eBlocks – Embedded Systems Building Blocks to Enable Project-Based Learning

Anuradha Phalke
Department of Electrical and Computer Engineering
University Of Arizona
aphalke@email.arizona.edu

Susan Lysecky
Department of Electrical and Computer Engineering
University Of Arizona
slysecky@ece.arizona.edu

ABSTRACT

We present an interactive platform to enhance STEM education through project-based activities that complement the existing curriculum, while drawing on problem solving skills and team work. The addition of hands-on projects is intended not only to reinforce the various concepts learnt through normal class readings and discussion, but also make these topics multi-dimensional by enabling students to apply these principles in their everyday surroundings. The types of courses that can take advantage of the eBlock platform range from middle school math and science classes to university level engineering courses. In this paper we present the eBlocks platform which can be utilized to implement a wide variety of projects without requiring users to have programming or electronics experience. Specifically, we summarize the preliminarily usability results and discuss possible applications of the eBlock platform.

Categories and Subject Descriptors

H5.2 [**User Interfaces**]: User-centered design; Evaluation/methodology, Interaction styles; Prototyping; H5.m [**Miscellaneous**]

General terms

Design, Human Factors

Keywords

eBlocks, user interfaces, STEM education, interactive embedded systems building blocks

1. INTRODUCTION

The math and science skills of K-12 students within the United States are continuously evaluated [1][11][12], with concern that student rankings are dropping with respect to their international counterparts. Beyond secondary schools we see engineering enrollment and retention at the college and university level is adversely effected [22], fewer students are pursuing science and engineering degrees, with only about 6% of undergraduates majoring in engineering [4]. Even further reaching, is the continued debate as to whether there is an "engineering gap" in the United States in that many engineers are retiring but there are not enough engineers to fill these spaces [13][33][35]. The U.S. Bureau of Labor Statistics projects 15 of the 20 fastest growing occupations through 2010 will require significant math and science backgrounds [31]. Given these statistics, we look for possible methods to enhance STEM (Science, Technology, Engineering and Mathematics) performance, as well as increase interest in pursuing STEM careers.

The benefits of project based learning have been studied for decades [2][14][15][17][25][28], where students to participate in the learning process rather than passively listening to a lecture, reading a book, or watching a video. Studies have shown that students who are able to investigate the world around them and topics that are relevant to their everyday surrounds not only achieve higher scores than students in traditional classrooms [9][30] but are also more engaged [3][21]. While these learning models have been shown to be effective, traditional STEM courses struggle to provide an active learning environment. Incorporating advanced learning technologies into the curriculum are often times not feasible because test equipment is too expensive and school budgets are already spread thin, while building a custom solution requires a large breadth of programming and engineering expertise. Thus, we are faced with a challenge to develop a platform that is usable by a wide range of teachers and students, cost-effective for schools, and support a variety of STEM learning goals.

We propose to utilize and extend the eBlocks platform [5], a platform composed of numerous fixed-function building blocks, initially developed to enable non-expert users to create customized monitor/control systems. The eBlock platform has the potential to provide an interactive and customizable project-based learning environment that allows students to see how the topics studied in class relate to real-world applications by enabling students to create their own customized systems. In this paper we provide the motivation behind the eBlocks platform, a glimpse of the underlying platform implementation, and initial usability studies to demonstrate the feasibility of the eBlocks platform within STEM education. Lastly, we discuss potential applications of the eBlock platform and future directions.

2. RELATED WORK

While a great deal of effort has been put forth to improve STEM education, we only highlight a few such projects that seek to include technology as a mechanism to captivate participants. Logiblocs [16] are one such technology consisting of small plastic blocks that users snap together to build various systems and consist of light sensors, buttons, AND, OR, NOT, speakers, bleeps, LEDs, etc. Logiblocs are targeted for primary education and educational toys. Children are able to design various systems and gain insight into the design process by trial and error. While many systems are viable such as flashing warning lights, light breaker alarms, or simplistic robots, the simplistic interfaces limit the possibility to build more advance systems to accompany STEM projects beyond the primary school level. Electronic Blocks [37] also utilize small building blocks consisting of

Figure 1: eBlocks provide a platform which enables non-experts to build a variety of interactive, hands-on projects.

processors incorporated inside of LEGO Duplo Prima blocks. Users simply stack the correct combination of blocks to produce the desired output. Electronic Blocks consist of blocks that detect light, sound, and touch; produce physical output such as illuminating a light, producing sound, or movement; and intermediate blocks alter the expected action by inversion, toggle, or delay. Electronic Blocks are aimed at students between the ages of 3 to 8 years old and are again limited in use for older students due to the simplicity of the blocks.

