computing@tanet.edu.te.ua
www.computingonline.net

ISSN 1727-6209
International Scientific
Journal of Computing

REMOTE-CONTROL MEASUREMENT SYSTEM BASED ON THE IEEE-488

Petr Cesak [1], Jaroslav Roztocil [2]

[1] Czech Technical University, Technicka 2, 16627 Prague 6, CZ, cesakp1@fel.cvut.cz, measure.feld.cvut.cz
[2] Czech Technical University, Technicka 2, 16627 Prague 6, CZ, roztocil@fel.cvut.cz, measure.feld.cvut.cz

Abstract: *The paper presents a complete system for remote measurement. The system is based on the client-server communication structure over the Internet. One part considers the client application – it is used by a user to control the measurement. Second part deals with the server – the server application itself and local connection the instruments over the IEEE-488 interface bus. The communication between client and server is built on TCP connection.*

Keywords: *Remote-control, measurement system, IEEE-488.*

1. INTRODUCTION

There is a course called "Measurement systems and their programming" at the department of Measurement, FEE CTU in Prague. The students learn to create a complete system for remote measurement.

The system is based on the server-client communication structure over the Internet. The measuring part of the system is based on the IEEE-488 interface bus by which the instruments are connected to the server.

Also from the point of teaching, the system is described separately in two parts. One part considers the client application – it is used by a user to control the measurement. Second part deals with the server – the server application itself and local connection the instruments over the IEEE-488 interface bus. Each student develops both application – client and server.

The LabWindows/CVI from National Instrument is used to develop a client application. The server is running on Linux and so Linux programming techniques are taught and used to develop the server application (including communication with instruments via VISA library). The communication between client and server is built on a TCP connection.

2. MEASUREMENT TASK

The task of the system is to measure V-A characteristics of a TTL logic gate. TTL logic gate is the Device Under Test (DUT) for the application. The V-A characteristic is a function of the input and output voltage, where the output voltage is the dependent variable. The V-A characteristic is measured for constant power supply voltage.

$$output = f(input, power) \qquad (1)$$

where *input* is the input supply voltage of the DUT, *power* is the power supply voltage of the DUT and *output* is the output voltage of the DUT.

Fig. 1 shows setup of the system for measuring V-A characteristics of a DUT. The system is composed from two supplies, one voltmeter and a DUT. One supply (called *Input*) is used to setup the input voltage of the DUT. The other one (called *Power*) is used to setup a constant power voltage for DUT during measurement. The output voltage of the DUT is measured by the voltmeter (called Output).

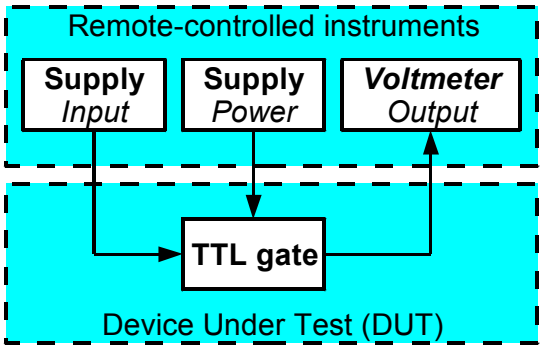

Fig. 1 – Setup of the system for measuring V-A characteristics of DUT.

3. MEASUREMENT SYSTEM

The system uses the client-server topology for communication (see Fig. 2). Two computers communicate over the Internet. The communication between client and server is built on TCP connection.

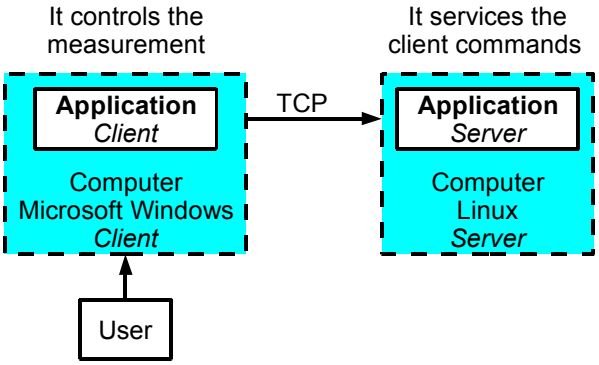

Fig. 2 – System based on the server-client communication.

The client application (on the computer with Microsoft Windows) controls the measurement. The server application is running on the computer with Linux. Server application services the client application commands and demands.

The server computer is physically connected to measurement instruments (two power supplies, one voltmeter) via the IEEE-488 interface bus (also known as GPIB). Fig. 10 shows this connection of measurement instruments. The server computer is also available as emulator – it will be described later.

