TELKOMNIKA, Vol. 11, No. 4, April 2013, pp. 1835~1840
ISSN: 2302-4046

Remote Control Techniques to the Digital Storage Oscilloscope by GPIB and VISA

Faquan Zhang*[1], Guofu Wang[2], Jincai Ye[3], Qingning Zeng[4]
[1,2,3,4]School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
No.1 Jinji Road Guilin City, Guangxi Zhuang Autonomous Region, China
*Corresponding author, e-mail: zhangfq@guet.edu.cn

Abstract

Some techniques of controlling remotely the digital storage oscilloscope were proposed including VISA, VXIplug&play drivers, TVC and IVI-COM drivers. By means of GPIB and VISA, several application development environments could be used to develop remote control techniques to the digital storage oscilloscope. The programming language of Visual C++ was used to develop software. With the help of VISA and the dynamic link library, remote control to the digital storage oscilloscope through network was completed. All operations to the digital storage oscilloscope including parameters setup, data acquisition, waveform acquisition and storage, data processing were implemented in a remote place. The results show that the techniques of remote control are convenient and efficient and fit for experiments of scientific research and practical projects.

Keywords: *digital storage oscilloscope, GPIB, remote control, data acquisition*

1. Introduction

The Digital Storage Oscilloscopes are high performance solutions for verifying, debugging, and characterizing sophisticated electronic designs [1-2]. Hence, they are utilized widely in many applications. Compared to the analog oscilloscopes, their key features include: (1) Trigger modes including edge, logic, pulse, and sequence. (2) Sample, envelope, peak-detect, high-resolution, and average acquisition modes. (3) Full programmability, with an extensive GPIB-command set and a message-based interface. (4) Powerful built-in measurement capability including histograms, automatic measurements, and measurement statistics. (5) Wide array of probing solutions. (6) Open connectivity to several design environments including Visual BASIC, C, C++, MATLAB, LabVIEW and LabWindows/CVI. In many scientific research experiments and practical projects, to be on the safe side, the operators were forbidden in the scene [3-5]. Therefore, the digital storage oscilloscopes for data acquisition need be connected with the central computer to acquire real-time data and process data. So remote control techniques to the digital storage oscilloscope are useful. In this paper, we research these techniques by taking the Tektronix digital storage oscilloscope for example.

2. Research Method

2.1. System Framework

Figure 1 shows connectivity between Application Development Environments and the digital storage oscilloscope.

Test and measurement applications require some kinds of I/O library to communicate with the test instrumentation, or the digital storage oscilloscope. As a step toward industry-wide software compatibility, the VXIplug&play Systems Alliance developed a common I/O library called the Virtual Instrument Software Architecture (VISA). VISA provides a common standard for software developers so that software from multiple vendors, such as instrument drivers, can run on the same platform.

TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API), is industry-compliant software, available with selected Tektronix instrument models, for writing (or drawing) interoperable instrument drivers in a variety of Application Development

Received December 26, 2012; Revised February 9, 2013; Accepted February 21, 2013

Environments (ADEs). TekVISA implements a subset of Version 2.2 of the VISA specification for controlling GPIB and serial (RS-232) instrument interfaces locally or remotely via an Ethernet LAN connection. TekVISA provides the interface-independent functionality needed to control and access the embedded software of Tektronix test and measurement equipment in the following ways: (1) Using virtual GPIB software running locally on Windows-based instrumentation. (2) Using physical GPIB controller hardware. (3) Using asynchronous serial controller hardware. (4) Over a Local Area Network (LAN) that uses VXI-11 protocol and one of the following: an AD007 LAN-to-GPIB adapter to GPIB controller hardware or a 10Base-T Ethernet connection together with virtual GPIB software running on Windows-based instrumentations [4].

