

Remote Laboratory for Supporting e-Studies in Electronics

Yevgeniya S. Sulema* and Vladimir M. Cvjetković**

*National Technical University of Ukraine “Kiev Polytechnic Institute”

**Faculty of Science in Kragujevac, Serbia and Montenegro

Abstract— E-learning becomes more and more popular today. It gives an opportunity for studying to persons who are not able to attend traditional courses and lectures for some reasons. In areas such as humanities or economical education, E-trainings provide a student with practically the same quality of learning materials. But in case of technical education a student must have traditional laboratory lessons to obtain practical knowledge on the subject. It concerns chemistry, physics, electronics and other technical disciplines. However, modern network technologies allow creating remote laboratories for supporting e-studies. The idea of using the remote lab for support of e-teaching electronics is discussed in the article. The structure and functionality of the remote laboratory are described. The software and hardware parts of the remote lab are considered and presented, as well.

Index Terms— E-learning, on-line experiments, remote laboratory

I. INTRODUCTION

Practical experimental work in laboratory is an important component of education process in general, and particularly in distance learning where special attention should be paid due to absence of classical management of learning process on one side, and specifically different surrounding and conditions characteristic for remote access to experimental equipment on the other. Remote access to experiment and experimental equipment has various limitations and demands that can be met up to a certain extent. The main idea and intention of this system is to make and provide the basic functionality of laboratory conditions and requirements usual for practical and experimental work, on-line, from the side of remote student that uses PC and internet browser as the main and only tool.

Two basic approaches are possible. One is the intention to make very similar conditions and user interface to instruments that are in the real lab, to make available all controls, switches and buttons existing in real equipment, also available for on-line student. Such approach is very demanding one, and it is essentially impossible to create identical conditions for remote user and student in the real lab working on real equipment.

The other approach is more flexible one, i.e. the intention is not to create as similar conditions as possible, but to make some kind of adapted remote laboratory system that will provide the functionality based on the logical interconnection with theoretical material, serving as support for easier understanding and also for

introduction and illustration of basic notions characteristic for measurements. The presentation and user interface are accommodated to different conditions imposed by remote access. Only the most important controls, measured values, results are shown, and the user interface can be quite different comparing to laboratory instruments, but logically quite functional and sufficient for complete understanding. In this paper, the second approach is used, as it is primarily for illustration of main characteristics, functionality and capabilities of implemented prototype of remote laboratory system.

The proposed system will be presented and discussed from the point of view of basic structure, software and hardware, reasons for adopting such structure, possible changes and flexibility of both the general system structure and implemented system especially from the point of view of scalability, various places and roles that such system can have, further development that can be carried out both by system designers and programmers on one side and students of undergraduate and graduate studies as part of their regular studying process or as important kind of their research work for specialist, masters or higher degrees.

II. SYSTEM STRUCTURE

System that was designed, developed and implemented for the purpose of remote access to laboratory equipment and experimental work for on-line support of e-learning is called RemoteLab. It is prototype system that is intended to support experiments in the area of electronics, but its structure poses no limitations on other types of experiments also, such as in physics, mechanical engineering, or some similar field of engineering or science.

The basic structure of RemoteLab system is given on UML [1] general class diagram in fig. 1 below. Each of the classes in fig. 1 corresponds to one of the layers in RemoteLab system. Classes and associations in fig. 1 form short sentences that express relations between them in simple way. Sentences can be read in the direction signed by small arrows in the following way:

- Web user interface displays experiment
- Programmable device performs experiment

Cardinalities of association “Displays” are designated and are equal to 1. Each web page corresponds to exactly one experiment, and vice versa. Cardinalities of other association “Performs” are different as one Experiment can be performed by at least one or more programmable devices, while each programmable device can be part of

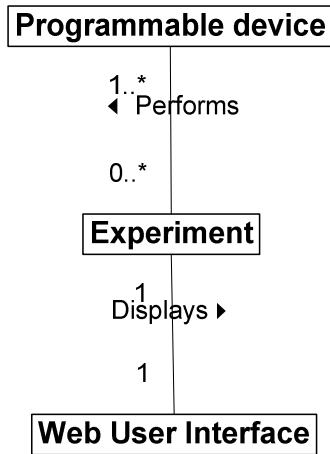


Figure 1 RemoteLab class diagram

more, one or temporarily not used by any experiment It is quite general structure that can be implemented using various software and hardware in different ways. The main goal and intention of such configuration is to fulfill number of requirements.

- Universal configuration suitable and easily adopted to various kinds of experiments
- Structured configuration that can be implemented using various hardware and software
- Easily scalable configuration that can be extended to meet the following requirements:
 - Increase in number of users
 - Increase in number of experiments
 - ❖ Introduction of new experiments
 - ❖ Increased number of existing experiments
- Consists of hardware / software that are not too specific and intended more or less for general use
- Control of users competing requests for the same experiment or the same experiment equipment

Clear distinction and defined communication between classes on different levels aids to minimum dependence between classes and also enables reusability and independent changes between classes, as long as interfaces between classes remain the same.

