

Development of remote control software for multiformat test signal generator

Yang Gao^{1,*}, Jiefeng Liu¹

¹Communication University of China, School of Science and Technology, Beijing, China

Abstract. The multi format test signal generator mentioned in this paper is the video signal generator named TG8000 produced by Tektronix Company. I will introduce the function about remote control for signal generator, how to connect the computer to the instrument, and how to remote control. My topic uses my computer to connect the instrument through the 10/100/1000 BASE-T port on the rear panel of TG8000. Then I write program to transmit SCPI (Standard Commands for Programmable Instruments) to control TG8000. The application is running on the Windows operating system, the programming language is C#, development environment is Microsoft Visual Studio 2010, using the TCP/IP protocol based on Socket. And the method of remote control refers to the application called TGSetup which is developed by Tektronix Company. This paper includes a brief summary of the basic principle, and introduce for details about the process of remote control software development, and how to use my software. In the end, I will talk about the advantages of my software compared with another one.

Keywords: TG8000, C#, Socket, SCPI

1 Introduction

Test signal generator is mainly applied in generating synchronization signal or test signal for final-stage production of TV program and broadcasting television system. TG8000 is a major player on test signal generator market especially for TV station and university. Due to the high cost of signal generator, is not easy to replace, some of the button function has been damaged. In order to solve the problem of the loss of function caused by the damage of the button, it is necessary to develop one way to remote control by computers. Tektronix developed TGSetup software for remote control. However, this software is not open source, the operating system version of the system has a lot of limitations, has a long latency time and so on shortcomings. Worst of all, operating system often crashes when TGSetup running. This paper designs an improved remote control method, based on the Visual Studio environment, almost no delay. It is worth mentioning that the main function of the software is to remotely control the signal generator to generate the required signal.

2 SoftwareDevelopment

2.1 The Way to Solve the Communication Problem Between PC and TG8000

With the rapid development of modern science and technology, information and technology updates faster and faster, to meet the people of higher quality, more content, more information and higher level visual auditory experience requirements, as carrying the arduous task of dissemination of information of the radio and television industry is also undergoing a change rapidly changes. Video signal generator is an important tool in the radio and television industry.

TG8000 is a sophisticated multi format analog and digital signal generator platform, it provides synchronous pulse and time code signal for radio and TV applications, and can provide reference for the application of video test signal testing equipment. TG8000 is mainly used to generate the pulse signal and test signal of the post production and broadcast television broadcasting system, to study and develop the test signal, and to design and maintain the equipment.

Because of the lack of information on how to connect PC with TG8000 to send commands, I must refer to TGSetup. First of all, I need to connect my PC to the TG8000 through LAN port with a cable [1]. There is a 10/100/1000 BASE-T port on the rear panel of TG8000

which is designed for remote control [2]. Then I get the IP address of TG8000 through its front panel and set the IP address of my computer to the same segment. Open TGSetup, enter the IP address of TG8000 and click the connect button in the dialog box that appears, about ten seconds after the LCD display of TG8000 turn from light to dark, and the TGSetup access control interface. This phenomenon represents a state of remote control.

From these steps, it is no difficult to find that after the establishment of a physical connection, the software send a series of commands packed into IP packets to TG8000 through the network. So I analyzed the commands sent by the software by using Wireshark software. This software is used to follow TCP stream. There two colour letters in the screen. The red letters matched the data sent from the PC to the TG8000 while the blue letters matched the feedback of the TG8000. The software sent packets to control the instrument, so it is no need to be focus on the blue letter. The TCP stream shows that SCPI is encapsulated in TCP data packet and sent to the TG8000 to make it response. SCPI (Standard Commands for Programmable Instruments) is a software standard independent of hardware, so many of these commands can be consulted. SCPI is a hardware independent software standard, the string can be transmitted through any instrument interface. It defines the content and structure of the controller to the instrument and the instrument to exchange information between the controller layer messages, which is a standard way to describe the various functions of the instrument; its purpose is to reduce the automatic test equipment program development time. The SCPI command consists of three parts:

- Grammar and style -a description of the basic principles of the application and development of real language,
- Command structure -describes the actual language structure in the instrument,
- Data interchange format -a standard representation of a data set that can be used between instruments and applications, between applications and applications or between instruments and devices.

