

A LabVIEW Based Remote DSP Laboratory

A. Kalantzopoulos, D. Karageorgopoulos and E. Zigouris
University of Patras, Greece

Abstract—In this paper the design and development of a Remote DSP Laboratory (R-DSP Lab) in the area of systems design with Digital Signal Processors is presented. The R-DSP Lab allows students to access and control real equipment in order to carry out laboratory exercises via Internet, through user-friendly Web Pages. For the implementation of this Remote Laboratory, the use of various technologies and tools such as Web Server Operations, Internet Communications, HTML, PHP programming, MySQL Database, Remote Measurements, and LabVIEW was necessary. The R-DSP Lab's structural elements are the Main Web Server and the Workstations. The number of Workstations can be dynamically increased or decreased depending on the needs. The architecture of the R-DSP Lab has a flexible and upgradable structure which does not depend on specific equipment. Due to this advantage, the proposed architecture can also be used to serve teaching needs of more cognitive fields.

Index Terms—DSP, LabVIEW, Remote Laboratories

I. INTRODUCTION

The tuition of many cognitive subjects includes both theoretical and practical parts. The practical part usually conducts in hands-on laboratories using the relevant equipment. These laboratories hold an interesting part in the educational procedure, encouraging students to deal with practical issues and observe various phenomena taking place in real-time conditions. However, in such hands-on laboratories, there are restrictions such as a limited number of workstations and specific working hours. In order to deal with the above issues, many universities develop remote laboratories [1-7].

Remote laboratories provide the students with the capability to perform laboratory exercises exploiting the relevant equipment any time of the day without their physical presence. Providing the ability for more than one student to use a single workstation, contributes to the reduction of the laboratory cost.

Turning to advantage the above and according to the needs of Master courses in the fields of DSPs (Digital Signal Processors) Systems Design and Signal Processing Systems with DSPs, we designed and developed a Remote DSP Laboratory. By using a Web Browser and by utilizing the R-DSP Lab, students have the ability to perform experiments on DSPs. For now, there is the capability to carry out laboratory exercises such as FIR, IIR digital filters and FFT as well as run any executable file developed by the student [8]. In any case the control of the real laboratory instruments (Oscilloscope and Function Generator) as well as the observation of the results is carried out through the use of specially designed Graphical User Interfaces (GUIs).

II. ARCHITECTURE

The structural elements of R-DSP Lab's architecture, Fig. 1, are the Main Web Server and the Workstations. The Main Web Server can be connected to one or more Workstations via Ethernet or Internet. This gives to R-DSP Lab the ability to use workstations that could be anywhere in the world, as long as there is an internet connection. The number of Workstations that can be used, depends on the occasional needs and may be dynamically increased or decreased without influencing the operation of R-DSP Lab. This is the reason that the R-DSP Lab's

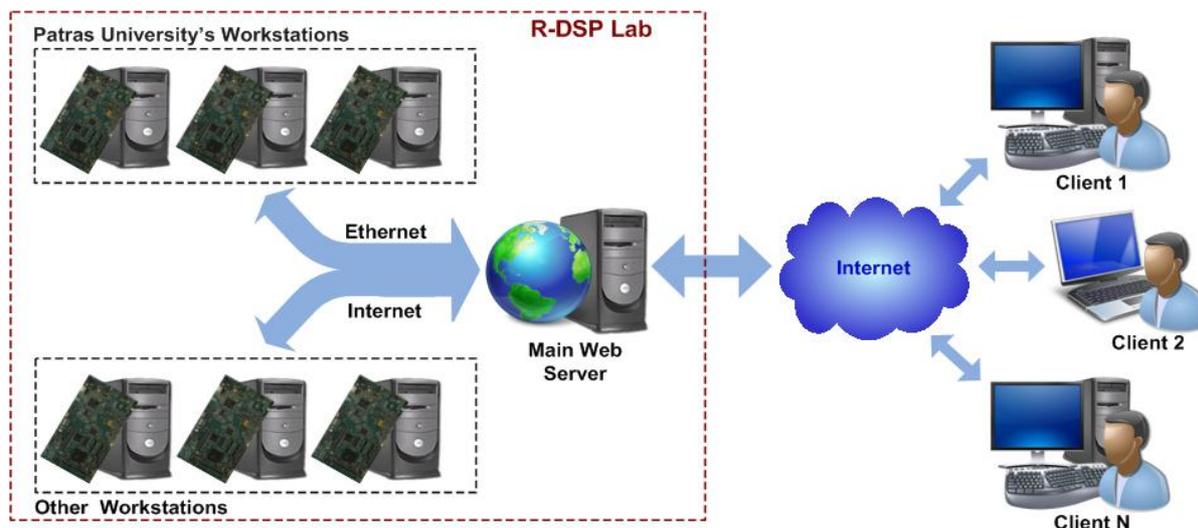


Figure 1. R-DSP Lab's Architecture

architecture is flexible and upgradable. Furthermore, the Workstations have been designed to serve more than one users at the same time. In this way, the number of users that can be served simultaneously by R-DSP Lab is increased without the need to purchase extra laboratory equipment. This cost-effective proposed architecture is ideal, especially for the implementation of Remote Laboratories that require high cost equipment.

A. Main Web Server

The Main Web Server undertakes the reception of visitors. In the home page of the R-DSP Lab, visitors are informed on the history of the laboratory and on the exercises that can be carried out with remote access. Access to informative material related to the object of laboratory, is also provided.

