IOP Conf. Series: Materials Science and Engineering **394** (2018) 052064 doi:10.1088/1757-899X/394/5/052064

Preliminary Exploration of Instrument Control Technology

Zhenjie Zeng, Xiaofei Zhu*, Shiju Qi, Xiaowei Shen, Shangjun Cai

Rocket Force University of Engineering, Xi'an, China

*Corresponding author e-mail: 1224048915@qq.com

Abstract. Instrument control technology is a key technology for building large-scale automated test systems. In order to construct a large-scale automatic test system more efficiently, the method of interconnecting and interoperating between a computer and an instrument and between an instrument and an instrument when constructing a system is discussed. The common bus for controlling the instrument is introduced. The establishment of the communication link, the control method of the instrument and the selection of the development platform are analysed. The characteristics and application occasions of the two instrument drivers are compared. Finally, several development principles that should be paid attention to when implementing a large-scale automatic test system to realize instrument control are summarized, which provide references for instrument control in large-scale automatic test systems.

1. Introduction

With the continuous expansion of the system scale, the direct use of existing instruments or modular equipment to build test control systems has increasingly become an effective means of shortening the R&D cycle, increasing equipment reutilization, lowering production costs, and increasing production efficiency. Therefore, how to effectively control in-use equipment and equipment to ensure interoperability and good interoperability between devices has become a key issue (1).

This article addresses these issues by introducing the establishment of common instrumentation buses, the establishment of software communication links, the selection of control platforms and development platforms, and the use of instrument drivers. The in-depth analysis of instrument control hardware and software techniques is given and instrument control is given. The general method of system construction and development provides a reference for instrument control in large automated test systems. It should be noted that the instruments discussed in this article are those stand-alone instruments or modular devices that have a built-in microprocessor that can be programmed and controlled through its I/O interface.

2. Instrument bus

The primary task of instrument control is to establish a direct or indirect physical connection to the system controller through the instrument's own bus interface. The instrument bus can be generally divided into two categories: device bus and system bus. The device bus is GPIB (general purpose interface bus), RS-232, RS-422, RS-485, USB, FireWire, Ethernet, Wireless, etc. Such as PCI, PXI, PCIe, VXI and so on. Each type of bus has different applications because of different communication speeds, communication distances, reliability, and mechanical characteristics. The diversity of instrumentation buses provides great convenience and flexibility for the construction, maintenance, and

IOP Conf. Series: Materials Science and Engineering **394** (2018) 052064 doi:10.1088/1757-899X/394/5/052064

expansion of communication links between system controllers and various interface devices in various complex applications. This article is in the device bus and system bus. Take each one for introduction.

2.1. GPIB bus
The GPIB bus is also known as the IEEE488 bus. It became an international standard in 1978. It is the most commonly used bus in the device bus and is widely used in the field of automatic testing.

GPIB is a parallel bus with 16 signal lines, including eight data lines, three handshake lines, and five management lines. A GPIB can connect up to 15 GPIB devices at the same time. The communication range must not exceed 20m (the cable length between the two devices should usually be less than 4m). The connected units can communicate directly without going through the intermediary unit. The most typical advantages are convenient connection, flexible configuration, and stable and reliable communication. For the test system, the reliability of data transmission is very high. The high reliability of the GPIB bus can meet the requirements of the test system. Therefore, today's bus technology is still the most popular type of bus for users. It is not a standard interface for PCs. A GPIB expansion card needs to be installed to establish a connection with the GPIB device.

2.2. VXI bus
The VXI bus is an extension of the VME bus in the field of instruments. It has the advantages of fast test speed, good reliability, strong anti-interference ability, and friendly man-machine interaction. The VXI bus-based automatic test system is widely used in the military industry and aerospace industry. , product testing and other fields. In the more than ten years after its emergence, the number of VXI bus instruments has increased by almost 50% every year. Nearly half of the instruments in the United States used VXI buses in 2000.

Although the current bus has great development, new types of buses such as PXI and PXIe have appeared. However, due to the mature technology, reliability, and anti-interference performance of the VXI bus, a large part of the test equipment used in many aerospace and military industries is currently used. It is still the VXI bus test equipment. Therefore, in the development of the automatic test system, the development based on the VXI bus is still a good solution.

3. Communication link and Control method
To control the control of many instruments, it can be divided into two steps. The first step is to realize the communication between the test instrument and the computer and between the instrument and the instrument. The second step is to send the instrument language to the instrument and control the instrument. Initialization and related tests.

