Undergraduate/Postgraduate Student Project Work to Support the Teaching and Learning of Remote Laboratory Design

I.A. Grout
University of Limerick, Limerick, Ireland

***Abstract*—In this paper, consideration is given to the types of skills required by engineers who would be tasked with designing, developing, implementing and maintaining remote engineering laboratories. In this, the engineer would be required to have a broad range of skills including electronic hardware design, computer software and information technology (IT) amongst others that are brought together in order to develop and support remote laboratories. The paper will consider these skills, along with the acquisition of these skills at primarily undergraduate level to acquire the basic skills, but also postgraduate level to acquire the advanced skills.**

***Index Terms*—Educational technology, remote engineering, programmable logic.**

I. INTRODUCTION

As the fields of **remote engineering** and **virtual instrumentation** [1, 2] mature to a state that they are now being included into mainstream higher education, there is an increasing desire and need to include these engineering (and technology) disciplines (referred to from now onwards in this paper as simply as *engineering disciplines*) into both undergraduate and postgraduate programmes of study. Such a need is based on the increasing use of Internet technologies in everyday life, including educational aspects such as distance education and life long learning. Both **remote engineering** and **virtual instrumentation** are important tools in modern engineering practice, where they are defined here as:

- **Remote engineering** refers to the area of engineering concerned with the design, development and deployment of resources (equipment and software) that can be accessed by multiple users who can be physically located in separate locations globally. Access is usually via an Internet connection and is then often referred to as **online engineering**. An increasing awareness of this field of engineering in leading to new applications. Two current key activities in the field of remote engineering are the International Conference on Remote Engineering and Virtual Instrumentation (REV) [3] and the International Association of Online Engineering (IAOE) [4].
- **Virtual instrumentation** refers to customisable software and (modular) measurement hardware to create user-defined measurement systems. These are generally referred to as *virutal instruments*. These *virtual instruments* essentially provide the same functions as traditional test and instrumentation equipment such as oscilloscopes, digital multimeters and spectrum analysers, except now much of the hardware is replaced by software running on a personal computer (PC). One of the leading software systems in this field is LabVIEW (**Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) with its in-built software drivers for a range of available modular hardware units. LabVIEW is provided by National Instruments [5].

Remote engineering and virtual instrumentation are important tools in modern engineering practice for use in four key areas:

1. In **education**, where they can be used to support and enhance both the student learning experience and the teacher content delivery methods,
2. In **research and development (R&D)**, where access to complex and expensive equipment can be increased to support globally distributed research groups providing access to equipment and software that researchers would not otherwise have access to due to cost constraints,
3. In **commerce**, where they can be used to provide support for customers to access products using Internet browser based, low-cost (to the potential customer) product evaluation purposes,
4. In **industry**, where teams of engineers within an organisation who are globally distributed can act as a single group utilising the same equipment and software, so removing the barriers that would otherwise exist in distributed teams working together on the same project.

The inclusion of such support technologies within the **education** arena can be either to enhance the student learning experience through the use of appropriate (remote/virtual) technology, or to provide support for the learning of remote engineering and virtual instrumentation concepts. In this paper, the focus will be towards the **remote engineering** area. In particular, the design of **remote laboratories** will be the focus of the discussions. The education move (in utilising the right mix of support technologies to enhance the educational experience) has started in the right direction, both in an ad-hoc manner where remote engineering is included as part of an existing programme of study, and also by the inclusion of dedicated programmes of study. For example through the European Union (EU) supported MARE project [6, 7] and its' masters level programme championed by the Carinthia University of Applied Sciences [8].

In this paper, consideration is given to remote engineering education and the types of skills required by engineers who would be tasked with designing, developing, implementing and maintaining remote engineering laboratories. The paper will commence with a broad discussion into the basic skills acquisition with reference to undergraduate schemes of study in electronic engineering, computer engineering and information technology (IT). Limitations of traditional schemes of study will also be considered. Then secondly, consideration to the acquisition of advanced skills at the postgraduate level will be considered. The third part of the paper will look at a case study design for remote engineering in electronic circuit design and test will be considered. In this, a developed hardware/software solution that can be built within an undergraduate programme of study at various stages in the curriculum will be presented and discussed.

