Developing and Improving a Multi-Element First-Year Engineering Cornerstone Autonomous Robotics Design Project

Mr. David Joseph Frank, Ohio State University

David J. Frank is a 4th year Computer Engineering honors student at The Ohio State University and an Graduate Teaching Assistant for the Fundamentals of Engineering for Honors program. He will graduated with his B.S.E.C.E in May 2017, and his M.S.E.C.E in May 2018.

Ms. Kelly Lynn Kolotka, Ohio State University

Kelly L. Kolotka is a third year Chemical Engineering honors student at The Ohio State University with a minor in Biomedical Engineering. She is currently the Co-Lead Undergraduate Teaching Assistant for the Fundamentals of Engineering for Honors program. She will graduate with her B.S. in Chemical Engineering in May of 2018.

Mr. Andrew H. Phillips, Ohio State University

Andrew H. Phillips is a University Fellow and Graduate Teaching Assistant at The Ohio State University. He graduated summa cum laude from The Ohio State University in May 2016 with a B.S. degree in Electrical and Computer Engineering and with Honors Research Distinction. Currently, he is pursuing a Ph.D. in Electrical and Computer Engineering with research in integrated nonlinear optics, but he is also interested in Engineering Education. As a Graduate Teaching Assistant for the Fundamentals of Engineering for Honors program, he is heavily involved with teaching and developing laboratory content, leading the in-house robotics controller maintenance, and managing the robotics project development.

Mr. Michael Schulz, The Ohio State University

Michael H. Schulz is a teaching assistant with the Fundamentals of Engineering Honors program at The Ohio State University. He is currently the lead developer of the robot course software development team, of which he has been a member for three years. As a Computer Science and Engineering (CSE) student, he will graduate in May, 2017 with his B.S.C.S.E and a minor in Music, Media, and Enterprise.

Ms. Clare Rigney, Ohio State University, Engineering Education Department

Clare has been working as a teaching assistant for Fundamentals of Engineering for Honors (FEH) program at Ohio State since fall of 2015. She began as a student in the program in fall of 2014, studying Biomedical Engineering.

Mr. Allen Benjamin Drown, Ohio State University

Allen is a third year Industrial and Systems Engineering Undergraduate student at The Ohio State University who is an Undergraduate Teaching Assistant for the Fundamentals of Engineering for Honors (FEH) Program. His interests include Engineering Education, Lean Logistics, and Humanitarian Engineering. He will graduate with his B.S.I.S.E in May, 2018.

Robert G. Stricko III, Ohio State University

Robert "Bob" Stricko is graduating with a B.S. in biomedical engineering from Ohio State in May 2017. He has worked as a undergraduate teaching assistant for the FEH program since Autumn 2014. Leading up to the 2017 robot project, Bob worked as a co-lead for the construction team, where he led in the designing and building of the robot course. Following graduation, Bob will continue to pursue his goals of working in the medical device industry as he travels to California to work on the da Vinci Surgical System for Intuitive Surgical.

Dr. Kathleen A. Harper, Ohio State University

Kathleen A. Harper is a senior lecturer in the Department of Engineering Education at The Ohio State University. She received her M. S. in physics and B. S. in electrical engineering and applied physics from Case Western Reserve University, and her Ph. D. in physics from The Ohio State University. She has been on the staff of Ohio State's University Center for the Advancement of Teaching, in addition to teaching in both the physics department and department of engineering education. Her research interests address a broad spectrum of educational topics, but her current foci are adapting problem-solving instructional techniques to first-year engineering and incorporating engineering elements into K-12 science courses.

Dr. Richard J. Freuler, Ohio State University

Richard J. (Rick) Freuler is a Professor of Practice and the Director for the Fundamentals of Engineering for Honors (FEH) Program in Ohio State's Department of Engineering Education in the College of Engineering. He teaches the two-semester FEH engineering course sequence and is active in engineering education research. He is also affiliated with the Mechanical and Aerospace Engineering Department and conducts scale model investigations of gas turbine installations for jet engine test cells and for marine and industrial applications of gas turbines at the Aerospace Research Center at Ohio State. Dr. Freuler earned his Bachelor of Aeronautical and Astronautical Engineering (1974), his B.S. in Computer and Information Science (1974), his M.S. in Aeronautical Engineering (1974), and his Ph.D. in Aeronautical and Astronautical Engineering (1991) all from The Ohio State University.