Perhaps the most closely related project targeting STEM education in primary and middle school is that of MIT Crickets, having evolved from the MIT Programmable Bricks project [18][19]. A Cricket is a tiny computer, powered by a 9-volt battery that can receive information from two sensors and can control two motors. A key principle in Crickets is that people program them to perform a variety of functions. To make programming simple, the Logo language is used [26][27] – a simple, graphical, highly intuitive language. Crickets provided the foundation for the Lego Mindstorm product [36], consisting of numerous sensor and actuator Lego blocks that can be connected to a central microprocessor block to build a variety of small robots, again programmed using a simple graphical language. Crickets provide a general programmable computer, requiring programming by users. While we can learn much from the MIT Cricket approach, we have to consider situations where learning programming languages or having access to computers are not an option. Our goal is to complement the current curriculum without requiring programming or electronics knowledge.

The Infinity Project [10] is an intriguing year long course available to high schools to expose students to, and excite students about, careers in engineering and technology. The course curriculum is designed for students who have completed Algebra II and at least one science laboratory. A graphical software design environment is provided to run on a PC accompanied by a Technology Kit including a central board built around a digital signal processor (DSP), multimedia peripherals, such as digital web camera, speakers, and microphones, and an accompanying text book. Students learn about engineering through the design and experimentation on topics including the Internet, wireless communications, digital imaging, and music and audio processing. The Infinity Project is a fascinating project but mandates prerequisites in math and science. Additionally, we want to involve students before they reach high school. Our goal is for students of all ages and background to be able to easily utilize the eBlock platform to build and configure a variety of systems, providing an opportunity for hands-on projects relating topics discussed in class. While the goals of each project differ,

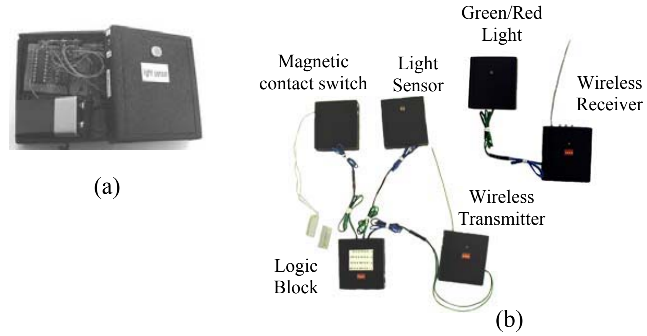

Figure 2: (a) Internals of an eBlock light sensor, (b) implementation of a Garage-open-at-night system using eBlock prototypes.

our proposed project can serve as a complement to the Infinity project. Studies have shown students need to be engaged early. The eBlock project will hopefully give middle school (and eventually elementary school) students the confidence to continue to pursue math and science and participate in programs, such as the Infinity project, throughout their education.

Numerous platforms are available which expose students to technology, with many focusing on the underlying platform. While an introduction to, and an understanding of, the technologies that makeup the digital age is important, we strive to enable students to utilize technology without having expert knowledge of the underlying platform.

3. EBLOCK OVERVIEW

The inspiration for the eBlock platform arose from a simple observation that there exist numerous applications across a variety of domains (residential, commercial, medical, and so on) that can be solved with a handful of sensors and some control logic. For example, a homeowner may want a system to monitor whether a garage door was left open at night. Figure 2(b) illustrates how this system can be implemented using a light sensor, contact switch, a LED (light-emitting diode), and control logic that turns the LED red when no light is sensed (daytime) combined with an open contact switch (indicating the garage door is open). Additionally, if the homeowner wants to place the red/green warning LED away from the sensors, wireless transmit and receive modules can be utilized. As another example, a homeowner might want to set up a system that detects if their child is sleepwalking in the dark. Figure 3(a) illustrates such a system involving a motion sensor block and a light sensor block, feeding into a logic block detecting the motion sensor outputting true and the light sensor outputting false, wirelessly feeding into an LED or buzzer block. Other examples include, but not limited to, a daytime doorbell, detecting motion on the property, detecting if a side gate is open before letting pets into the backyard, monitoring the availability of conference rooms within a company, or monitoring nocturnal activity by turning a recording device on when motion is sensed within a defined area.

With so many application possibilities, why aren't these systems more prevalent? One reason is cost; most applications are too specialized to be cost effective when one considers the many real costs (packaging, marketing, store placement, etc.) of introducing a new consumer product into the market. Thus, very few off-the-

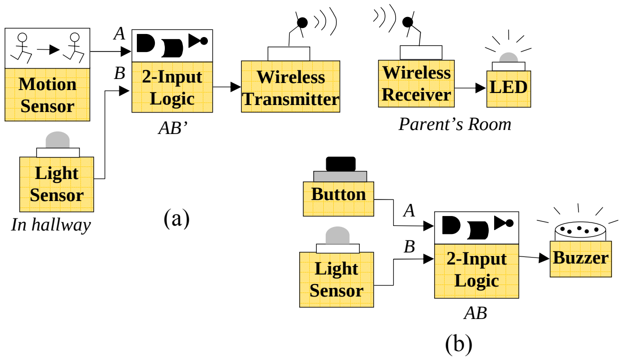

Figure 3: Various applications built with eBlocks, (a) Sleepwalking Detector and (b) Daytime Doorbell.

