

Integrating the Internet, LabVIEW™, and Lego Bricks into Modular Data Acquisition and Analysis Software for K-College

**Merredith Portsmore, Martha Cyr, Chris Rogers
Tufts University**

Abstract

Both data acquisition and analysis are used in different forms from kindergarten through college for teaching math, science, and engineering. The RCX™, the LEGO programmable brick, has a 10 bit A/D, can acquire points at up to 200 Hz and store up to 2000 points, making it well suited for simple to advanced data acquisition. ROBOLAB™, produced in partnership by Tufts University, LEGO DACTA, and National Instruments, allows users to program the RCX in a graphical environment powered by LabVIEW™. Initially designed to handle only robotics creations, the cross platform software has been extended to allow users to use a variety of sensors to take data and analyze it. The new component of the software, titled Investigator, accommodates the wide range of ages with 5 sections, each with a number of different templates, where users can write their hypothesis and conclusions, create their data acquisition programs, upload and view their data, and perform calculations and comparisons. The templates in each section allow for different levels to be available to the users -- ranging from templates with large icons and few options for elementary school students to completely open ended levels with all options available for college students. To encourage the exchange of data and results, the ability to share programs, data, and results over the web has been built into software. The web also serves to allow the software to expand and change as new templates can be downloaded and added to the software as needed. Finally, LEGO creations can be controlled remotely via the Internet. This paper will present some of the issues associated with developing software for such a diverse audience as well as assessment of how it has worked, both at the elementary school and college level.

I. Introduction

Data acquisition and analysis are introduced to students at an early age. In kindergarten, for example, students often record whether it is a sunny or cloudy day by crossing off the appropriate box on a piece of large graph paper. They use the information they gather to answer questions -- "Did we have more cloudy or sunny days this week?" or "Which month had the most sunny days?" Collecting data and using it to answer questions helps 5 and 6 year olds to learn about the weather as well as mathematical and comparison skills. At the college level, data acquisition and analysis are an essential part of the sciences as well as all engineering disciplines. The two topics play an important role in courses ranging from instrumentation to advanced fluid mechanics.

Our goal was to create a piece software that would encompass the needs of students from ages 5-25 to perform data acquisition and analysis quickly and easily without placing a ceiling on the capability of the software. Creating a piece of software that spanned this large age group could help reduce the amount of time spend learning different software packages thus, allowing more time to be spent on investigation and learning. The software needed to support the classroom environment and the methods put forth by teachers for scientific and mathematical investigations. Moreover, it was important to incorporate ways for students to give presentation, share data, and publishing web pages with their data. The final result, ROBOLAB 2.0, expands upon the earlier version of ROBOLAB to meet these goals. This paper will present some of the issues associated with developing the software, a detailed description of the final solution, as well as assessment of how it has initially worked, both at the elementary school and college level.

II. Background

In 1995 and 1996 grants from NASA and NSF funded the development of a K-12 aeronautics program and the development of an innovative undergraduate engineering curriculum at Tufts University.¹⁻⁵ Each program needed tools that were flexible, powerful, and affordable to implement their respective goals. Moreover, developers were looking for a software environment that would allow for easy simultaneous development of tools for the PC and Macintosh platforms. The most effective and accessible materials and tools in both instances were LEGO bricks and LabVIEW. LabVIEW's graphical environment combined with the familiar building materials made developing creations that incorporated science and engineering principles easy and fun. The success of teachers, students, and developers with this combination led to a partnership between Tufts University, LEGO DACTA, the educational division of the LEGO company, and National Instruments, the makers of LabVIEW, to commercially develop software and supporting curriculum. The partnership focused on software that would allow users to program the RCX (Figure 1), the programmable LEGO brick, in a graphical, icon based environment.



Figure 1. The RCX has three ports (1,2,3) that can take input from sensors and three ports (A,B,C) that can output power to motors, lights, or sound makers. It runs on 6 AA batteries and can hold up to 5 different programs: each executing up to 10 simultaneous tasks. Programs are downloaded to the RCX via an IR tower connected to a computer's serial port.

ROBOLAB, first released in Sept 1998 for both the PC and Macintosh, is powered by National Instrument's LabVIEW. This initial release of the software has only one component, Programmer, which allows users to create programs to control their robotic creations using the LEGO touch, light, temperature, and rotation sensors. Programmer is divided into two sections, Pilot (Figure 2) and Inventor (Figure 3), which address the needs and abilities of kindergarten and college students by varying the options available, the complexity, and the size of the icons. The graphical environment makes programming easy and eliminates the hassle and difficulty of debugging lines of code. It also allows pre-kindergarten and kindergarten students to program even though they cannot yet read and write. Moreover, by using LabVIEW as the base, the software can easily be adapted for changes in the RCX or other computer hardware.