Developing and Improving a Multi-Element First-Year Engineering Cornerstone Autonomous Robotics Design Project

Abstract

For the past 23 years, The Ohio State University's College of Engineering has offered to first-year engineering honors students the Fundamentals of Engineering for Honors (FEH) program. In that time, The Ohio State University has worked to incorporate and develop best practices for student development in engineering. With student immersion and growth in mind, the program has developed to include an engineering cornerstone project in which students work in teams to design, build, and program autonomous robots to complete tasks on an interactive robotics course. In support of the project, a variety of technologies were designed and polished as the program grew. Classroom methodologies were also evaluated and improved with time in response to student feedback and research on best practices.

This paper provides a historical review of practice for the program with an emphasis on the technologies and methodologies that have been most effective in the program as it has developed.

Introduction

For 23 years researchers in autonomous robot design have worked on the advancement of a freshman honors engineering design program. In the program, students, typically on teams of four, develop autonomous robots that complete a series of tasks within a two minute time period. The project is designed to be challenging and effective while also emulating a realistic engineering problem. As a result of the continuous pursuit of these goals, technological systems have been developed as well as classroom and teaching methodologies.

The advancements made in achieving the program's goals can be broken down into five sections: the classroom component, project requirements and resources, student resources, performance evaluations, and public outreach. The classroom component lays the groundwork the design project being feasible. Utilizing an inverted-classroom approach, the program incorporates rapid development of programming and CAD skills, in addition to the soft skills developed from working in a challenging and long-term group project.

Beyond this groundwork is the project itself, which is based on an interactive robotics course. The course is controlled using a modular network of partially custom hardware and monitored using an array of cameras which provide positioning information for robots. Students are loaned a custom controller designed by the program, and access to the program's store of building materials from which they can use their team budget to buy components from. Interfacing with the store using a catalog like website, students are able to order parts as well as submit designs for laser cutting or 3D printing. They are able to utilize nearly 40 hours a week of staffed lab and machine shop time. The students are evaluated in a series of weekly performance tests that lead up to a final public competition in which teams compete in single-elimination tournament.

Background

The university was part of an engineering education coalition in the 1990s committed to engaging engineering students with real engineering experience immediately in their first year of college. For example, this coalition has increased the number of first-year students participating in a design experience, as well as increasing how many participate in courses which give focus to engineering ethics and technical communication. The coalition effort has proven that engineering education methodologies such as cooperative learning and taking a student-centered approach have improved first-year engineering student engagement and retention into the second year[1].

Specifically, one major aim of the coalition is for first-year engineering students to participate in a full design project. The freshman honors engineering program at this university includes a 10 week-long robotics design project in the second semester which follows these guidelines of the coalition. This honors robotics design project is unique among other design projects by the large scope and infinite possibilities for students to tackle the problem and design a solution. For example, students have the opportunities to build their own robot structures and mechanisms from a variety of materials and using multiple shop tool, laser cutting and 3D printing. These technologies vastly increase the creativity and uniqueness of each student team and the robot. In comparison, another university within the same coalition has a robotics design project in the second term of the first year which utilizes specific and already made LEGO Mindstorms robotics parts[2]. This limits the amount of creativity for a student team in designing and building a robot, while the FEH robot program enables students to formulate solutions to the tasks in their own unique ways.

Even within The Ohio State University, this honors robotics project is on a much larger scale than, for example, the standard track's design project. This standard project is to build and program an alternative energy vehicle (AEV) to run on a monorail track. The AEV standard course project was developed to be a smaller version of this honors robotics project in that there is much less designing and programming involved.

Although this honors robotics design project is unique and large-scale now, it has certainly been the product of much advancement, especially in the last ten years[3]. For example, the laser cutting and 3D printing student capabilities were not implemented until a few years ago when those technologies became more commonplace. Years ago, students used the MIT Handy Board as the controller for the robot and programmed in Interactive C. This controller became difficult to maintain due to its age, and so, a new in-house controller was developed for use in the robotics project[4]. Many of the features and tools students have access to now, such as a positioning system on the course and access to example robots for testing, were not developed in full until a couple years ago. Additionally, the first-year engineering program used to be housed in a university academic center within the College of Engineering, but in November 2015, the center transitioned into the Department of Engineering Education focusing on engineering education, multidisciplinary capstone program, engineering technical communications, and the first-year engineering program. Therefore, with more engineering education researchers involved, even more pedagogical content has been added to the design project.

Pedagogical Practices

Many best practices of engineering education have been implemented into this design project. The mutual learning methodologies of cooperative-based learning and collaborative-based learning are utilized throughout the experience since students work with each other in teams and with other teaching assistants. This is enveloped by instructional team approach to the classroom environment, whereby each class of 36 students has an assigned instructional team of one instructor, one Graduate Teaching Assistant (GTA), and three Undergraduate Teaching Assistants (UTA). Thus, these mutual learning models are supported by peer teaching and mentoring from GTAs and UTAs, as well. Additionally, the course gives students experience with project-based learning and problem-based learning in different aspects of the course. The project overall is open-ended and gives students a large starting point while there are also smaller tasks and problems which teams must complete along the way.