SCPI can be divided into two commands: the general command and the instrument specified command. For example, “*IDN?” is a general command. It asks the instrument to return the version information and if the TG8000 receives this command successfully, it will reply” TEKTRONIX, TG8000, 0, 0”. Besides these request commands, there are also some control commands. They end with a period, while those request commands end with a question mark [3]. Obviously, we can use these control commands to achieve remote control. The most important work of our software is to control the TG8000 to send signal that we need. In order to accomplish this, we will use following commands:

“*OPC?” It is a general command, and its main function is to stop receiving any commands until all the commands before had been completed. This command ensures the complete execution of the key commands.

“MMEMory:LOAD:SIGNal<module_name>,<signal_name>”,This is a control command without a return

value. It’s no difficult to see that this command is used for loading signal. When analyzing this command, I found that the type and the format of a signal appear as a whole.Forexample,MMEM:LOAD:SIGN3,"NTSC/OTHER/NTC7^Combination.cmp.

This provided a clear idea for my software. The signal format of AVG module includes NTSC, PAL-1, PAL-2, NTSC NO SETUP, 625 YPbPr, 525 GBR, 625 GBR, 525 YPbPr, 525 Beta [4]. These formats are corresponding to different specific signals. And the signals (expressed in the form of a SCPI parameter) are:

- OTHER/NTC7^Combination.cmp,
- OTHER/FCC^Multiburst.cmp,
- OTHER/NTC7^Composite.cmp,
- OTHER/Matrix 1.cmp,
- OTHER/FCC^Composite.cmp,
- OTHER/SNG^ColorBars.cmp,
- SWEEP/100%^Sweep.cmp,
- SWEEP/Chroma^Frequency^Response.cmp,
- SWEEP/60%^Sweep.cmp,
- COLOR BAR/100%^ColorBars.cmp,
- COLOR BAR/75%^ColorBars.cmp,
- COLOR BAR/SMPTE^ColorBars.cmp,
- FLAT FIELD/10%^Flat Field.cmp,
- FLAT FIELD/Field^SquareWave.cmp,
- FLAT FIELD/100%^Flat Field.cmp,
- FLAT FIELD/Black Burst^with^FieldRef.cmp,
- FLAT FIELD/40%^Flat Field.cmp,
- FLAT FIELD/0%^Flat Field.cmp,
- FLAT FIELD/50%^Flat Field.cmp,
- FLAT FIELD/Black Burst.cmp,
- MULTIBURST/60%^Multiburst.cmp,
- MULTIBURST/Multipulse.cmp,
- MULTIBURST/100%^Multiburst.cmp,
- MONITOR/White^Window.cmp,
- MONITOR/2 Level^Pedestal^&Pluge.cmp,
- MONITOR/Monitor^Setup.cmp,
- MONITOR/Convergence^Pattern.cmp,
- MONITOR/Safe^Area.cmp,
- MONITOR/75% Red^Field.cmp,
- MONITOR/Gray^Window.cmp,
- MONITOR/4 Level^Pedestal^&Pluge.cmp,
- MONITOR/100% Red^Field.cmp,
- LINEARITY/5 Step.cmp,
- LINEARITY/Modulated^5 Step.cmp,
- LINEARITY/Ramp.cmp,
- LINEARITY/Modulated^Pedestal.cmp,
- LINEARITY/10 Step.cmp,
- LINEARITY/Shallow^Ramp.cmp,
- LINEARITY/Modulated^Ramp.cmp,
- PULSE BAR/Window 2T^Pulse &Bar.cmp,
- PULSE BAR/SinX`sX.cmp.

These are all the signals that could be used for software. I can take advantage of these combinations to create the signals that we need.

2.1.1 The Design Idea of the Software

After Part 1 had understood the working principle of TGSetup, Part 1 focused on the improvement method. TGSetup can only be used on Windows7 or Windows XP, so I write my program on Visual Studio so that it could be used on most operating systems. The biggest shortcoming of TGSetup is the long delay. It takes at least ten minutes to connect, and worst of all, it costs more than thirty minutes to set a signal. So I analysed data by Wireshark.

My conclusion is that two-way communication takes too long, so I tried to communicate one way most time. Two-way communication is to ensure each command is received correctly, so the software each sends a command an additional request, but this greatly reduces efficiency. My software retains the request that follows the control command, if the return value is an error, it will inform the user control error. If any other kind of error makes the computer fail to receive the return value, it will automatically report the error and terminate the program after a period of time. On the basis of the error reporting function, this method greatly speeds up the execution of the program.