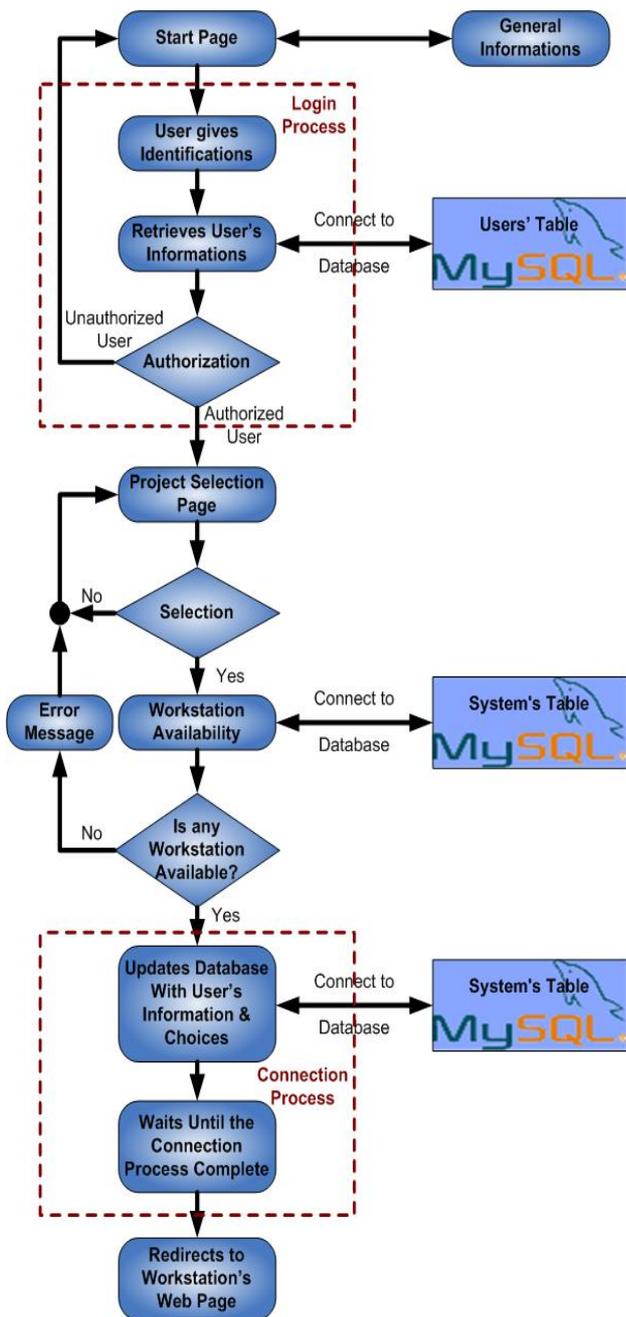


Figure 2. Block Diagram of the Main Web Server's Operation

The functions of the Main Web Server were developed using the features of PHP and HTML [9]. For the hosting of Web Pages the HTTP Server (Apache) was used. Furthermore, a database (MySQL) has been installed in the Main Web Server [9, 10]. Inside the database, the users' information, the executable files that have been uploaded and certain statistical information are stored.

Users that wish to implement an exercise or to verify the functionality of their own executable code, should follow the Login process. During this process, the users are asked to give their Username and their Password. Then the Main Web Server will try to identify the user, accessing information stored into MySQL database. The users that have been successfully identified have the ability, through the appropriate Web Page, to select any of the predefined laboratory exercises or to upload executable files that they have created.

In case the user wishes to upload his executable file he will be redirected to a new Web Page. From this Web Page he has the ability of selecting the executable file that has been previously created by the user. This file will be stored into the database after an available Workstation is found. If no Workstation is available, the Main Web Server informs the user and asks him to try again later.

Provided that a Workstation is available, the user's information (Username, IP and Connection Time), his choices and the executable file that he has previously uploaded are stored into database. Then the Connection process with the corresponding Workstation begins. After this process the user will be redirected to the corresponding Workstation's Web Page. Through this Web Page the remote user can exploit the laboratory equipment in order to verify the correct function of any of the predefined laboratory exercise or any executable file.

In Fig. 2 the block diagram of the Main Web Server's operation is presented.

B. Workstations

Workstations allow the communication between the authorized user and the laboratory equipment (instruments and development platforms). The design and the development of a Workstation are determined by the object of the laboratory exercises that are carried out and by the laboratory equipment provided. Workstations of R-DSP Lab, were developed and designed in order to serve the needs of Master courses in the fields of DSPs Systems Design and Signal Processing Systems with DSPs. Each Workstation, Fig. 3, is an entity which consists of the hardware and the software part.