All of these practices combine to help form a community of practice for the robotics project. This community is composed of other first-year teams who are starting to learn, teaching assistants who are experienced in the many aspects of the course, and instructors who provide structure to the project. In addition, teams are exposed to an environment full of tools and resources to choose to use and skills to develop. Thus, this honors design project certainly fits into the situative framework of learning[5]. Students start as peripheral participants in the community when they are given the specifications and structure for the project. As they ask questions and begin to explore the possibilities of the robotics design, they begin to integrate more into the community. Talking with other teams and getting advice and help from teaching assistants further integrates them. Finally, after learning new problem solving and design skills and having weeks of practice, the students become fully participant in the community and have less reliance on the teaching assistants and instructors for guidance. Hence, student learning and growth in this project are partly demonstrated by increased participation within the larger community of practice.

In-class material

The foundation of the robot design project is the classroom content and experiences that prepare students to take on the project. This begins in the first semester when students are introduced to basic coding via Matlab, C, and C++. Matlab is considered easier to learn than C or python. The syntax is more forgiving, and the compiler gives easily understood information when there are errors in the code. This aids in teaching students programming concepts while minimizing time spent on syntax specifics. Once the concepts of types, arrays, functions, loops, and switch cases are taught in Matlab, the students move on to learning them in C and C++. Lessons go on to include C/C++ related elements such as structs, classes, and object oriented programming[6].

In the second semester students rapidly develop their CAD and graphics skills before embarking on the design project. Students are introduced to drawings. They learn of different views as well as how to convert between them and how to draw in them. They expand on this with missing views, constraint-based modeling, section views, feature based modeling, and dimensioning. While doing this CAD work exploring these concepts is slowly introduced. Using Solidworks, students create part models based on engineering drawings as well as creating assemblies and learning how to properly document them. They then learn about distributed tolerancing, and start to review the engineering design process in anticipation of the design project.

Together the in-class material and design project material focus strongly on the design process. On average each student team in the design project visited each aspect of the design process at least once per week and many elements of design were visited three or more times. This was also shown to be significantly more than similar first-year engineering design projects at The Ohio State University[11].

TAs and Instructional Team

The instructional team is a primary way that mutual learning methodologies of cooperative-based learning and collaborative-based learning are implemented. In each classroom, an instructor, graduate teaching assistant (GTA), and three undergraduate teaching assistants (UTAs) are present each day to assist 36 students[7]. The program utilizes an inverted classroom approach in order to encourage staff and student interactions[6]. With the inverted classroom approach, the students complete reading, watch videos, and practice the material prior to the class, so that the majority of the time can be used to work on the project when in the classroom. Each class opens with a short lecture over any new material, and the rest of the time is given to the students to work within their teams. This allows them to ask questions of the teaching team as they arise since they are working hands on and testing during class time while the assigned staff is present.

The instructor of each class manages the responsibilities of each member of their teaching team, assists in problem solving for students, and leads the class through lecture materials. The graduate teaching assistant is the main point of contact for the robotics project for his/her class and is trained to assist the students in all aspects of the project throughout the semester. The undergraduate teaching assistants are previous students of the program and are able to provide advice, examples, and support based on previous experience. Generally, teaching teams are balanced by incorporating undergraduate teaching assistants of different strengths such as documentation, computer programming, CAD model and design, or construction.

The teaching assistants, at both levels, work to create the robotics course each year. A large percentage of TAs are involved to better their technical learning, and do so by developing the technologies used by the program[7]. The out-of-class hours provided to students to work on their robots is also staffed with teaching assistants in order to answer questions, manage the store, and provide guidance. Altogether, this structure provides the program with a large number of TAs who are intimately familiar with the details of the program and student experience, whose work is fueled primarily by enjoyment and interest in the program[7].

Team formation and team working skills

The team formation process is designed to create teams of equal skill with equal chances of success on the robot course. The current system was made with some aspects of McGourty's Team Developer in mind[12]. Its goal is to be more thorough and specific to The Ohio State University's robot design project. Students take a self-assessment survey to identify their strengths and weaknesses, and to pick six “preferred teammates,” at least one of which they are guaranteed to be on the same team. Next, the formation process is automated using software developed in house. This software creates many possible team formations, and selects the most balanced formation by considering factors such as GPA and the self-assessment survey. The self-assessment survey yields a rough idea of how strong students are in the following areas: technical writing, mechanical construction, electrical engineering, software development, problem solving, team dynamics, CAD, and general robotics experience. The team formation software uses this data at the end of the team formation process to determine which team formation yields the most balanced teams.