Pilot and Investigator each have 4 levels that gradually add complexity and capabilities. Capabilities include FOR and WHILE loops, subroutines, inter-RCX communication, on board calculations, and the ability to compose and play music. The levels allow for the user to learn how to use the software and the control the RCX by mastering different capabilities and elements of the interface at each level. Users are motivated to continue learning the software as they want to create more complex creations.

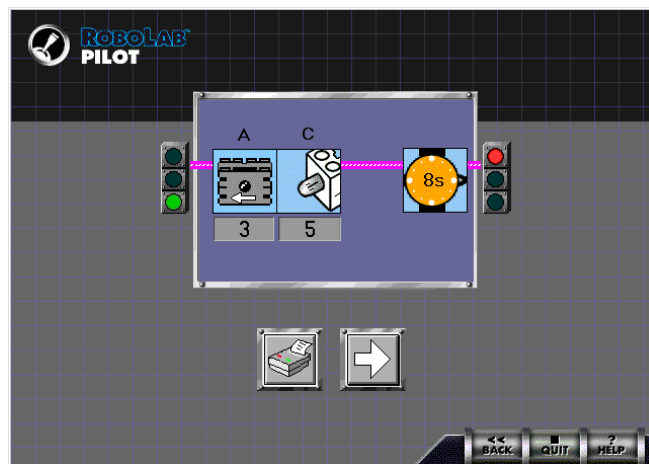


Figure 2. Pilot programs allow users to select options from pull down menus. This simple, Pilot Level 2 program turns a motor and a light on for 8 seconds.

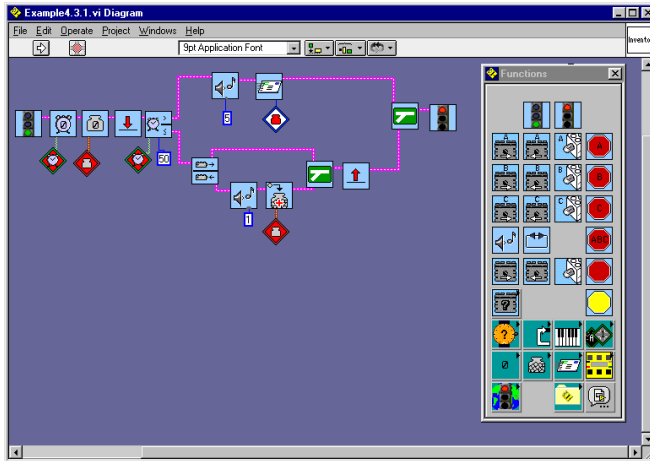


Figure 3. Investigator programs are created by picking icons from a palette, placing them in the window, and wiring them together. This Investigator Level 4 program sets a variable to the number of times a touch sensor has been pressed and then sends that value to another RCX.

ROBOLAB 2.0, released in early 2000, has an additional component entitled Investigator, which allows the RCX to be programmed to acquire data. Investigator was designed to make the process of capturing and analyzing data accessible to kindergartners but also useful and powerful for college students. The diverse needs of this broad user group made it necessary to design a product that would allow for user customization and modular expansion. It is the development of this part that is the main work of this paper.

III. Investigator Development Issues

In developing the Investigator component of ROBOLAB 2.0, the permeating questions were “In what context will people use this software?” and “What capabilities are needed for the different users?” Robotics programs, created in the Programmer section, essentially only involve unidirectional communication. The program is downloaded to the RCX, via the IR tower, the user runs the program and only returns to the computer to debug or modify the program. However, to perform data acquisition, users need to create programs that acquire data and then return to the computer to upload and view the data they have acquired. Once they have uploaded their data they may also need to upload additional related data, and perform calculations, manipulations or comparisons. Initial testing and demos of the idea indicated that a method was needed to connect the RCX program with the data so that problems or changes could be analyzed together. The testing also showed that in a classroom environment multiple groups with different ideas could be using one computer so data and programs needed to be encapsulated in a way that would be obvious and helpful. While the general process was similar across the ages, kindergartners measuring the temperature to determine the peak value and college students looking to find the distribution of strain gauge measurements had very different needs.

IV. Our Solution

To facilitate the bi-directional communication and the related tasks, the Investigator section has projects instead of individual programs. Each project is composed of 5 areas - the Journal

Area, the Program area, The Upload area, The View and Compare Area, and the Compute Area. This created one environment for the program, the data, subsequent analysis, as well as a place for comments and photos. Each student or groups of students could create a project for their experiment or investigation. In each of the areas, users can choose different templates that vary in their capabilities and complexities. Each area can have a number of pages so that different kinds of templates can be used within a single area.