3.1. Establish communication
The first step is to establish communications. As mentioned above, with the development of automatic test technology, a variety of instrument interfaces have emerged, and each interface has its own characteristics, so the measurement and control program must be written for a specific interface bus. This kind of monitoring and control program is difficult to transplant to other interfaces. When you need to transplant, you need to change a lot of code. To solve this problem, in 1993 VXI plug play Systems Alliance developed a bus-independent software specification for instrument I/O. The full name is virtual instrument software architecture (VISA), which defines a set of common APIs that can be used for almost all instrument bus communications. . As shown in Figure 1, the appearance of VISA makes the measurement and control program no longer dependent on the specific bus interface. The program written based on VISA can be well transplanted between different buses, which greatly simplifies the development of monitoring and control procedures. The generality of the program (2).

It should be noted that VISA is only a specification and cannot be used directly. Specific instrument manufacturers or other third-party software developers based on this specification can use it directly. The most widely used is the Agilent VISA library developed by Agilent and NI Corporation. Developed NI VISA library. VISA can communicate with different I/O interfaces, not because he owns this capability, but because it encapsulates the low-level I/O routines of the common bus. When communicating with the instrument, it calls the internal underlying I/O. O program to achieve. When a new bus interface emerges, it only needs to be encapsulated into the underlying I/O of the new interface to be able to communicate with instruments that have this interface, and to maintain consistency in use, so VISA will never be out of date.

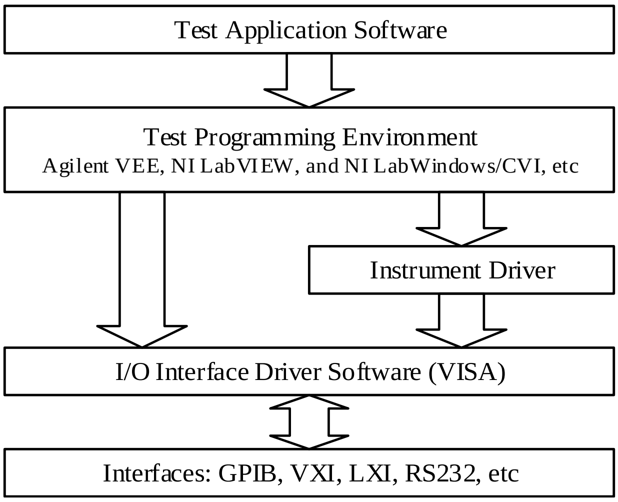

Figure 1. Virtual instrument software architecture

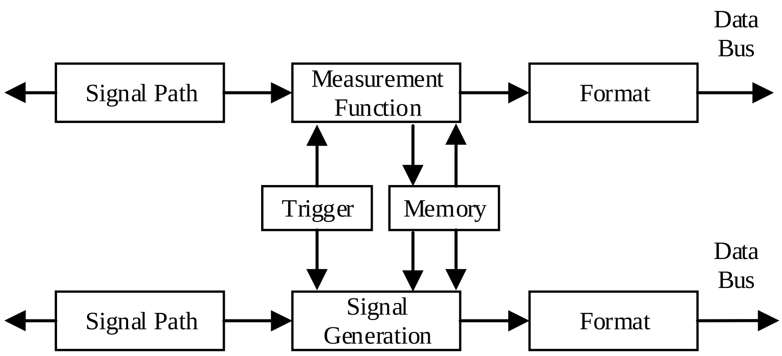

Figure 2. SCPI program-controlled instrument model

3.2. Controlling

The second step is to achieve control. There are two kinds of bottom control methods most commonly used by computers to control the instrument. They are register-based programming and string-based programming.

When using the programming mode of the register, the control computer sends to the target instrument the information such as the address and data of its internal register, so the instrument MPU can directly read and rewrite the related register. The advantage of this mode is that it is flexible and efficient, and the instrument control technology is used in the early stages of development. The disadvantage is that it requires a high level of programmers. The programmer must be very familiar with the hardware structure and instrument working principle to write the program. This is for most programmers. Not realistic.

When using the string programming method, the control computer sends a string of codes to the target instrument, so the instrument MPU needs to parse the strings before completing the underlying operations. Compared to register programming, this kind of programming the implementation of the method is less efficient. However, the string programming method encapsulates the underlying operations related to the instrument operation, so that the programmer can program and control the instrument without understanding the hardware of the instrument, and the character string is very intuitive and specific, and is easy to understand and grasp.

Due to the large number of instrument manufacturers, a specific instrument to unify the program control command is also difficult to achieve, this problem until SCPI (Standard Commands for Programmable Instruments) appeared before the turning point. The programmed commands of the instruments in the SCPI specification are no longer oriented toward specific instruments, but are instead directed to a general instrument model that describes the function of the instrument.