Due to the complex nature of team formation, given a class of 36 students with teams of four, there are $1.4*10^{29}$ ways to make teams. While computers have increased in processing power recently, this is still too many combinations to feasibly calculate. The software makes some assumptions to decrease running time, including separating the class into two groups of 20 and 16, and forming teams on each section. This alone brings the number of formations to $1.9*10^{19}$ combinations (Equation 1).

$$N2 = \prod_{i=1}^{5} \frac{4 * i}{4} * \prod_{i=1}^{4} \frac{4 * i}{4} \tag{1}$$

There are several more optimizations that reduce the number of possible acceptable formations by checking if a team doesn't have a strong programmer or builder, or if each team member does not have one of their preferred teammates. Many team formations are then made by this modified brute force approach. The best formation is selected by using the student's self-assessment responses to balance skills across teams. This is to prevent certain teams having an advantage over the others in experience or skill, ensuring a more level playing field.

Labs

In addition to the classroom material taught before and alongside the cornerstone design project there are also labs. The labs occur in both semesters of the first year program. The labs in the semester preceding the design project happen weekly and aim to explore different engineering majors in meaningful hands on experiences[8]. In these labs many of the skills required for the design project are first introduced to students. Basic wiring, soldering, and Boolean logic is taught during a lab in which students adapt toys to be used by children with special needs. Interfacing with microcontrollers is taught during a lab in which students use the program's proprietary controller to control a miniature LED stoplight. Lastly, students learn to control servos and dc motors as part of a lab where different motor types are explored.

In the second semester of the program when the design project occurs, students partake in three robot centric labs, which aim to explore the basic concepts of robot design and control. In the first lab students use CdS cells to detect the color of a light and adjust a servo motor in relation to that color. They also program premade robots (called Crayola Bots) to navigate a small boxed in area using only microswitches. A figure showing this can be seen below in Figure 1. In the second lab students learn to make robots follow lines using the Crayola Bots' array of three analog optosensor. They also use the built in shaft encoders on the Crayola Bots to perform precise navigation and turning. In the last lab, students explore guided navigation and data logging. Using a Robot Positioning System (RPS) which simulates GPS, students program Crayola Bots to navigate to a specific location on the robotics course only using x-y coordinate and heading information. The students also use the system for data logging of positional data.

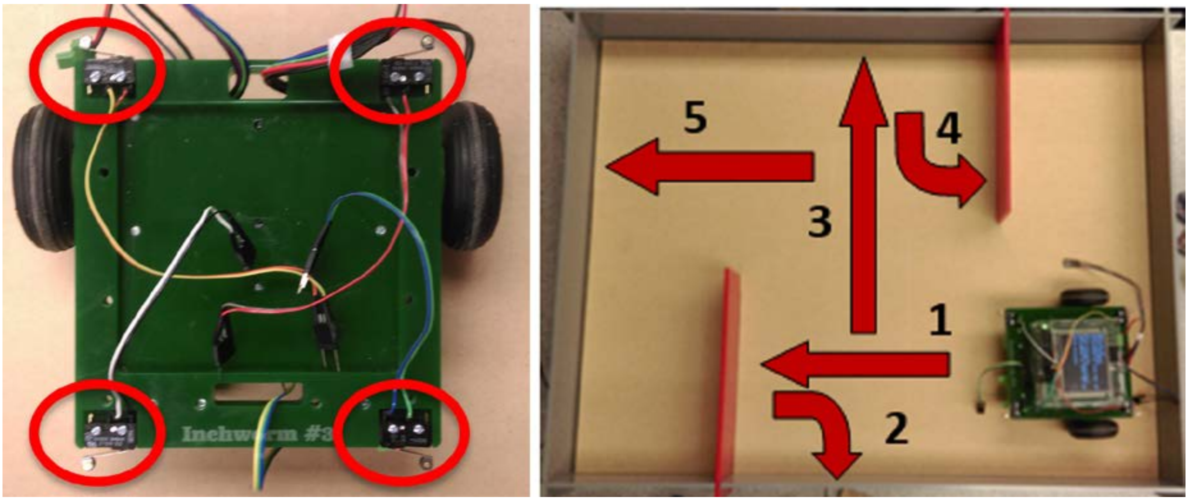

Figure 1: Crayola Bot with bump switches (left) and navigation box (right)

The labs provide a strong framework for the students to work from. The first semester teaches skills needed to build and program the robots, and the second semester applies these skills in focused lab exercises. The second semester labs are structured to build up essential skills in robotics as the design project progresses.