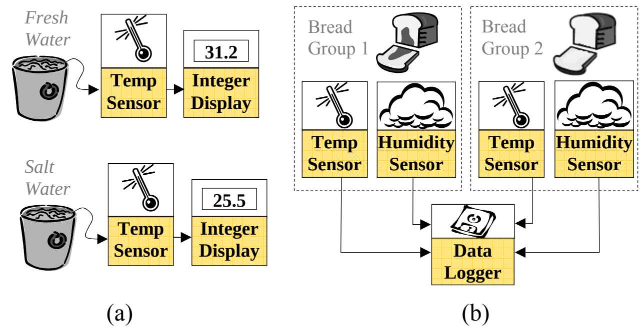

Figure 4: Sample eBlock projects. In (a) eBlocks are used to observe the freezing of fresh and salt water. In (b) eBlock temperature sensors are used to observe the effects of temperature and moisture effect mold on a piece of bread.

shelf applications appear in the market place. Furthermore, while these applications are seemingly simple, building a custom working system from standard electronic components is beyond the skills of ordinary people (homeowners, teachers, office workers, etc). In fact, building these systems is even a challenge for most engineers who haven't been specifically trained in embedded system design. Lastly, hiring an engineer to build a custom solution is possible, but the cost is seldom justifiable.

The eBlocks platform [5][8] emerged from a desire to empower regular people, having no programming or electronics experience, to build custom electronic systems. Rather than creating specialized systems for each application scenario, the eBlock platform is composed of commonly used building blocks that users connect blocks together like Legos™ to form an application. Because the platform consists of building blocks, the development cost of each block can be amortized over a larger volume, yielding lower costs. Furthermore, through the use of hardware wrappers the underlying platform implementation is made transparent to users, alleviating the need for programming or electronics experience.

3.1. EBLOCK PLATFORM

The key to the eBlocks platform is to integrate compute intelligence with previously "dumb" sensors and actuators, as shown in Figure 2(a). In our case, we utilized a PIC16F690 microprocessor [20]. The processor hides low-level interfacing details and communication between nodes so that users simply connect components together to specify the functionality of the application.

A variety of building blocks have been identified, with approximately 20 different blocks physically implemented. These building blocks can be classified into one of the following four categories.

- Sensor blocks - monitor the environment, including motion sensors, light sensors, buttons, contact switches, and so on
- Compute blocks - perform basic logic transformations (e.g. AND, OR, NOT) or basic state functions (e.g. prolong, toggle, trip, pulse).
- Output blocks - provide stimuli, and include light-emitting diodes (LEDs), beepers, electric relays, etc.
- Communication blocks - provide wireless point-to-point communication or replicate a signal to send to multiple blocks.

The initial eBlock platform contains both fixed function and programmable blocks. Fixed function blocks have a specific predefined function that may have slight configurability (i.e. which logical operation to perform, the length of time a signal is prolonged). A generic 2-input, 2-output programmable block is also supported, the programmable block can be configured to perform any user specified function. Using the eBlock simulator, the desired functionality of eBlock system is defined. The simulator interface partitions the desired functionality unto a programmable block (or multiple blocks depending on the desired functionality) and downloads the new function to the programmable block using a PC interface. The programmable block is easily reprogrammed without requiring compilation or device programming by the user. While we have strived to keep the interface and tool chain accessible to novice users, we note that it is more likely that advance users or users with some programming background would opt to utilize the programmable block.

The initial version of eBlocks operate on Boolean data, blocks send and operate on "yes" or "no" packets. Applications are built by snapping blocks together and the order in which blocks connected specifies the functionality of the system. Extensive usability testing has been performed on Boolean blocks interacting with close to 500 participants [5][6][7]. Findings indicate that over 50% of participants, with little or no training, are able to successfully create a variety of applications within 10 minutes of being introduced to the platform. Boolean values however limit the types of projects that can be built, in particular STEM projects require building blocks that operation within the integer domain. Thus, extending the current platform to incorporate integer based eBlocks is essential.

4. EBLOCK EDUCATION KIT

eBlocks have the potential to enhance STEM education by providing students with the opportunity to build customizable and interactive projects that complement the specific topic studied without the burden of first having extensive programming or electronics experience. Course work becomes more interesting, memorable, and other skills such as problem solving and team-