After reviewing a number of projects done in elementary and middle school as well as college, templates were designed to meet the fundamental requirements of the activities. Acknowledging that it is impossible to account for the potential needs of all groups, the software was designed so that additional templates can be downloaded from the web and added at any time. The software's general interface is very similar to a web page with navigation bar on the left (Figure 4).

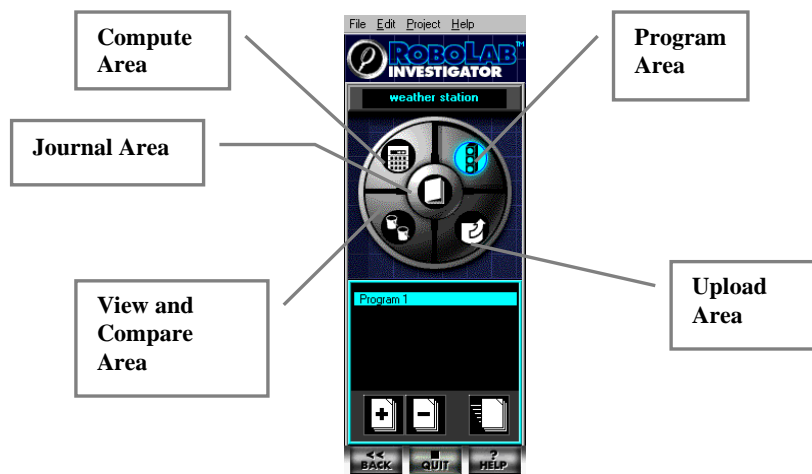


Figure 4. The Navigation Bar – Every Investigator Project has a navigation bar on the left side of the screen that allows users to switch between areas as well as add, subtract, or reorder pages in that area.

The main goal of the Journal area (Figure 5) is to provide users with a place to document information about their investigation. This area was developed to meet the needs of teachers who want to emphasize methods and report writing skills. In this area there are templates for writing predictions or conclusions, as well as the ability to import a picture or page from their project and explain it.

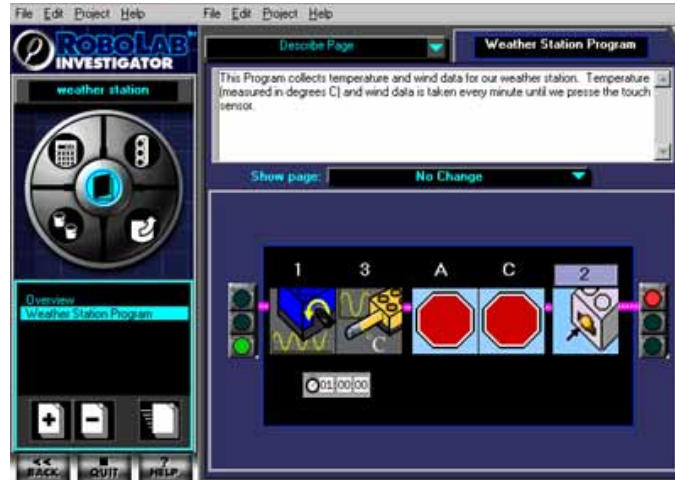


Figure 5. The Journal Area – This template allows you to import a page from another area of your project and annotate it. All templates can be named by the user in the upper right hand corner. The title is show in the navigation bar on the left.

The Program area has 5 templates that vary in level of complexity and capability. The Program Level 1 template allows only data logging to be done with one sensor (Figure 6). This level illustrates the basic concept of data acquisition in Investigator. Users select the sensor they want to use, the interval between points, and the condition when data logging is terminated. They can choose from the 4 LEGO sensors (Touch, Temperature, Light and Rotation) as well as third party sensors that use an adapter (Humidity, Temperature, Sound, pH, Pressure, Position, and Voltage). The basic idea for this area was to present data acquisition with simple templates and gradually adding complexity in the form of more sensors and mobility options.

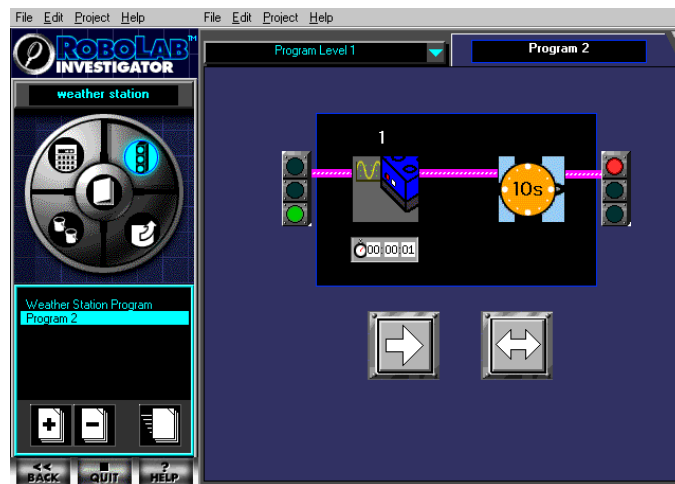


Figure 6. This Level 1 program takes light data from the sensor on Port 1 every second for 10 seconds.