After solving these problems, my design idea is completed. First, I will build a virtual panel in C#. The virtual panel is similar with the real panel but more simple and intuitive. Then I will connect a computer to the TG8000 by using the Socket programming. The last step is to write functions to send SCPI. Once TG8000 receives the message, it will make response.

2.1.2 The Process of Writing of the Software

The first step is to build a virtual panel. Since there are three parts in the command, my software is made up by three windows as well, as shown in Figure 1, Figure 2, and Figure 4.

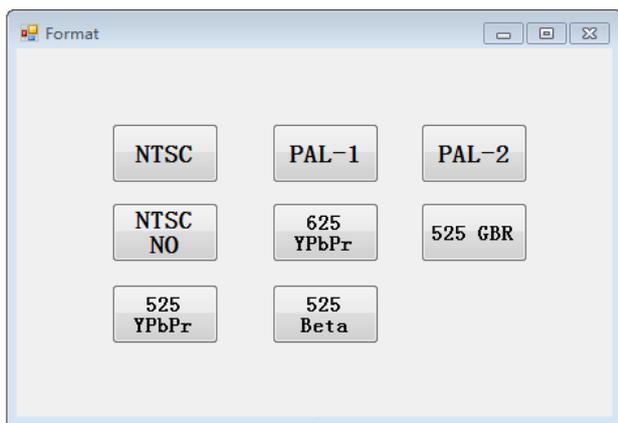


Figure.1. Format interface

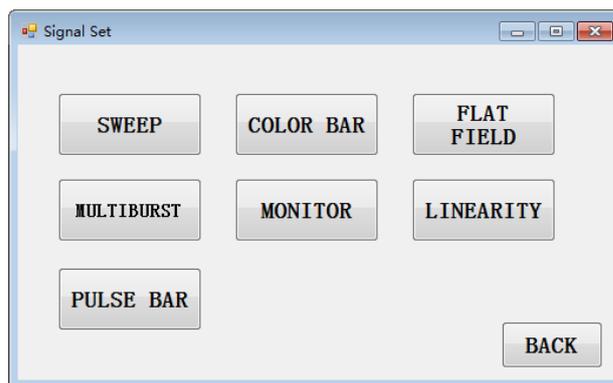


Figure.2. Signal Set interface

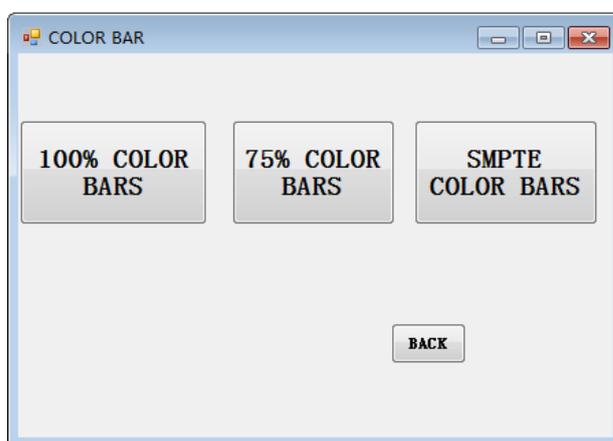


Figure.3. Specific signal interface

These three parts can give command “MMEMory:LOAD:SIGNal<module_name>,<signal_name>” a clear parameter. The only thing left to do is writing functions for the buttons. The software need to send commands through network, so I chose to use Socket to achieve this process.

Socket uses the client / server mode to solve the problem of establishing the communication. The connection process between the sockets can be divided into three steps: server monitoring, client request, connection confirmation.

- Server monitoring: the server socket does not locate a specific client socket, but is in the state of waiting for the connection,

- Client request: refers to the client socket connection request, the target is to connect to the server socket. To do this, the client socket must first describe the socket of the server to which it is connected, point out the address and port number of the server socket, and then make a connection request to the server socket,

- The connection confirmation: refers to the server socket to listen to or received by the client socket connection request, it will respond to the client request socket, create a new thread, the server socket description to the client. Once the client confirms this description, connection is established. While the server socket continues to be in the listening state, continue to receive other client socket connection request [5].

For my software, I know the IP address and the port number of the TG8000 and my computer, so I can easily build their connection. Besides, there are two Socket functions playing a key role. After setting the parameters, function “clientSocket.connect()” make my computer connect to the TG8000. Then I write the SCPI into string, and use function “clientSocket.send()” to send SCPI to the TG8000 [6].

