Structured education begins with the K-12 curriculum. When considering the dearth of Computer Science (CS) curriculum included in K-12 classes in the U.S. [16], it is not difficult to conclude that outreach efforts in the K-12 arena must be enhanced. In recent years, several successful efforts to develop K-12 outreach programs have been reported [5,8,10,14,17]. According to Thies et al. [15], these and other efforts can be classified into one of three categories. The first includes those designed to refute the classic, all male, “nerd” image of a computer scientist. The second includes programs designed for more serious students that have an inherent interest in CS. Finally, the third includes programs that provide interactive activities centered around teaching computing, with or without the use of computers. Our program falls into Thies’ third category. The existing programs provide a start, but there is significant work still needed to meet the anticipated future demand for CS related jobs.

Networking concepts are pervasive in our daily lives. We are born into a network of family, and eventually develop networks of friends and co-workers. The postal service operates a network that connects us to people across the world. And of course, networks play an important role in today's world of technology. Like the postal service, the Internet connects people throughout the world, providing a fast and efficient gateway for data transmission, as well as face-to-face communication.

Observing the importance of networking concepts in computing, we have developed a curriculum module designed to introduce middle and high school students to the fundamentals of networks, protocols, and algorithms. The approach combines lecture, physical demonstrations, and hands-on activities. The program was designed to engage all students, regardless of their learning styles. The hands-on activities rely on a small embedded hardware platform – a “serious toy”. As an example of one activity, students are given a set of these devices, instructed on how they are connected, and tasked with forming a small network. This activity provides the instructor with the opportunity to introduce basic networking concepts, and gives students the opportunity to explore and ask questions about the hardware devices.

We piloted a series of such activities with six middle school classes – two each in the sixth, seventh, and eighth grades. Our goal was to introduce the basics of networks, protocols, and algorithms in an engaging and interesting manner, and to have a positive impact on students' perception of CS as a discipline. In this paper, we present the curriculum module, the design of the “serious toy”, and the evaluation results from six pilots.

ABSTRACT
Networking concepts have been in use for centuries. The human body is a network of organs that must coordinate to survive. The postal service is an example of a network that connects individuals world-wide. It is only natural that networks play an important role in computing — from networks of sensors collecting and recording data, to social networks, to the most complex network of all, the Internet. Observing the importance of networking concepts in computing, we have developed the second in a series of “serious toys” to use in the K-12 curriculum. In this case, the toy is an embedded hardware device designed to enhance a lecture titled “Learning Networks, Protocols, and Algorithms”, by engaging visual and kinesthetic learners. In this paper, we describe our curriculum module and its use in an outreach program involving six middle school classes. We conclude with a summary of evaluation results that show the program produced positive results in terms of content understanding and attitudes toward Computer Science.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education—CS education, Curriculum

General Terms
Experimentation, Human Factors

Keywords
Networks, protocols, algorithms, computer science outreach, middle school curriculum, experimental evaluation

1. INTRODUCTION
The amount of available knowledge in the science and engineering fields doubles every ten years [12]. This growth drives job growth in science, technology, engineering, and math (STEM) related areas, increasing the demand for STEM graduates. As beneficiaries of this trend, STEM educators have a responsibility to offer the next generation of students the knowledge and skills needed to succeed.
2. RELATED WORK

Outreach is essential for computer science to keep pace with the growing job market. There are several successful CS outreach programs, many of which are based on popular programs such as CS Unplugged [4], Computing Science Inside [3], Scratch [9], and Alice [2].

CS Unplugged [4] offers several activity modules that demonstrate the need for developing step-by-step instructions for computers to follow. Consider Marching Orders. In this activity, a student is given an image and instructed to give the rest of the class step-by-step instructions on how to draw the image. Similarly, Harold the Robot involves a volunteer playing the part of a robot, while the class is tasked with giving step-by-step instructions on how to build a structure from blocks. These activities demonstrate the difficulty of creating an algorithm for a computer to follow. Muddy City—Minimal Spanning Trees introduces familiar networks and some basic networking concepts. Like our program, each of these activities demonstrates either the importance of networks, or the necessity to create correct step-by-step instructions for the computer to follow. However, none of these activities are reinforced with hands-on activities, as in our module.