Level 1 is similar to logging software already on the market. The unique part of using the LEGO elements and ROBOLAB is that logging can be combined with motor control and reaction to sensors as well. Users can drive to a location, take data and then drive back thus exploring issues of control, design, and physics. Integrating these ideas provides the opportunity for richer and more realistic experiences. In Program Level 2 the ability to log an additional sensor as well as limited motor control is added. Users can even have the motor speed determined by the value of a sensor. Program Level 3, the highest of the Pilot style levels, gives users 3 steps of motor control and the ability to choose when their two sensors are capturing data (Figure 7.).

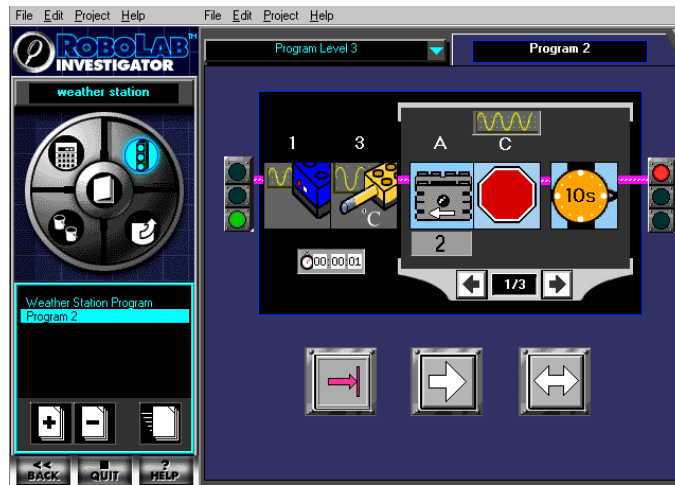


Figure 7. Program Level 3 allows users to have 3 motor steps and to enable and disable logging during each step.

Program Level 4 and 5 give the user an Investigator style interface. At the highest level users have access to every capability of the RCX as well as advanced programming structures (while loops, conditional statements) and the ability to control another RCX over the internet. Users also have access to LabVIEW functions and loops.

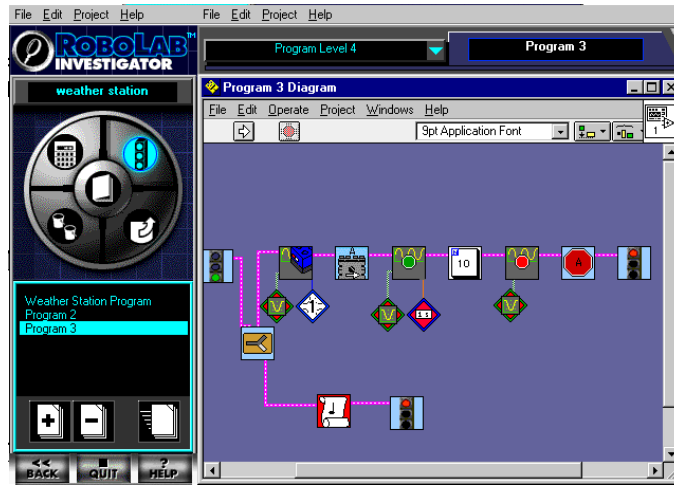


Figure 8. This level 4 program turns on motor A and takes light data every second until it has collected 10 data points. At the same time it also plays a song recorded on the red scroll.

Once a program has been created and downloaded into the RCX it can be run at any time by the user. When the user has acquired the data, they then return to the computer and go to the Upload area to retrieve their data from the RCX (Figure 9). Data is uploaded into colored bins. Each bin with data in it is initially labeled with the name of the sensor that was used to take the data. Subsequent data sets are, by default, uploaded into the bin that corresponds to the sensor they used.