4. CLIENT APPLICATION

Each student develops both applications – client and server. First, a student starts with developing the client application. The LabWindows/CVI from National Instrument is used to develop a client application. Table 1 describes the variables that can be controlled by users from their client applications.

Table 1. Available variables in client application

Variable	Description
Input voltage	User can set up the range of the input voltage. This is done by entering values – minimum, maximum and count of steps.
Power voltage	User can set up the range of the power voltage. This is also done by entering values – minimum, maximum and count of steps.
TCP address	The IP address of server computer.
TCP port	The listening TCP port of server application.

The client application has a user interface to control and visualize the measurement (see an example of user interface in Fig. 3 – created as a result of their work by students). It shows graphical results of measurements. For easier debugging, the communication log between the client and server applications is also displayed.

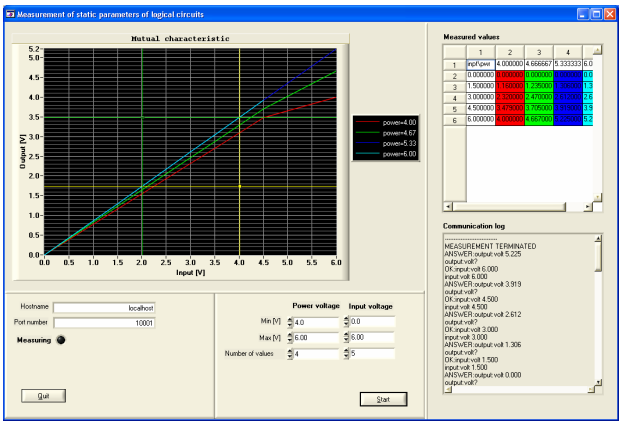

Fig. 3 –Client application user interface.

The client application is written as multithreaded application. There are two threads in the application (see Fig. 4):

1. **DISPLAY thread**. This thread shows graph with V-A characteristics of DUT. It also shows the communication log of TCP connection (data transferred between the client and server application). This log allows easy debugging when client or server application is being developed. The user interface is periodically refreshed during measurement of the V-A characteristics.
2. **COMMUNICATION thread**. This thread is started by user pressing the start button. The thread is responsible for communication with server and stores measured values into memory shared by all threads. Measurement setup is done by the user via controls in the user interface panel.

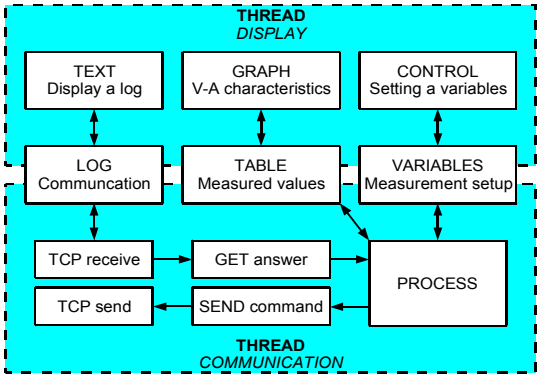

Fig. 4 – Threads in the client application.

The application uses mutex objects to protect shared resources. This allows student to understand that access to shared variables (or sources of

functionality) within an application with more than one thread of execution is restricted and has to be handled correctly.

5. SERVER EMULATOR

The server emulator application for Microsoft Windows was developed to help students in development of their client application. It is an emulator of the system for remote measurement (server-side part). This server emulator application includes also virtual devices – so it completely replaces the whole server measurement system. This application is available at website [1].

Fig. 5 shows the user interface of the server emulator application for Microsoft Windows. User can select a TCP port, start or stop the server. The additional information (like input voltage, output voltage, power voltage, communication log, count of commands) is displayed.

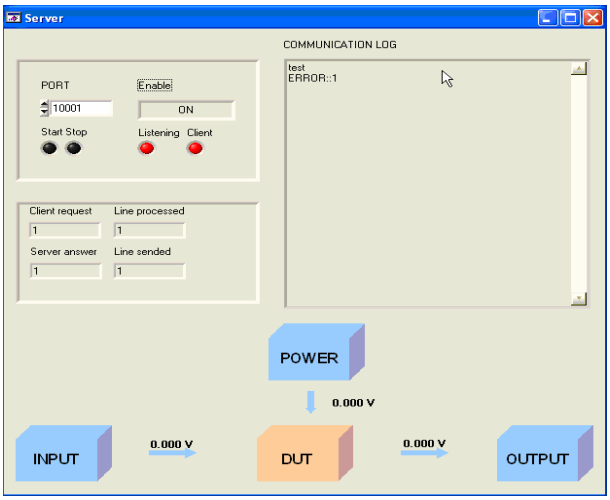

Fig. 5 – User interface of server emulator application for Microsoft Windows.