Computing Science Inside [3] also offers modules designed to teach basic networking and algorithm concepts. Tablets of Stone focuses on introducing students to networks and protocols by discussing examples of networks familiar to students (e.g., Internet, postal service, cell phone). They also discuss the importance of each network having a common set of rules – protocols – governing communication. Rather than using an embedded device for the hands-on activity, this program introduces a communication scenario and requires students to develop a protocol for the scenario. The algorithm module compares an algorithm to a recipe. As an activity, students are given a set of ambiguous instructions and asked to carefully follow the instructions. As expected, the pictures drawn are typically all different. As an exercise, students are tasked with rewriting and testing the algorithm. Although many of the concepts covered in this module are similar to ours, our approach relies on hands-on activities with an embedded device, which is engaging to the kinesthetic learner, rather than paper and pencil exercises.

Johann Heinrich Pestalozzi, during the early 19th century, was an early pioneer of learning through “hands-on” activities. Inspired by Pestalozzi, Fredrich Froebel, known as the father of kindergarten, developed a set of “gifts” – balls, blocks, and sticks – for children to use for playing and learning [13]. Manipulatives continue to be useful in today’s classroom to engage kinesthetic learners, be it as simple as the gifts offered by Froebel, or advanced digital manipulatives.

Resnick et al. [13] discuss the use of four digital manipulatives used in K-12 outreach. The first is LEGO Programmable Bricks (PBricks). Students learn to write programs using Logo, a programming language for LEGO PBricks. The PBricks have controllable motors and lights and are able to acquire data through sensors and communicate via infrared. The second manipulative is Programmable Beads (PB), small embedded devices that have a microprocessor and LED that can be strung together to form a necklace. Using inductive coupling, the PBs communicate; depending on the program provided, they can create colorful light patterns. The third manipulative is called BitBall, a rubber ball the size of a baseball. It is embedded with a LEGO PBrick, an accelerometer, and colored LEDs. Using infrared communication and a modified version of Logo, students can program the ball to turn on its LEDs based on readings from the accelerometer. The last manipulative is Thinking Tags. These devices were originally developed as conference name tags that contained information about the wearer and transmitted this information to other Thinking Tags. Later, these Thinking Tags were incorporated in schools as part of “participatory simulations”. As an example, students wear the Thinking Tags to simulate the spread of a virus. In the scenarios described, students were given the devices, shown how to use or program them, and left to explore and create. Our goal is to teach specific networking concepts, while at the same time allowing students to play, explore, and ask questions.

3. TOY ARCHITECTURE

Inspired by the MoteStack stackable sensing platform [6], we developed the “serious toy”, a hardware platform that consists of three printed circuit board (PCB) layers, to engage students in hands-on networking activities. Each PCB layer is connected using two sets of Hirose connectors [7]. The assembled toy is shown in Figure 1a; Figure 1b shows the individual layers of the toy.

Power (bottom) Layer.

As shown in the lower part of Figure 1b, the bottom layer of the device is designed to enable device interconnection, exposing power, ground, and communication pins using male and female headers placed on each side of the board. To prevent accidental reverse polarization by the students, the ground pin is connected to a male header, while the power pin is connected to a female header. To make the device extendable for future toys, this layer also has two sets of headers that expose the unused microcontroller pins, as well as two tactile switches connected to input pins on the microcontroller.

Control (middle) Layer.

As shown in the top-left part of Figure 1b, the middle layer provides power regulation and computation support; it is the core of the device. It consists of a 5V regulator, an ATmega168 microcontroller [1], three LEDs – red, green, and yellow – as well as a header used for programming. (We omit supporting circuitry in our discussion.)

Sensor (top) Layer.

As shown in the top-right part of Figure 1b, the final layer provides three sensors used during hands-on demonstrations.
Each sensor is connected to an Analog-to-Digital Converter (ADC) pin on the microcontroller. One is a simple galvanic skin response (GSR) sensor that consist of two pennies soldered to two wires. One wire is connected to the 5V power supply, while the second is connected to an ADC pin, grounded through a 10K resistor. A .1uF capacitor provides basic filtering. The second sensor is a broad spectrum photosensor, and the third is an electret condenser microphone. This layer also provides three LEDs – red, green, and yellow.

4. APPROACH

The curriculum module is designed to span two sessions of forty minutes each. The first session consists of a lecture and a series of questions designed to engage students. The second consists of demonstrations and hands-on activities to reinforce the lecture concepts.