In general, the design of a remote laboratory, be it predominantly electronic hardware based, computer software or mechanical (e.g., telerobotics), requires a number of skills for the engineer (or team of engineers) to possess in order develop and implement a workable solution for the particular system requirements. A number of the key skills are shown in Fig. 1. Here, a number of skills are brought together to form the overall remote laboratory design. This requires the development of both theoretical and strong practical skills. It is the **practical skills** side of the educational requirements that are of primary interest here. This requires educational programmes that support a high level of laboratory and practical work without compromising the student acquisition of the necessary level of theory. Other key skills acquisition requirements might also be considered, depending on the focus of the particular programme of study. For example, it could be easily argues that suitable programmes of study should also consider skills such as:

- Design ergonomics,
- Project planning, costing and management,
- Experiment design,
- Documentation and presentation skills,
- System archiving and management,

In fact, the list could become long and detailed!

In this paper, the skills requirements are considered with reference to the design and development of suitable case study project work. Section II will discuss remote laboratory design and section III will consider the development of suitable programmes of study. Section IV will discuss a case study development system that would span across a number of different subject areas in a programme of study and which can be designed, developed and utilised by students as they progress through their programme of study. Section V will conclude the paper.

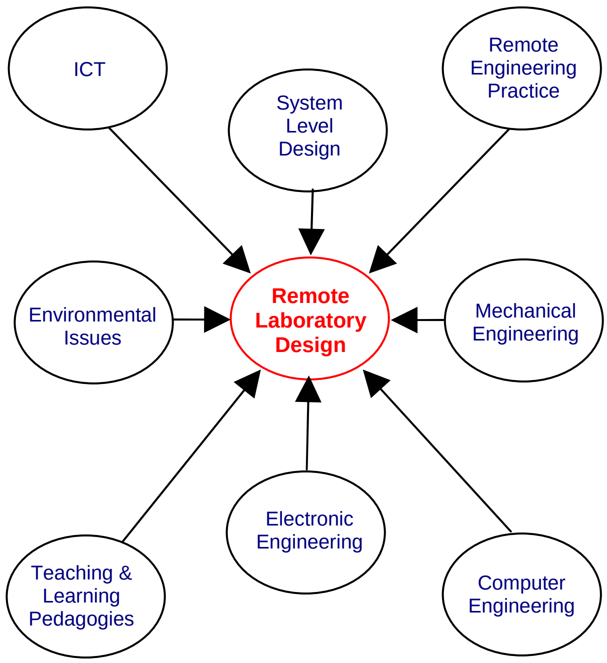

Figure 1. Remote laboratory design: skills requirements

II. REMOTE LABORATORY DESIGN

There would normally be a number of reasons for developing a remote laboratory facility within a higher education teaching and learning context:

- To provide an enhanced electronic hardware experiment capability,
- To provide an enhanced computer software simulation experiment capability,
- To provide an on-line course notes delivery, timetabling and student assessment system,
- To provide an additional mechanism for student study support,
- To provide students with the right set of enabling technologies to enhance their learning experience,
- To introduce students to experiments (academic) and industry processes that they would not otherwise have access to. This could be used to support the students in creating a context for their studies (helping to answer the question "*why am I studying this?*") and also to provide a means for the students to have access to "real world" data.

Whilst these are provided to a certain level within a normal programme of study in a "traditional sense" (i.e., local based student learning with a limited use of technology enhanced learning), the addition of the remote laboratory capability provides potential benefits in terms of:

- Increasing access to laboratory facilities where only limited access time is available, and often restricted, timetable slots. Students may benefit from the potential of 24-hour access to laboratory facilities, to times not normally available outside the timetable slots, whilst potentially alleviating some of the timetable scheduling difficulties,

- Part-time students are required to work on module coursework away from the university. This facilitates such work,
- There is a marked increase in the requirement to access laboratory facilities leading up to coursework deadlines leading to bottlenecks in the availability of the required hardware and software. This arrangement can allow for additional access to be provided and if suitably defined, students will be allocated equal access time,
- Many students have Internet access via desktop and laptop PCs at home and may wish to work on coursework in this context,
- Providing access to "real world" data that can be used by students in their own experiments (for example, in a control engineering context then real plant data can be obtained for modeling, simulation and control system design activities).