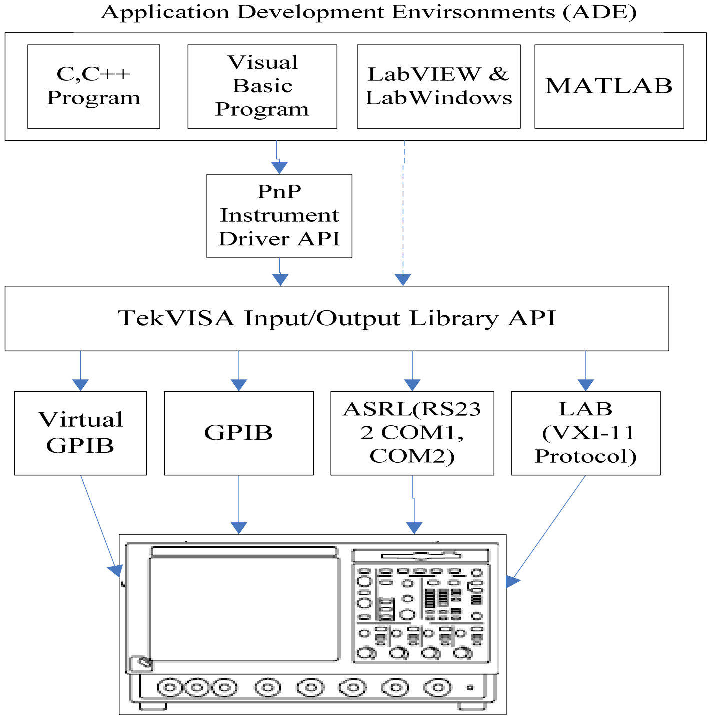

Figure 1. System Framework used in many Applications

TekVISA also provides language interface libraries for programmers using multiple Application Development Environments as shown in Figure 1, including Microsoft C/C++, Microsoft Visual Basic, LabVIEW graphics software using the G language, and MATLAB analysis software [5].

Plug&Play (PnP) instrument driver API is a VXI Plug&Play standard instrument driver for the Tektronix series of oscilloscopes. The driver consists of a number of functions that mirror the knobs and controls on the oscilloscope and the menu selections on the oscilloscope software. It is basically a library of ANSI C functions. These functions can set up, communicate with, acquire data from, and otherwise control features of the oscilloscope. We can call the run-time functions from the test programs you write. Since the driver is delivered as a standard dynamic link library, we can easily incorporate these Plug&Play driver functions into programs that you build using the following application development environments and popular test automation packages like Microsoft Visual C++, Microsoft Visual Basic, Borland Delphi, NI LabWindows, NI LabVIEW, HP VEE, and other ADEs which support DLLs.

The Plug&Play drivers can be used directly on the scope itself and also on a client PC controlling the instrument through GPIB, USB-GPIB adapter, Tektronix VXI-11 LAN server or GPIB-LAN adapters, such as Tek AD-007 and equivalent National Instruments product [6-9].

In this system, we adopt the system structure schematic diagram shown in the Figure 2.

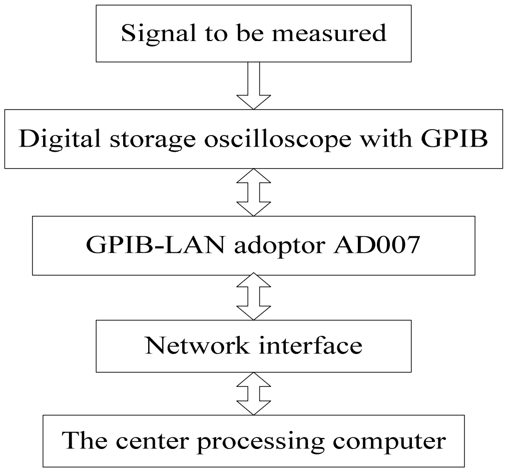

Figure 2. System Structures Schematic Diagram in the Application System

2.2 Software Implementation

We use Microsoft Visual C++ language to program the digital storage oscilloscope. Visual C++ is a good programming environment for all 4 programming interfaces including VISA, VXIplug&play drivers, TVC and IVI-COM drivers.

We take an example of VISA driver. First we use viOpenDefaultRM() function to get access to get access to the Default Resource Manager. This function must be called before any VISA operations can be invoked. The first call to this function initializes the VISA system, including the Default Resource Manager resource, and returns a session to that resource.

After opening the Default Resource Manager, applications use the viOpen() operation to get access to a particular instrument resource. This operation opens a session to a device resource using VISA descriptor. In this case we are opening the string identifies the virtual GPIB device which allows the program to run on the scope. After a VISA session is opened, the session can be used by viWrite() and viRead() functions to send and read GPIB responses to/from the instrument.

Finally we use viClose() to close that session and free up all the system resources associated with it.

Software flow chart is shown in Figure 3.

The following steps need to be carried out in a practical instance.

1) Initialize the oscilloscope

Initialize the oscilloscope using tktds6k_init() function with format:
Tktds6k_init(ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, Visession *instrumentHandle).