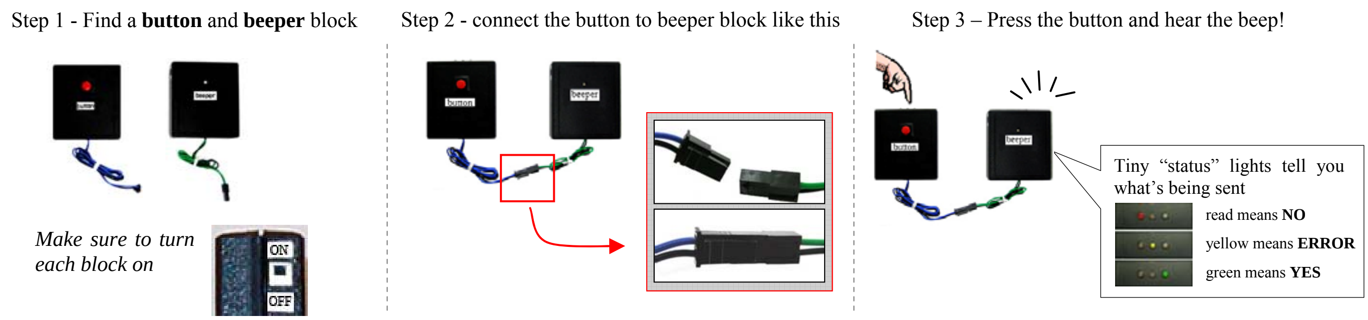

Figure 5: A one page starter guide is provided to participants to quickly illustrate how eBlocks are powered on, connected to one another, and how to determine what the blocks are communicating to one another.

work are utilized and developed. Thus, we set out to define an eBlock Education Kit to target STEM education starting with middle school STEM education with future extensions reaching to the high-school and university level.

To begin we needed to identify which building blocks (i.e. sensor, intermediate, and output blocks) to include in the eBlock Education Kit so that students are able to build a wide variety of applications. Yet, at the same time we must balance the number of components included as not to overwhelm students with a gigantic catalog of eBlocks. A set of interactive projects for middle school students were identified by studying the topics included in the existing STEM curriculum listed on the local school district’s webpage [34] as well as several other websites that provide descriptions of middle school science fair projects [23][24][29]. Based on these project ideas, we can determine the types of building blocks required in the initial set.

As we discover the project possibilities, we are also compiling a project guide booklet. The booklet is intended to get teachers and students started by illustrating sample systems that can be implemented with the eBlock Education Kit. Each project begins by posing a question. For example, why don’t oceans freeze as quickly as lakes? Next, steps are outlined to help students connect various eBlock components together to build a project that tests the posed question. Figure 4(a) illustrate how temperature sensor and integer display eBlocks can be snapped together to observe the freezing point between containers of fresh and salt water. Students are then asked questions that require interaction with the platform, such as placing both containers into a freezer and recording the temperature of each container every 10 minutes. Additionally a series of follow up questions are provided, requiring students to reflect on the experiment. Figure 4(b) illustrates the setup for another experiment intended to monitor the effects of temperature and moisture on mold. While the application is different we can see that some of the blocks can be reused to create different projects. Furthermore, while the project guide booklet provides numerous projects it serves as a starting point for students and teachers to create their own projects.

As we identify and develop each building block within the platform, we must also ensure all of the resulting physical prototype have intuitive interfaces such that user can readily understand the functionality of that block without extensive training. Thus, as part of the iterative design process, we have also begun testing to determine the effectiveness of the newly proposed interfaces.

5. USABILITY EXPERIMENTS

5.1. Prototype-based Usability Testing

Utilizing the new integer-based eBlock prototypes, we conducted a series of informal experiments to see what aspects of the eBlock Education Kit were readily understood and what aspects were confusing. While we ultimately plan to perform usability testing with the target audience (middle school students), we started usability testing with small groups of college level students to provide an initial glimpse into the interface design and provide an opportunity for refinement before deploying the eBlock platform into middle schools. If an interface is not usable by university students, the interfaces developed would have no hope in a middle school setting. These experiments provide an initial evaluation of the various interfaces and a second testing phase is planned to target middle school students.

Participants were categorized into two groups, non-expert and expert. Non-expert participants are students at the university who have no engineering background, students in non-engineering majors such as pharmacy, accounting, or political science. Expert participants are students who have had at least one semester in an engineering major such as computer engineering, chemical engineering, or hydrology.

All participants were provided with a handout containing a brief introduction to eBlocks, as shown in Figure 5. The flyer quickly illustrates how blocks are powered on, connect to one another, and communicate with one another. Additionally, participants were provided with a set of physical blocks and an eBlock catalog containing descriptions of each block’s functionality and interface. These handouts were the only materials provided to participants; we did not provide an oral introduction or answer any questions during testing.

A written survey was also provided that asked participants to build the two systems shown in Figure 6. The first application was not utilized to determine the usability of the integer interfaces (as no integer interfaces were utilized). Rather, this example merely served to indicate if users were at least able to understand how to utilize the eBlock platform. Figure 6(b) illustrates the integer-based application students were asked to construct. Note, the block diagram was not provided for the second system, rather a

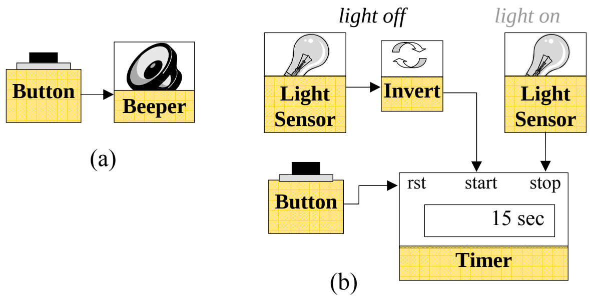

Figure 6: Systems built by participants in prototype-based usability testing, (a) corresponds to doorbell application of System 1 and (b) corresponds to light usage monitoring application of System 2.

