Client-server based education tool for measurement applications

Petr Cesak, Jaroslav Roztocil

Czech Technical University in Prague, Faculty of Electrical Engineering
Technicka 2, CZ-16627 Prague 6, Czech Republic
Phone: +420-224 352 058, Fax: +420-233 339 929, Email: cesakp1 / roztocil @fel.cvut.cz

***Abstract*-**This paper describes an application called server emulator. It is one part of the system for remote measurement. The system is based on the server-client communication structure over the Internet.

TOPIC: TC4 – 19: E-learning and Education in Measurements and Instrumentation

I. 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. The LabWindows/CVI from National Instrument is used to develop a *client* application. The server is running on Linux and so Linux programming techniques have to be taught and used to develop a *server* application (including communication with instruments via VISA library). The communication between client and server is built on TCP connection.
The task is to measure V-A characteristics of 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.

II. System description

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

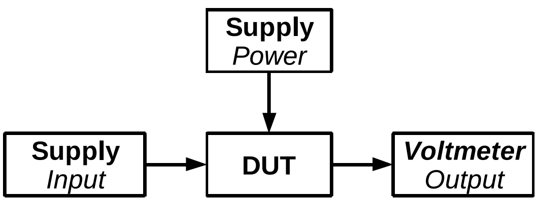


Figure 1. System for measuring V-A characteristics of DUT

Figure 2 shows a typical implementation of the system for measuring V-A characteristics of DUT. Two computers communicate over the Internet. The *client* application (on the computer with Microsoft Windows) controls the measurement. The *server* application is running on the computer with Linux. This computer is physically connected to measurement instruments (two power supplies, one voltmeter) via the IEEE-488 interface bus (also known as GPIB). *Server* application services the *client* application commands and demands.

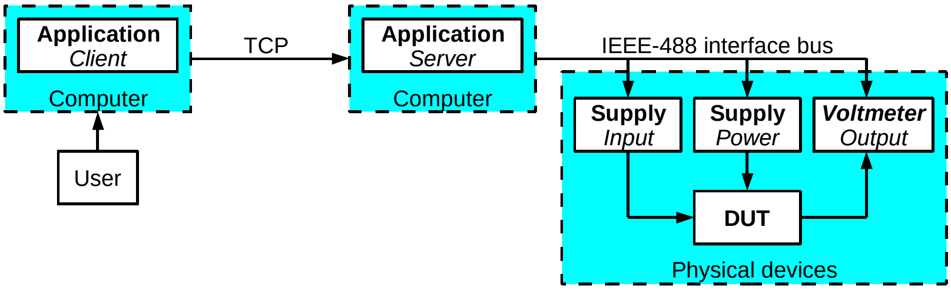


Figure 2. Implementation of the system for measuring V-A characteristics of DUT

III. Server emulator application for Microsoft Windows

Each student develops both application – *client* and *server*. First, a student starts with developing of the *client* application. The s*erver emulator* application for Microsoft Windows (Figure 3) was developed to help students in development of their *client* application. This *server emulator* application include also virtual physical devices – so it completely replacing the whole server measurement system. This application is available at website [1].

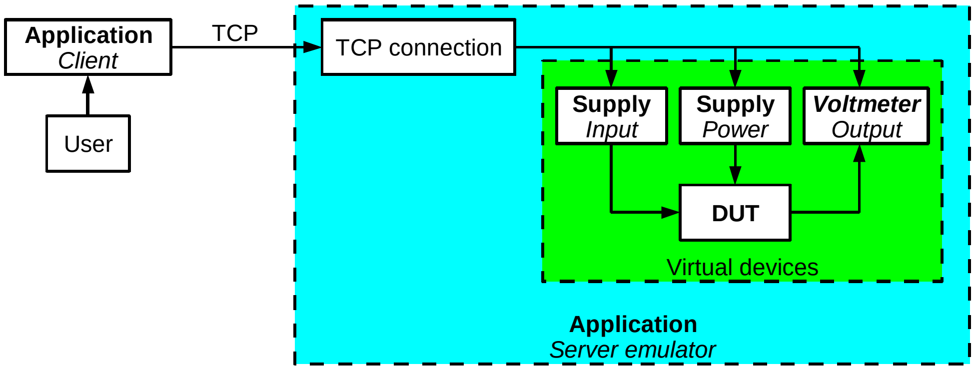


Figure 3. Server emulator application for Microsoft Windows

A. Communication protocol

The TCP connection is used to exchange information between the *client* and *server* application. It is a simple text based protocol – this allows using application like *TELNET* to communicate with the server application. Server application replies to each command received from a client. Each command is separated by characters CR, LF (identifying a new line – a new command). This communication protocol is available at website [2].

B. Software implementation

Figure 4 shows the implementation of the s*erver emulator* application for Microsoft Windows. There are three threads in application:

1. **TCP thread**. This thread accepts incoming TCP connection. TCP data are placed into FIFO memory. There are two FIFO memory block – one is for sending TCP data, the other one is for receiving TCP data. This allows sending a command from *client* application by one character (can be used *TELNET* terminal).