However, what form the laboratory should take and how the laboratory will be designed, developed, maintained and upgraded, would normally be specific to the particular institution, to the particular national (country) needs, and even the particular laboratory developer. There would not be a "one laboratory fits all scenario", rather there would be a common structure (underlying laboratory architecture supported by standards) that can be localised to the particular needs. In many cases, the underlying software [9] side of the laboratory access mechanism is based on WAMP (Windows, Apache, MySQL and PHP) or LAMP (Linux, Apache, MySQL and PHP). An additional concern would be as to what level within a programme of study should remote engineering be integrated (be it to teach the concepts of remote engineering, or to utilise the technology to enhance the teaching and learning activities). This would also need to be considered as either a single institution concern, or an inter-institution concern. An example set-up for an *imaginary* remote laboratory is shown in Fig. 2. Here, remote learners [10] would log into the web server via a standard Internet connection (via an Internet browser) and, depending on their access privileges, would have access to one or more experiments, course notes, timetabling and assessment system(s). Such a scheme could be custom built incorporating only those features considered of use (which are coded by the laboratory developer), or would be a combination of custom software scripts (experimentation access) and a suitable content management system (CMS) or learning management system (LMS).

The overall laboratory, or parts of the laboratory set-up, might also adhere to specific standards (e.g., SCORM [11]). Additionally, the experimentation might be custom built [12] or based on available systems such as National Instruments LabVIEW [5].

The actual hardware/software system architecture and the amount of design and development to be undertaken would be dependent on the focus of the work to be undertaken.

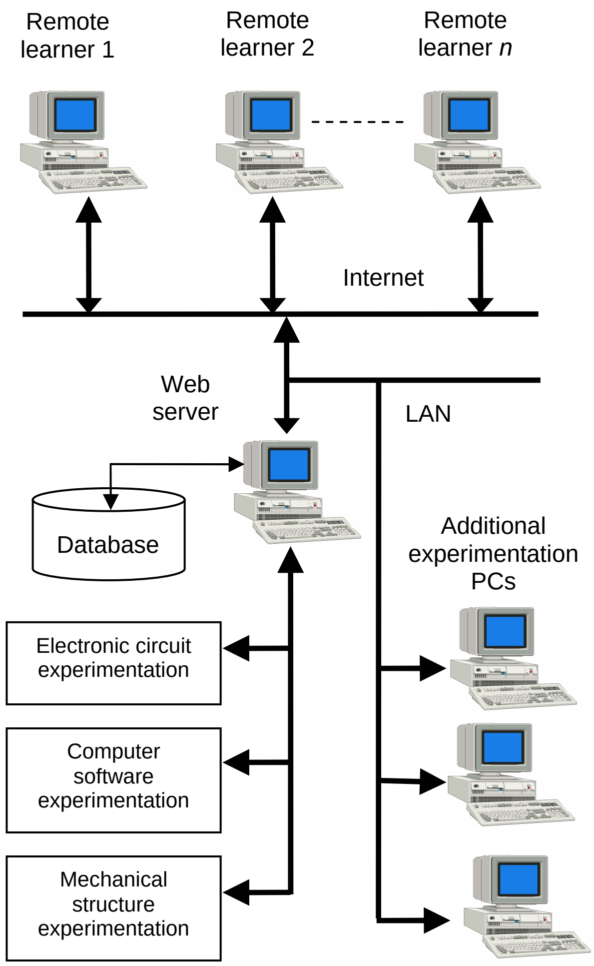

Figure 2. Remote laboratory structure (single laboratory)