Workstations of R-DSP Lab were designed and implemented using the existing hardware of the hands-on DSP Laboratory. The hardware used for each workstation is a Windows based PC, a DSK C6713 development platform of Spectrum Digital based on TMS320C6713 DSP of Texas Instruments (TI), a TDS1012 Oscilloscope of Tektronix and a TG 2000 DDS Function Generator of TTI. These instruments completely cover the needs of the development platform DSK C6713 which is mostly designed for audio applications. This development platform has an onboard codec with sampling frequency up to 96 kHz. Moreover, the low cost of the above instruments, makes them a perfect choice for a hands-on DSP Laboratory. What can be considered as a disadvantage, is the relatively low connection speed with

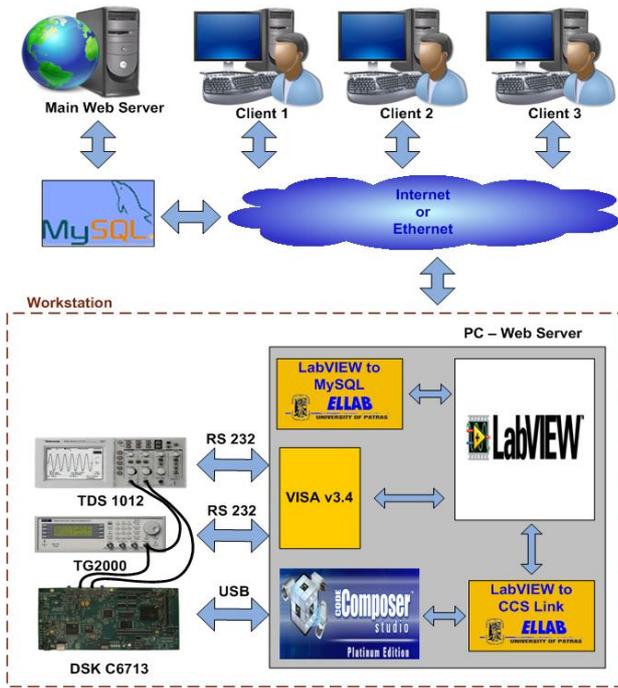


Figure 3. Block Diagram of the Workstation's Architecture

the PC, since this connection is established through serial protocols. As a result, these instruments are a good, low-cost choice for the implementation of Remote DSP Laboratory. The total cost for the purchase of the proposed hardware for the implementation of a Workstation of the R-DSP Lab is less than 3000 €.

For the control of each Workstation and the communication between the users and the available laboratory equipment, a series of applications based on LabVIEW v8.0 has been developed [13]. Each of these applications has a user-friendly GUI that represents the real function generator and oscilloscope and can serve only one user. These applications have been designed in such a way that more than one applications can be simultaneously run in each Workstation. In this way, more than one users can have access to the same laboratory equipment and carry out different laboratory exercises at the same time, according to a priority logic. In each R-DSP Lab Workstation PC, three such applications run. As a result, three different users can have access to the Workstation simultaneously. Consequently, the number of the available workstations is increased whereas the total cost per Workstation remains the same.

With Workstation's applications, the complete control of the Code Composer Studio (CCS) v3.1 and DSK C6713 is achieved, using the features of the toolkit LabVIEW to CCS Link that was created from scratch. These applications can connect to a MySQL database utilizing the potentials of the toolkit LabVIEW to MySQL. Furthermore, they have user-friendly GUIs through which the control of TG2000 Function Generator and TDS 1012 Oscilloscope using VISA v3.4 (Virtual Instrument Software Architecture) of NI is achieved. Moreover, these applications give the users the capability to control the Workstation through the environment of suitably designed Web Pages. These Web Pages are hosted by embedded LabVIEW Web Servers.