Figure 2 shows the SCPI general instrument model. The first part describes the measurement function of the instrument. The role of signal routing is to select the test channel and the internal signal transmission path according to the received command; the measurement function is the core of this part, different the instrument can provide different measurement services. Under normal circumstances, it needs external triggering. The data obtained by the test needs to be stored. Therefore, it needs to interact with the internal memory of the instrument. The second half describes the function of the source instrument. The formatting part converts the external incoming data into a data format. The signal generation is the core of the lower part. Different instruments can generate different signals; the signals that the signal routing will generate select the path output to the external (3). An instrument may contain all the functions in the instrument model, but in most cases, it only contains some of the functions in this diagram. For example, a calibration source contains only the lower part of the function.

For the first time, SCPI has realized the unification of instrument control methods. Since its appearance, SCPI has been widely used. Nearly all instrument manufacturers have produced program-controlled instruments according to this standard. SCPI is a continuously improving standard. With the appearance of new instruments, new program control directives have been added to the standard. In 2003, the SCPI Alliance joined the IVI Foundation, which means SCPI will continue to develop in the coming period of time. Is widely used. It should be noted that SCPI is upwardly compatible and that the new version of the standard contains commands that are not applicable to the instrument prior to that.

4. Software development platform

There are many development environments that can be used to develop instrument control software. The main ones are NI LabVIEW and Lab Windows/CVI, Keysight (renamed by Agilent) VEE, and Microsoft Visual Studio (4).

4.1. Lab VIEW

Lab VIEW and VEE are graphical development-based development environments. Their easy-to-learn features make them widely used in the development of small-scale measurement and control software. However, it is worth noting that National Instruments is moving LabVIEW to the .NET Framework and releasing the first version of LabVIEW NXG (Next Generation: Next Generation). This upgrade of LabVIEW NXG 1.0 is not an unusual upgrade of LabVIEW 2017. Although it uses the previous graphical programming style, it differs from the previous version in technical architecture. LabVIEW NXG 1.0 is an exploration of NI's migration to the .NET architecture and is currently not mature enough to be used to develop complex test systems. Previous versions were not in the .NET Framework. Under the current trend, it can be used to develop a small test system with a short maintenance period. In view of the poor readability and low execution efficiency of the graphical design development environment, the development of large-scale measurement and control programs does not recommend the use of LabVIEW.

4.2. Visual Studio

The Visual Studio environment (including Visual C#.NET, Visual Basic .NET, Visual Basic 6.0 and Visual C++, etc.) is a general-purpose programming tool, and measurement and control software development can be achieved with NI's Measurement Studio toolkit. Measurement Studio is an integrated suite for the Visual Studio .NET and Visual Studio 6.0 environments, including a variety of commonly used measurement and automation controls, tools, and class libraries with ActiveX and .NET controls, object-oriented measurement hardware Interfaces, advanced analysis libraries, scientific user interface controls, measurement data networking, wizards, interactive code designers, and highly scalable libraries. Measurement Studio brings great convenience to the development of instrument control programs using Visual Studio, attracting many developers to use Visual Studio as a development tool. At present, there are many instrument manufacturers in China that are vigorously promoting C# as a development tool. In particular, Jianyi Technology is using C# as a programming tool, and is committed to building a complete monitoring and control software development platform, the Open View monitoring and control platform, although it is currently in the beginning stage, many places are still not perfect, but it has the advantages of open source and has great potential for development.

4.3. Lab Windows/CVI

As a software development platform that has been active in the field of test and measurement, LabWindows/CVI has undergone considerable development and has been designed and optimized for measurement and automated application development. It is a complete ANSI C programming environment, has a simple and intuitive graphical user interface, uses input function parameters in the function panel, and uses event-driven and callback function programming techniques, making the engineering design not only efficient but also stable and reliable. It uses object-oriented programming ideas, rich controls and buttons, and a powerful library of functions (such as signal processing libraries, interface libraries, etc.) and additional toolkits that make it powerful, flexible, and efficient. And stability, it is suitable for the development of large-scale measurement and control software, mainly used in areas with high stability and reliability requirements, such as aerospace industry, military industry (5) (6).

5. Instrument driver

Although SCPI largely achieves the unification of control methods, when developing software, it is still necessary to query the instrument commands and their formats through the instrument manual (or SCPI manual) and format each input and output task into a corresponding command string. . Obviously, in large-scale measurement and control applications, using SCPI programming is still a very tedious task.

In order to solve this problem, you need to use instrument drivers. Instrument drivers are a set of advanced function libraries that can be called directly by the user to control a certain type or class of instruments. It encapsulates underlying operations such as communication links (implemented using VISA), register programming, and SCPI programming, liberating software developers from the complex underlying programming protocols associated with a particular instrument. With the help of instrument drivers, the instrument control is changed to a general function call problem. The user only needs to provide appropriate entrance and exit parameters for the tuned function according to the measurement and control requirements. By using the instrument driver, without understanding the specific operation flow within the instrument, matching the instrument driver with the corresponding interface can realize the interaction between the instrument and the computer. At present, there are two kinds of instrument drivers used most widely, namely VPP driver and IVI driver.