III. DEVELOPING PROGRAMMES OF STUDY

All academic institutions are continually involved in improving/updating existing programmes of study, and developing new programmes of study that introduce modern engineering concepts. There are many different teaching and learning styles adopted in such programmes. As with any engineering discipline, the need to build on practical skills (design, build and test) to support the learning of the necessary theory, and to build the skills required by the practicing engineer is a major concern. When considering the development of a programme of study to teach the concepts of remote engineering, then this can be either at the undergraduate level, or the postgraduate level, depending on the particular set of required programme outcomes for the students. The undergraduate programme (either a 3-year or 4-year programme) would need to take students from the basics through to tackling more advanced topics, whereas the masters programme (either 1-year or 2-year) would aim to develop the students understanding of the more advanced topics. However, both levels of programme would need to have as a primary programme outcome, the students' ability to independently design, develop, maintain and upgrade a remote engineering laboratory.

The focus of the programme of study would need careful consideration. For example, if the purpose is to enable the students to develop their own remote laboratory, from design concept through to implementation, then they would need to be able to undertake all aspects of development themselves, however with the student concentrating on a particular area of focus (e.g., digital electronics). If however the purpose of the programme is to develop efficient experiments for a range of remote engineering scenarios, then the focus would be towards instrumentation and experiment design, which would then require the use of existing remote laboratory equipment, and would have a focus away from the electronics, mechanics, computing and IT system design. It could be envisaged that an undergraduate programme of study would aim to provide the basic concepts that would allow the graduate to utilise these concepts directly in industry, or would be equipped to continue onto further education at the advanced masters level.

For example within the University of Limerick, then undergraduate programmes are 4-years in duration and masters level programmes are 1-year duration. Hence, the following considerations would be based on this form of *4+1* study structure which fits into the 5-year masters study duration under the European ECTS (**E**uropean **C**redit **T**ransfer **S**ystem) structure. Student project work has always been and is integral to the student learning experience, be it individual project work or team based project work. One point to note is that these projects have always been associated with a particular subject (module) and have not normally linked across modules. However, given the nature of the remote engineering discipline, with its' multiple skills requirements (see Fig. 1), then consideration must be given to the development and deployment of project work that spans multiple modules, and which also spans across subject areas and years of study. This idea is shown in Fig. 3. Here, the idea is that there are two types of activity that span across the four years of study:

1. **Subject streams** – these are the theoretical parts of the programme and will be based on lectures, laboratories and tutorials, considering both local and remote based learning,
2. **Project streams –** these are linked into the subject streams and are projects that are undertaken by the students in each module within the subject stream. However, the projects are not necessarily only associated with a particular module. Rather, they can span across subject streams and years of study. In this approach, then the students can ultimately build a complete system where the individual building blocks of the system are created within each module.

Given this cross-module and cross-subject approach to project work, then careful consideration must be given to:

1. Setting-up the problem and hence the project specification such that the student can utilise a successful outcome of the project immediately,

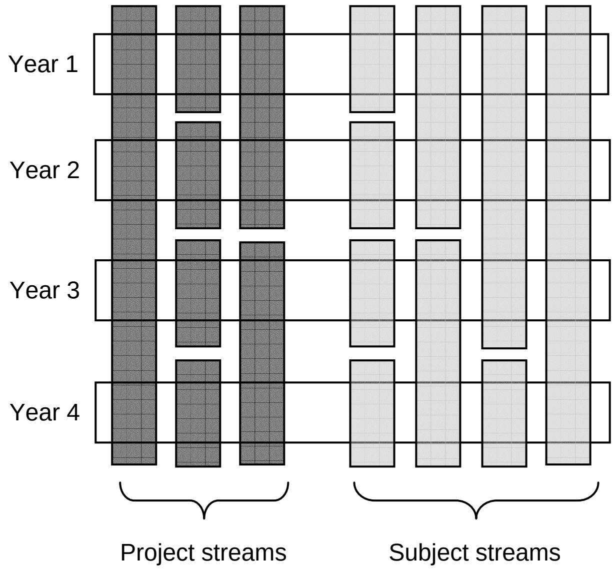

Figure 3. Project streams integrated into an engineering programme (considering a 4-year undergraduate programme)