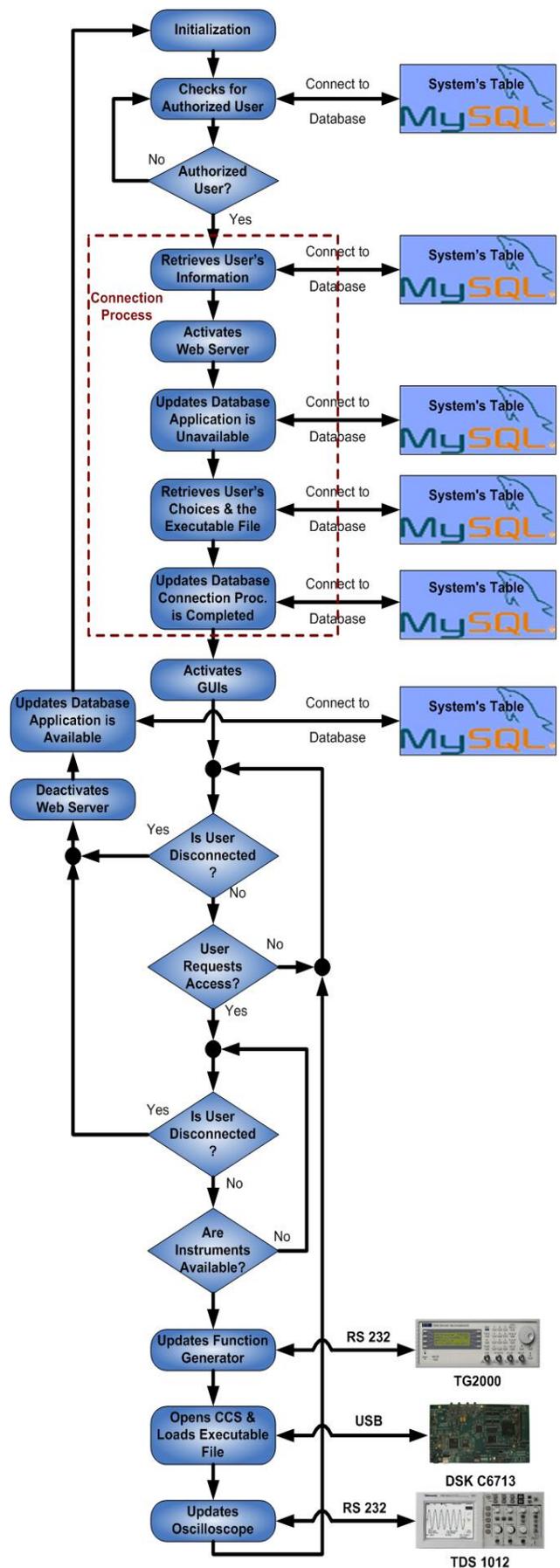


Figure 4. Block Diagram of the Workstation's Operation

Fig. 4 presents the block diagram of the operation of each Workstation's application. When one of these applications runs, it is initially connected to the MySQL Database and monitors the existence of an authorized user. In case an authorized user is found, the application retrieves the user's information and the connection process begins. During this process, the embedded LabVIEW Web Server is activated. The application updates the Database and the Main Web Server that it is occupied by a user. Then the user's information, his choices and probably the executable file that has previously uploaded are retrieved by the application. At this point, the Connection Process is completed and the user is redirected into a properly formatted Web Page where the GUIs of the laboratory instruments are activated.

From now on the remote user can exploit the laboratory equipment in order to verify the correct function of any of the predefined laboratory exercises or any executable file. When a user requests access to the equipment in order to refresh the GUIs, the application initially checks whether another user has access to the instruments. In case the instruments are available, it updates the function generator with the user's choices. Then the CCS is activated and the executable code of the predefined laboratory exercise or the one created by the user is loaded into DSK C6713. When the DSP begins the execution of the above code, the application updates the oscilloscope with the user's choices. Then the oscilloscope updates the application and the corresponding GUI with new data.

In case a user requests access to the equipment while another user is having the control, the Workstation's application automatically sets the user on hold until the current procedure ends. The control of the laboratory equipment is set to 20 sec at most, depending on the user's choices. This is the time needed for the Web Page's GUIs to update the data that comes from the laboratory equipment.

When the user completes the exercise that carries out and disconnects from the R-DSP Lab, the application deactivates the embedded LabVIEW Web Server and updates the database for the availability of the current Workstation's application.

C. Security

The access to the R-DSP Lab from remote users requires that the Workstation is heavily secured. At this time, this can be accomplished with certified identification procedures.

During the Login process the Main Web Server detects and stores the IP of any authorized user to the Database. At the Connection process each Workstation's application retrieves the IP from Database and activates the corresponding embedded LabVIEW Web Server only for this IP.

The Workstation's application periodically checks if the authorized user is still connected to the application. It also monitors if any non-authorized user attempts to connect to the application from the same IP (using the same router for example). In this case the application prohibits the access to the non-authorized user. When the authorized user is disconnected for any reason, the application updates the Database and deactivates the corresponding embedded LabVIEW Web Server.

Using firewall, the access to the Database is permitted only for IPs that belong to R-DSP Lab's Workstations. The connection to the Database also requires a Username and a Password as a second level security.

III. LABVIEW TO CCS LINK

Following the design philosophy of NI's Test Integration Toolkit for TI DSPs and in order to overcome its weak points, a new toolkit, LabVIEW to CCS Link, created from scratch [11-13]. This toolkit has the advantage of fully controlling the CCS v3.1 and communicating with TI's DSPs. Additionally it provides the opportunity to control CCStudio Setup v3.1. Therefore the hardware with which the CCS will communicate, is software defined through LabVIEW. This toolkit also supports reading and writing numbers and tables of all kind (floating-point single or double precision, signed and unsigned 1-, 2-, or 4-bytes integers) as well as strings by direct DSP memory access. Moreover it supports number and table (floating-point single or double precision, signed and unsigned 1-, 2-, or 4-bytes integers) transfers from/to DSP, using the RTDX (Real Time Data Exchange) technology.

The LabVIEW to CCS Link requires that LabVIEW v7.1 or later and CCS v3.1 or later are already installed. In Fig. 5 the connection between LabVIEW and CCS in block diagram is presented, using the LabVIEW to CCS Link.

Using LabVIEW to CCS Link one can create in a fast and easy way VIs (Virtual Instruments) that will actually work as GUIs for controlling and managing DSPs' applications. The LabVIEW to CCS Link is a very useful tool not only for educational purposes but for DSP system designers as well, due to the reduced amount of time needed, to develop GUIs.

The LabVIEW to CCS Link is a set of programs which are called sub Virtual Instruments (subVIs). These subVIs are divided into the three following categories, according to their usage:

- CCS Setup
- CCS Automation
- CCS Communication

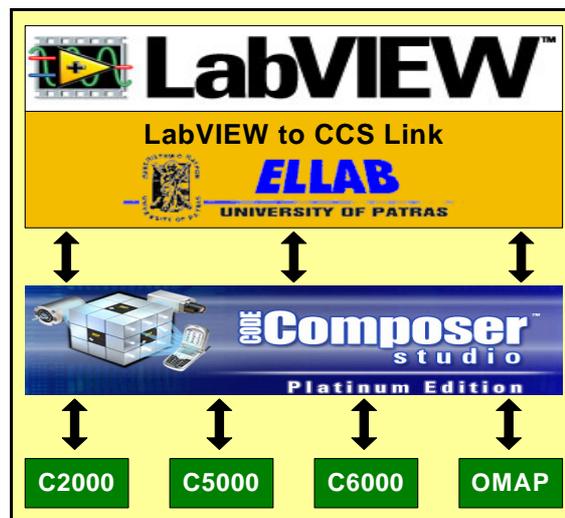


Figure 5. LabVIEW connection with CCS

TABLE I.
CCS SETUP CATEGORY

Icon	Name	Icon	Name
	CCS_Setup_Open.vi		CCS_Setup_Remove_Board.vi
	CCS_Setup_Close.vi		CCS_Setup_Rename_Processor.vi
	CCS_Setup_Clear.vi		CCS_Setup_Boards_Processors.vi
	CCS_Setup_Add_Board.vi		CCS_Setup_Save.vi
	CCS_Setup_Rename_Board.vi		

