

## Implementing Remote Laboratories for Digital Electronics: Foundations for Distance Learning and Resource Sharing

*Mais Nijim* (Corresponding author)

Department of Electrical Engineering and Computer Science, Texas A&M University Kingsville

700 University Blvd, Kingsville, TX, 78363, U.S.A.

E-mail: mais.nijim@tamuk.edu

*Hisham Albataineh*

Department of Physics, Texas A&M University Kingsville

700 University Blvd, Kingsville, TX78363, U.S.A.

E-mail: hisham.albataineh@tamuk.edu

*Abayomi Obawomiye*

Department of Electrical Engineering and Computer Science, Texas A&M University Kingsville

700 University Blvd, Kingsville, TX78363, U.S.A.

E-mail: abayomi.obawomiye@tamuk.edu

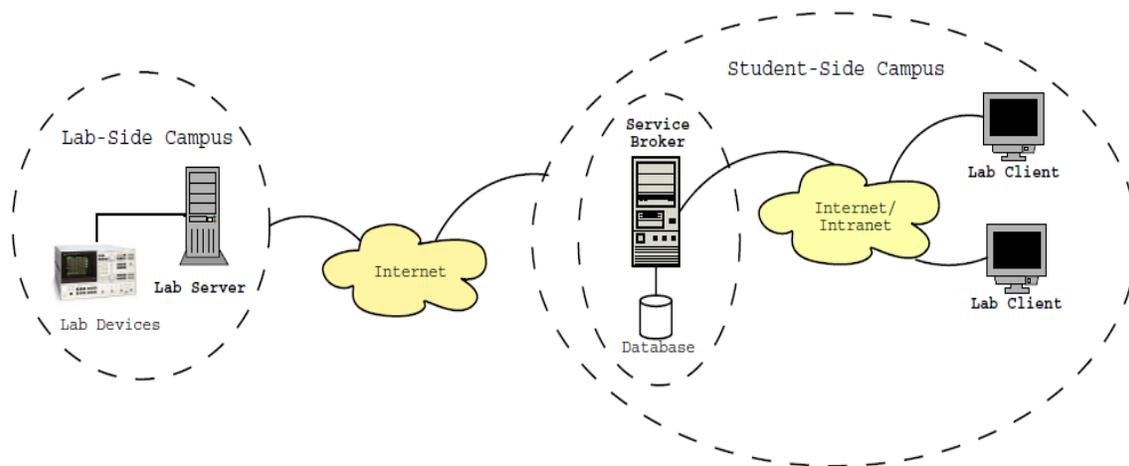
**Abstract:** This paper describes the design, implementation and development of a remote laboratory for digital electronics. The laboratory uses the MIT three-tiered iLab architecture, a framework for remote lab development and deployment. This specific experiment involves using a simple encoder-multiplexer circuit connection where users will be required to determine an unknown input to the multiplexer. Not much emphasis is placed on the details of digital electronics, but rather it serves as a proof of concept for how remote/online laboratories can be implemented and adopted in any engineering/science discipline. This paper also introduced a new approach to presenting remote experiments to distance learning students in order to give them an experience similar to what students who have access to the physical laboratories have.

**Keywords:** iLab Architecture, Remote Lab

## 1 Introduction

Remote laboratories are not new, there have been many papers which have discussed and implemented online experiments for over 16 years and in fact a number of universities have developed their own laboratories but many of them do not have a technique for in place for ensuring that students or users adhere to the same trainings as they would get if they were to perform such experiments in a real academic laboratory. In this paper, an approach has been introduced to ensure that the students go through a short mandatory introduction to the lab experiment before they perform their work. There is an authentication phase where the student has the sufficient information in accessing the lab by observing a video. Moreover, an advantage regarding this is the student cannot skip or ignore this video until they watch it at least once. Why this is important is because this remote laboratory is supposed to be a replica of a real laboratory, therefore students should be able to partake in what it feels like to be in a typical laboratory session. A simple experiment was implemented in this thesis so that students with some science/engineering background can partake in it without any form of supervision. With the remote laboratory technology, students can have access to the laboratory not only in the convenience of their dormitory or apartment but also at any time comfortable for them. Apart from the benefits the students derive from using the remote laboratory, universities can also leverage on the remote laboratories in order to share laboratory equipment and resources. Another benefit of using the remote lab is the effect on faculties and laboratory assistants, with the remote laboratory; students can start their experiment and send results over the Internet without necessarily requiring the presence the physical presence of an instructor.