2. The scenarios that if a student does not produce a successful outcome to the project, then there is a "fall-back" situation where the student can be provided with a basic working solution that allows him or her to continue, but would not be fully credited in the assessment as would someone who successfully completed the project in the particular module,
3. Allowing for the student to investigate the operation of the developed system by supporting independent learning to take the design further, but with a "fail-safe" position in that if problems occur, then the last known working system would be restored. This would require the development of a centrally administered student project storage capability where the student project work would be deposited,
4. Producing projects that are manageable, meaningful, challenging and the relevance readily understood by the students,
5. Building into the assessment mechanisms a way in which the student work can be verified as independently created, and not simply a copy of previously completed projects. This can be achieved, for example, by changing the design specification each year, to build into the assessment individual student project work reporting and presentations, and to build into the programme of study an ethical dimension where the students learn about ethical behaviour.

In addition, then the projects undertaken would also require specific learning outcomes. Therefore the learning outcomes could be considered at the following levels:

1. Subject module,
2. Subject stream,
3. Individual project,
4. Project stream,
5. Study year,
6. Programme.

Examples of project stream outcomes include:

1. A promotion of the application of scientific method in the analysis and development of solutions for engineering problems,
2. A promotion of the ability of individual problem solving,
3. A promotion of the ability of team based problems solving
4. A promotion of effective communication with the engineering community and wider community.

The multiple years (3- or 4-year) of an undergraduate programme of study provide a natural mechanism whereby projects that cross subject boundaries and years can be readily developed. This would be an ideal situation that would allow for the different subjects (electronic engineering, computer engineering, IT, teaching and learning pedagogies, etc.) can be introduced to the students and then built into the project.

> *Over time, the students then would naturally end-up building a complete system where each part of the system has been developed and implemented for a particular purpose.*

In one (or two) year masters degree programmes, then the multiple year project capabilities might not exist and so any project work would need to be based on the advanced subject nature of the programme and have a different focus. Additionally, the advanced programme would need to consider whether it would be aimed at holders of primary degrees in remote engineering, or would be aimed at graduates with a relevant (engineering/technology) primary degree who would want to work in the field of remote engineering. The types of subjects and the level would need to be set for the particular target audience. However, individual and team based projects that are interesting, academically robust and challenging can be brought into such masters level programmes.

IV. CASE STUDY DEVELOPMENT SYSTEM

In this section, a case study hardware-software development system is described. The hardware development board, as shown in Fig. 4, is based on CPLD (complex programmable logic device) technology and implements a hardware interface between a remote laboratory server (PC) and the electronic *circuit under test* (CUT).

The CPLD used was the Lattice Semiconductor MACH4A5-64/32 CPLD. The choice here was based on the availability of the devices and EDA (Electronic Design Automation) tool (i.e., ispLEVER) and prior knowledge of the device. However, given that the design was created in such a way that it could be readily transferred to target other programmable logic devices (PLDs) such as the FPGA (Field Programmable Gate Array), then alternate devices and device vendors could be used. This is important as different academic institutions utilise different PLDs, EDA tools and PLD vendors in the teaching and learning activities. This is an indication of the relevance and usefulness of PLD technology to the teaching and learning experience.

The required *circuit under test* would be either configured within the CPLD itself, or would be attached to the digital I/O ports via jumper leads. With this arrangement, the student would have full access and control of each part of the design. The design, development and use of this development board will require a number of skills that would need to be acquired to implement a successful solution. These skills would be acquired over time in a suitable programme of study at the undergraduate level which then can lead to more advanced acquisition of skills at the postgraduate level. Such skills include:

- Electronic hardware design (digital, analogue and mixed-signal),
- Printed circuit board (PCB) design,
- Design of digital circuits and systems using programmable logic,
- Design of digital circuits and systems using hardware description languages (HDLs) – primarily VHDL and Verilog®-HDL,
- PC interfacing (and communications standards),
- Design ergonomics,
- Power supply design,
- Electronic component selection and reading/interpreting datasheets,
- Computer programming languages,
- Web server installation and development,
- Web services design and development,
- Database design and development,
- Project management,
- Design archiving and maintenance,
- Environmental issues,
- Ethical issues,
- Presentation and documentation skills.