A. CCS Setup

The subVIs that belong to the CCS Setup category, provide the user with the capability to create new VIs with LabVIEW that will control the CCS v3.1 Setup and define automatically the board (-s) that will be used. The subVIs included to this specific category are presented in Table I.

TABLE II.
CCS AUTOMATION CATEGORY

Icon	Name	Icon	Name
	CCS_Open.vi		CCS_Reset.vi
	CCS_Close.vi		CCS_Run.vi
	CCS_Open_Project.vi		CCS_Restart.vi
	CCS_Close_Project.vi		CCS_Halt.vi
	CCS_Connect.vi		CCS_Is_DSP_Running.vi
	CCS_Disconnect.vi		CCS_RTDX_Enable.vi
	CCS_Build_All.vi		CCS_RTDX_Disable.vi
	CCS_Build_Result.vi		CCS_RTDX_Logfile_Configuration.vi
	CCS_Download.vi		

TABLE III.
CCS COMMUNICATION CATEGORY

Icon	Name	Icon	Name
	RTDX_Channel_Disable.vi		MEM_Read.vi
	RTDX_Channel_Enable.vi		MEM_Write.vi
	RTDX_Channel_Status.vi		Leds_Read_(DSK6713).vi
	RTDX_Read.vi		Leds_Write_(DSK6713).vi
	RTDX_Write.vi		Switches_Read_(DSK6713).vi
	MEM_Get_Address.vi		

B. CCS Automation

The subVIs that automatically control the CCS v3.1 belong to the CCS Automation category. These subVIs are shown in Table II.

C. CCS Communication

The subVIs that are used for data exchange between LabVIEW 7.1 and CCS v3.1, belong to the CCS Communication category. With these subVIs the user can create VIs so that data will be sent to and received from the DSP, either by direct DSP memory read/write operations, or by taking advantage of the RTDX technology. Table III presents the subVIs of CCS Communication category. The RTDX_Read.vi and RTDX_Write.vi are polymorphic VIs that can read and write floating point numbers and arrays of single and double precision, as well as signed and unsigned 8-, 16- and 32-bits integers to RTDX channels. The MEM_Read.vi and MEM_Write.vi subVIs are polymorphic too, which are capable of reading and writing floating point numbers and arrays of single and double precision, signed and unsigned 8-, 16- and 32-bits integers as well as strings, directly to the DSP memory.

IV. LABVIEW TO MYSQL

LabVIEW to MySQL is also another new toolkit, for accessing MySQL databases [10]. This toolkit was created in order to allow the connection between the R-DSP Lab and a MySQL database, utilizing the features of .NET framework [14]. LabVIEW to MySQL contains a set of high level subVIs for performing the most common database tasks and advanced functions for customized tasks. These subVIs are presented in Table IV.

LabVIEW to MySQL toolkit requires that MySQL v5.0, MySQL Connector/.NET v5.0, .NET Framework v2.0 and LabVIEW v8.0 or later versions are already installed.

In order to access data in a table or execute SQL statements, a connection to a MySQL database must be

TABLE IV.
SUBVIS OF LABVIEW TO MYSQL

Icon	Name	Icon	Name
	Open_Connection.vi		Update_Blob_Data.vi
	Close_Connection.vi		Read_Blob_Data_&_Save_In_File.vi
	Change_Database.vi		Update_Blob_Data_From_File.vi
	Read_Column_Field.vi		Execute_Query.vi
	Update_Column_Field.vi		Execute_Non_Query.vi
	Read_Blob_Data.vi		

established using the Open_Connection.vi. This toolkit supports multiple and simultaneous, local or Internet connections to a single database or multiple databases. After the connection to the database is successfully established, the following operations can be performed: change of the current database, retrieve or update data to a database table, read Blob (Binary Large Object) data and save in a file or write Blob data in Database from a file. More advanced operations can be performed by executing SQL statements, using Execute_Query.vi and Execute_Non_Query.vi. The disconnection to MySQL can be performed using the Close_Connection.vi.

V. R-DSP LAB'S GUIs

The GUIs of the R-DSP Lab were carefully designed in order to provide the most accurate representation of the features and the operation of the real laboratory instruments. Due to this, students have the opportunity both to familiarize with the real laboratory equipment and effectively carry out the laboratory exercises.