The MIT iLab architecture was designed as a distributed software toolkit and middleware service infrastructure for lab development and deployment, this architecture has made network communication easy for the developer. With the web-service this architecture employs, there is a client-server communication, which is based on traditional method calls over HTTP [12]. The iLab architecture separates the more generic administrator tasks such as user authentication and authorization from the lab-specific operations such as running the experiment; this has proven to be helpful in efficient manner. Added to the client-server interface is the service broker. This handles all administrative tasks, this way remote laboratories can be developed without worrying about implementing custom made administrative solutions for each of the new labs [12].



**Figure 1** An overview of a three-tiered iLab architecture.[12]

## 2 Related Work

Since the development of the iLabs Shared Architecture, many universities around the world have adopted it. Some few universities in Africa have found use for the architecture and have developed different labs with it. The goal most of the remote laboratory projects seek to achieve is to make real laboratory instruments available for students and educators regardless of the user's geographical location and time-zone. Unlike the previous approaches which utilize simulations to introduce practical concepts to students, the remote laboratory avail users access to real laboratory instruments by leveraging the power of the internet.

There has been numerous works on the topic of remote laboratories. Most of them have focused been focused on the application of remote laboratories in the various engineering and science fields. Aboluwarin et al. carried out a research on how to improve on an already existing laboratory experiment. In their work titled Remote Realistic Interface Experimentation Using the Emona DATEx Board [12], they were able to perform a remote reconfiguration of the circuit on an actual Emona DATEx board. The Emona board is used for implementing some basic telecommunications experiment. They were able to make the same board was made available for multiple experiments and multiple users. In the past, only one experiment can be performed at any point in time because the hardware could not be dynamically reconfigured. They were also able to implement an interactive and realistic user interface which gives the user the feel of interacting with the real equipment. The aim of their research was to show the benefits of remote laboratories for universities in terms of resource sharing and cost. By using the iLab shared architecture, they were also able to prove that remote laboratories are not just useful for Electrical Engineering alone but can also be implemented in Communications Engineering.

In the publication titled An Improved Operational Amplifier iLab with a More Realistic Looking Interface [11], the authors also focused on the development of a realistic looking interface

for an operational amplifier remote laboratory. This experiment was used to introduce the concept of operational amplifiers to electronics and electrical engineering university students. They reported that with their improved interface, students were able to familiarize better with the real laboratory instruments and there was also an improvement on their understanding of the topic after performing the experiment.

Osakue, Chen, Ahmed, Shahryar, Olowokere (2012), in a case study focusing on the use of virtual and remote laboratory framework development for engineering technology education [15], aimed at developing a software and hardware framework that will allow users to control physical instruments via an internet web browser. They came up with a remote Smart Vibration Platform experiment for teaching active vibration control. In their work, they gave detailed explanation on web based experiment development with the consideration of firewall and security issues.

There are quite a number of work both unpublished and published that discussed different technologies for implementing remote laboratories and how they can be integrated into different domains in Science, Technology, Engineering and Mathematics (STEM). Just a few of these works have been examined in this paper, there are still a lot more work that have been done and a lot more that will still be done in the future.

This work has focused on implementing a remote laboratory for digital systems using a multiplexer and encoder to demonstrate a simple experiment. The aim is to look into ways of implementing remote laboratories, with an emphasis on the user experience. A mandatory introductory video was included which users must watch before they can gain access into the laboratory. This is to replicate the experience of the interaction laboratory instructors have with the students before the beginning of laboratory experiments. By using the information available in this paper, current and aspiring students interested in the design of remote laboratories can look into many areas of remote laboratories they can improve on. Many remote laboratories have focused on computer web clients, not a lot has been done on implementing remote laboratories for access on smart phones and devices. Future research can emphasize on how to get remote laboratories on these devices that students use the most. Another area is to include a platform where users can interact with the laboratory administrators. This way, they can ask questions and get answers as they would in a traditional laboratory. More attention should also be given to making the experiments attractive enough for students to have fun performing them. In the future, there should be dedicated mobile apps which give students access to different remote laboratories that are available. This way, access to the remote laboratories can be further increased and expanded.

### **3 The iLab Architecture**

The iLab framework is a three-tiered topology useful for batched experiments. It utilizes web services, which are the Lab Client, Service Broker and Lab Server as show in Figure 1. The Service broker was developed in such a way that different groups can be created specific to a particular experiment and different access levels such as membership and management can be granted by the remote lab administrator. Students do not need to have direct contact with the laboratory setup, this is the task the Service Broker performs.