Crayola Bots

As was mentioned, the program makes use of a set of premade robots for students to use. This use not only happens as part of the robot labs, but also occurs during open lab hours. These robots, called Crayola Bots, provide a resource for teams to test code when their robot is under construction. They also provide a good demonstration of a variety of sensors available to students.

The Crayola Bots are a replacement to previously pre-assembled robots used by the program, and were designed and built by the program's TAs. Originally the robots used by the program for labs and student practice were hand made out of bent aluminum sheets and erector set pieces. While basic in their assembly and made out of robust materials, the robots proved difficult to maintain and replace. The entire design of the Crayola Bots was to address these issues while also providing updated sensors. The design can be seen in Figure 2. The chassis are made out of layered acrylic. The acrylic has cutouts that allow screws to be embedded in the layers allowing the layers to easily be locked together and to other pieces, as well as cutouts for sensors and wires. The design also utilizes 3D printed parts which interface with the Crayola Bots motors, analog optosensors, and CdS cells. On the top of the chassis there is also a mount in which a micro QR code can be placed for use with the Robot Positioning System. Given that custom parts of the Crayola Bots are either laser cut or 3D printed, they are exceptionally easy to maintain as they can be printed or cut without much technical knowledge.

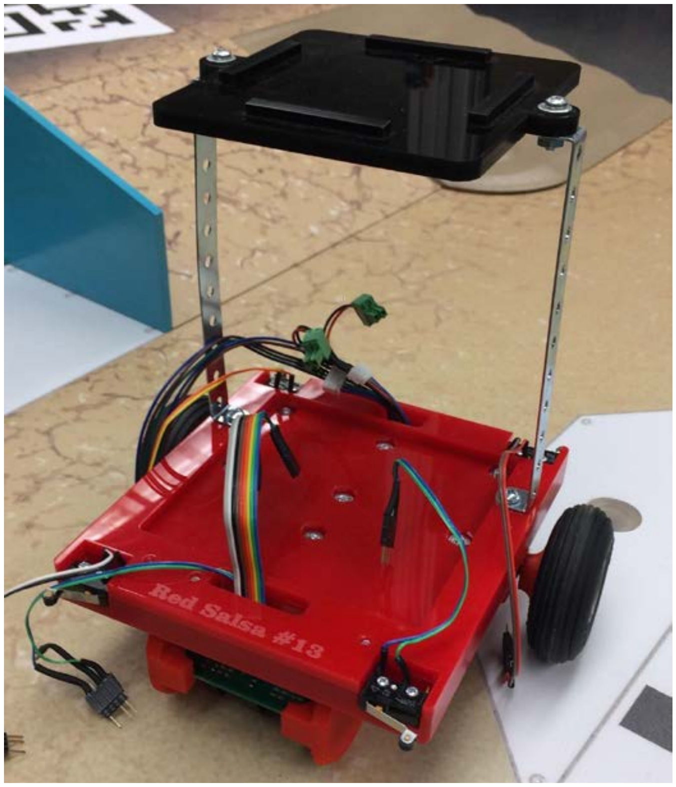

Figure 2: Full Crayola Bot

Project Requirements and Resources

At the core of the project is the robot course. The robot course is rebuilt every year to include new tasks and challenges that student robots must complete. Centered around a different theme every year, the course establishes a reasonable scenario that calls upon the development of autonomous robots to complete a series of tasks. Many course tasks are simple in nature, but each year at least one task is included that requires more complex control and interaction from a robot.

The next five sections will outline the resources provided to students that aid them in accomplishing the course tasks. The resources, which have been developed over the years, range from the hardware and software that control the course, allowing it to interact with students as well as allowing students to control it for refined testing sessions, to resources such as the Robot Positioning System, which provides a GPS like navigation system for student robots to tap into. The program also uses its own controller design and has its own store of robot parts.

Course Theme and Objectives

A theme is provided for the students to give real world context to the objectives, or tasks, faced within the course. Students must create a prototype of an autonomous robot that will perform tasks such as a robot in industry might have to perform. Autonomous robots can be used in a variety of settings, thus enabling different scenarios to be used. A new theme is provided every year in order to ensure that students have a new set of challenges. This forces students to use their creative problem solving skills as they cannot rely on the documented design of robots developed in previous years to complete the tasks of the current year. The 2016 theme was "Rocket Launch." A figure of the course and its objectives can be seen in Figure 3 below.

Figure 3: Rocket Launch themed robotics course from 2016

The course tasks are designed to range in difficulty so that 100% of students can be at least 75% successful. An example simple task from the Rocket Launch theme was reading a light embedded in the course to choose which of two buttons needed to be held for 5 seconds to allow the correct liquid to flow through a fuel pipeline. A more advanced task was wirelessly communicating with the Robot Positioning System to determine the direction that three switches needed to be flipped. In each of these cases and for many tasks in general, lines are painted on the course to allow robots to line follow to the tasks.