Through these GUIs, students can control the parameters both of the function generator and the

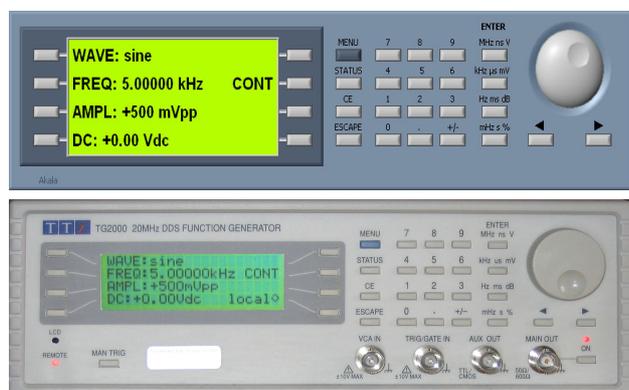


Figure 6. GUI of R-DSP Lab's function generator and TG2000 DDS function generator of TTI

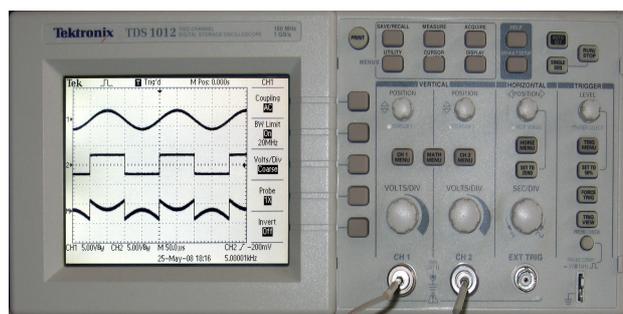
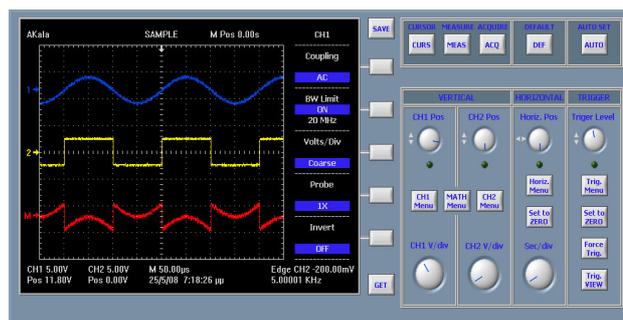


Figure 7. GUI of R-DSP Lab's oscilloscope and TDS 1012 oscilloscope of Tektronix

oscilloscope. From the GUI of the function generator students can choose the input waveform (sine, square and triangle), the frequency, the amplitude and the DC offset. Fig. 6 presents both the appearance of the GUI and the real function generator.

The GUI of the oscilloscope provides the students with the capability to control the real instrument. Through this GUI's controls, students can define the parameters for each channel as well as Volts/Div, Time/Div, waveform positions (vertical and horizontal), Trigger mode etc. They can also utilize the full functionality of real oscilloscope like autoset function, cursors readouts, automatic measurements, waveform averaging and peak detection and math functions. Fig. 7 presents both the appearance of the GUI and the real oscilloscope.

VI. LABORATORY EXERCISES

The students are asked to carry out a set of laboratory exercises which involves the design and implementation of both FIR, IIR digital filters (low pass, high pass, band pass and stop band) and a 512-point FFT, using the R-DSP Lab [8]. Utilizing the R-DSP Lab's predefined exercises, students can observe the correct operation of the algorithm that they are asked to design and implement, without the need of writing their own code.

At first, the student must design his solution using open source scientific software like Scilab or any other commercial software like Matlab by Mathworks. Utilizing one of the above packages, the student computes the coefficients of a digital filter. Moreover he can simulate the behavior of his proposed solution. After that, the student using the CCS writes in C or Assembly language the code that implements his solution. Then he creates the executable file and upload it to the R-DSP Lab. Provided that there is an available Workstation's application and the executable file is successfully uploaded, the student can verify the correct function of his solution.