Providing the students at the undergraduate level with exposure to remote laboratory design, development, implementation and maintenance requiring the above skills would allow for the increased and more effective use of remote engineering in everyday life.

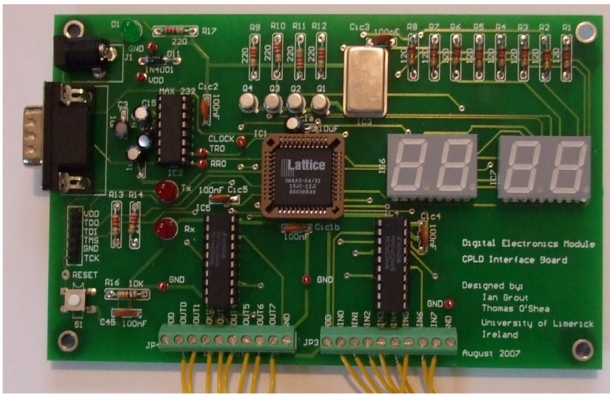

Figure 4. CPLD development board for learning of remote engineering concepts

In fact, it would be possible for each student to implement their own complete remote laboratory on PCs within their department, or even on their own PC (desktop or laptop) at home. The more motivated students could also independently set-up and develop their own set of distributed on-line laboratories that combine their own laboratories into a laboratory grid. If carefully considered, then such potential for independent student learning would naturally enhance the student learning experience and would have long term benefits for the students on completion of the programme of study (and then entering industry).

The example of a hardware/software design that could be used as a prototype for a remote laboratory shown in Fig. 4, could provide a basis for a range of project. This board can be designed, manufactured and tested by the students at part of their programme of study, and would have the capability of being used once successfully tested in an overall remote engineering laboratory set-up which would also be designed, manufactured and tested by the students. The basic electronic system architecture for the CPLD development board is shown in Fig. 5.

The ability also exists to allow for the students to add on circuits via the digital I/O port and to develop interesting experiments. For the size of CPLD chosen here, it also provides the students with exposure to PC interfacing of electronics via the RS-232C serial communications protocol and also to:

1. The need to develop efficient HDL code in order to fit the required design into the device,
2. To show limitations of CPLD size (hardware),
3. The issues around HDL code simulation and synthesis,
4. The fitting of designs (or place & route in field programmable gate arrays (FPGAs)) to particular devices,
5. Logic interfacing of programmable logic to external digital logic circuits.

Within the CPLD, then the configured digital system was developed by describing the different system functional blocks using VHDL as the hardware description language (HDL) of choice, although it would be possible to re-code the design using Verilog®-HDL if required.

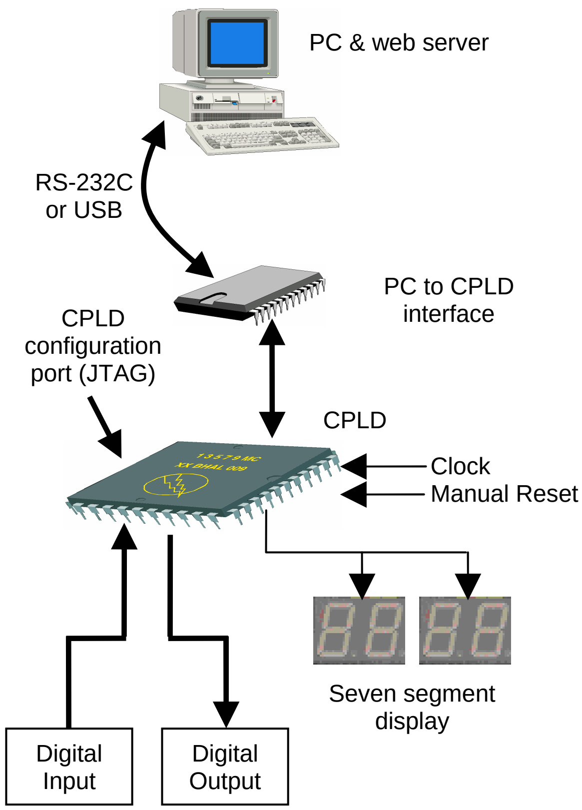

Figure 5. CPLD development board architecture

Each of the functional blocks was then integrated into a top level schematic. This arrangement is shown in Fig 6.