Where parameters of resourceName is the device alias. Parameters of IDQuery is ID of the oscilloscope to be initialized. Parameters of resetDevice determines whether the oscilloscope is reset or not.

2) Setup parameters for the oscilloscope

Many parameters of the oscilloscope can be setup by dint of different functions provided by DLL library, for example:

SetWfmDisplayStatus(int, int, int*). This function is used to set display status of waveform.

SetTriggerLevel(int, double*). This function is used to set trigger level to capture input signal when the oscilloscope is triggered.

SetVerticalScale(int, int, double*). This function set vertical scale of the oscilloscope for display.

There are many similar functions to set all kinds of parameters for the oscilloscope in remote style.

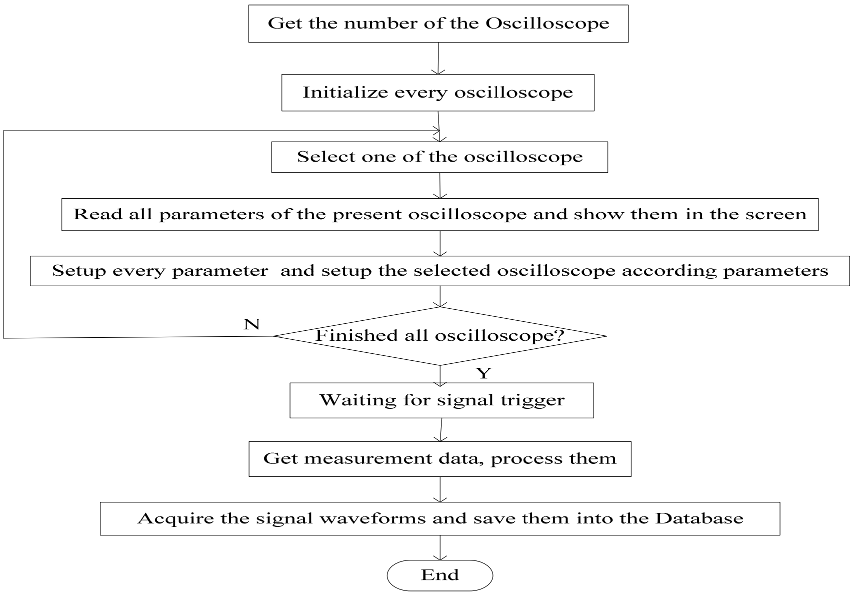

Figure 3. Software Flow Chart in the Application System

3) Waveform acquisition and storage

After triggered by the assigned parameter, the input signal waveform was grabbed and shown on the screen of the scope in assigned style. Waveform acquisition can be completed by GetWaveform() function. Now the original data displayed in a real oscilloscope can be reached in the computer by network. Therefore we can utilize SaveWaveform() function to save the original data in specific file format in the database.

3. Research Results and Analysis

When hardware system and software system introduced in the above section are completed, operational interface to the oscilloscope in a remote place is shown in Figure 4.

The remote control system allows several oscilloscopes in the practical projects. So we need select one of the oscilloscopes to operate. In the drop-down box of select oscilloscope all oscilloscopes is list to be selected.

Trigger parameters including trigger channel, trigger level and trigger delay are in the left-top part. Every oscilloscope has several input channels, so we need select which channel signal trigger acquisition. Trigger level and trigger delay are inputed according to the original signal.

The drop-down box of Time Resolution list different horizontal scale values to determine horizontal display style.

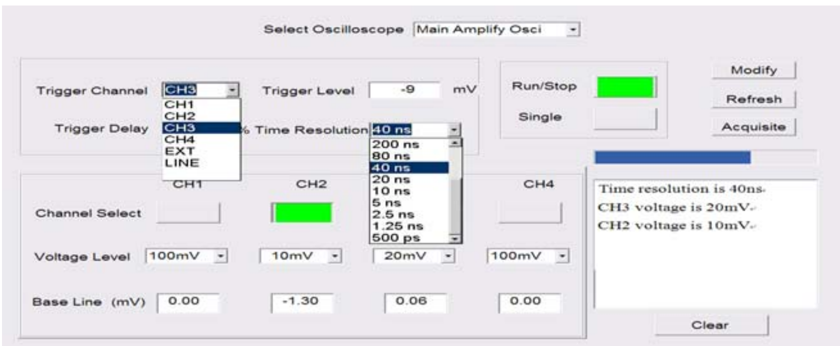

Figure 4. Setup Interface of Parameters for the Oscilloscope in the Initial Stage

There are four drop-down boxes of Voltage Level and four edit boxes of Base Line. They are correspond respectivly to four channels to determine vertical display style of the oscilloscope.