textual description of the desired system functionality was provided. In this experiment, we were interested if users would be able to utilize the timer depicted in Figure 6(b). When start = 1 (or "yes") the timer is activated and tracks the number of seconds, when stop = 1 the timer is deactivated, and when rst = 1 the timer display zeros out and the internal counter is reset to zero. Table 1 illustrates the results of these experiments. All participants were easily able to construct the first system, indicating a basic understanding of the eBlock platform. The second system which is more complex, mixing both Boolean and integer blocks, yielded slightly lower success rates. However, 40% of the non-expert participants were able to successfully construct the system, with another 40% close to correct with a minor error in the connection of blocks. It is not surprising to see that expert participants had higher success rates, with all expert participants able to successfully build the corresponding systems. On average experts finished constructing both systems in 8 minutes, while non-experts required 20 minutes.

Because this was the first time participants had encountered eBlocks, we also decided to test if training made any difference in the success rate. In these experiments participants were additionally provided with a training example which included a system description, a block diagram with step-by-step instructions indicating how to connect and configure each block, and a description of how each block contributes to the desired functionality. Many of the blocks used in the training example could then be reused to implement the application illustrated in Figure 6(b). Success rates for non-expert participants increased from 40% to 50%, with another 38% close to constructing a working project. Surprisingly, we found that success rates for expert participants decreased compare to expert participants with no training, as illustrated in Table 2. These results warrant further investigation. The small sample size may have contributed to these irregularities or it may simply be that the training example provided too many details and caused users to second guess themselves or ignore the training phase altogether.

5.2. Simulator-Based Usability Testing

While physical block prototypes are an ideal testing platform, the eBlock simulator shown in Figure 7 was also utilized to enable a larger number of student to participate, as well as enable rapid testing of a numerous interfaces. The simulator is a Java-applet-based graphical user interface (GUI) for the design entry and simulation of various eBlock systems. A pallet located on the right edge of the simulator contains a variety of blocks users can

Table 1: Success rates for individuals who did not received training, with participants who were close to correct in parenthesis.

	% Success		Total # Participants
	System 1	System 2	
Non-expert	100 % (0 %)	40 % (40 %)	5
Expert	100 % (0 %)	100 % (0 %)	2

Table 2: Success rates for individuals who received training, with participants who where close to correct in parenthesis.

	% Success		Total # Participants
	System 1	System 2	
Non-expert	100 % (0 %)	50 % (38 %)	8
Expert	100 % (0 %)	57 % (43 %)	7

select to utilize within a system simply by dragging blocks into the workspace. The blocks are organized as sensor, output, or computation/communication blocks, and can be viewed by clicking on the corresponding tabs located at the top of the pallet. Users establish connections between blocks by drawing lines between circular representations of blocks' input and outputs. Additionally, users can configure blocks by clicking on the DIP switch image, toggling individual switches between "yes" and "no". eBlocks that sense or interact with their environment, such as light or motion sensors, include accompanying visual representation of their environmental stimuli/interaction to simulate the corresponding environment. For example, each light sensor is accompanied by a light bulb icon that a user clicks to toggle the light on and off. The light bulb indicates the sensor's external environment (e.g., whether the sensor detects the presence or absence of light). The electric relay is connected to a lamp icon that is on or off depending on the block's input. In the bottom left corner of the simulator, a gray text box provides context-sensitive help. If a user places the mouse cursor over blocks in the simulator, the block's description and interface automatically appears in the text box. While the eBlock simulator is behaviorally correct, the simulator does not capture all low-level timing details.

Utilizing the eBlock simulator to gauge usability is expected to

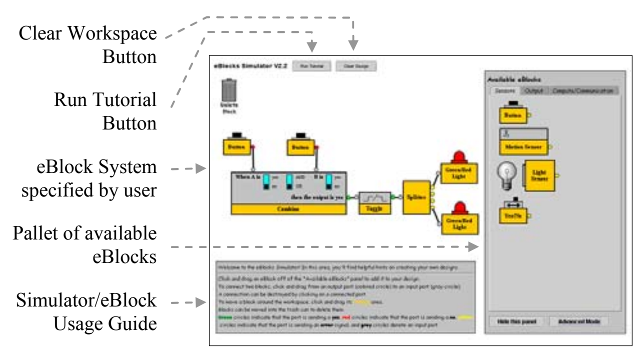

Figure 7: Java-based eBlock simulator that enables participants to build, test, and refine various eBlock systems without the overhead of physical blocks.