4.1 Lecture

We begin the lecture by reviewing the definition of a network and motivating what makes a network a network. This is done by asking students to identify familiar networks. Students' responses range from radio networks to a pack of dogs. We conclude the discussion by defining a network as a group of entities connected and capable of communicating with each other. We next discuss the main components of two well understood networks: the postal service network and traditional computer networks. We talk about the importance of assigning unique addresses in a standard format to each house and each computer for reliable communication. We also point out that for some destinations, more than one path may be available. Students are invited to discuss the importance of choosing a standard address format and the factors that should be considered in choosing the best path. We then steer the discussion to the example of a scenario where the standard protocol is not followed. We show that this can result in chaos and confusion. Despite being connected to each other, network components will not be able to communicate. We conclude by reiterating that it is essential for all entities in a network to abide by the same set of rules or protocols.

Next we discuss the concept of an algorithm. First, we define an algorithm as a plan of action, then invite students to explain common tasks they perform. We ask what will happen if they do not follow the algorithms correctly. Using their responses, we explain that to complete a task, a correct algorithm must be followed. We then discuss the relationship between networks, protocols, and algorithms, and why following standard protocols and algorithms is important within a network.

Our last discussion is a brief introduction to the concepts of centralized and distributed computing. We discuss the responsibility distribution in each case, and how failure of one or more entities will affect the operation of the network. We discuss different types of networks and ask students to classify them as centralized or distributed. We conclude the lecture with examples of real life networks and introduce associated protocols.

4.2 Activities

The second session of the module begins with a demonstration of the sensors on the embedded toy. (The architecture of the toy is described in Section 3.) After the sensor demonstration, students are divided into two groups and rotated through two additional activities. One activity introduces students to a bus-based hardware platform that allows sensing devices to be added to and removed from a network without interrupting communication among the devices. In the second activity, students use the toy to create a network of devices that can communicate with each other.

4.2.1 Sensor Demonstration

The sensor demonstration begins with a description of the sensors and their capabilities. A desktop Java program provides a graphical view of the real-time readings from the sensors.

The GSR sensor was designed to allow students to actively participate in the demonstration by volunteering to come forward and place their fingers on the contact points described in Section 3. We explain to the students that one contact applies voltage to their finger and the other contact measures the voltage on the other finger. We explain that an emotional stimulus, such as fright, pain, nervousness, happiness, or anger, can cause the the skin to produce a tiny amount of sweat that will change the amount of natural resistance in the skin, reflected in a change in the graph being displayed. When the student volunteer has placed her two fingers on the contacts, they are instructed to stay calm for a moment to allow the graphical readings to level out. Next, she is instructed to pinch her ear, causing a change in the graph due to the physiological change caused by pain. We then allow the class to ask the volunteer (reasonable) questions, or to make comments that might cause an emotional response from the volunteer.

This demonstration exposes students to the sensors on the network toy, as well as graphs, and the concept of measuring GSR. After a brief period of question and answer about the demonstration, students are placed in their appropriate groups and directed to the next activity.

4.2.2 Serial Bus Network

The serial bus network activity begins with a demonstration of a microcontroller network composed of our toys, communicating through a serial bus. One of the devices is connected to a desktop computer. This device is responsible for identifying all of the devices in the network and collecting the data they send. The desktop computer uses a Java program to display the data.

We explain to the students that these devices are communicating through a serial bus, exposing them to the concepts of serial and parallel communication. We also discuss the associated communication protocol. We explain further that the serial bus is like a one lane road allowing traffic in both directions. We then ask, how can more than one car use such a road? Using students' responses, we discuss the concept of a multi-drop bus and contention issues for bus access in a serial bus network.

Next, we point out that each device has a unique ID; devices appear and disappear from the computer screen as they are added or removed from the network. Students are shown readings on the computer screen characterizing the noise level in the room. We mention that this is a simple example of a sensor network. Next, two student volunteers are selected to speak loudly or softly near a given sensor. All of the other students then observe the changing noise level on the computer screen. This activity demonstrates that devices are communicating with each other.
We next remove all of the devices from the network, except for the one communicating with the computer. Then add one device at a time, each time asking students to identify the new ID on the computer screen and to remember it. Then we play a game, removing one device at a time, asking which device will disappear from the screen. This exercise builds the foundation for explaining how devices connected to the computer learn about the arrival and departure of other devices.