Figure 8. Function generator's GUI through R-DSP Lab's Web Interface

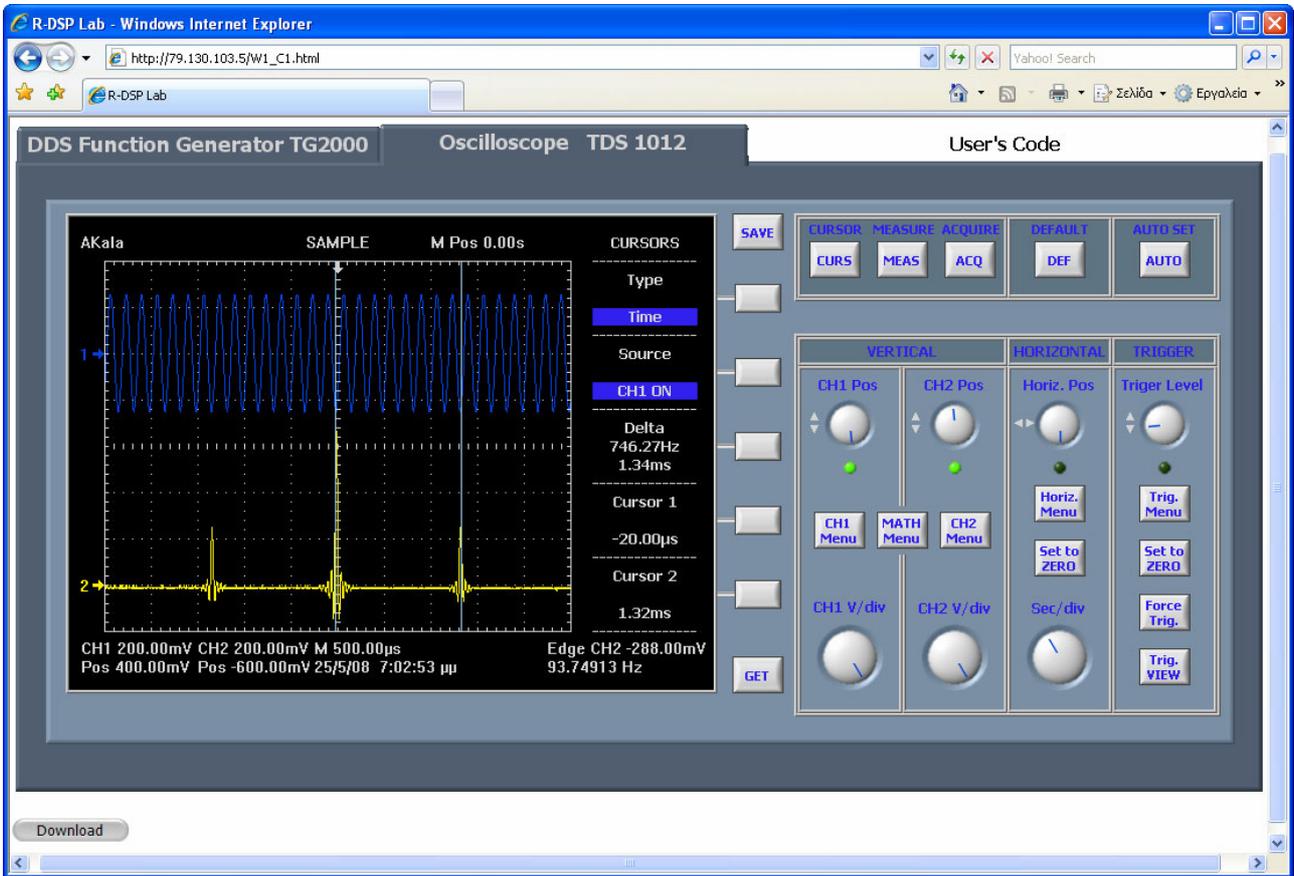


Figure 9. Oscilloscope's GUI through R-DSP Lab's Web Interface

As an example the student is asked to design, implement and observe the response of a 512-point FFT with 48 kHz sampling frequency (F_s), using the R-DSP Lab. Following the above procedure, implements and uploads the executable file into R-DSP Lab. In order to verify the FFT's functionality, at first, defines the parameters of the input signal. Then he observes the FFT's response and calculates the input signal's frequency from the above response.

Utilizing the function generator's GUI through R-DSP Lab's Web Interface, Fig. 8, in order to verify FFT's functionality, a sinusoidal input signal with 6 kHz frequency and 500 mVpp amplitude is used. Fig. 9 shows the oscilloscope's GUI through R-DSP Lab's Web Interface. Through this GUI the student can change the oscilloscope's parameters, request the update of Web Interface and save the captured screenshot in JPEG format into Workstation. The student can also download the above image, in order to use it in his report. Fig. 10 and Fig. 11 present the screenshots both of the R-DSP Lab oscilloscope's GUI and of the real instrument. The oscilloscope's GUI is an accurate representation of the real laboratory instrument.

The verification of the FFT's functionality can be easily tested by comparing the frequency of the input signal and the frequency which is calculated from the FFT's

response. In the FFT's response the central spike represents the DC component. The distance (Dt) between the central spike and one of the other two spikes corresponds to the input signal's frequency. Equation (1) calculates the frequency of the input signal utilizing the sampling frequency (F_s), the distance Dt and the number of FFT's points (N).

$$f = \frac{F_s^2 \cdot Dt}{N} \quad (1)$$

The distance Dt in Fig. 10 and Fig. 11 is 1.34 msec. The frequency which is calculated using (1) is 6.03 kHz. This value is very close to the input signal's frequency.

VII. CONCLUSIONS

In this paper the architecture, development and operation of the R-DSP Lab was presented. Following the above architecture, the conduct of laboratory exercises with TI's DSPs from distance, utilizing existing equipment is achieved. The implementation of the R-DSP Lab required the use of various technologies and tools such as Web Server Operations, Internet Communications, HTML, PHP programming, MySQL Database, Remote Measurements, and LabVIEW. In the next academic year the Master Degree Students of Electronics and Computers Division of the Physics Department of Patras University will use the R-DSP Lab in order to carry out their laboratory exercises in the fields of DSPs Systems Design and Signal Processing Systems with DSPs. The R-DSP Lab can be easily extended to fields such as digital and analogue electronics, microprocessors etc. At this moment, we are focusing on the implementation of a Web Based Editor that will allow users to develop their own DSP applications. Utilizing this Web Based Editor the users don't need to install the CCS. The use of instruments with faster communication interfaces will definitely result in the reduction of the delay time.

ACKNOWLEDGMENT

This work was financed by the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program "New graduate programs of University of Patras".