### **3.1 Web Services**

Since the iLab technology is intended to be flexible and available for reuse, the need of implementing the framework using web services became important [12]. Since there is a high possibility that the client-side services and the lab-side services will run on different hardware and software platform, it is important that the architecture should be able to support them.

### **3.2 The Lab Client**

The user interface (lab client) of this experiment is made up of three main sections. The left hand side of the screen is an image representing the schematic block diagram of an encoder connected to a multiplexer. Four inputs lead into the encoder and the user can select any one of the four inputs per experiment session. There are two outputs from the encoder, which leads into the multiplexer. There are also four inputs that lead into the multiplexer. For this experiment, these four inputs are already predetermined for the user. There is an output that leads out from the multiplexer. On the right hand side is a screen where the user can see a live feed of the experiment of the experimental setup.

On the lower part of the user interface, there is a “RUN” button which the user needs to click to send the experiment specification to the experiment engine for execution. There is also a screen display to the right of the run button which indicates when the experiment has been sent for execution. Since Adobe currently maintains a hold over the market with Flash – the Flash Player is on 99% of hosts with web browsers installed, therefore using an Adobe Flex user interface seems to be a good idea.

The Lab Client was developed with Adobe Flex. This is because the Adobe framework deploys consistently on all major browsers which our lab users will be using since we are targeting desktop computer users. Previous Lab Clients developed, require the user to have the Java Plugin and the Java Runtime Engine before they can be launched. This is not so with Adobe Flex because the interface will be rendered as an Adobe Flash movie and most computers already support it.

As earlier mentioned, for the sake of simplicity since this work is focused mainly on a proof of concept to show the possibility of remote digital laboratory experiment which can be exploited for distance learning, much emphasis won't be placed on the realistic outlook of the user interface but rather we would use something which many students with some understanding of electronics are familiar with. The reason for this is because there is a section for live video feed on the user interface. The user can enjoy viewing a live feed of the experimental set up and also see a simplified circuit diagram of their connection as they would deal with it in theory.

The first step for any user would be to gain access to the Service Broker by entering the web address into a browser. They will be required to provide their username and password. The Service Broker will authenticate them and direct them to a dynamically generated webpage where the lab client can be launched. See Figure 2.1 and Figure 2.2 for the image of the service broker login page and the launch page.