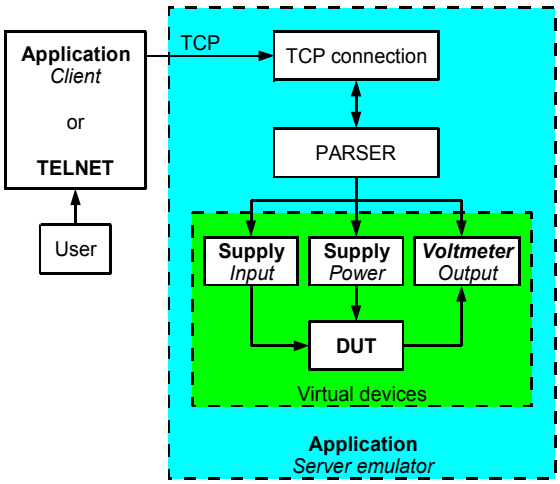

Fig. 6 – Internal structure of the server emulator application.

Users can use a client application or application like TELNET to make a TCP connection to the server emulator application (see Fig. 6). The communication protocol is described later in the paper.

There are implemented three modes of the DUT in the server emulator application (see Fig. 7-9):

1. **Mode 1: DUT is O.K.** DUT is implemented as the TTL logic gate. The output voltage is depended on the input and power voltage.

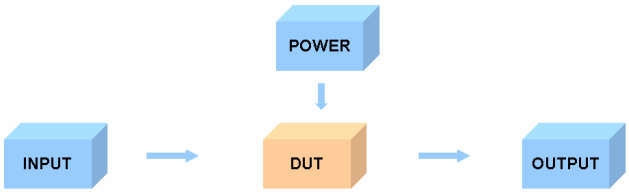

Fig. 7 – Mode 3 of DUT in the server emulator application. DUT is O.K.

2. **Mode 2: DUT is overloaded**. When the power voltage is higher than should be, the DUT output is depended only on the power voltage. User can return from this mode back to mode 1 by decreasing power voltage with no damage to the DUT.

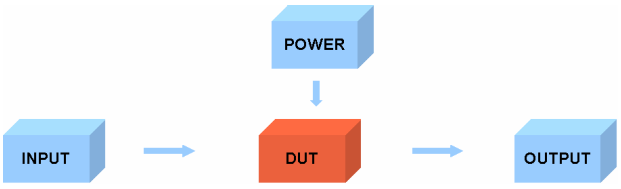

Fig. 8 – Mode 2 of DUT in the server emulator application. DUT is overloaded (higher power voltage).

3. **Mode 3: DUT is broken**. When input voltage is higher than power one – the DUT is broken. The DUT output voltage is constant value (zero). The user can not return from this mode back to other one – a restart of application is required.

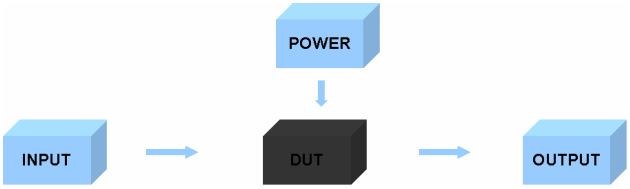

Fig. 9 – Mode 3 of DUT in the server emulator application. DUT is broken (wrong input and power voltage).

More information about the server emulator application has been published in [2].

6. SERVER APPLICATION

After successful development of the client application and testing it with server emulator, the students solve a server application. The server computer is physically connected to measurement instruments (two power supplies, one voltmeter) via the IEEE-488 interface bus (also known as GPIB). Fig. 10 shows this connection of measurement instruments.

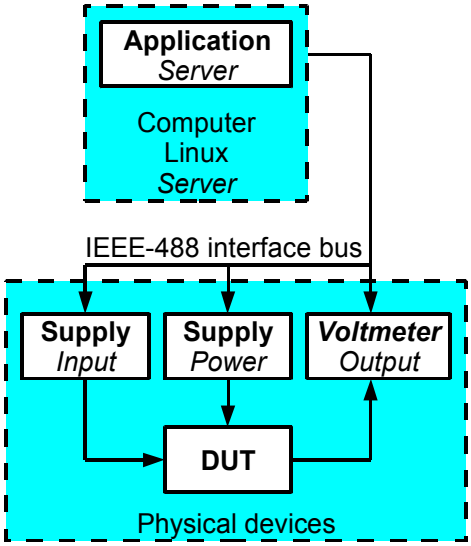

Fig. 10 – Connection of measurement instruments to server computer.