REFERENCES

- [1] C. Gravier, J. Fayolle, B. Bayard, M. Ates and J. Lardon, "State of the Art About Remote Laboratories Paradigms – Foundations of Ongoing Mutations " *International Journal of Online Engineering*, vol. 4, issue 1, pp. 19-25, February 2008.
- [2] Dr. Gokhan Gercek and Dr. Naveed Saleem, "Transforming Traditional Labs into Virtual Computing Labs for Distance Education", *International Journal of Online Engineering*, vol. 4, issue 1, pp. 46-51, February 2008.
- [3] M. Chaabene, K. Mkaouar and M. Ouali, "A Web-based Interactive Real Laboratory for Process Engineering Education", *Journal of Computer Science*, vol. 3, No. 7, pp. 540-545, July 2007.
- [4] Z. Nedic and J. Machotka, "Remote Laboratory NetLab for Effective Teaching of 1st Year Engineering Students", *International Journal on Online Engineering*, vol.3, No. 3, August 2007.
- [5] D. Hercog, B. Gergic, S. Uran and K. Jezernik, "A DSP-Based Remote Control Laboratory", *IEEE Transactions On Industrial Electronics*, vol. 54, No. 6, pp. 3057-3068, December 2007
- [6] D. Karadimas and K. Efstathiou, "An Integrated Educational Platform Implementing Real, Remote Lab-Experiments for

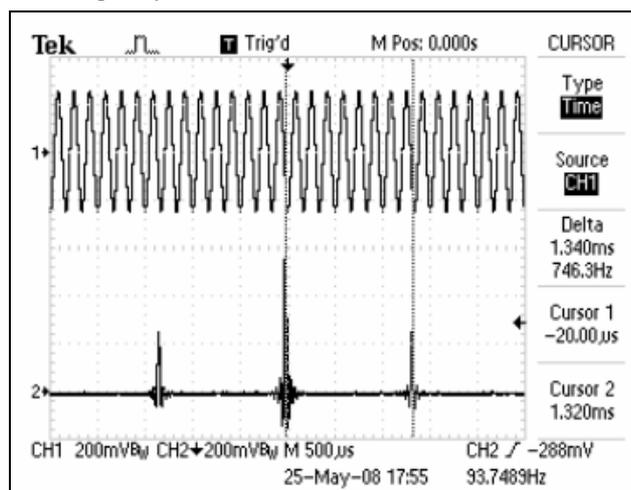


Figure 10. Screenshot of TDS 1012 oscilloscope

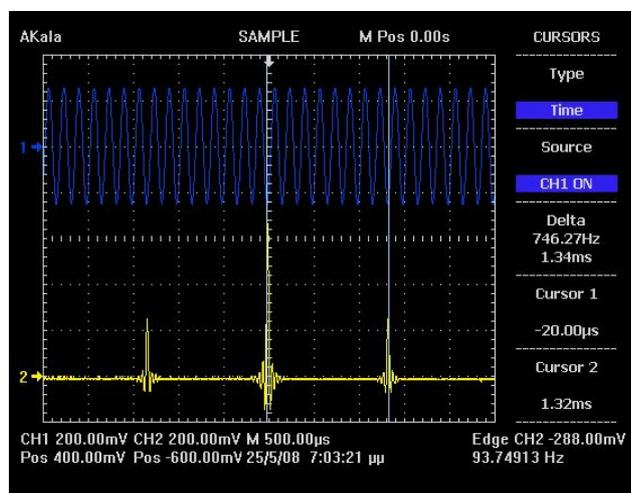


Figure 11. Screenshot of oscilloscope's GUI

Electrical Engineering Courses”, *Journal Of Computers*, vol. 2, No. 2, pp. 37-44, April 2007.

- [7] F. Schauer, M. Ozvoldova and F. Lustig, “Real Remote Physics Experiments Across Internet – Inherent Part of Integrated E-Learning”, *International Journal on Online Engineering*, vol.4, No. 1, February 2008.
- [8] R. Chassaing, “Digital Signal Processing and Applications with the C6713 and C6416 DSK”, John Wiley & Sons, 2005.
- [9] W. Jason Gilmore, “Beginning PHP 5 and MySQL 5: Novice to Professional”, 2nd Edition, Apress, Springer-Verlag, 2006.
- [10] C. A. Bell, “Expert MySQL”, Apress, Springer-Verlag, 2007.
- [11] E. Zigouris, A. Kalantzopoulos and E. Vassalos, “LabVIEW to CCS Link for Automating Digital Signal & Image Processing Applications”, *8 International Symposium on Signals, Circuits & Systems, ISSCS 2007*, pp 445-448, 12-13 July 2007, Iasi, Romania.
- [12] National Instruments, “LabVIEW Test Integration Toolkit for TI DSP”, September 2003.
- [13] R. Bitter, Taqi Mohiuddin & Matt Nawrocki, “LabVIEW Advanced Programming Techniques” 2nd Edition, CRC Press, Teylor & Francis Group, 2007.
- [14] A. Chakraborti, U. Kranti and R. J. Sandhu, “.NET Framework: Professional Projects”, Premier Press, 2002.

AUTHORS

A. Kalantzopoulos is with the Electronics Laboratory, Electronics and Computers Div., Department of Physics, University of Patras, Rio Patras, GR-26500 (e-mail: kalan@upatras.gr).

D. Karageorgopoulos is with the Electronics Laboratory, Electronics and Computers Div., Department of Physics, University of Patras, Rio Patras, GR-26500 (e-mail: dimitriskarageorgopoulos@gmail.com).

E. Zigouris is with the Electronics Laboratory, Electronics and Computers Div., Department of Physics, University of Patras, Rio Patras, GR-26500 (e-mail: ez@physics.upatras.gr).

Manuscript received 20 June 2008. Published as submitted by the authors.