A. Web user interface

Web user interface is on the highest level, and serves for communication with remote user – student. Web interface for RemoteLab consists of two parts. The first part is theoretical introduction and explanation for experiment that will be performed. It also includes precise instructions on how to perform the experiment, the meaning of all values that should be entered or selected, and values that are obtained as the result of experiment. Besides textual information about experiment, it supports graphical presentation of measured data in experiment that is important for full understanding of performed experiment. With graphical presentation it is very easy to see that presented data are from the real world, obtained on real equipment with all the influences that can not be avoided in real experiment.

Together with basic shapes of obtained lines that characterize the observed dependencies, the unavoidable noise is also present, that can be seen on screen shots that follow. Web interface should also contain means for control of the user access to the system. The structure of Web user interface is shown in fig. 2. It consists of classes for each experiment that inherit top class.

B. Experiment

Class or layer named Experiment is the middle layer of RemoteLab system. It contains the so called “business rules” or in this case the coded logic needed for each experiment. The logic is used for:

- collecting the user input data from the corresponding web page
- checking the data consistency
- preparing and sending data to programmable devices for equipment management
- full control and management of the experiment process using the programmable devices
- accepting the data measured in experiment from programmable devices
- data processing and calculation of indirectly measured quantities
- preparing and sending data to web server

Structure of the Experiment layer is given on the class diagram in fig. 3.

This level is the central part of the whole system. Each of the classes in fig. 3 that inherit from Experiment class is devoted to particular experiment. As every experiment must use one or more programmable devices, the class for each experiment has coded knowledge for data it receives from the web page, for programmable devices it uses, for preparation and setting devices for measurement, devices activation, control and management of the experiment process, receiving data from experiment, processing data and sending processed data to web page for display.

C. Drivers for programmable devices

Software drivers for programmable devices are at the lowest level of RemoteLab system. Depending on the type of programmable devices, they directly access the hardware or use manufacturer’s software libraries for accessing hardware. They are designed for efficiency and reuse, as the every experiment must use one or more programmable devices. Every programmable device is represented by one class or hierarchy of classes.

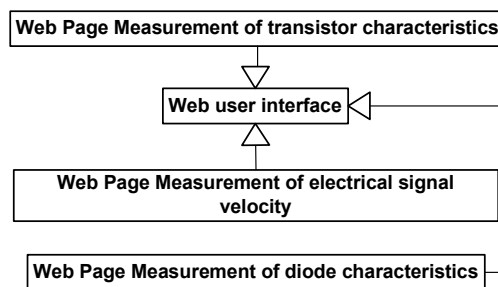


Figure 2 Classes for web user interface

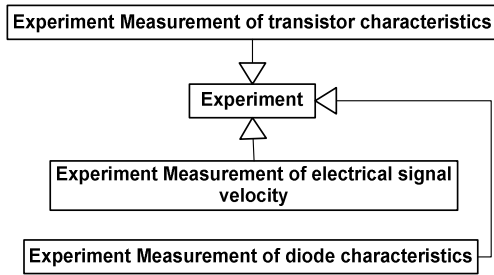


Figure 3 Classes for Experiment layer

The structure of the device determines the corresponding structure of classes that represent the programming model for each type of programmable device. Top class represents common features of the instrument, while the descendent classes that inherit top class correspond to logical or physical parts of device. Setting the devices and functional properties of devices are modeled as class methods, which are accessed by calling corresponding methods of instantiated classes.

III. PROGRAMMABLE DEVICES

Programmable devices that are used in prototype system consists of Data acquisition system (DAS) and Digital oscilloscope (DO).

DAS is Burr – Brown PCI 20000 [2] compatible. It has four standard sections:

- Analog digital (A/D) section with 8/16 differential/single ended inputs, 12 bit resolution, with programmable gain amplifier (PGA). A/D is used for direct voltage measurements in experiment circuits, and can be also used for indirect measurement of non electrical quantities by using the appropriate converters with standard voltage output. Measurement of electrical current with A/D is also possible, but indirectly by measuring voltage on resistor with known value.
- Digital analog (D/A) section with two channel voltage/current outputs, 12 bit resolution. D/A can be used for generation of necessary voltages and currents in experiments for two basic purposes. One purpose is to provide necessary conditions for experiment performance. Depending on particular experiment, voltages/currents can be constant acting as parameters, or changing when it is important to measure and show the functional dependence between observed values, with generated voltage as independent variable. The other purpose is to control some experimental equipment that is not programmable, but that can be managed by using voltage or current signal.
- Digital input/output (DIO) section with four digital 8 bit I/O ports. Each port can be programmed either as input or output. When acting as output, it can be used similar to D/A section for controlling equipment with standard TTL digital signals. Output digital signals can be

also thought of as communication with outer equipment. Digital inputs are mainly used for communication purposes, to read some register on piece of equipment that is connected to.