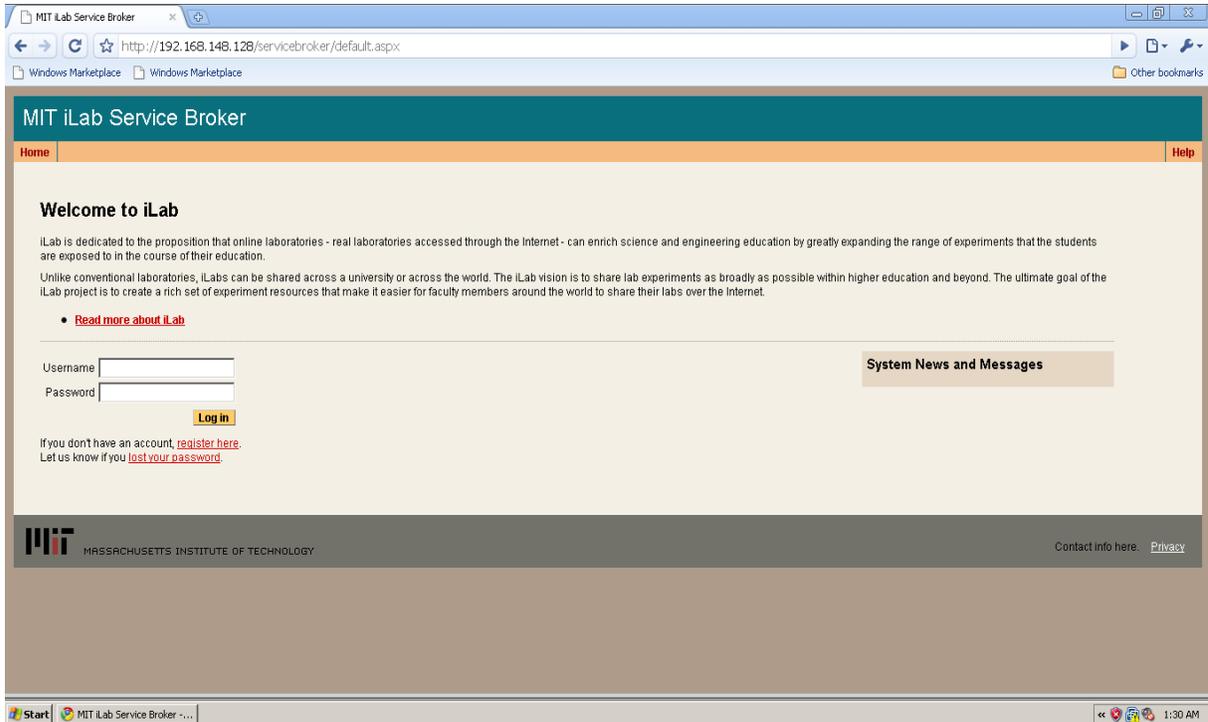


Figure 2.1 Service Broker login page where user needs to enter a username and password

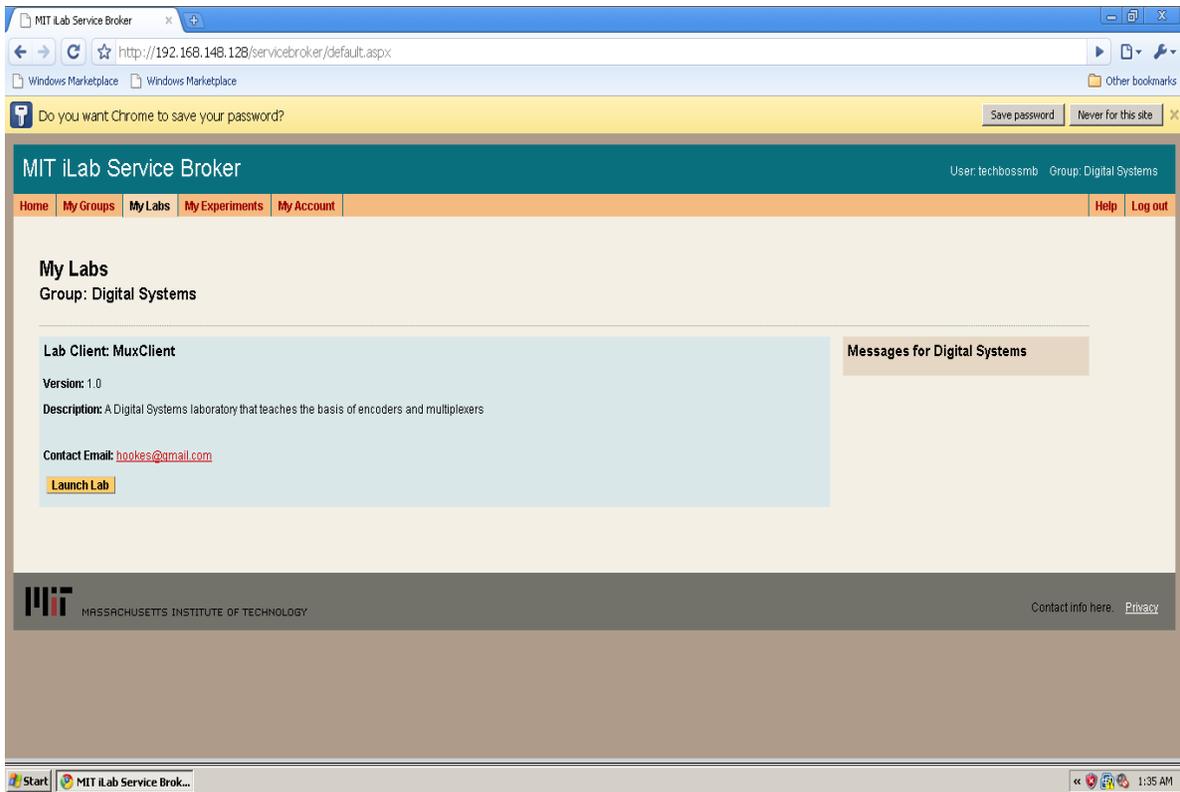


Figure 2.2 Remote laboratory service broker launch page

### **3.3 The Lab Server**

The Lab Server is operated by the lab's owner and deals with the actual operation of the experimental set up. It communicates with the Service Broker through an ASP.NET web service running over Internet Information Services (IIS). The Lab Server runs an experiment engine separate from the web services module which periodically checks whether a new experiment is available in the lab server database. The experiment engine prioritizes the loading of the experiment on the MS SQL database based on the order in which they were submitted. Then it processes the experiment specification sent from the user through the Lab Client when the hardware connection requirements are met. Once the required hardware connection has been completed by the user and the user starts interacting the setup through the Lab Client, the data sent from the user is sent to the data acquisition device through the LabVIEW dynamic-linked library (dll) by the experiment execution engine. The result is sent back to the experiment engine, which then forwards it to the MS SQL database.