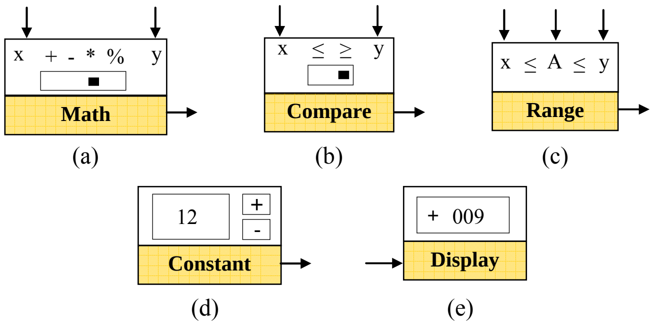

Figure 8: Additional integer-based eBlocks.

yield slightly higher success rates than usability experiments conducted with the corresponding physical prototypes. In the initial testing phases we found that users with little or no engineering background did not know how to use components such as DIP switches or slide switches. Additionally, users frequently forget to power on each block, had trouble connecting blocks and attempted to connect two outputs or two inputs together, or crossed the ground and data wires. While these issues are not fundamentally related to the blocks' functionality, they nevertheless reduced the resulting success rates. Most of the initial experiments provided no training for users, thus over time we believe the usability obtained from physical testing would increase to match those obtained in the simulator testing environment.

5.2.1 Additional Integer eBlocks

In addition to the timer block shown in Figure 6(a), we also expanded the eBlock library to include the five new integer based eBlocks shown in Figure 8. The math and compare blocks include a slide switch to enable a user to configure the block to perform various operations. The range block is a multi-function block. If all three inputs are connected, the range block performs the comparison $x \leq A \leq y$. If the x and A inputs are connected, the range block performs the comparison $x \leq A$. Lastly, if the A and y inputs are connected, the range block performs the comparison $A \leq y$. The constant block consists of a display indicating a constant integer value outputted by the block. A "+" button is provided to increase the block's value, and a "-" button is provided to decrease the block's value. Lastly, the display block receives an integer value and displays the corresponding value.

5.2.2 Integer Block Testing

To determine the usability of the various integer-based blocks, we performed experiments with 11 participants who were incoming engineering freshmen and 19 undergraduate students who were in non-engineering majors. Participants utilized the eBlock simulator to build various integer-based systems. Students were asked a total of 4 questions, one question from each of four problem classifications listed in Figure 9. Questions classified as "Simple Math" requires construction of a math formulation such as 5 + 4, requiring a single math block. Questions classified "Compare" requires construction of a comparison such as 4 > 2, the comparison can be implemented utilizing either the compare block or by connecting two out of the three inputs of the range block. Questions classified as "Cascaded Math" is a slightly more difficult math formulation requiring two operators such as 2 * 4 – 5, where the output of one math block is used as the input to a

Q1. Simple Math (12-3)

Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates "12 – 3" and displays the result on a Display eBlock.

Q2. Simple Math (4*3)

Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates "4 * 3" and displays the result on a Display eBlock.

Q3. Compare (6>=3)

Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 6 is greater than or equal to 3.

Q4. Compare (6<=15)

Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 6 is less than or equal to 15.

Q5. Cascaded Math (5+2*4)

Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates "5 + 2 * 4" and displays the result on a Display eBlock.

Q6. Cascaded Math (7-(6/3))

Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates "7 – (6 / 3)" and displays the result on a Display eBlock.

Q7. Range (2<5<9)

Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 5 is greater than 2 and less than 9.

Figure 9: Questions posed to participants to determine usability of newly added integer-based eBlocks.

second math block. Questions classified as "Range" requires construction of a range formulation such as 3 < 6 < 8. The range operation can be implemented by either using all three inputs of the range block or by utilizing two compare blocks and a 2-input combine block. One comparison block evaluates 3<6, the second comparison block evaluates 6<8, then both outputs are fed to the 2-input combine block to see if both comparisons are true. Participants were allotted 25 to 30 minutes to complete all 4 questions.

The majority of participants were able to utilize the math block. As shown in Table 3, 82% of users are able to successfully utilize a single math eBlock, while 72% of users able to cascade two math blocks. Within the current integer based block definitions there are two methods to construct a comparison, the comparison block or the range block. Because the comparison and range block have overlapping functionality, we wanted to determine if a single block could be provided to perform both operations. Participants asked to utilize the comparison block to perform a compare operation yielded an average success rate of 81%, as shown in the "Compare using Compare eBlock" entry. Participants asked to utilize the range block to perform a compare operation experienced a significantly lower success rate of 54%, as denoted in the "Compare using Range eBlock" row. Similarly, the range operation can be implemented utilizing either a range or comparison block. Participants using the compare block to perform a range operation yielded a success rate of 13%, as shown in the "Range using Compare eBlock" entry. Participants

Table 3: Success rates for non-expert participants in simulator-based usability experiments given various integer-based interfaces.

	Percent Success	Number of Participants
Simple Math	82 %	49
Cascaded Math	72 %	60
Compare using Compare eBlock	81 %	26
Compare using Range eBlock	54 %	24
Range using Compare eBlock	13 %	15
Range using Range eBlock	53 %	15

provided with a range block yielding an average success rate of 53%, as listed in the “Range using Range eBlock” row.