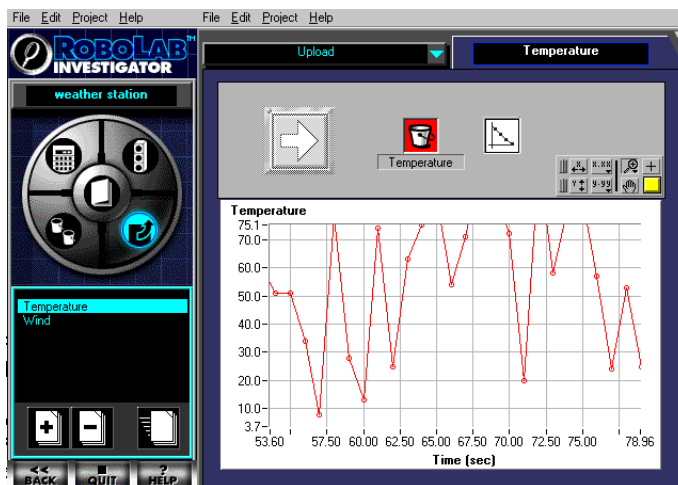


Figure 9. The Upload Area allows users to retrieve data from the RCX. Users can organize their data into bins and select how it is displayed (lines, points, bars, or numbers).

In designing the software, the issue of how to sort uploaded data was important so that it was clear what data came from what sensor and so that it could be used easily in computations. If a class, for example, had built a weather station and were taking temperature and wind data over multiple days – how could they keep data from different days separated but still average all the temperature data? This scenario helped us develop the bin system. In the weather station scenario, every day of temperature data would have its own page in the upload section but all of the runs of temperature data would be in the red bin so they could be reduced later on. Similar to the issues with the templates, it was necessary to make this a flexible system so that if users had different

needs they could assign bins as they wished. Users are free to change the bin location of any data set and the name of the bin.

After uploading data, users have two sections that can help with analyzing and manipulating their data. The View and Compare area provides the tools to extract basic information from the data (min, max, mean) as well as ways to view different buckets of data. This area is designed to help users answer basic questions like “What is the maximum temperature recorded?” or “Is there a relationship between wind speed and temperature?”.

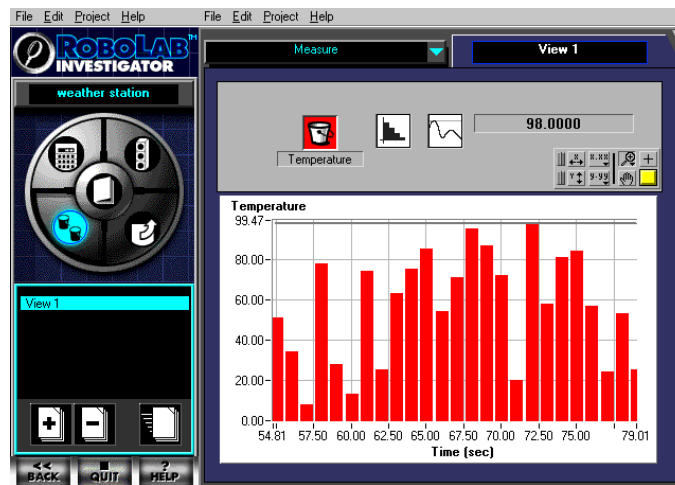


Figure 10. In the View and Compare Area users can select a bucket to analyze as well as several different function and displays. In this example the user has chosen to find the maximum reading in the red temperature bin with a bar graph.

The Compute area allows users to do more complicated manipulations and calculations. The first 3 Compute Levels feature pull down menus that allow users to quickly and easily select options. At these levels, users can perform operations on their data ranging from simple addition and subtraction to finding the sine or tangent of the data set. They can also plot one bin versus another, fit a line to data, and average a set of lines. The examination of activities done in schools and colleges helped us to come up with the functionality of the first three templates. Activities using the temperature and rotation sensor necessitated that there be an easy way to perform operations on the data to change from Celsius to Fahrenheit or determine the distance traveled. Basic physics activities required there be a method of taking integrals and derivatives. To provide for the higher level high school and college activities Compute Levels 4 and 5 allow the user to create their own calculation and manipulations using Inventor Style programming. This gives the users the ability to create histograms from multiple data sets, do exponential curve fits, find peak and well time, and write their own advanced functions.

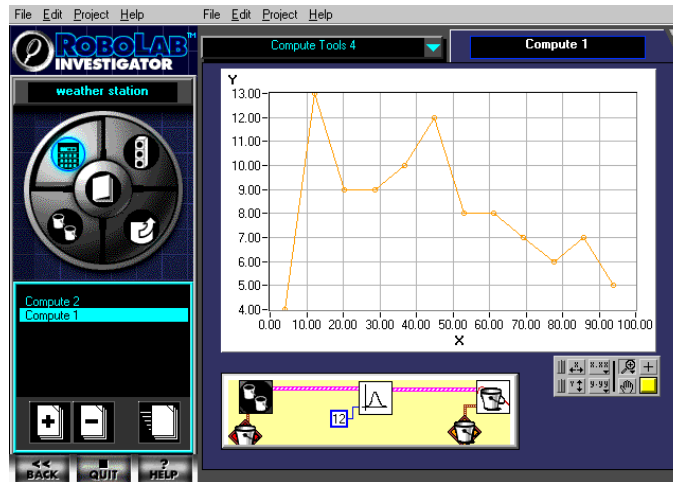


Figure 11. This Compute Level 4 program builds a histogram of the temperature data in the red bin and assigns it to the orange bin. This histogram could be displayed in the View and Compare Area in bar graph style.

