEW.

2.1.1 Virtual Instruments (VI)

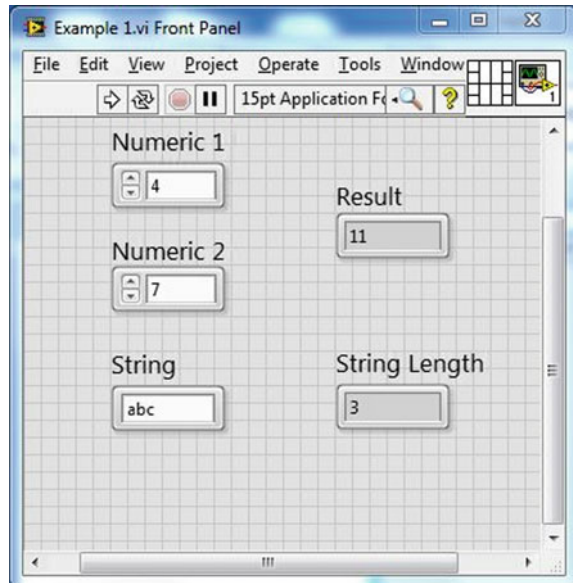
Virtual Instrument (VI) is the file on which LabVIEW coding is done. It comprises two components—front panel and block diagram. The front panel is used to provide

the control components such as a regulatory knob, string input, numerical input or string display. The front panel serves as the graphical user interface and can be easily designed. The block diagram, on the other hand, consists of the programming functions such as algebraic operations, instrument input/output, signal processing, data communication, and error handling. The programming logic is implemented on the block diagram using the different functions available with it.

2.1.2 Front Panel

The front panel which also serves as the graphical user interface for LabVIEW application is used to simulate the input and output mechanism. The input from the user which is to be sent to the instruments can be entered through the numeric controls, text controls, or Boolean input. Similarly, for the displaying the output numeric, text, and Boolean indicators are used on the front panel. Figure 2.1 shows the numeric and string controls and indicators that can be used to design the front panel. The controls are driven by the user, and the indicators are used to display the execution results. In the figure, Numeric 1 and Numeric 2 are the numeric controls, String is the text control, Result is the numeric indicator, and String Length is the text indicator.

Fig. 2.1 Front panel

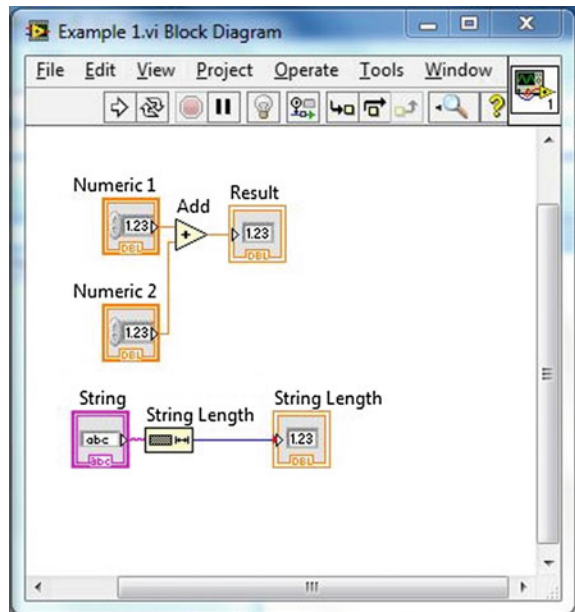


2.1.3 Block Diagram

The front panel shown in Fig. 2.1 will automatically be accompanied by the block diagram. Pressing Ctrl + E opens the block diagram. For the previous example, the node used for the controls and indicators will automatically appear on the block diagram. Figure 2.2 shows the corresponding nodes of the control and indicators used in Fig. 2.1. There are various functions available on the block diagram, for example, the arithmetic operation of addition is used to add Numeric 1 and Numeric 2. Also, programming operations on text can also be done, for example, string length function is used on String Control to calculate the string length indicated on the String Length indicator.

It can be realized by the LabVIEW user that LabVIEW provides an easy and shortcut methods to basic programming problems. As in the given example, the string length function is an easy way that takes off the burden from the programmer, unlike the other programming languages, to write a detailed code for retrieving the length of the string.