- Counter timer section (CTS) with referent 8MHz crystal controlled time base and four 16 bit counters. It is possible to generate stable signals of frequency from 2 MHz to 2 mHz, as the module of counting is $2^{16+16}=2^{32}$, that makes approximately $4 \cdot 10^9$. It is also possible to count some events in experiments, or to measure time intervals.

Program access to DAS is possible using software drivers for programming languages, specialized software such as National Instruments Lab View, Visual Designer or direct access from program to registers in memory map. Registers control the behavior, determine the status of DAS and contain the measured values. Programming with registers is actually used in RemoteLab system. In fig. 4, the structure of classes that represent programming model for DAS is presented. It can be seen from fig. 4 that class structure directly corresponds to structure of DAS.

Second programmable device - DO is Hewlet Packard (HP) 54502A [3, 4] with two channels and 400 MHz maximal frequency. DO can be thought of as very fast A/D converter with graphical display and lot of various controls. When possible, direct analogy with classical analogue oscilloscopes is kept. One of the basic advantages of digital oscilloscopes to analogue is that they have digital memory and therefore can be used for measurement of very short transient processes, not just periodic processes as analogue oscilloscopes. It also has built in functions for various standard measurements, such as frequency, period, time delay, rms, rise time, fall time, max and min values, etc. Another advantage is that it can be connected with PC computer, in this case using HPIB (Hewlet Packard Interface Bus) or GPIB (General Purpose Interface Bus) [5] and completely controlled from PC. It can also transfer all measured values to PC in selected measuring units.

Mentioned programmable devices DAS and DO enable large number of measurements and control of experimental equipment.

IV. SOFTWARE FOR REMOTELAB IMPLEMENTATION

Software that was used for implementation of RemoteLab system was chosen on the basis of requirements of project and requirements of used programmable devices [6]. The basic requirement was to use flexible software that could cover all aspects of

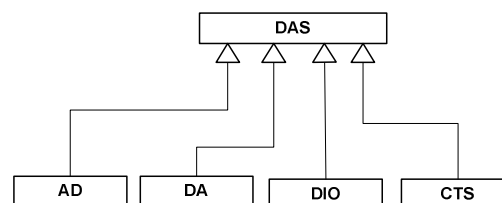


Figure 4 Classes for DAS

project and that could be easily integrated into fully functional application. More detailed, sub requirements for project were following:

- Programming on the side of web server for implementation of the web interface
- Direct access to memory map of DAS
- Programming language that can be used with command library for HPIB interface and DO
- Up to date platform that supports powerful programming languages for rapid application development (RAD) and with integrated development environment (IDE)

As the requirements are quite different and wide, it was not possible to select only one platform and programming language for all tasks and requirements, but it was possible to choose the basic platform that fulfils the majority of requirements and that could be easily integrated with parts implemented in other software. The basic platform is Microsoft .Net and Visual Studio .Net 2003. It offers powerful web server programming with ASP.Net class library [7] that can be used with .Net languages of which C# [8, 9] was chosen. Classes for DAS were implemented in C++ [10] as it offers direct memory access using pointers. Programming the DO however, required using some obsolete software, as the command library for DO and HPIB was very old, and couldn't be used from MS C++. Command library was designed for some old versions of programming languages, of which only Turbo C++ could be used. .Net platform and C# were used for implementation of two top layers - Web user interface and Experiment, while the lowest layer Programmable devices was implemented in MS C++ 6.0 and Turbo C++ 1.0.

V. REMOTELAB PROTOTYPE

Implementation of the prototype RemoteLab system directly follows from the logical system model presented with class diagrams. Used software – programming languages and platform itself does not put any limits for deployment on contemporary PC computers. However, the used hardware, namely the programmable devices put substantial limits on PC computers that can be used. DAS requires the PC computer of previous generation, as it was designed for ISA PC slot, and PC card for HPIB also requires the same type of slot. At the same time, PC's with ISA slots are not suitable for .Net platform, as they lack necessary computing power. In order to find a proper solution, a deployment system with two PC's was used. One of the PC's, the PC 2 that can be contemporary PC without ISA slots, was used as web server with installed .Net platform. Top two layers, the Web user interface and Experiment are physically deployed on PC 2, and implemented in C# for .Net. The third layer, Programmable devices, was deployed on PC 1 that is PC of previous generation, with ISA slots. On that PC, both cards for DAS and HPIB were installed, together with implementation of the third layer in MS C++ 6.0 for DAS and Turbo C++ for DO. In order for this distributed application to work, some kind of connection for communication between PC1 and PC2 was necessary. The used solution is based on TCP/IP protocol, and windows sockets interface as communication standard supported both on .Net platform and Win 98 on PC 1.

Described solution is presented on UML deployment diagram in fig. 5.