In both the compare and range experiments the users achieve higher success rates utilizing the block most closely matching the desired functionality. Soloway, et al. performed studies indicating that programming constructs need to “cognitively fit” with the user to be effective [32], thus it is not surprising that we similarly find users achieve higher success rate when a block which closely fit the desired functionality is provided. As part of future work, we plan to conduct additional experiments in which users are provided with both the comparison and range blocks, to determine if higher success rates are actually achieved, or if too many blocks lead to confusion.

6. ROLE OF EBLOCKS IN EDUCATION

While the eBlock Education Kit is presently being developed and tested with middle school STEM courses in mind, we foresee the use of this platform at the high-school and university level. The flexibility of the eBlock platform enables use across a wide range of age groups and disciplines.

At the middle school level the eBlock platform can complement the existing curriculum by enabling students to build complementary monitor/control systems and test out the ideas studied in class. In addition to improving STEM performance, we hope students will begin to think about how technology is integrated into their everyday surroundings and begin to question how it all works. Because the eBlock platform is intended to complement the existing curriculum, rather than redefine the existing curriculum, we do not anticipate any low-level discussion about the underlying implementation or discussion of the actual engineering principles. An afterschool environment would however enable greater flexibility and may provide an opportunity for student to learn more about engineering and high-level engineering concepts.

At the high school level we similarly envision eBlocks complementing the existing curriculum, which would limit discussion of the underlying platform. However, at the high school level courses such as the Infinity Project [10] are intended to expose students to engineering. Depending on the type of courses available, expertise of the instructors, and the availability of resources, the eBlock platform can be similarly be utilized as an introduction to embedded systems. At a higher abstraction, students can practice problem solving by building customized monitor/control systems for various applications, utilize a variety of sensors and actuators, and learn about data collection and analysis. At the lowest level students can utilize eBlocks as a low-cost platform to start investigating communication, interfacing, and usability issues.

At the university level students in engineering disciples can utilize eBlocks in a variety of programming and hardware courses. Many engineering students spend weeks or months learning theory before every being able to build a working system. The eBlock platform provides a quick and easy starting point for engineering students to build a variety of computing systems. Then as students learn about various topics, such as interfacing or interrupt service routines, the platform can be dissected to demonstrate the topic studied within a larger working system. By utilizing a top-down methodology students have a better perspective of how each of the topics studied relate to one another. Similarly, the logic and state blocks can be utilized by students in an introductory programming course as a physical platform or within the simulation environment to setup and test Boolean equations and state-based relationships through a trial and error methodology. Students can relate the functionality of these Boolean and state operators to everyday situations, such as detecting light or button presses to control a doorbell, before applying these operators to programming constructs. Lastly, the eBlock platform has the potential to enhance other university STEM programs. Similar to the middle school application, these students can utilize the eBlock platform to develop complementary monitor/control systems, and illustrate the role of technology within a particular discipline.

7. CONCLUSIONS AND FUTURE WORK

To stimulate interest and improve performance in STEM education we have proposed the use of the eBlock platform to enable the construction of interactive projects that complement the existing curriculum. Initial experiments show that roughly half of the novice participants can readily utilize these building blocks with little or no training. While the initial results are promising, many questions remain unanswered. As future work we must consider a more diverse group of participants, including students at the middle and high school level. We are current working with local middle schools and community based organizations to begin testing the usability and feasibility of the eBlocks platform. Furthermore, the eBlock systems constructed by participants contained only a handful of blocks. We must also investigate the usability of the eBlock platform given increasingly complex applications, as well as the ability for users to define and design their own customized system without the aid of a project booklet.

8. REFERENCES

[1] Arizona Department of Education. Arizona's Instrument to Measure Standards (AIMS) Results, State Report Card, http://www.ade.state.az.us/srcs/statereportcards/StateReportCard06-07.pdf, 2007.

[2] Barron, B., D. Schwartz, N. Vye, A. Moore, A. Petrosino, L. Zech, J. Bransford. Doing with Understanding: Lessons from Research on Problem- and Project-Based Learning. The Journal of the Learning Sciences, 7, pp. 271-311, 1998.

[3] Blumenfeld, P. C., T. Kempler, J. S. Krajcik. Motivation and cognitive engagement in learning environments. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences, pp. 475-488, 2006.

[4] Committee Prospering in the Global Economy of the 21st Century: An Agenda for American Science and Technology; Committee on Science, Engineering, and Public Policy. Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future. Washington DC: National Academy Press, 2007. http://www.nap.edu/catalog/11463.html

[5] Cotterell, S., K. Downey, F. Vahid. Applications and Experiments with eBlocks - Electronic Blocks for Basic Sensor-Based Systems. Sensor and Ad Hoc Communications and Networks, October 2004.

[6] Cotterell, S., F. Vahid. A Logic Block Enabling Logic Configuration by Non-Experts in Sensor Networks.CHI, April 2005.