Fig. 2.2 Block diagram



2.2 LabVIEW Features

From the introduction above, it is evident that LabVIEW is an easy-to-learn programming language based on data flow used for the implementation of the modern measurement setup. There are some applications in which LabVIEW is the most preferred choice, such as

- Instrument control
- Acquiring and analyzing measurement data
- Designing of embedded systems and field-programmable gate array (FPGA)
- Automated test and validation systems.

In this book, only instrument control and automated test systems are implemented and explained. You will see in Chap. 4, how an example of attenuation measurement system is automated to provide reduced test time and efficient result analysis.

2.3 Virtual Instrumentation

2.3.1 Virtual Instrumentation Foundation

Virtual instrumentation is a modern technique used to emulate the manual execution of a measurement with the help of a computer system [4]. The basic idea of the virtual instrument is shown in Fig. 2.3. The virtual instrument aims to provide better accuracy and precision with minimum manual interference in the measurement. LabVIEW is one the powerful languages which can be used to build virtual instruments.

2.3.2 General Purpose Interface Bus (GPIB)

The General Purpose Interface Bus (GPIB) is a parallel communication interface generally used for instrument control. This bus when connected to the PC at one

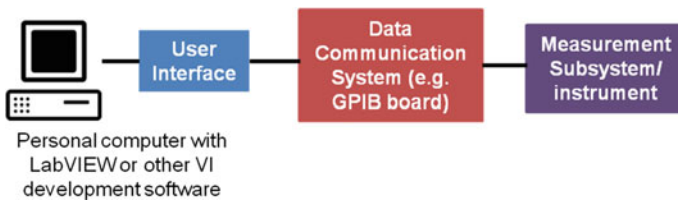


Fig. 2.3 Basic idea of virtual instrument (VI)

end and GPIB compatible instrument(s) at the other end, then the PC can be used as a system controller for the connected instrument(s). The main features of this bus are the following:

- Standardized as IEEE 488
- 8-bit parallel communication using asynchronous handshaking protocol
- The bus has 24-pin configuration where eight are data lines, five are bus management lines, three are handshake lines, and eight are ground lines
- One system controller and 14 instruments can be connected to a single GPIB
- The GPIB instruments on the bus are uniquely identified by 5-bit GPIB (Primary) address ranging between 0 and 30.

Using the NI Measurement and Automation Explorer (MAX), one can detect and configure the connected GPIB instruments on the PC. Figure 2.4 shows a GPIB instrument detected by the NI MAX. This is possibly the first step toward the learning of instrument control and virtual instrumentation.

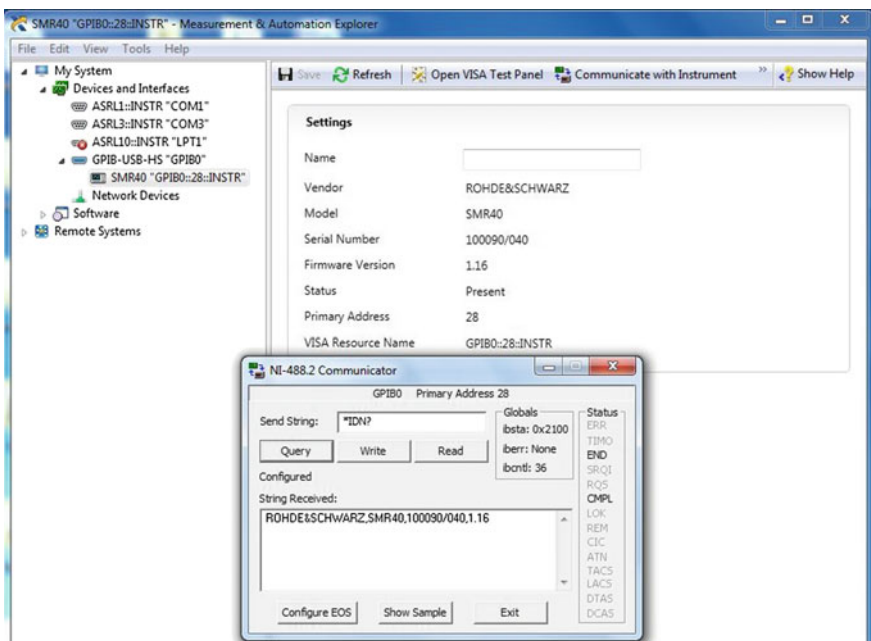


Fig. 2.4 NI measurement and automation explorer (MAX) for detecting GPIB instrument