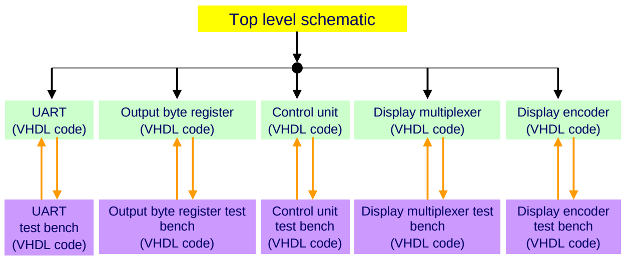

Figure 6. CPLD configuration

This forms the overall design architecture as shown in Fig 7. This view shows the digital system implemented within the CPLD and shows:

1. Serial communications via the RS-232C communications protocol with the PC (UART Receiver and UART Transmitter),
2. Internal signal generation (Control Unit) created from the master clock (1.8432 MHz) and master reset (asynchronous, active low),
3. Data capture register (Register),
4. User display (Multiplexer and Seven Segment Display Encoder) that be viewed locally by the user, or remotely via a web cam.

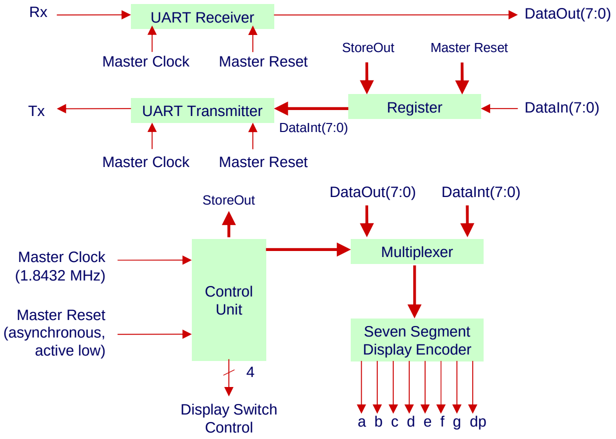

Figure 7. System block diagram

When the CPLD development board has been built and tested (verifying that it works), then it can be connected to a web server (such as the Apache web server) by connecting the board to the web server (PC) serial port. When the web server is run, a basic set-up would be to create and run a PHP script that allows the user to submit data to the board. A simple web page that is generated by an example PHP script is shown in Fig. 8. Here, the user can select a value to send to the board (byte of data) and then this is displayed on the seven segment display on the board itself. In this case, the local address (127.0.0.1) is used and the web page is accessed with:

`http://127.0.0.1/rev_2008.php`

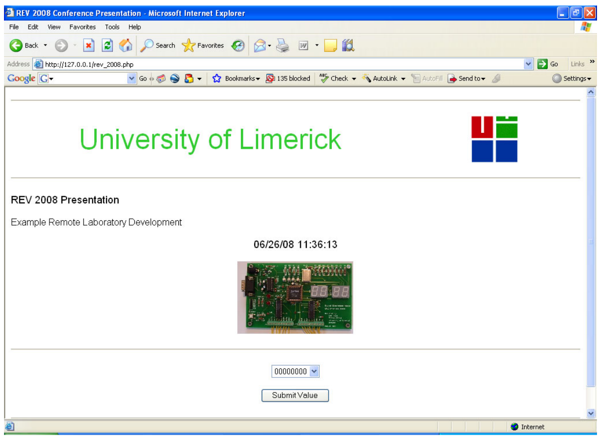

Figure 8. Example hardware access web page

Once the `Submit Value` button has been pressed, then the value set in the drop down menu is sent to the web server, and a second PHP script saves the value to temporary file on the web server. The PHP script then calls a local executable file which reads the temporary file and sends the value to the CPLD development board. The response of the board are then transmitted back to the web server and saved to a second temporary file. This results file can then be accessed.