Linux server is used to develop and run the server application. Each student has an account on Linux server. Anjuta DevStudio is installed on Linux server to enable remote development of server application. This application can be download from website [3]. An example of the user interface is showed in Fig. 11.

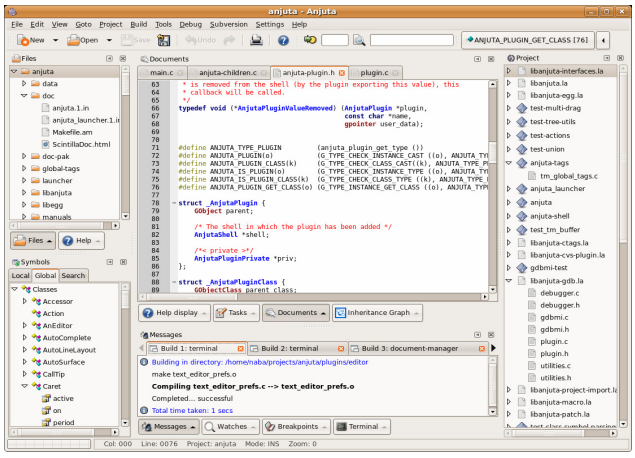

Fig. 11 – Anjuta DevStudio is used to server application development by the students.

Table 2 describes the variables which are controlled by user in a server application. The server application is running as a console application. Users can specify the variables' values before execution as command line parameters or in running application from console. Also additional information can be showed – communication log with connected clients and status of connected instruments.

Table 2. Available variables in server application

Variable	Description
Input voltage	User can set up the maximum of input voltage. This limits the maximum value of input voltage which can be set up by client application.
Power voltage	User can set up the maximum of power voltage. This limits the maximum value of power voltage which can be set up by client application.
Input supply	The IEE-488 bus address of the device setting input voltage of DUT.
Power supply	The IEE-488 bus address of the device setting supply voltage of DUT.
Output voltmeter	The IEE-488 bus address of the device measuring output voltage of DUT.
TCP address	The listening IP address of server application.
TCP port	The listening TCP port of server application.

The server application is a multithreaded program. There are two threads in the application (see Fig. 12):

1. **INSTRUMENT thread**. This thread is used to communication with instruments. It is sending command to instruments via VISA library. A reply is then written back to queue.
2. **CLIENT thread**. This thread is listening on TCP port. When client is connected to TCP port the thread services all client commands. New connection to TCP port creates a new CLIENT thread – so each client is serviced by a separated thread.

Mutex objects are also used here to protect shared queue.

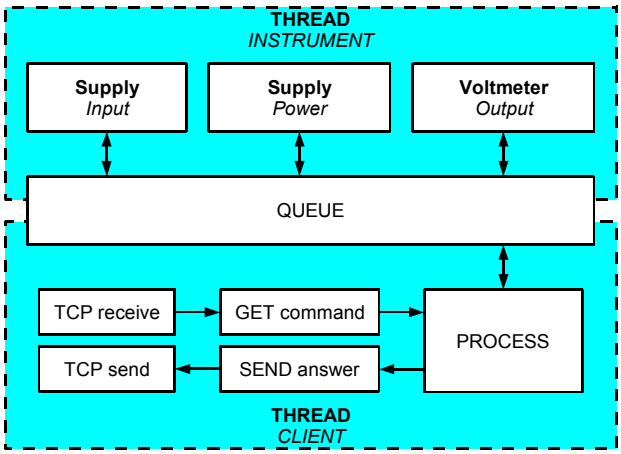

Fig. 12 – Threads in the server application.

7. COMMUNICATION PROTOCOL

TCP connection is applied to exchange information between the client and server application. Data transfers use the message protocol specially designed for this purpose. The message protocol contains commands and replies (Table 3). Both commands and replies are represented by strings. Each command (answer) is separated by characters CR, LF (identifying a new line). This allows using application like TELNET to communicate with the server application.

Table 3. Message protocol

Order	Sender	Data
1	Client	*Command*<CR><LF>
2	Server	*Answer*<CR><LF>

Server application sends an answer to each command received from a client. Table 4 shows type of commands and Table 5 shows type of possible answers:

1. **Type C1**. Client is writing a value to the device. The server can reply using type S1 or S3.
2. **Type C2**. Client is reading a value from the device. The server can reply using type S2 or S3.

Table 4. Type of commands

Type	*Command*
C1	*device*:*request value*
C2	*device*:*request*?