### **3.4 Lab view dll**

The experimental setup was mounted on a breadboard and a data acquisition device (NI myDAQ) was used for selecting the inputs to the encoder. LabVIEW is used to communicate with the DAQ. LabVIEW is a graphical programming language developed by National Instruments. As discussed earlier, the lab client was developed using C#, therefore we need to figure out a way to enable communicate with LabVIEW VI through the experiment engine. This way we can have access to the DAQ through the experiment engine. After the LabVIEW VI has been converted into a dll function, a C# method call was used to control the VI from the visual studio environment. This was helpful when the experiment engine needed to interact with the VI.

### **3.5 System under Test**

In order to imitate events that happen in a real laboratory, a short video was introduced to summarize the laboratory experiment to the user. The essence of this was to represent the lab instructor who gives a short "lecture" to the students before the commencement of the laboratory exercise. Of course, because we know some students might not really have interest in viewing the video and might want to skip the video, the skip mechanism in the video was deactivated. This way, access to the lab will only be possible upon the completion of the video.

To achieve the development of this remote laboratory, a Lab Client was developed alongside the design of a Lab Server after which the experiment was deployed on the Service Broker. The Lab Client as discussed in the previous chapter is the user interface and it is situated at a remote location away from the real laboratory. The development of the Lab Client was done using the Adobe Flex Framework. The Lab Server consists of a C# program that coordinates the experiment execution and it is the first point of interaction with the hardware. There is also a MS SQL Database which will handle storage of data. The actual lab setup which is a connection of two integrated circuit chips (an encoder and a multiplexer) connected to a data acquisition device. The Service Broker is the MIT iLab batched experiment Service Broker developed by the MIT iLab team. There is not

much to be done at the Service Broker except configuring it to suit our application.

To make an existing traditional laboratory experiment an iLab, a series of questions needed to be answered. The first most important question we need to answer is can we control the lab equipment from a personal computer? To what extent do we have control over the parameters and quantities involved in the experiment? If the answer to the first question is no, then it's basically impossible to make such an experiment an iLab. If the answer to the first question is yes then we start thinking of how to make that computer a server so that a client computer can have access to such system. So it's like a (user) computer controlling another (server) computer that controls the equipment.

Making an experiment an iLab involves sending and receiving of electrical signal for configuring and measuring of quantities and parameters associated with the experiment. Hence most iLabs are in the domain of electronic/electrical and computer engineering, this has been proven not to be a major setback as demonstrated in the paper development of strength of materials remote laboratory.

In the design of the strength of materials laboratory, the lab developer was able to show that with the right combination of sensors and actuators it is possible to convert physical variables/quantities into electrical signals. With all the necessary hardware in place, a functional fully automated Lab with absolutely no human intervention was designed