In general case, with some newer generation of programmable devices, the simpler structure is possible that consists of only one PC. One PC in principle can handle all the jobs and requirements of the basic RemoteLab configuration. Although configuration in fig. 5 was necessary, it also has some advantages when compared to configuration with one PC. PC for measurement and PC with web server can be placed anywhere on quite arbitrary places, as they communicate with TCP/IP protocol, and the only condition is to be connected to Internet. Besides convenience of arbitrary placing, it has benefit of distributing computation load on two PC's that is good solution in the case of more intensive use by many users and many experiments. Configuration of RemoteLab system with three layers is easily scalable and can be deployed in many different ways depending on actual load, usage of the system and needs for computational power. Possible scalability of the system will be discussed later.

VI. EXPERIMENTS

Prototype system has some experiments implemented that can serve as illustration of system functioning and for describing of some specific characteristics, that were not mentioned in general system overview. The system is under development, and currently three experiments are implemented. Experiments are from the area of electronics, and taken from the set of laboratory experiments for students of the third year of Physics in subject Electronics with telecommunications. Current implementation of experiments is also prototype, which can show some system characteristics, but needs to be further developed for real usage. Implemented experiments are:

- Measurement of electrical signal velocity through the coaxial cable
- Measurement of semiconductor diode characteristics
- Measurement of transistor characteristics

The home web page for RemoteLab system with experiment list is presented in fig. 6

A. Measurement of electrical signal velocity through the coaxial cable

Measurement of electrical signal velocity is based on using the DO as the main programmable device and measurement instrument. The idea is to measure the time necessary for electrical signal to run through the coaxial cable of known length. Actual coaxial cable that is used is

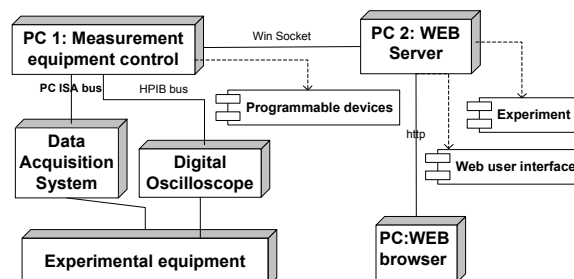


Figure 5 Deployment diagram of RemoteLab prototype system



Remote LAB



[Measurement of electrical signal velocity through the coaxial cable](#)

[Measurement of semiconductor diode characteristics](#)

[Measurement of transistor characteristics](#)

Figure 6 Home web page for RemoteLab system

standard 50Ω cable used for Ethernet LAN. Measurement setup consists of DO and function generator (FG) with variable frequency and rectangular waveform. Output of function generator is connected to channel 1 of DO using the standard T connector for LAN. The other side of T connector is connected to one end of coaxial cable for measurement. The other end of coaxial cable is connected to channel 2 of DO. Described setup is given in fig. 7. Coaxial cable of given length is connected between channel 1 and channel 2 of DO. Triggering and time base should be adjusted so that the rising or falling part of rectangular waveform is selected on both channels. If the time base is short enough, the time delay between channels caused by the finite velocity of electrical signal can be seen and measured. There are two different ways of measurement the time delay. One is to use the built in function of DO for time delay measurement between channels, and the other is to calculate the time delay in software - Experiment layer from acquired data from DO. Once the time delay is known, velocity v of electrical signal can be obtained as $v=l/\Delta t$, where l is the length of cable, while the Δt is measured time delay.

In combo box, various values of measurement duration can be selected. Selection of shorter time base gives better measurement, with lower absolute and relative measurement error.

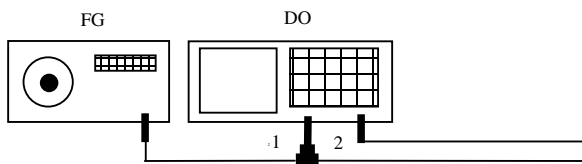


Figure 7 Experiment setup for electrical signal velocity measurement

Measured values for time delay and velocity with corresponding error are displayed after measurement. After clicking the button “Start measurement”, “Measurement status” goes to “In process”, and remains up to the moment when measurement results are ready and displayed.

After that “Measurement status” changes to “idle” as can be seen in fig. 8. If the measurement equipment is not turned on, after clicking the “Start measurement” button, text that equipment is turned off is displayed as measurement status. Image with graphical presentation of data obtained from DO can be displayed by clicking the hyperlink on the bottom of the web page. Web page with image of measured data is presented in fig. 9. Image with data in this experiment and all other experiments is obtained with methods of class that is developed for the purpose of automatic creation of bmp type image file from measured data obtained in experiment. It is designed to be universally used for displaying data from any experiment. There are methods for adding the first line – curve, all additional lines if necessary and for creation of file header. The image automatically adapts to maximum and minimum values and number of measured points. There are also methods for drawing vertical and horizontal lines, as can be seen in figure 9. Obtained results for electrical signal velocity correspond to data from text books. For the delay of 1 μs, cable length of approximately 200m is necessary. For 10.8 m cable length in this case, measured time delay is 55 ns, which is good agreement.