2. **LINE thread**. This thread gets a command from FIFO memory block. An answer to command is also placed to FIFO memory block.

3. **KERNEL thread**. This thread processes a command and always creates an answer to this command. It also contains the virtual devices – it is emulating a real system.

All shared memory variables are accessed using the locks (mutex).

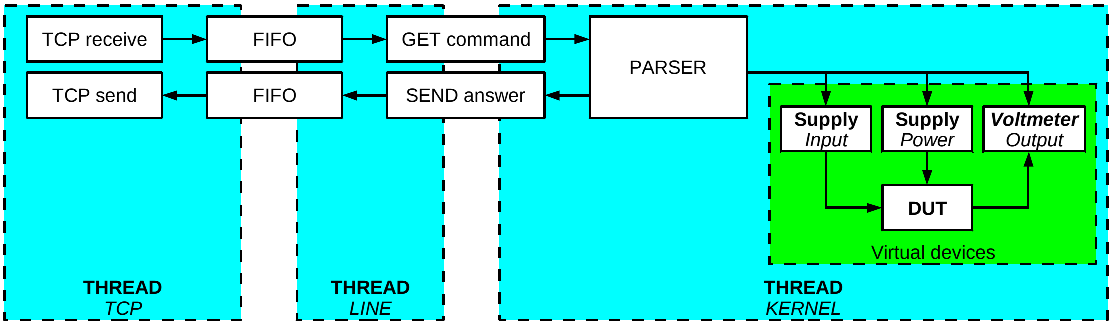


Figure 4. Implementation of the server emulator application for Microsoft Windows

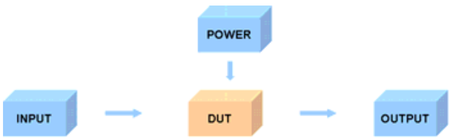
C. DUT Implementation

The DUT can be in one of the three modes (Figure 5):

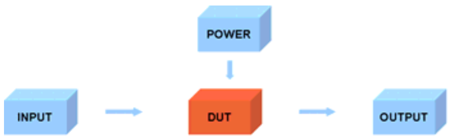
1. **Mode 1: DUT is O.K.** DUT is implemented as the TTL logic gate. The output voltage is depended on the input and power voltage.

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.

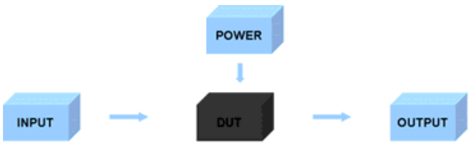
3. **Mode 3: DUT is broken**. When input voltage is higher than power one – the DUT is broken. The DUT output voltage is constant value (zero). User can not return from this mode back to other one – a restart of application is required.



Mode 1: DUT is O.K.



Mode 2: DUT is overloaded (higher power voltage)



Mode 3: DUT is broken (wrong input and power voltage)

Figure 5. The modes of DUT

D. User interface

Figure 6 shows the user interface of the s*erver emulator* application for Microsoft Windows. User can select a TCP port, start/stop server. The additional information (like input voltage, output voltage, power voltage, communication log, count of commands) is showed.

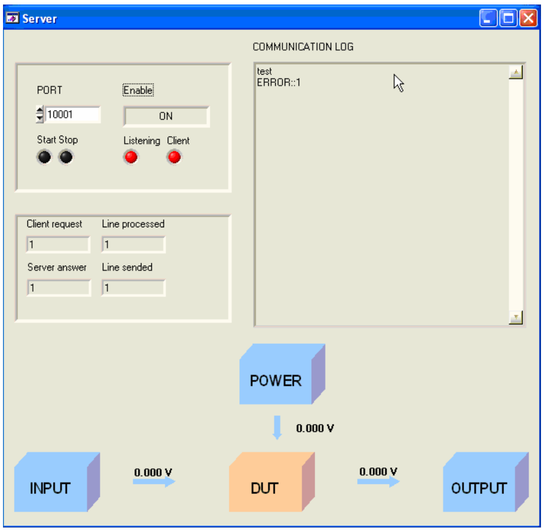


Figure 6. User interface of server emulator application for Microsoft Windows

IV. Conclusions

There were developed the s*erver emulator* application [1] 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 in development of their *client* application.
The s*erver emulator* application communicates with client via TCP connection. Also *TELNET* application can be used to send a command to the s*erver emulator* application. The communication protocol [2] was also developed. The DUT has three modes – to emulate a real situation (for example power voltage overload).

References

[1] CESAK, P.: Server emulator application. http://pck338-48.feld.cvut.cz/soft/server_.zip
[2] CESAK, P.: Communication protocol. http://pck338-48.feld.cvut.cz/vyuka/X38PLS_Protokol.pdf