Once a student has completed a project, there are a number of ways to share the results. The main publish screen allows users to select which pages they will share and in what order.

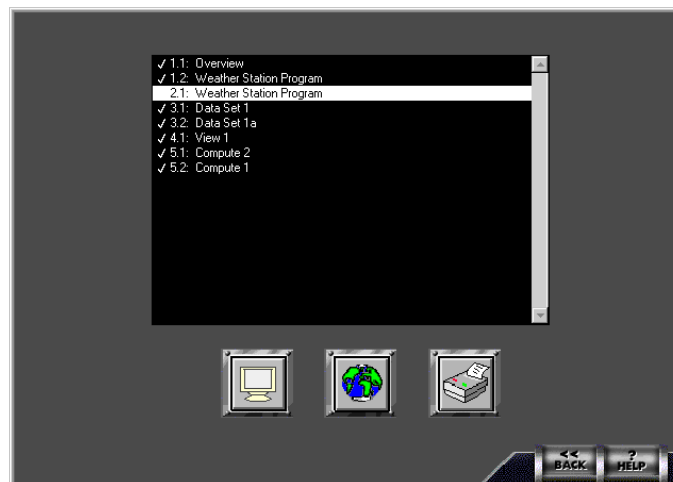


Figure 12. The Publish section allows the user to present, publish to the web or print their project. Double clicking on the page names allows the user to include or exclude specific pages.

Several teachers and professors indicated it would be useful to have a quick and easy way for students to give presentations of their data and results. Figure 13 shows the present option in the publish section.

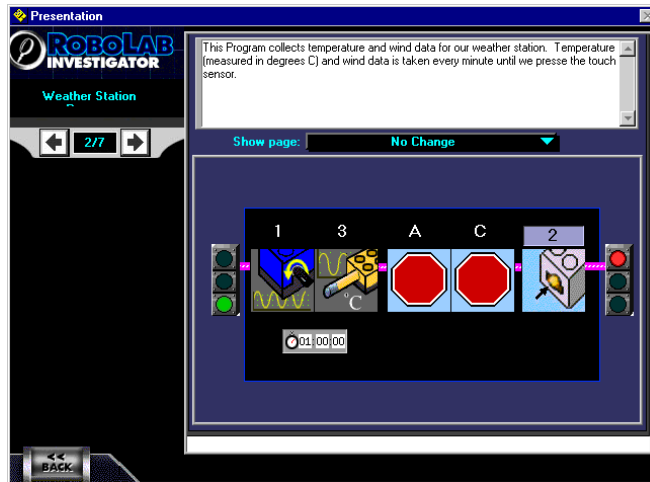


Figure 13. The present sections displays a snapshot of the project page, the title and a method by which to advance to the next slide.

As the Internet becomes a fixture in more and more classrooms, the ability to quickly and easily publish web pages to share information and results is needed in nearly any software application. Investigator projects can be turned into a set of web pages that feature the project pages and the ability to download the project (Figure 14).

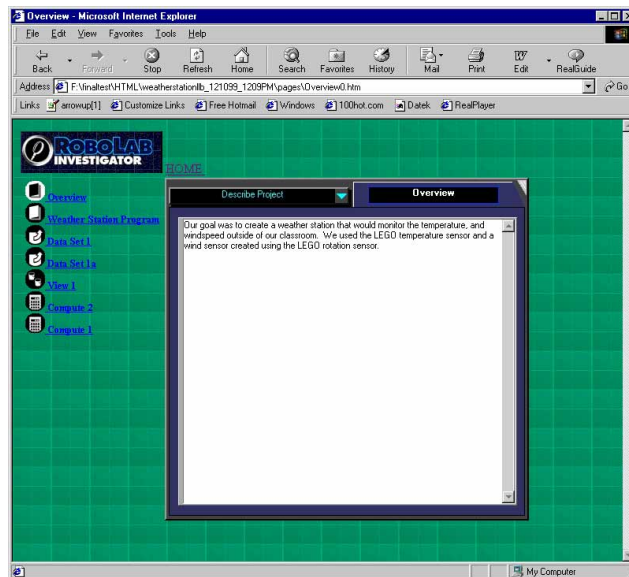


Figure 14. The set of web pages includes all project pages selected by the user. Pages from different areas are differentiated by icon.