B. Measurement of semiconductor diode characteristics

In experiment for semiconductor diodes, two types of diodes can be measured, rectifier and zener diodes. This experiment is performed by using the DAS only.



Measurement of electrical signal velocity through the coaxial cable

Time per div 500ns

Measurement status: idle

Coaxial cable length (1082 +/- 1) 0.01 m

Time delay (550 +/- 5) 1E-10 s

Signal velocity (197 +/- 2) 1000000 m/s

[Image of the measured data on digital oscilloscope](#)

[Home](#)

Figure 8 Web page for velocity measurement

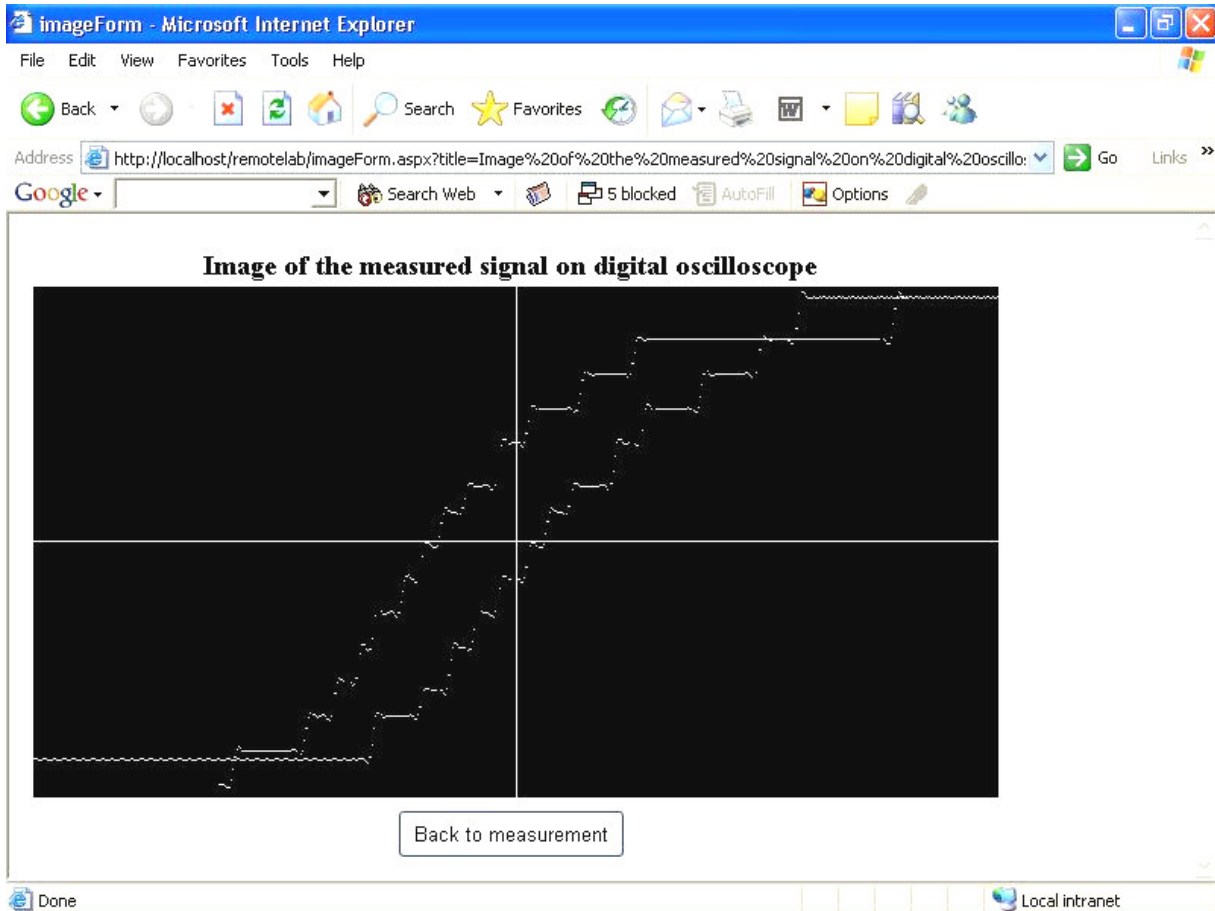


Figure 9 Web page with image of measured data from DO

As the measurement of analogue voltages is required, and strict timing of measurement is not important, it is performed by using the AD and DA sections of DAS. Characteristic diode voltages are measured in this experiment. It is based on recording the $I(v)$ dependency, that gives changes of diode current I from diode voltage v . Interval of change for diode voltage is symmetric, from approximately $-10V$ to $+10V$. Negative voltages correspond to inverse diode polarization, while the positive correspond to direct. Voltage change is obtained by generation with DA section, with specified number of points, in $\pm 10V$ interval. Diode current is measured indirectly, by measuring the voltage on serial resistor of known resistance of $1k\Omega$. Web page for diodes is presented in fig. 10. Type of diode rectifier or zener can be selected from the combo box. In other combo box, the number of measurement points with DAS can be selected.