After a brief question and answer period, students are guided to the next activity.

4.2.3 Creating a Network

The network activity begins with a discussion on the process of creating the network hardware. We specifically discuss how the network toy started as an idea, was prototyped using a breadboard, transferred to a PCB, programmed, and tested. A quick demonstration is provided using a set of breadboarded devices connected and pre-programmed, as shown in Figure 2a. During the demonstration, basic definitions are revisited.

Next, each group is given a set of network devices pre-programmed with a unique peer-to-peer protocol. The differing protocols allow for a discussion on the communication problems created if one or more of the devices are interchanged between networks with different protocols. Students in each group are instructed on how to connect the devices using jumper wires. Students are then tasked with connecting the devices to form a network. Once they are satisfied that all devices are connected properly, the network is powered up, as shown in Figure 2b. Each device is programmed to turn on or blink its LEDs, depending on the state of the device. Using the LED pattern, an explanation of the algorithm running on the network is given. Students are given the chance to swap individual devices between networks, demonstrating problems that arise with communication due to differing protocols. Throughout this activity, students are posed with questions designed to reinforce the concepts covered during the activity.

The activities provided in our module appeal to students of all learning styles. Students who are kinesthetic learners are able to create a network by connecting the devices with jumper wires. They learn about GSR through a sensor created using two pennies. Visual learners are able to visualize the change in sensor readings through a graphical user interface and are able to visualize the network communication through the pattern of blinking LEDs. Finally, students who prefer auditory learning are able to learn through the discussions, as well as the numerous question and answer segments throughout the activities.

5. PILOT

The pilot group consisted of six classes at a local middle school – the largest in the state, with approximately 1,350 students. During the 2011 school year, the middle school achieved an overall rank of “Good”, indicating that its performance exceeds the standards for the state’s “2020 Performance Vision”. The vision states that by “2020, all students will graduate with the knowledge and skills necessary to compete successfully in the global economy, participate in a democratic society, and contribute positively as members of families and communities.” [11]

The pilot group consisted of two classes each from the sixth, seventh, and eighth grades. For the remainder of this paper, the individual classes are referred to by their grade and class section. As an example, 6A refers to sixth grade, section A. As shown in Table 1, these classes have a total of 160 students enrolled, with 136 participating in the first day of the program, and 118 completing both days of the program. Note that there were additional students who participated on the second day; however, they were only included in the evaluation results if they completed the evaluation for both days of the module. Table 1 also shows the number of students in each class identified, by the school, as needing special learning considerations. The evaluation process was anonymous; these students received no additional help completing their activities or evaluation instruments.

6. EVALUATION

The module was evaluated using two evaluation instruments – pre- and post-surveys and content quizzes.

6.1 Survey

The survey consisted of 13 Likert-style statements, shown in Listing 1. For each statement, students were asked to rate their level of agreement by choosing from strongly disagree, moderately disagree, disagree, agree, moderately agree, and strongly agree. The statements were weighted from 1-6, with 1 being strongly disagree, and 6 being strongly agree. The statements were designed to evaluate the students’ level of interest in computer science, their perceived level of content understanding, and their perception of the outreach program. With the exception of statements 10-13, a statistical analysis was completed to determine if there was a significant change in the data1. A two sample F-test was performed to determine if the variance was equal. Depending on the variance determination, the appropriate t-test was

1Students were unable to rate these statements when complete pre-survey; therefore only the post-survey data was considered.
performed to determine if the changes between the pre- and post data sets were significant (p-level was 5%).

Figures 3a, 3b, 4a, 4b, and 4c summarize the average scores for the pre- and post-survey responses for each group, for statements 1, 2, 3, 7, and 9, respectively. (Similar results are omitted due to space constraints.) For each of these graphs, the X-axis represents the group, and the Y-axis represents the average statement score. Figure 3c summarizes the average score for the pre- and post-survey results for statements 10-13, for each of the six participating groups. In this graph, the X-axis represents the statement, and the Y-axis represents the average score.

1. Computer science seems like it would be fun.
2. I might be interested in majoring in computer science in college.
3. I think I know the definition of a network.
4. I think I understand the concept of a network.
5. I think I understand the value of networks in computer science.
6. I think I understand the relationship between networks and the Internet.
7. I think I know the definition of protocol.
8. I think I understand the relationship between networks and protocols.
9. I think I understand the definition of an algorithm.
10. I think the teacher did an appropriate job explaining the material.
11. I like the format of this outreach program.
12. I would like to attend more outreach programs related to computer science.
13. I liked learning about networks, protocols, and algorithms.