[7] Cotterell, S., F. Vahid. Usability of State Based Boolean eBlocks. HCII, July 2005.

[8] eBlocks Science Kits. http://www.ece.arizona.edu/~uecs/eb_kits/index.html, 2008.

[9] Geier, R., P., Blumenfeld, R. Marx, J. Krajcik, B. Fishman, E. Soloway. Standardized test outcomes of urban students participating in standards and project based science curricula. Proceedings of the sixth international conference of the learning sciences, 310-317, 2004.

[10] Infinity Project, The. http://www.infinity-project.org/, 2006.

[11] Institute of Education Sciences, U.S. Department of Education. International Outcomes of Learning in Mathematics Literacy and Problem Solving: PISA 2003 Results from the US Perspective. http://nces.ed.gov/surveys/PISA/PISA2003highlights.asp, 2007.

[12] Institute of Education Sciences, U.S. Department of Education. Trends in International Mathematics and Science Study (TIMSS) 2003, http://nces.ed.gov/timss/Results03.asp, 2007.

[13] Jackson, S. The Quiet Crisis: Falling Short in Producing American Science and Technical Talent. Building Engineering and Science Talent (BEST) Report, 2004.

[14] Kovac, J. Student active learning methods in general chemistry. Journal of Chemical Education, Vol. 76, No. 1, pp. 120-124, 1999.

[15] Krajcik, J., C. M. Czerniak. Teaching science in elementary and middle school: A project-based approach (3rd ed.). Mahwah, NJ: Lawrence Erlbaum, 2007.

[16] Logiblocs, http://www.logiblocs.com/, 2007.

[17] Lunsford, B., Herzog, M. Active learning in anatomy and physiology: Student reactions and outcomes in a nontraditional AP course. The American Biology Teacher, Vol. 59, No. 2, pp. 80-84, 1997.

[18] Martin, F., et. al. The MIT Programmable Brick. http://lcs.www.media.mit.edu/groups/el/projects/programmable-brick/, 2006.

[19] Martin, F., et. al. Crickets: Tiny Computers for Big Ideas. http://lcs.www.media.mit.edu/people/fredm/projects/cricket/, 2006.

[20] Microchip Technology Inc. PIC16F631/677/685/687/689/690 Data Sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/41262E.pdf, 2008.

[21] Mistler-Jackson, M., N. B. Songer. Student motivation and internet technology: Are students empowered to learn science? Journal of Research in Science Teaching, 37, pp. 459-479, 2000.

[22] National Science Board. Science and Engineering Indicators 2006, http://www.nsf.gov/statistics/seind06/, 2006.

[23] Newkirk, K. Selah Intermediate School Science Projects, Science Project 2006 Index, http://www.selah.k12.wa.us/SOAR/SciProj2006/index. htm, 2006.

[24] Phillipsburg Christian Academy Science Fair Projects, http://www.fellowshipch.org/pcasciencefair.html, 2007.

[25] Price, S., Y. Rogers. Let's get physical: the learning benefits of interacting in digitally augmented physical spaces. Computers & Education, 43(1-2), August 2004.

[26] Resnick, M. Behavior Construction Kits. Communications of the ACM 36, No. 7, pg. 64-71, 1993.

[27] Resnick, M., S. Ocko, S. Papert, LEGO, Logo, and Design, Children's Environments Quarterly 5(4), pg. 14-18, 1988.

[28] Richards, L., et. al. Promoting active learning with cases and instructional modules. Journal of Engineering Education, Vol. 84, No. 4, pp. 375-381, 1995.

[29] Rubin, J. Science Fair Projects Resources, http://www.juliantrubin.com/schooldirectory/fairresources.html, 2007.

[30] Schneider, R. M., J. Krajcik, R. W. Marx, E. Soloway. Performance of students in project-based science classrooms on a national measure of science achievement. Journal of Research in Science Teaching, 39(5), pp. 410-422, 2002.

[31] Science, Technology, Engineering, and Mathematics Education (STEMEd) Caucus Steering Committee. K-12 STEM ED Report Card: How Arizona Ranks. http://www.stemedcaucus.org, 2006.

[32] Soloway, E., Bonar, J., Ehrlich, K. Cognitive Strategies and Looping Constructs: An Empirical Study. Communications of the ACM, vol. 26, issue 11, pp. 853-860, 1983.

[33] Teitelbaum, M. The U.S. Science and Engineering Workforce: An Unconventional Portrait. National Academies' Government University Industry Research Roundtable (GUIRR) Summit, 2002.

[34] TUSD Curriculum Resource Guide, http://instech.tusd.k12.az.us/standards/index.asp, 2006.

[35] Wallice, K. America's Brain Drain Crisis. Reader's Digest, December 2005.

[36] Wallich, P. Mindstorms Not Just a Kid's Toy. IEEE Spectrum, Vol. 38, No. 9, September 2001.

[37] Wyeth, P., H. Purchase. Tangible Programming Elements for Young Children. Extended Abstract, Conference on Human Factors in Computing Systems, 2002.