Courses are built using the same materials and design concepts that students use. Throughout the autumn semester, the teaching assistants design the entire course using Solidworks. When those designs are finalized, undergraduate teaching assistants build each task, laser cutting or 3D printing certain pieces[10]. Assembly and construction of parts is done using the machine shop that students have access to during the design project. Additionally many parts from the student robot store are used as well. This is helpful to the cooperative-based learning atmosphere as TAs become a knowledgeable resource to students on the intricacies of the material available to them.

Course Software

The software that runs the course has evolved significantly over the past ten years. Originally, it was rewritten each year, providing minimal functionality and no automatic scoring. Eventually, a framework was developed that could be built upon, but it was somewhat bloated and not reliable. Updates regarding the state of the course were often lost, and students were impacted by the lack of consistency. Finally, the current framework was developed with the goal of providing a lasting framework with a consistent and reliable update pattern[13]. The new framework was also developed to provide automatic scoring, which allows students to test their design in an environment similar to that of the final competition.

The course itself is comprised of a network of devices, including a Raspberry Pi, Arduino, and several types of IO boards for each of the four regions on a course. For each course, there is a main course computer with a user interface with which the students can interact. The software allows the students to choose the requirements for their robot to complete during testing, either deterministically or randomly. The course computer works as a server, and each Raspberry Pi works as a client. When the course is started, the clients connect to the server via a TCP connection, which is reliable and does not result in dropped network packets (and thus lost information about the robots' completion of tasks). The Raspberry Pis communicate with the course components, such as buttons and LEDs, via the I2C protocol, which operates at approximately 100kHz. These properties together result in an update cycle that lasts just a fraction of a second, meaning the state of the user interface and other course components reflects the actions of the robot immediately without delay.

Because of the modular design of the software, the students are able to focus on their own design goals without accounting for errors in the course itself. During testing and competition, the software provides real time updates of the score that the robot has achieved, meaning students can design focused and intentional tests of specific aspects of their design and be presented with objective, real time data on which to base their conclusions. Additionally, the course computer maintains a timer for each region, allowing students to see the time it takes their robot to complete each task, with millisecond resolution. During competition, the course timers are used to break ties in score.

Finally, the course software is built on a framework that easily allows for new types of tasks to be added. Software for new types of electrical components can be added in a short period of time. A simple XML document is used to add components to the course year-to-year. Changes to this document can be made quickly, and all of the necessary runtime information for a given component (such as I2C bus, resistor pull-up/pull-down configuration, pin number, etc.) is included. A major design goal for this system was configurability, along with reusability. All networking information is present in the XML document as well, so should any networking parameters change, those changes can be reflected in the XML document and no changes need to be made to the software itself. Therefore, this system facilitates the development of innovative and creative tasks in the future without any barriers.

Robot Positioning System

In addition to the core software and hardware that allows students operate and test on the courses, there is also the Robot Positioning System (RPS). The RPS is a system of cameras that track student robots using QR codes and then wirelessly transmits that data to the robots using RF transmitters[9]. The system directly aids in many of the programs goals. It provides a realistic and modern system that parallels GPS, it aids in adding variety to possible robot design by providing another means for navigation, it allows communication with robots beyond coordinate data, and it provides a user interface that can be used by students and the general public.

The system in its current state has existed for two years within the program. Originally tracking was done using hacked Wii remotes tracking pairs of infrared LEDs. However, this system was abandoned as its reliability and accuracy were considered too poor a parallel to GPS to be used in the classroom. The current system makes up for those faults. To provide accuracy the system utilizes a 1080p HD webcam mounted above each individual course section to track 13 by 13 block micro-QR codes. As part of its user interface it also allows calibration points to be set to help create a coordinate plane that is accurate within a quarter of an inch within the 18 square foot course. The system is also very quick and reliable. It is programed in LabVIEW and run using a Windows PC. The PC and software supports the running of four cameras and course sections at the same time, and it can do so while maintaining a refresh rate of at least 10 Hz. The system is also networked with the course computers and receives data on the status of all course elements. This allows the RPS to communicate to the robots, status information about current course tasks as well as the heading and position data that would expected from a GPS like system. It also allows the RPS to display a UI over live image feeds of each course. This UI provides a visual indicator of the status of each task and is used to quickly gauge a robot's completion of the course. In terms of reliability, the system ran for a three-month period during the 2016 design project period and only two software crashes were recorded.