With this arrangement, then the student would be able to see each of the main basic functional blocks that would be required to build a basic remote laboratory. They could then improve on this basic set-up and enhance the functionality using their imagination and personal skills.

It should be finally noted that the above considerations are aimed to be just at starting point, and that once a basic set-up has been developed and is seen to be working, then the opportunities for developing projects that could excite the students would naturally occur. Really, there should be no limitations to the potential for developing interesting and challenging projects for inclusion within both undergraduate and postgraduate programmes of study. Imagination should have no boundaries.

V. CONCLUSIONS

This paper has discussed a number of issues related to the teaching and learning of remote engineering concepts, and the need to incorporate effective student project work into programmes of study. Such project work can be used to enhance the student learning experience by incorporating projects that the students participate in that spans multiple modules, and which also spans across subject areas and years of study. An example CPLD based system for interfacing to a PC based web server that could be completely developed by the students was considered. The resulting electronic hardware and software that would be developed at each stage within a programme of study would be used in the next stage of the programme. This provides the ability for the students to learn and practice the required concepts and see the relevance and use of the theory learned. It also provides a natural mechanism whereby the results (outcomes) from each module (subject) undertaken by the student has a direct relevance and impact on the student, and which are utilised in the follow-on modules that the student is to study.

VI. FURTHER INFORMATION

Further information and design details for the CPLD development board, and an example set-up arrangement can be found at the following URL:

http://www.ral.ul.ie/CPLD_Development_Board/index.htm

ACKNOWLEDGEMENTS

The author would like to acknowledge the work of Dr Thomas O'Shea in the development of the CPLD development system when working in the Department of Electronic and Computer Engineering at the University of Limerick.

REFERENCES

[1] International Journal of Online Engineering, http://www.online-journals.org/i-joe/

[2] J. Murphy, I. Grout, J. Walsh, and T. O'Shea, "Local and Remote Laboratory User Experimentation Access using Digital Programmable Logic", *International Journal of Online Engineering*, vol. 1, October 2005

[3] International Conference on Remote Engineering and Virtual Instrumentation (REV), http://www.rev-conference.org/

[4] International Association of Online Engineering (IAOE), http://www.online-engineering.org/

[5] National Instruments, LabVIEW, http://www.ni.com

[6] ISOC Socrates Project Database, MARE - Joint European Master Degree Program "Remote Engineering", http://www.isoc.siu.no/isocii.nsf/projectlist/29298

[7] R. Šafaric, A. Rojko and D. Hercog, "European Joint Master Degree Remote Engineering", *Proceedings of the REV 2007 conference*, Porto, Portugal, June 2007

[8] Carinthia University of Applied Sciences, http://www.cti.ac.at

[9] Meloni J., *SAMS Teach Yourself PHP, MySQL™ and Apache in 24 Hours*, SAMS, 2003, ISBN 0-672-32489-X

[10] Palloff R. and Pratt K., *The Virtual Student - A Profile and Guide to Working with Online Learners*, Jossey-Bass, 2003, ISBN 0-7879-6474-3

[11] SCORM® 2004, Advanced Distributed Learning, http://www.adlnet.gov/scorm/index.aspx

[12] I. Grout and M.J. Burbidge, "Evolution of a Remote Access Facility for a PLL Measurement Course", *2nd IEEE International Conference on e-Science and Grid Computing*, Amsterdam, 4th–6th December 2006, The Netherlands, http://www.escience-meeting.org/eScience2006/index.html

AUTHOR

I. A. Grout is with the University of Limerick, Department of Electronic and Computer Engineering, Limerick, Ireland (e-mail: Ian.Grout@ul.ie).

This article was modified from a presentation at the REV2008 conference in Düsseldorf, Germany, June 2008. Manuscript received 30 June 2008. Published as submitted by the author.