The overall Investigator solution is, at the highest level, a graphical solution that incorporates project management, with programming and analysis. Project management includes the encapsulation of investigation so that the software can be easily used in a class and management within the individual project for data, analysis, notes, and images. Being almost entirely graphical

allows the younger students to use it without needing to know how to read. The large icons at the lower levels are similar to their advanced counterparts allowing students to adjust quickly as they advance through the levels. The early “click and choose” levels mimic the structures needed to use the advanced levels that require selecting icons from a palette and wiring them together. Attempting to design software for this diverse audience forced the development of new solutions and approaches to data acquisition that should benefit all users.

V. ROBOLAB 2.0 Issues

The main obstacle and difficulty involved in bringing ROBOLAB 2.0 to any classroom is cost. The software itself requirements (Figure 15) preclude it being used on a number of older computers.

PC	Macintosh
85 MB free hard drive space 32 MB RAM 33 MHz (166 recommended) PC processor Windows 95 or Windows NT	85 MB free hard drive space 32 MB RAM 60 MHz (120 recommended)Power PC processor System 7 or higher

Figure 15: Hardware and software requirements for ROBOLAB 2.0

Ideally, each group of 3 or 4 students requires an RCX, some LEGO bricks and sensors, and a computer. This is difficult in many elementary and secondary school settings where computers are not plentiful in the classroom. Moreover, the cost of \$200-\$300 dollars per group for the LEGO elements and RCX can be prohibitive.

VI. In the Classroom

ROBOLAB 2.0 was beta tested in elementary and college classrooms. In Barbara Bratzel’s 8th grade course “Physics By Design”, students were introduced to Investigator through a temperature scavenger hunt. They logged the temperature as they walked inside and outside of the school. Upon uploading the data, they looked at the data and tried to determine their path based on their readings. This simple introduction used Program Level 1 to give students a basic understanding on how data is acquired and uploaded. With that understanding, they were able to progress to more complex activities. Motor control was introduced as students investigated how fast their cars could go. In this activity they investigated different ways to determine speed. With a light sensor attached to the car (which gives different readings for light and dark), students taped start and finish lines to the floor. They could tell from the data when the car passed over the lines and knowing the distance could calculate the velocity. They also investigated other methods of computing the speed by attaching rotation sensors to the axle. Using the diameter of the wheel they were able to calculate the distance and then differentiate the data to find the velocity.

The software and the RCX empowered the students to collect their own data and see the effects their design or position had on it. Students seemed to have a deeper investment and connection with the data. Their closer relationship to the data gave them a better understanding of the graphing and of the principles related to it. Moreover, the students were very involved and excited about the process. The teacher reported a high level of enthusiasm among students, who

even wanted to come in at recess. Several parents also indicated that the Physics by Design course was discussed more at home than other classes.

Two college level classes, Introduction to Instrumentation and Fluid Mechanics, have used the RCX in their coursework. The goal of Introduction to Instrumentation, a junior level course, is to experimental design and report writing. Using the RCX and the advanced ROBOLAB 2.0 icons, students get a basic introduction to the concept of data acquisition. The RCX is an easy way to introduce the concepts of sampling error, sensor response, and measurement uncertainty. The limitations of the RCX, sampling speed (up to 200 Hz) and buffer size (2000 points), help the students to understand the impact of hardware configuration and capabilities on their ability to take accurate data. The integration of data acquisition with LEGO sensor and motor control also allows the students to complete complicated projects quickly and easily as they only have one interface to use and learn.

This past year students were challenged to build a TANG dispenser that would allow the user to specify the concentration and temperature of the drink. Students first took data with temperature sensors and strain gauges to determine the best configuration for their system and to test the repeatability of their components. The final automated system, controlled by one RCX, required students to have mastered and addressed many experimentation and automation issues. students also had to construct an analytical model of their setup and compare it to their actual results. Students worked for hours perfecting their design for the final test and worked many additional hours on non-required features that made their design “cooler”. The junior level engineers also came up with creative names for their dispenser and devised marketing campaigns to promote their devices. The course and software serves as a jumping off point for students to go on and use LabVIEW and high end data acquisition boards

In the Fluid Mechanics class, ROBOLAB 2.0 and the RCX were used because they provide the quickest and easiest method for running basic experiments. Students in the class needed only one lab session to understand how the RCX acquired data and how to use the software. They were then able to conduct several experiments over the course of the semester in a relatively short amount of time. To study frictional losses in pipes, for example, students used a graduated cylinder, a light sensor, and a piece of Styrofoam to determine the time it took for water to flow from various heights through a piece of Tygon tubing. (Figure 16).