Proteus Controller

Each team is given one Proteus controller for operating the robot. The Proteus is an in-house designed robotics controller based on ARM computer architecture[4]. Students program the Proteus for the robot to autonomously move, gather information, and make decisions to complete tasks. The Qt Creator open source integrated development environment is used to program the Proteus with the C/C++ programming language. Programming the Proteus is still based on the core C/C++ concepts learned by students in the first semester course, but Proteus-specific programming functions and objects keep the programming at a high level for easy interaction with the robot course environment. All of these unique functions and objects are documented on a program-run website for student access.

Hardware wise, the Proteus is a robust controller with multiple capabilities. It has 32 flex digital and analog input/output ports, eight servo motor ports, and four DC motor ports. These ports can be used to operate sensors, servo-controlled robot arms, and drivetrains. The large number of available ports encourages the use of many components and the need to plan how these components will interact with the robot and the Proteus. Also, the Proteus receives RPS data through an XBee wireless module, and it has a 320 by 240 pixel color LCD for text output and program debugging. This allows students to fully engage with the test, validate, and redesign cycle. These main features are highlighted in Figure 4 below.

In addition, more advanced features have been developed for the Proteus in the last two years. First, an on-board accelerometer has been implemented to allow students to determine the orientation of their robot. Second, the LCD has been implemented as a touch screen. Third, microSD card support has been added for logging data to text files. These advanced features open up new testing strategies and increase the amount of information students can get to help inform future design revisions.

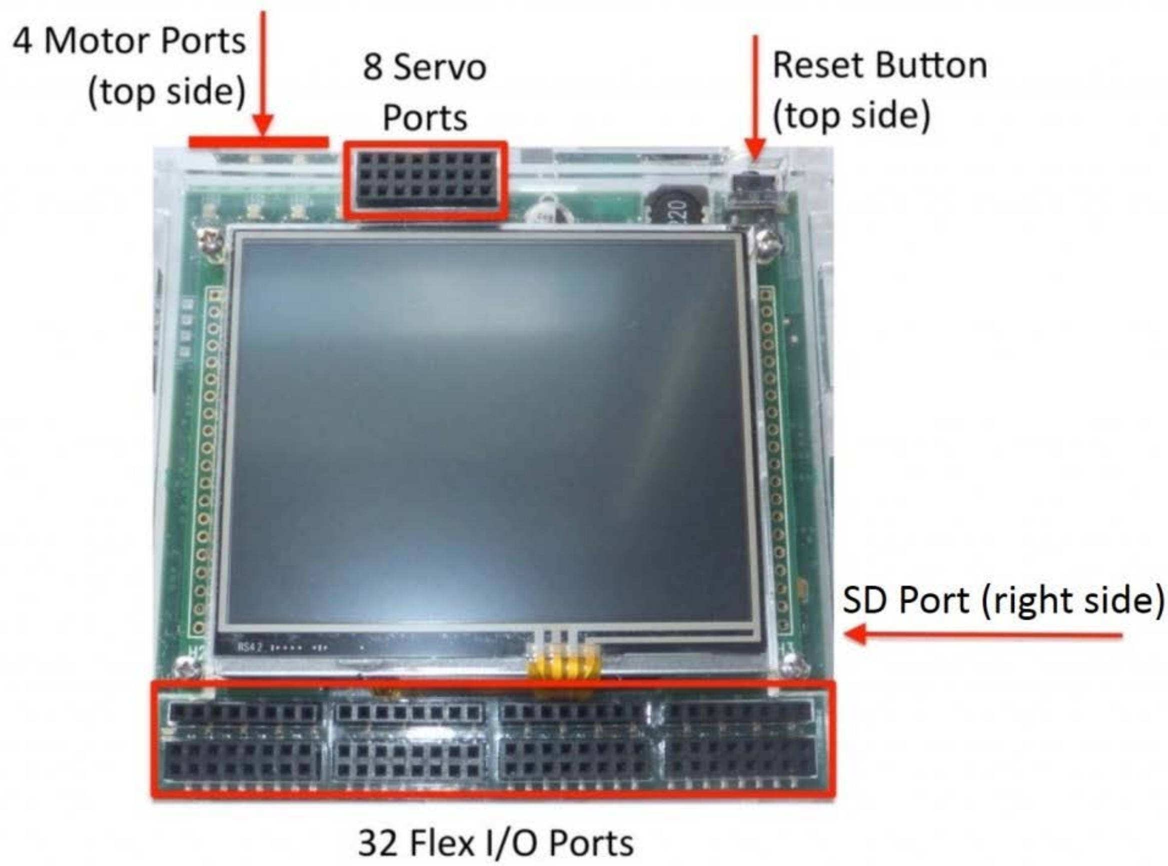

Figure 4: Proteus controller front side with labelled ports

The Store

The majority of the materials used by the students for their robots are obtained from the robot store. Each team is allocated $160 to complete their project that they can spend at the store. The store has everything needed support a variety of robot designs. It has a variety of chassis materials, motors, erector set pieces, electronics/sensors, wheels, fasteners, etc. If there is a part not stocked by the store, the students can have the store order the part or buy the part themselves and have the cost deducted from their budget. For objects included on robots that do not have a price associated with them (e.g., a rock for weight), the cost is estimated and subtracted from the team's budget. Periodic technical inspections are made to ensure all parts on a student robot are in their budget.