Table 5. Type of answers

Type	*Answer*
S1	OK:*device*:*request value*
S2	ANSWER:*device*:*request value*
S3	ERROR:*device*:*error*

Table 6 describes the meaning of variable in commands and answers.

Table 6. Variable in commands and answers

Name	Description
device	Device name.
request	Request to device.
value	Appended value to request.
error	Error number.

The list of device names is in Table 7. Table 8 shows the type of request for the supply instrument. Table 9 shows the type of request for the voltmeter instrument.

Table 7. Device name

Device	Instrument	Description
POWER	Supply	Power voltage of DUT
INPUT	Supply	Input voltage of DUT
OUTPUT	Voltmeter	Output voltage of DUT

Table 8. Type of request for supply instrument

request	Description
VOLT X.YY	Set supply voltage to value X.YY
VOLT?	Get supply voltage.

Table 9. Type of request for voltmeter instrument

request	Description
VOLT?	Measure a voltage

It error occurs in command server will reply with type of answer S3. The type of error is described in Table 10.

Table 10. Type of errors

error	Description
1	Syntax error.
10	Unknown device.
11	Device communication fails.
20	Unknown request.
21	Unsupported request.
30	Value or ? is missing.
31	Incorrect value.
32	Value not required.
33	Value out of range.

Table 11 is example of communication between the client and server – where no problem occurs.

Table 11. Example of valid communication

Index	Sender	Data
1	Client	power:volt 5.1
2	Server	OK:power:volt 5.100
3	Client	input:volt 1.23
4	Server	OK:input:volt 1.230
5	Client	output:volt?
6	Server	ANSWER:input:volt 4.683

All possible example of invalid communication is showed in Table 12.

Table 12. Example of invalid communication

Index	Sender	Data
1	Client	Client
2	Server	ERROR::1
3	Client	blabla:
4	Server	ERROR:blabla:10
5	Client	power:blabla
6	Server	ERROR:power:20
7	Client	power:volt
8	Server	ERROR:power:30
9	Client	power:volt 5.aa
10	Server	ERROR:power:31
11	Client	power:volt 5.00?
12	Server	ERROR:power:32
13	Client	power:volt 99.0
14	Server	ERROR:power:33
15	Client	power:volt 5.0
16	Server	ERROR:power:11
17	Client	output:volt 5.0
18	Server	ERROR:power:21

8. CONCLUSION

The remote-control measurement system was described in this paper. Students at the Czech Technical University use this model to learn and create a complete system for remote measurement. The measuring part of the system is based on the IEEE-488 interface bus by which the instruments are connected to the server. Overall the described system should be a model that student use during their work.

The system is based on the client-server communication topology. A TCP connection is applied to exchange information between the client and server application. The communication protocol was developed and described.

There were developed the server emulator application [2] for Microsoft Windows in LabWindows/CVI from National Instrument. It is an emulator of the system for remote measurement (server-side part). It helps students to develop their client application.

9. REFERENCES

[1] P. Cesak. *Server emulator application*. http://pck338-48.feld.cvut.cz/soft/server_.zip.

[2] P. Cesak, J. Roztocil. *Client-server based education tool for measurement applications*. 15th International Symposium on Novelties in Electrical Measurements and Instrumentation. Iaşi, Romania, 19 - 21 September 2007, in press/accepted.

[3] Anjuta DevStudio. http://anjuta.sourceforge.net.

Petr Cesak *was born in Hradec Kralove. He studied Computation and Automation Engineering during 1995–1999 on SPSE in Pardubice (www.spse.cz). In 2005 he graduated at CTU (Czech Technical University) in Prague at the Department of Measurement. He started his Ph.D. studies after his Master State examinations in March 2005. His current research interests include testing AD converter in microcontroller and development of measurement system software. His hobbies are programming and building hardware. He is familiar with microcontrollers like MSP430, ATMEL x51 and AVR, DSP, ARM. His website is www.cesak.com.*

Jaroslav Roztocil *was born in 1957. He received the M.S. and Ph.D. degrees in measurement technology from the Faculty of Electrical Engineering, Czech Technical University (CTU), Prague, Czech Republic, in 1981 and 1988, respectively. Since 1981 he has been working in the Department of Measurement, CTU, and was appointed as an Associate Professor in 1999. His current research interests include dynamic testing AD converters and modules, time and frequency metrology, development of measurement system software. Mr. Roztocil is an official Member of the URSI, a Member of the IEEE and a Member of the Technical Committee for Time and Frequency of the Czech Office for Standards, Metrology and Testing.*