From measured data obtained in experiment, characteristic diode voltages for direct and inverse polarization – zener are calculated and displayed on web page.

As with previous experiment, graphical presentation of data measured in experiment are available on the same web page for image, but now with different data. Graphical presentation of data measured in experiment for diode – zener diode BZ7, is given in fig. 11.

Calculated values of direct and zener voltage in figure 10 correspond to two “knees” in diagram in figure 11. The left “knee” corresponds to zener voltage, and the other to direct voltage.

C. Experiment for measurement of transistor characteristics

Experiment for measurement of transistor characteristics is similar to measurement of diode characteristics. In this case, transistor current amplification with common emitter is obtained in experiment. DAS is used very similar to previous experiment for diode, with exception that it is necessary to additionally generate base current as parameter in measurements of dependence of collector current $I_c(v_c)$ from collector voltage v_c with base current I_b as parameter. As previous, collector current is measured as voltage on collector resistance. Base current is generated with second DA channel. Web page for transistor measurement is presented in fig. 12. In the first combo box, type of transistor can be selected. In the case of PNP and NPN transistors, different voltage ranges for collector voltages are used, positive for NPN, and negative for PNP type. Selection of base current means difference between values of I_b as parameter, starting with value of $0\mu A$.



Measurement of diode characteristics

Diodes list

Number of points

Measurement status:

Measured direct diode voltage 0.723 V

Measured inverse zener voltage -6.598 V

[Image of the measured data for diode](#)

[Home](#)

Figure 10 Web page for semiconductor diodes

Selection of number of lines is equal to number of different parameter values for I_b . Number of points have the same meaning as in previous experiment. Diagram for measured data obtained in transistor measurements is presented in fig. 13. It can be seen that basic shape of lines is well defined, but that some points are distracted from the basic shape due to influence of unavoidable noise that accompanies every measurement.

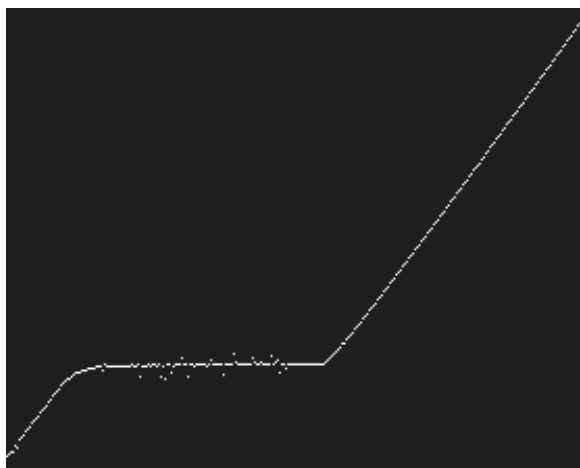
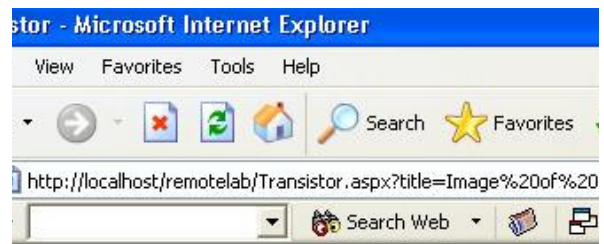


Figure 11 Graphical presentation of measured data in experiment for zener diode



Measurement of transistor characteristics

Transistor types

Base current uA

Number of lines

Number of points

Measurement status

Current gain 286

[Image of the measured data for transistor](#)

[Home](#)

Figure 12 Web page for transistor measurement

If the basic shape is well defined, that means the average value is good representation of the set of points.

The transistor current amplification is calculated as averaged value in the region of linear dependence of collector current I_c from base current I_b .

D. Concurrent requirements for same experiment

As the RemoteLab system is designed to be used by many independent users, it is quite possible that two users