2.3.3 Standard Commands for Programmable Instruments (SCPI)

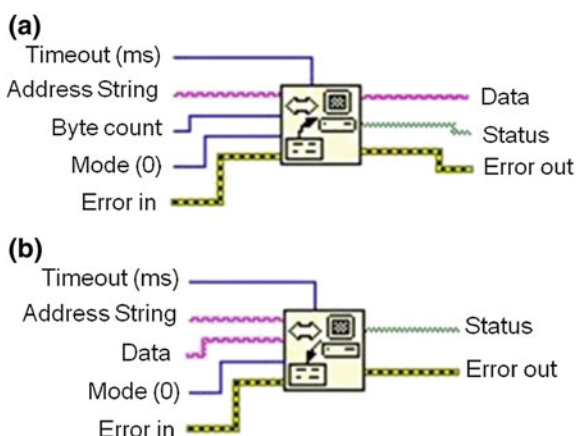
Fig. 2.4, the GPIB instrument with primary address 28 is connected to the PC. Using the NI-488.2 Communicator, one can communicate with the instrument by sending the query. For example, when string `*IDN?` is sent as a query, the detailed identity of the connected instrument is returned. Here, `*IDN?` is an SCPI command.

2.3.4 Instrument Control in LabVIEW

To start instrument control using LabVIEW, one must be acquainted with the GPIB or Virtual Instrument Software Architecture (VISA) functions. Figure 2.5 shows the input and output wires of GPIB read and write functions. GPIB read function is used to retrieve the information from the GPIB device to the controller, i.e., the PC and GPIB write is used for sending the command from the controller to the connected GPIB instrument.

The use of these two functions is shown in Fig. 2.6. The `*IDN?` command sent to the source of GPIB address 28 using NI MAX is shown in Fig. 2.4. The same operation may be programmed on LabVIEW, as given in Fig. 2.6, using simple GPIB write and read functions. In the same manner, other SCPI commands can be sent and bus data can be retrieved for the development of LabVIEW application for instrument control. In Chap. 4, you will see that GPIB read and write functions are used to develop the attenuation measurement suite on LabVIEW virtual instrument.

Fig. 2.5 GPIB functions
a read and b write



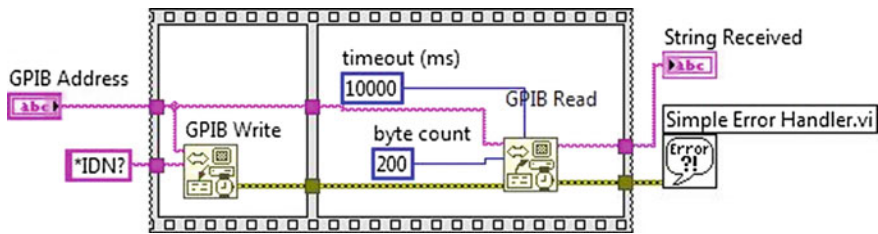


Fig. 2.6 Simple GPIB read and write functions used to communicate with the connected GPIB device

2.4 Advanced LabVIEW Programming Techniques

Although the programming capabilities of LabVIEW are large in number, only the techniques used in the subsequent chapters are discussed in this section. The major focus of this book is to provide the basic introduction to the programming techniques required to automate the commonly used microwave tests and measurements. Previously, the use of GPIB read and write function is demonstrated which plays an important role in GPIB instrument control. Additionally, the advanced file functions, dialogs, and user interface functions facilitate the development of user-friendly virtual instrument. The use of these advanced LabVIEW features will be demonstrated in Chap. 4.

References

1. R. Bitter, T. Mohiuddin, M. Nawrocki, *LabVIEW: Advanced programming techniques*, 2nd edn. (CRC Press, Florida, 2006). ISBN: 0-8493-3325-3
2. G.W. Johnson, *LabVIEW Graphical Programming*, 4th edn. (Tata McGraw-Hill Education, New York, 1997). ISBN: 0-07-145146-3
3. J. Travis, J. Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun (National instruments virtual instrumentation series)*, 3rd edn. (Prentice Hall PTR, New Jersey, 2006). ISBN: 0131856723
4. H. Goldberg, What is virtual instrumentation? *IEEE Instrum. Measur. Mag.* **3**(4), 10–13 (2000)

LabVIEW based Automation Guide for Microwave
Measurements

Dubey, S.K.; Narang, N.; Negi, P.S.; Ojha, V.N.

2018, XIV, 45 p. 31 illus. in color., Softcover

ISBN: 978-981-10-6279-7