Figure 16. Students used the RCX to start and stop the water flow and log the time it took for the Styrofoam wedge to cross in front of the light sensor and measured the amount of water in the graduated cylinder.

The fluid mechanics students were shown a design that would clamp and unclamp the Tygon tubing and told they could use it if they wished. Much to the amazement of the professors and instructors, nearly ninety percent of the students choose to build their own device and spend hours refining their design. Frustration reigned as students had difficulty getting their simple valve control to work reliably, however they were vested in their designs and spent extra time in the lab to get a working set up. The ability to quickly upload and analyze their data also allowed them to quickly see where they had made mistakes or if part of their device wasn't working properly and make adjustments or repeat the run.

The experiences in these classes indicate that students are willing to put in extra time and effort to do coursework that allows them hands-on ownership of the process. The product of their effort and time is a more complete understanding of the specific topics addressed as well as the peripheral elements. Other engineering classes have started using ROBOLAB 2.0 to look at Hooke's law, creating sensors, and developing anemometers. Additional examples of ROBOLAB 2.0 can be found at the web site for the Center For Engineering Educational Outreach (<http://www.ceeo.tufts.edu>).

VII. Conclusions

Providing students and teachers the tools to perform data acquisition and analysis gives them the power to explore and understand math, science, and engineering. Creating a cross platform software package that can be used from kindergarten through college helps to eliminate the time spent learning the software and allows the focus to be on learning and understanding. With varied curriculums, teaching styles and goals, creating a product that meets everyone's specific needs is impossible. Therefore creating a product that is easy to expand and customize is an effective way of developing a universal product. The ability to create your own tools also empowers the user and gives them a better sense of how data is acquired.

A number of extensions to ROBOLAB are currently in development. Two new templates currently in development allow for easier uploading of data over the Internet and help users to plot data as a 2-D surface plot. Methods to create distributed intelligence systems using multiple RCXs are also being examined. In addition, using the RCX to introduce music theory is also being looked at.

Acknowledgements

The authors would like to thank Ben Erwin, Robert Rasmussen, Mike Giuggio, Cathy Fett, Regina Cadillac, and Barbara Bratzel for all their work developing ROBOLAB 2.0.

Bibliography

1. Capozzoli, P. and C.B. Rogers, *LEGOs and Aeronautics in Kindergarten Through College*, AIAA 96-2245, 1996.

2. Erwin, B., Cyr, M., Osborne, J., Rogers, C. *Middle School Engineering with LEGO and LabVIEW*. presented at VI in Education day at MIT, June 8. Outstanding Application Semi-Finalist, Education Category, National Instruments Week 1998.
3. Martha N. Cyr, Chris B. Rogers, *Enhancing Education with LEGO Bricks and Paperclips*, FEDSM98-5137, Proceedings of FEDSM'98 1998 ASME Fluids Engineering Division Summer Meeting, June 21-25, 1998, Washington, DC.
4. Martha Cyr, V. Miragila, T. Nocera, C. Rogers, *A Low-Cost, Innovative Methodology for Teaching Engineering Through Experimentation*, J. of Engineering Education, Vol. 86, No. 2, pp. 167-171, 1997.
5. John Paul Osborne, B. Erwin, M. Cyr, and C. Rogers, *A Creative and Low-Cost Method of Teaching Hands-on Engineering Experimentation Using Virtual Instrumentation, Laboratory Robotics and Automation*, Vol. 10, No. 2, pp. 63-66, 1998.

MERREDITH PORTSMORE

Merredith Portsmore is the Educational Technology Coordinator for the Center For Engineering Educational Outreach at Tufts University in Medford, Massachusetts. Her main efforts are concentrated on the development and implementation of tools that help teachers bring technology and education into the K-12 classroom. She has a strong interest in distance education via the Internet and professional development for educators. Ms. Portsmore received her B.A in English and B.S in Mechanical Engineering in 1998 and her M.A. in Education in 1999 from Tufts University.

CHRIS ROGERS

Chris Rogers is an associate professor of Mechanical Engineering at Tufts University. He got all his degrees at Stanford University. He spends much of his time either playing with LEGO brick or looking at the behavior of particles in a turbulent airflow.

MARTHA CYR

Martha Cyr is a research assistant professor with the Mechanical Engineering Department at Tufts University in Medford, Massachusetts. She is also Director of the Center for Engineering Educational Outreach at Tufts University and serves on the State of Massachusetts Science and Mathematics Advisory Council. Her main area of interest is effective engineering education for all ages. Dr. Cyr received her B.S. in Mechanical Engineering from the University of New Hampshire in 1982, and her M.S. and Ph.D. in Mechanical Engineering from Worcester Polytechnic Institute in 1987 and 1997.