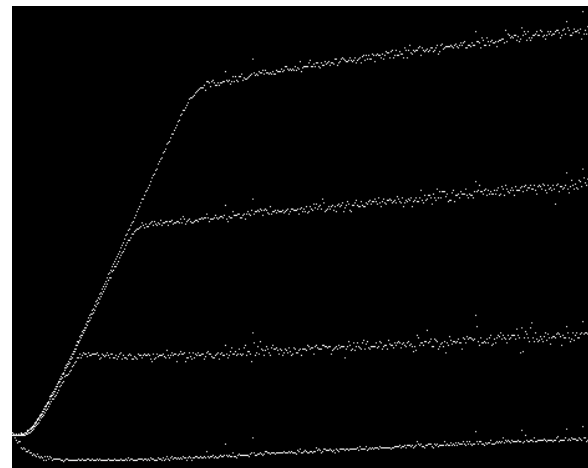


Figure 13 Graphical presentation of measured data for transistor measurements

try to start the same experiment simultaneously, or to try to start experiment while it is already in process – started by some other user. In the case of basic configuration described here, there is physically only one set of programmable devices for each experiment. Depending on the type of programmable device, it may be possible to use the same device simultaneously for different purposes. The word simultaneously means of course parallel tasks that are executed on multitasking operating system. Measurement on DO is specific, as once started, it can not be interrupted, in order to start some other measurement. However, the DAS is different, as it has four independent sections that can be used simultaneously. The Win socket connection between PC's is also important for this consideration. On the side of PC 1, when acting as server – listening for the call, one port is used for all experiments. PC 1 receives the call from PC 2 when the experiment is started.

PC 2 acts as client when initiates the experiment on PC 1, and also acts as server, when receiving data from experiment that are sent by PC 1, acting as client in that case. PC 2 has ports devoted to each experiment, on which PC 1 acting as client sends data to PC 2. In this implementation, simultaneous access to the same experiment is not allowed, while simultaneous access to different experiments is allowed, provided that they do not use the same hardware components. It is achieved on the side of web server, by using application variables. Application variables keep the current state of each experiment, and in each user session the value of application variables are examined. Depending on the state of experiment, user can start the experiment if it is not already started, or get the message that the

experiment is in process, and to try later. That situation is presented in fig. 14.

Explorer window on the right has focus as it has deeper frame color. That means the Explorer in the left window started the experiment before the Explorer in the right window. The result is obtained in the left window, while in the right, there is message that equipment is in use.

VII. SCALABILITY OF THE SYSTEM

Deployment of the prototype system shown in fig. 5 is necessary due to hardware requirements of used programmable devices. With different, more up to date equipment, it is possible to put everything just on one PC. But, the distributed configuration of the RemoteLab system has some advantages, as the total workload is distributed on more than one PC. It also has benefits that PC's do not have to be physically close, as they have network connection. Basic configuration as presented here has limitations with number of hardware devices that can be controlled. In order to control and manage many experiments, each must be fully controlled by programmable devices. Programmable devices can manage finite number of experiments. For new experiments, new programmable devices must be installed and used. It can be performed in various ways.

There are two main strategies to increase the number of experiments controlled by RemoteLab. One is to increase the number of programmable devices I/O for measurement and control, which will enable more experiments to be performed, as the I/O are devoted just to one experiment. The other is to use existing I/O for measurements and control, and to introduce switching