Listing 1: Survey Statements

**Interest in Computer Science.** Statements 1 and 2 were designed to measure student attitudes toward computer science, both as a discipline and a career choice. Figures 3a and 3b show that the module did have a positive impact on student attitudes for the majority of the groups. The average score for both statements decreased, though not significantly, for group 8B, and statement 2 showed no change for group 7B. We believe group 8B’s scores could have been influenced by the timing of the class period in which this group participated. (This group participated during the last class of the day, and the post-evaluation documents were completed the last full day of classes before the start of the winter holiday.) For statement 2, groups 6A and 6B both experienced a significant increase; however, the changes for these statements were not significant among the remaining groups.

**Content Understanding.** Statements 3-9 were designed to evaluate students’ level of perceived content understanding. Figures 4a, 4b, and 4c are representative of the results obtained for all statements related to content understanding. With respect to statements 3, 7, and 9, all groups, except group 8B, statement 3, showed a statistically significant increase in perceived content understanding. Overall, 88% (37 out of 42) of the content understanding statements showed a statistically significant increase.

**Structure of Outreach.** Statements 10-13 were designed to measure whether students enjoyed the format of the outreach program. The survey data summarized in Figure 3c indicates that all six groups felt the instructor did an appropriate job explaining the material. Three of the six groups moderately agreed that they liked the format of the program, with the remaining groups, 7A, 8A, and 8B,
entertainment devices. They enable communication — by phone, by email, and by snail-mail. Most workplaces cannot operate without computer networks. Motivated by the importance of networks in our daily lives, and the crucial role the networks play in CS, we have developed a curriculum module designed to introduce K-12 students to basic concepts of networks, protocols, and algorithms. Our module includes lecture, demonstration, and hands-on activities supported by an embedded toy. The curriculum is designed to engage students of all learning styles — visual, auditory, and kinesthetic. We piloted the program with six groups of middle school students, two each from sixth, seventh, and eighth grades. The evaluation results are largely positive across all six groups.

The curriculum module and embedded platform are the second in a series of "serious toys" developed to support computer science education. The curriculum modules and toys have had a positive effect on student attitudes toward computer science as well as their understanding of basic computer science concepts.

Acknowledgments

This work was supported by the National Science Foundation through awards CNS-075846 and DUE-1022941. Yvon Feaster is an NSF Graduate Research Fellow, DGE-0751278. We would also like to thank Dr. Brian Richard, our instructional partner from the participating middle school, for his help in making this program possible.

8. REFERENCES


6.2 Content Quiz

The content quiz consisted of 10 multiple choice and true-false questions designed to gauge students’ understanding of the material. Students were asked to complete the quiz at the beginning and end of the program. As shown in Figure 5, all of the groups showed an increase in quiz scores. With the exception of group 6A, all increases were statistically significant. However, we noticed group 6A scored highest in the pre-quiz data, possibly explaining why the increase in scores were not significant. Overall, evaluation results for this component suggest students understood the material presented.

7. CONCLUSION

Networks are pervasive in our daily lives. We use them in our homes to connect our phones, our computers, and our

![Figure 5: Content Quiz Results](image)

Listing 2: Quiz Questions

1. A group of entities connected together using a communication channel is called a _____.
   a) Forest b) Post Office c) Network d) Station
2. A(n) ____ is a set of planned actions to complete a given task.
   a) Algorithm b) Program c) Job d) Instruction
3. A set of rules for exchanging messages between two entities is called a(n) _____.
   a) Algorithm b) Program c) Network d) Protocol
4. If an unexpected event happens, a computer is smart and can determine by itself what action it needs to take. a) True b) False
5. Each computer in a network can use a different protocol and still be able to communicate with all other computers on the network. a) True b) False
6. You must have a computer to form a network. a) True b) False
7. It is important for each device/computer in a network to have a unique identity. a) True b) False
8. There are no problems associated with allowing one computer in a network to control the actions of all other computers. a) True b) False
9. A cable TV network is an example of a centrally managed network. a) True b) False
10. The Internet is an example of a distributed computer system. a) True b) False