Reaction of the real oscilloscope to every parameters setup is shown on the edit box in the right-bottom part.

Waveform acquisition result and indicating interface are shown in the Figure 5.

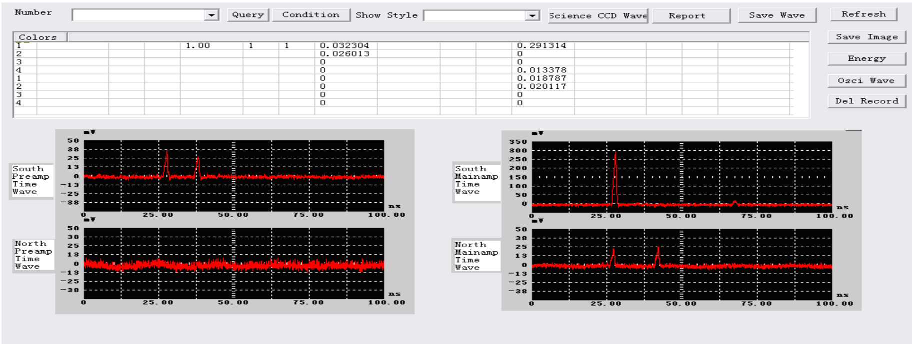

Figure 5. Waveform Acquisition Result and Indicating Interface

Original data acquired by the four oscilloscopes were transfer to the center control computer by network. We can process respectively every group data to extract rise time, impulse width and impulse count and so on.

5. Conclusion

We implement remote control techniques to the digital storage oscilloscope by using the principle of virtual instrument but avoid to program using virtual instrument language. By dint of dynamic link library provided by the manufacturer of the oscilloscope and Visual C++ language, we designed remote control system in which data acquisition can be completed by the central computer. We can setup all parameters for the oscilloscope in another place through the network and get the original signal. Then we develop data processing algorithm and retrieve the processing results. The techniques are introduced by the Tektronix oscilloscope, but they fit absolutely for the other type oscilloscope through using the corresponding dynamic link library.

Acknowledgements

The authors wish to thank the helpful comments and suggestions from my teachers and colleagues in Guangxi Key Lab of Wireless Wideband Communication & Signal Processing at Guilin. This work was supported in part by NSFC under Grant No. 61102115 and Guangxi Key Lab of Wireless Wideband Communication & Signal Processing under Grant No. 11109.

References

[1] Shengli Dong, Junfeng Han, Shenghui Pan, Jianfeng Deng. *Design of virtual oscilloscope based LabVIEW*. Communications in Computer and Information Science. Qinhuangdao, China, 2011, 244: 34-41.
[2] Lixiao Zhang, Xin Li, Yanshuang Li. *Labview-based design and realization of virtual oscilloscope*. Lecture Notes in Electrical Engineering, Hefei, China, 2012, 177: 269-274.
[3] G Matsunaga, M Takechi, K Toi. Fast data acquisition system based on digital oscilloscopes for fluctuation measurements in a long pulse JT-60U tokamak plasma. *Fusion Engineering and Design*. 2007; 82(2): 207-213.
[4] KPS Rana, R Singh, KS Sayann. Correlation based novel technique for real time oscilloscope triggering for complex waveforms. *Measurement*. 2010; 43(3): 299-311.
[5] Lombardi, Mark R. The most flexible system component: High speed digital oscilloscope. *IEEE Aerospace and Electronic Systems Magazine*. 2010; 25(7): 20-24.

[6] Haruo Saito, Toshio Hyodo. Improvement in the gamma-ray timing measurements using a fast digital oscilloscope. *Radiation Physics and Chemistry*. 2003; 68(3-4): 431-434.
[7] KPS Rana, SH Khan. A DAQ card based mixed signal virtual oscilloscope. *Measurement*. 2008; 41(9): 1032-1039.
[8] James Miall. *Impedance of active devices using a real time digital oscilloscope and quarter-wave technique*. CPEM Digest (Conference on Precision Electromagnetic Measurements). Washington, DC, United states. 2012: 524-525.
[9] Jian Shi, Mian Guo. Embedded digital oscilloscope based on STM32 and C/OS-II. *Applied Mechanics and Materials*. 2012; 190-191: 1129-1135.