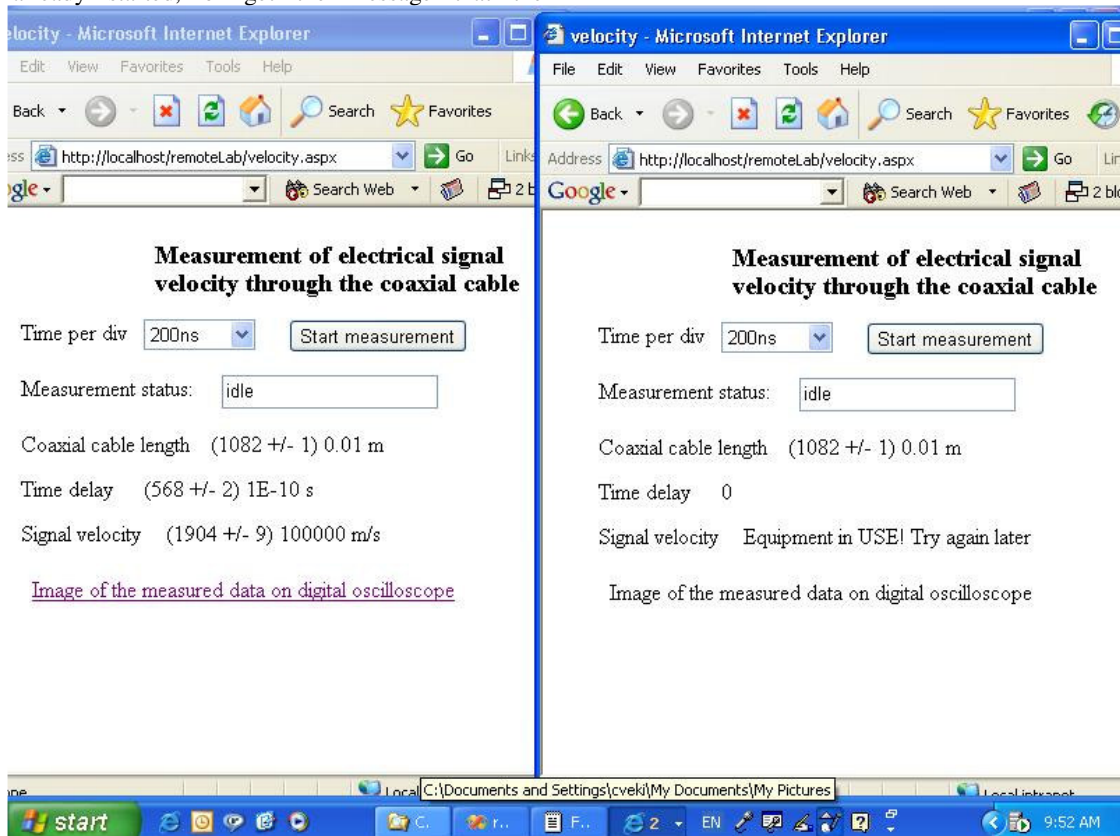


Figure 14 Simultaneous requests for the same experiment

circuits that will use existing I/O for more than one experiment. That approach also requires to increase the number of I/O for management of switching circuits. In both approaches it is necessary to install new PC cards with programmable devices. As each PC can support just finite number of expansion slots, introduction of new PC's might be necessary. But, adding new PC's for control of programmable equipment is very easy in distributed configuration as presented in figure 5. It is necessary to connect new PC's on network, and to add software for programmable devices and new experiment.

VIII. SYSTEM USAGE AND DEVELOPMENT

RemoteLab system can be useful and used by many categories of users ranging from regular students at various levels of education system, to remote users that relay especially on remote access to education materials. RemoteLab could be used as addition to regular laboratory exercises, or if significantly developed as replacement of part or all exercises in particular subjects. It can be also used as part of educational process not just for remote access to experiments, but as system to be studied from various aspects – web server programming, software experiment management, management of programmable devices, network connections of PC's that constitute the RemoteLab system, etc. This approach can lead to further system development by advanced students as part of their seminal, graduate or postgraduate activities, exams or some other kind of engagement. Finally, this system doesn't have to be used only for educational purposes, as its structure is general enough to be used in other areas also, such as for monitoring and control of remote systems, or for control and management of some other kind of experiments in real research.

IX. CONCLUSION

The presented RemoteLab system is intended to support e-learning systems enabling remote access to experiments with opportunity to control the experimental equipment, to obtain real data from experiment that was performed on user request. The obtained results are from the real experiment, and have unavoidable influence of noise that accompanies every measurement and every real experiment. In this particular case, the presence of noise is desirable as it is characteristic of real experiment, and remote student can see and feel the real experiment. Some experiment examples are from the area of electronics, but it is not the limitation of the system as it can also support experiments from other areas provided that experiment can be set up and performed only using programmable devices that actually control and manage the experimental equipment. Experiments that require some human presence in order to make some movements or settings not possible by programmable devices are not good candidates for remote access.

Basic configuration of RemoteLab system consists of three main parts or levels. The top part of the system is web user interface for experiment display, middle part for full control of experiment, data checking, processing of experimental results and at the lowest level are drivers for programmable devices. Such configuration is general enough to be supported and implemented by various kinds of software and deployed on various kinds of hardware. The choice of used software and hardware for RemoteLab prototype was discussed, from the point of general requirements for the system and hardware limitations.

Implemented experiments in the prototype are more for the illustration of some system characteristics, and need further refinement and elaboration. Simultaneous access to experiments is possible in basic configuration, but only for different experiments. As the demands grow with increasing number of experiments, system can be easily scaled by adding new programmable devices and PC's. System scaling is also possible on the other two layers, for experiment organization and web server in the case of increased number of users.

Using of the system can be for educational purposes, but also for some real life tasks such as monitoring and control of remote systems or for experiments in scientific research.

X. REFERENCES

- [1] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modelling Language User Guide, Addison Wesley Longman, Inc. 1999
- [2] Burr-Brown PCI 20000 reference
- [3] HP 54502A 400MHz Digitizing Oscilloscope Front-Panel Reference
- [4] HP 54502A 400MHz Digitizing Oscilloscope Programming reference, 1989
- [5] HP Using the HP-IB Interface and Command Library
- [6] A. M. Zikic, Practical digital control, Ellis Horwood Limited, 1989
- [7] A. Richard et al, Professional ASP.NET 1.0, Wrox Press Ltd, 2002
- [8] Jason Price, Mike Gunderloy, Mastering Visual C#.NET Sybex, 2002
- [9] Microsoft corporation with Jeff Webb, Developing Web Applications with Microsoft Visual Basic .NET and Microsoft Visual C#.NET, 2003
- [10] Stanley B. Lipman, Josee Lajoie, C++ Primer, Addison Wesley Longman, 2000

XI. AUTHORS

Yevgeniya S. Sulema is with National Technical University of Ukraine "Kiev Polytechnic Institute", Department of Special-purpose Computer Systems, P.O. box 36, Kiev-55, Ukraine, 03055, (email: sulema@scs.ntu-kpi.kiev.ua)

Vladimir M. Cvjetković is with Faculty of Science, Radoja Domanovica 12, 34000 Kragujevac, Serbia and Montenegro (email: vladimir@kg.ac.yu)