## **4 The Experimental Results**

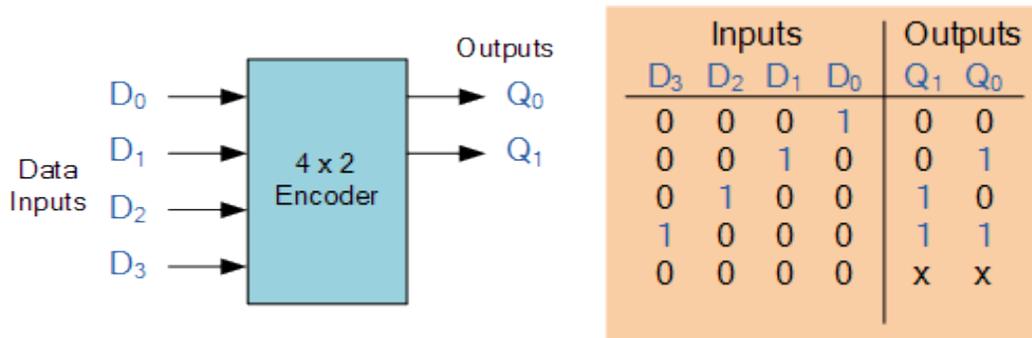
### **4.1 The Data Acquisition**

In order to be able to send data/signals from the circuit setup from and into the computer, a data acquisition device is needed. A data acquisition device is a hardware setup that is used for measuring real world physical conditions, sample such physical conditions, and then convert them into a format that can be processed by a software which runs on a computer. The data acquisition device is a DAQ that was used for this experiment. It is a low-cost portable data acquisition (DAQ) device designed by National Instruments that uses LabVIEW-based software instruments. The choice of the hardware was based on the need for an inexpensive data acquisition hardware that can provide us with at least 8 digital input/output line, a 5 volts power supply and digital ground. Although there are many more additional features on the NI myDAQ, much emphasis is not placed on them. Any DAQ that provides enough digital outputs and also a power supply would be sufficient but one of the main reasons for selecting the DAQ from National Instruments is due to the ease at which LabVIEW communicates and integrates with the hardware. This way, there won't be much time wasted on how to communicate with the hardware. It is important to bear in mind that different experimental setup will require different data acquisition need and different sensors or actuators might require more sophisticated/complex data acquisition system. The NI myDAQ has 8 digital input/output lines, the digital I/O lines are 3.3 V LVTTTL and are tolerant to 5 V inputs which makes it suitable for our application. To be able to communicate with the DAQ from the LabVIEW

software, the NI ELVISmx driver needs to be installed. NI ELVISmx supports LabVIEW (32 bit). To use NI ELVISmx with LabVIEW on a 64-bit operating system, you must have LabVIEW (32 bit) installed.

### 4.2 The Encoder

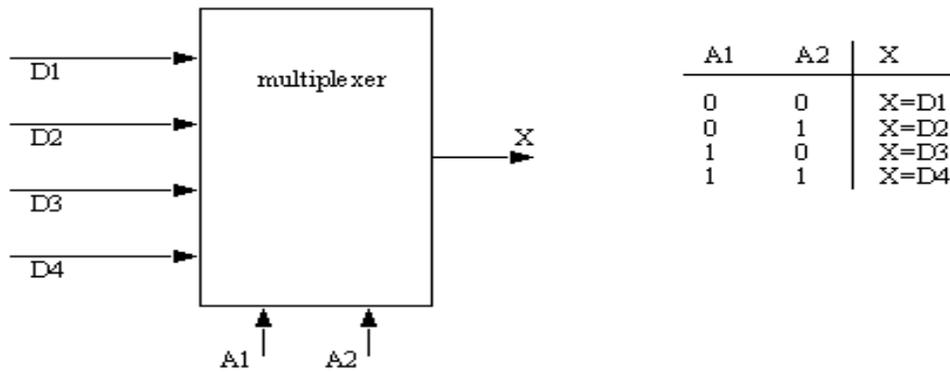
An encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller number of outputs. To achieve the aim of this experiment, an encoder known as the priority encoder was used. Priority encoders are often used to control interrupt requests by acting on the highest priority request [1]. They produce an output that is the binary representation of the original number starting from zero of the most significant input bit. If two or more inputs are given at the same time, the input having the highest priority will take precedence [1]. Figure 3.1 shows the image of a 4 to 2 encoder, this means that there are four inputs into this encoder and depending on the configurations of the four inputs, two possible outputs can be derived. So for example when a HIGH is sent into the first line  $D_0$ , and a LOW sent into  $D_1$ - $D_3$ , the two outputs  $Q_1$  and  $Q_0$  will both yield a LOW. Another configuration would be to send in a HIGH into  $D_2$ , and a LOW into the rest, the result of this is a HIGH on  $Q_1$  and a LOW on  $Q_0$ . This has an advantage as this means that the encoder can be used as a switching device or selector. The encoder used for this experiment is the 74F148 which is an 8-Line to 3-Line Priority Encoder. It allows eight data lines to be encoded into it in priority and provides 3-bit binary priority code. It also has input enable capability. It produces signals when data is present on any input. What we are trying to achieve is that the output from the encoder will be used to determine the output derived from the multiplexer. These two encoder outputs will serve as the inputs to the “SELECT” of the multiplexer as would be shown shortly.



**Figure 4.1** A 4-2 priority encoder with the truth

### 4.3 Multiplexer

Connected to the outputs of the encoder is the multiplexer. The multiplexer used for this experiment is the SN74151AN. It has eight digital inputs, one digital output and three select inputs. A multiplexer is a device that selects one of several analog or digital input signals and forwards the selected input into a single line [2]. For this experiment, the multiplexer would be functioning as an input selector.