5.1. VPP drive

In the early days of the development of instrument drivers, many instrument manufacturers also developed corresponding interface drivers while introducing new hardware. However, most of these drivers are unitary and only apply to one device or one type of interface, resulting in the same the poor hardware interchangeability between instruments of the type has brought inconvenience to the entire operation process.

In order to solve this problem, in 1993, Tektronix and other internationally renowned instrument companies jointly established the VXI plug play alliance. At the same time, the alliance developed the VXI plug play specification, referred to as the VPP specification. This specification addresses the commonality of VXI systems from different instrument manufacturers, making the VPP driver not only suitable for VXI instruments, but also for PXI instruments, GPIB instruments, USB instruments, and serial port instruments (7). In addition, the VPP specification also proposes a series of standards for VPP systems and instrument drivers. All instrument manufacturers develop instrument drivers in accordance with uniform specifications and requirements, which enhances the versatility of instrument drivers and reduces the burden on software programmers. . The VPP driver generally supports a device of the same manufacturer or the same series of devices, and cannot interchange the devices of different manufacturers.

5.2. IVI driver

From the current status of the development of instrument driver technology, according to the VPP software standard, the relevant technical problems of the instrument driver have established a unified standard. In addition, the VISA interface software is defined in the VPP specification and provides a unified interface for the operation of various bus instruments. However, VPP instrument drivers are closely related to specific instruments. When replacing different manufacturers or different models of the same manufacturer, not only the instrument driver but also the test program must be modified to accommodate the new instrument and instrument driver (8).

In order to achieve the interchangeability of the instrument and improve the efficiency of the test program, the IVI Foundation was established in August 1998 by 9 companies including NI, Tektronix, etc. The foundation established a new instrument driver program based on the VPP. The programming interface standard, designed on the basis of the standard driver program, realizes the independence of the instrument. In addition, the standard instrument driver program also increases the instrument simulation, state cache and other mechanisms, so that the test program design, debugging and operating efficiency have greatly improved. Although IVI technology is based on the VPP specification, it does not lead to additional complexity and performance degradation, but adds many features that improve the efficiency of system implementation and shorten development time. It draws on the advantages of VPP technology. The development cycle and development cost of the test software in the test system are reduced, and the update adaptability of the test system is greatly improved, providing an important way to eliminate redundancy from the software and increase the test rate. The IVI driver has more powerful functions and better program versatility than the VPP driver.

Conclusion

To sum up, when establishing instrument control, first, there is no need to consider the increase of programming difficulty caused by different buses, and it is necessary to select the appropriate bus construction system according to need; secondly, develop small-sized devices that have no versatility and no instrument interchangeability requirements. When monitoring and controlling the program, you can choose to use VPP driver to control the instrument to shorten the development time and reduce the development difficulty. Moreover, if you develop a small-scale measurement and control program that is versatile and reusable, use VSI-based SCPI string programming is a A good choice; Finally, to develop a general-purpose, reusable, large-scale measurement and control system, it is recommended to use IVI-type driver programming to reduce development difficulty, and use VISA-based SCPI string programming to increase system flexibility.

References

[1] ZHANG Zhi-li, HOU Chuan-xun, JIANG Yi. Discussion on Instrument Interconnection and Control Technology [J].Process Automation Instrumentation, 2011, 32 (07): 1-5+10.
[2] LIU Tian-shu, JIN Cong-jun. Development of automatic calibration program based on VISA and SCPI [J]. Modern Defense Technology, 2013, 41 (03): 175 - 179.
[3] HUANG Meng-tao,QUAN Ding-ke, FU Xiang-yang.Intelligent Instrument Driver Design Based on SCPI Command [J].Control and Instruments In Chemical Industry, 2010, 37 (04): 103 - 105.
[4] SONG Bo,CHEN Yi-min.Comparison and Selection of Virtual Instrument Development

Tools[J].Foreign Electronic Measurement Technology, 2006 (08): 1 - 5.
[5] TAO Xiao-liang, NIU Zhen. Application of LabWindows/CVI Multithreading Technology in Servo Testing Software [J]. China test, 2011, 37 (01): 81 - 83.
[6] HAN Hui-lian, WANG Hao. Design and Implementation of Missile Measurement and Launch Control System Based on Virtual Instrument [J].Computer Measurement & Control, 2011, 19 (03): 621 - 624.
[7] ZHANG En-feng. Design of automatic instrument calibration system based on LabVIEW and GPIB [D]. North University of China, 2017.
[8] IVI Foundation. Getting started with IVI drivers [EB/OL]. [2010-05-01]. http://www.ivifoundation.org/.