2.1.3 Software Demonstration

After Part I have finished all the code, the last thing I need to do is to test the software. For example, if I want to set 100%SWEEP in PAL-1 format, I should run the program and chose PAL-1, as shown in Figure 4.

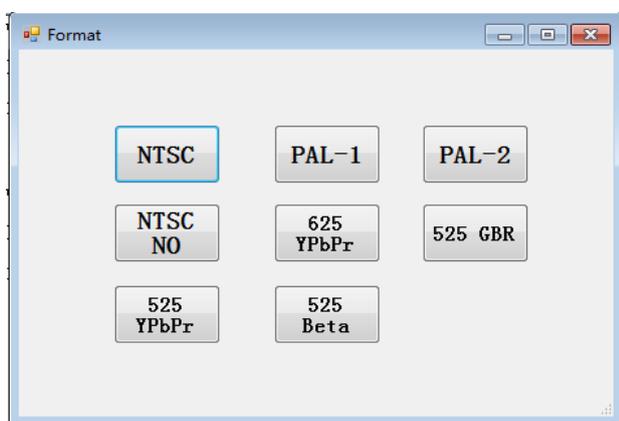


Figure.4. Final test 1

Then I will enter next screen and chose SWEEP, as shown in Figure 5.

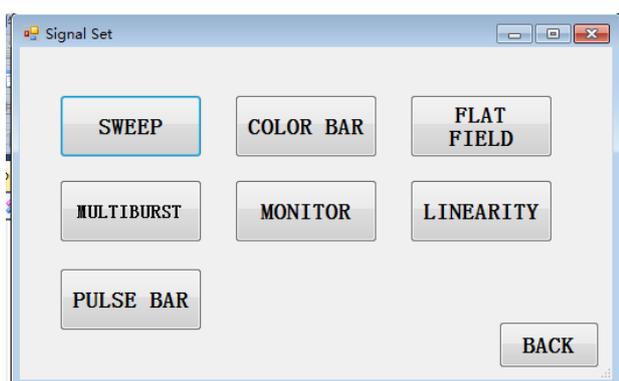


Figure.5 Final test 2

I chose 100%SWEEP, as shown in Figure 6.

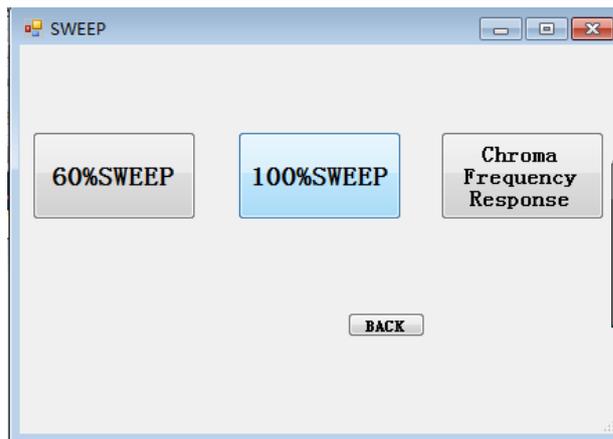


Figure.6 Final test 3

There is one way to verify this result. We can connect the TG8000 to an Oscilloscope, like VM700, as shown in Figure 7



Figure.7 Result in VM700

3 CONCLUSION

Table 1. Delay (ten times average)

	TGSetup	My software
Connect(s)	9.2	<1
Change signal(s)	32.8	3.6

We can see from the comparison that there is almost no delay for this remote control software. It is worth mentioning that there is no running error or system breakdown in my test.

Test signal generator is now widely used and a lot of problems followed. Now this software has been put into use in our university. Some instruments with not working button could be reused because of the remote control software. It is also used in experimental lessons when teachers start to demonstrate operation so that the

students can watch the operating process clearly. So I can say the remote control software expand the use of signal generator and improve the efficiency of users.

REFERENCES

1. Tektronix.TG8000 Multiformat Test Signal GeneratorPC Tools Technical Reference (2014)
2. Tektronix.TG8000-Multiformat-Test-Signal-Generator-Programmer-Manual (2014)
3. Liu Zhen. Design and implementation of SCPI parsing platform [J] (2012)
4. Tektronix.TG8000-Multiformat-Test-Signal-Generator-User-Manual_0 (2014)
5. Wang Wei Guo, Yang Guang Yu, Zhou HaiFeng. Application driven development of programmable instrument [J] (2010)
6. Carter. Socket principle and programming foundation [J] (2011)