**Figure 4.2** A 4-1 multiplexer with the truth table

Digital outputs DO<sub>4</sub>-DO<sub>7</sub> (line 4-7) of the DAQ were connected to the inputs D<sub>0</sub>-D<sub>3</sub> (pin 1-4) of the encoder. The rest of the encoder inputs I<sub>3</sub>-I<sub>0</sub> are left unwired since we only require four of the inputs and from the truth table we have found that their values are irrelevant therefore they can be left unwired. Pin 8 of the encoder which is the ground (GND) is also connected to the digital ground (DGND) of the DAQ. Pin 16, the input voltage (V<sub>cc</sub>) is connected to a 5V power supply from the DAQ. This is used to power on the encoder. Pin 5 which is the enable input (EI) was connected to the line 4 (DO4) of the DAQ which according to the experimental setup will always be LOW. The outputs of the encoder Q<sub>0</sub>-Q<sub>2</sub> (pin 9, 7 and 6) are then connected to A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub> (pin 11, 10, 9) of the multiplexer. These pins are the selectors of the multiplexer and they determine what input is passed to the output. D<sub>3</sub>-D<sub>0</sub> (pin 1-4) of the multiplexer are connected to the lines 3-0 (DO3-DO0) of the DAQ. Pin 8 which is the ground (GND) of the multiplexer is also connected to the DGND line of the DAQ. While being guided by the datasheet of the multiplexer, the strobe, G (pin 7) is connected to line 4 (DO4) of the DAQ. Pin 16, V<sub>cc</sub> of the multiplexer is connected to the 5V power supply of the DAQ while pin 15-12 (D<sub>4</sub>-D<sub>7</sub>) are all connected to line 4 (DO4) of the DAQ which according to the experimental setup will always be a LOW. The output of the multiplexer, Y sends out one of the inputs from D<sub>3</sub>-D<sub>0</sub> depending on the selector configuration.

The DAQ is connected to the computer via a USB cable and it is interfaced with via the LabVIEW software. A simple program is written to send data into the DAQ digital pins which in turn are used as inputs and outputs for the encoders and the multiplexers.

#### 4.4 Lab View Program

LabVIEW is short for Laboratory Virtual Instruments Engineering Workbench. It is a graphical programming language developed by National Instruments. LabVIEW uses icons that resemble flowcharts unlike lines of codes in conventional programming languages to implement its programs. LabVIEW was chosen as the software for the interfacing with the hardware because of the ease of communicating with the DAQ which is also a National Instruments manufactured hardware.

The Lab Client was developed using Adobe Flex. The interface consists of three main

components as shown in Figure 3.5. To the left of the interface is an image representing the schematic block diagram of an encoder connected to a multiplexer. Four inputs lead into the encoder and the user can select any one of the four inputs per experiment session. There are two outputs from the encoder which leads into the multiplexer. There are also four inputs that lead into the multiplexer. For this experiment, these four inputs are already predetermined for the user. There is an output that leads out from the multiplexer. On the right hand side is a screen where the user can see a live feed of the experiment of the experimental setup.

On the lower part of the user interface, there is a “RUN” button which the user needs to click to send the experiment specification to the experiment engine for execution. There is also a screen display to the right of the run button which indicates when the experiment has been sent for execution. It also outputs a message if the user has not selected any input but tries to run the experiment. The reason for the video feed is to give the student a visual representation of the events taking place at the remote lab location, they also get to see the experiments they are interacting with. This way the student is convinced they are not working on simulations but rather the real lab components. The client makes a connection to the server via a service broker web service and subsequently submits the specification to the lab server. The lab server executes the experiment and returns the output value to the lab client. The live video stream which is visible to the right of the lab client displays the activities of the lab server streamed live from the server. The Adobe Flash Media encoder was used for capturing and streaming the video to a server which is the Adobe Media Server Development edition. The Lab client streams directly from this server.

**Encoder input and output**

I = Input

X = Output

“0” represents 0-volt input and “1” represents a 5-volt input

I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	X <sub>1</sub>	X <sub>0</sub>
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

**Multiplexer output**

X = Select inputs

Y = Output

I = Data input

The following input combination was connected to the multiplexer  $\{I_0 = 0, I_1 = 0, I_2 = 1, I_3 = 0\}$

Where “0” represents 0-volt input and “1” represents a 5-volt input.

The following output was obtained when the select input was as shown below:

$X_1$	$X_0$	Y
0	0	0
0	1	0
1	0	1
1	1	0

The following data inputs were also tried:  $\{I_0 = 0, I_1 = 1, I_2 = 0, I_3 = 0\}$

The result also came out as expected as shown below:

$X_1$	$X_0$	Y
0	0	0
0	1	1
1	0	0
1	1	0

This result supports the theory behind the functioning of a multiplexer. The select inputs functioned as a “selector” and selected the data input based on the “selector” combination.

## 5 Conclusion

This thesis focused on the development of a remote laboratory which demonstrates simple concepts in digital electronics using an encoder and a multiplexer. A typical experimentation process to go through an authentication process after which the student must watch an introductory video at least one time before access to the experiment is finally granted. By following the instructions provided in the introductory video, the student should be able to gain sufficient knowledge in order to perform the lab experiment without any form of assistance. In this way, some part of the scenario which occurs during a real laboratory experiment session was successfully imitated. It was observed that despite having an instructional video, the presence of a physical instructor was not completely imitated. Future labs should be able to achieve the possibility of students asking questions in real-time. This present remote lab has also been limited to clients running the application on a personal computer. In the future, more remote labs should be developed for an

extended platform such as smart phones, this way students will be unrestricted by when they can access the labs. Presently, the introductory video in this current design provides an option for students to enter a generated token number in order to skip watching the video. An improvement to this would be a video designed in such a way that once a registered user watches the video, they can chose to watch or ignore the video at their next lab access without entering a token.

This digital electronics lab is intended to serve a proof of concept which explains the simple setup of an encoder and multiplexer. In the future, this can be improved on and more complex/advance labs can be developed using the iLab shared architecture and therefore incorporated into the university's engineering lab curriculum in teaching hands-on experience to students. The role of remote laboratories really is not to replace the traditional labs but to complement them. With the remote labs, the time students spend on routine hardware setup and configuration is also reduced, this way they can focus on the real goal of the experiment.

### **References**

- [1]. M. Morris Mano, Michael D. Ciletti, Digital Design 4th ed., Prentice Hall, 2006
- [2]. Debashis, De (2010). [Basic Electronics](#). Dorling Kindersley, 2010, pp. 557.
- [3]. R. Callahan. Purdue study: Hands-on learning better. Internet:  
<https://www.purdue.edu/discoverypark/learningcenter/assets/pdfs/HandsOn.pdf>, Feb. 2, 2009 [Nov. 10, 2014]
- [4]. Embedded Systems Lab "Reaches" Out to Distance Students. Internet:  
<http://www.colorado.edu/engineering/academic-programs/embedded-systems-lab-reaches-out-distance-students>, [Nov. 10, 2014]
- [5]. X. Chen, G. Song, Y. Zhang (2010). "Virtual and Remote Laboratory Development: A Review", Presented at the 12th Biennial International Conference on Engineering, Construction, and Operations in Challenging Environments and the 4th NASA/ARO/ASCE Workshop on Granular Materials in Lunar and Martian Exploration, Hawaii, United States, March 2010.
- [6]. V.J. Harward et al., "The iLab Shared Architecture: A Web Services Infrastructure to  
a) Build Communities of Internet Accessible Laboratories," Proc. IEEE vol. 96, pp. 931- 950, June 2008
- [7]. V. J Harward et al., iLab: A Scalable Architecture for Sharing Online Experiments, Presented at the International Conference on Engineering Education, 2004, Gainesville, Florida, October 2004
- [8]. University of Michigan, Center for Research on Learning and Teaching. Internet:  
[http://www.crlt.umich.edu/gsis/gsi\\_guide](http://www.crlt.umich.edu/gsis/gsi_guide) [Nov. 10, 2014]

- [9]. J. Pratt et al., Collaboration Between Universities: An effective way of sustaining community-university partnerships?" in Gateways: International Journal of Community Research and Engagement. doi: <http://dx.doi.org/10.5130/ijcre.v4i0.1773>
- [10]. ViedmaNúñez, Gerardo (2005). "Design and Implementation of the Feedback Systems Web Laboratory", Massachusetts Institute of Technology. Internet: <http://dspace.mit.edu/handle/1721.1/33369>
- [11]. B.I. Ishola et al., "An Improved Operational Amplifier iLab with a more Realistic Interface," American Society for Engineering Education, 2012
- [12]. O.O. Aboluwarin et al., "Remote Realistic Interface Experimentation Using the Emona DATEx Board," American Society for Engineering Education, 2012
- [13]. MIT (2011, Sep. 06). The iLab Project. [Online]. Available: <http://icampus.mit.edu/projects/ilabs/#architecture>.
- [14]. National Instruments (2009, Jan. 14). Integrating LabVIEW Code into Other Programming Languages. [Online]. Available: <http://www.ni.com/white-paper/3517/en/>