The store can also be used to check out tools for student use. These are handheld tools such as soldering irons, wrenches, heat guns, and hot glue guns. Tools like these are ones that do not require extensive training to use and save students a trip to the machine shop. Students are allowed to keep the loaned tools for a period of two hours. If the loaned items are not checked back into the store in that time period, each student on the team will be locked out of the store website until the item is returned. Students can also check out non-electrical components to evaluate before buying.

The store simulates many of the elements of developing a prototype for a real business. Students are able to browse store parts from different vendors and must make decisions with respect to their budget on which parts make the most economical sense. Teams also must place formal orders and wait for said orders to be ready. There is also the ability for a team to request a refund which may or may not be approved.

Student Resources

The resources discussed so far define and support the scope of the design project. The following four sections will cover resources available to students that can be used to their discretion. These resources largely define how students interact with the program.

Store website

The store website is used by students to browse different store parts, and make orders. It has been developed to do more than just list available parts. The store website allows students to access CAD models of available parts. This gives the students access to a part's precise dimensions, allowing them to easily interface it with laser cut or 3D printed designs. It is similar to many online catalogs which increasingly feature CAD models of all listed parts. The website also gives wiring diagrams for each sensor type that it sells. While less detailed than the common data sheets, the wiring diagrams still provide basic information about a sensors correct implementation.

All purchases are ordered through the store website. When an order is received, the undergraduate teaching assistants fill the order. When an order is filled, the teaching assistants mark it on the website, which sends an email to the team giving them a notice that their parts are ready to be picked up.

Student facing materials

Although there are many standard parts available to students in the store, there are instances where a student requires a custom part for their design. Students can either fabricate parts on their own in the machine shop or they can utilize 3D printing and laser cutting resources. Laser cutting and 3D printing techniques are very valuable to students because of their highly precise methods of fabrication. Furthermore, these options are encouraged because they allow the students to further develop their skills in Computer Aided Design[10].

Laser cutting is often used as a method for building a chassis. The students are provided with the options of cutting a two-dimensional vector out of a sheet of either ¼" acrylic, ⅛" acrylic, or ¼" medium density fiberboard (MDF). The benefits and drawbacks of laser cutting these materials are provided to the students by the instructional team through a set of slides. These slides also outline effective methods for creating the files necessary for a laser cutting order. Students typically find that their time was well invested in developing a laser cut chassis because it produces stable and durable structures that provide a foundation for a consistently performing robot.

Students often use 3D printing to make custom tailored tools for specific tasks as it is easier to create irregular shapes when compared to other fabrication techniques. Students are allowed to 3D print up to 100 grams of PLA filament in creating parts for their robots, which allows a moderate use of 3D printing, but not enough to create an entire robot chassis. Importance is placed on design variability within a team's design as well as variability from team to team.

Machine shop

Students have access to a variety of machining tools and power tools in the machine shop. The shop provides fabrication tools to teams that do not use laser cutting or 3D printing. It is also resourceful to any assembly or modification work that a team may need to do. Students have access to drill presses, belt sanders, band saws, scroll saws, a shear, a break, and various hand tools. At the beginning of the project, students are taught how to safely use these tools to make their robots. Teaching assistants also supervise the machine shop space and offer assistance to students while they try to figure out how to make parts or assemble their components.

Open Lab

Student spend a total of 6 hours and 15 minutes in the class each week. Even though most of the that time is set aside for students to design, construct, code, and test their robots, most teams would not be able to complete all of the tasks with the time given in class. The program sets aside open lab times where students from any class section not only have access to the robot course, but also to have access to the teaching assistants. Each period of open lab is staffed by one undergraduate teaching assistant stationed at the store, and two or three other undergraduate teaching assistants working in "open lab" designated classrooms. A graduate teaching assistant is also present during these times and in charge of student safety. In a typical week there are nearly 40 hours of these open lab times. This amount is far in excess of what additional time a team may need outside the classroom. It exists solely to provide ample flexibility for students' schedules.

Performance Evaluations

The design project provides continuous opportunities for students to showcase and document their robot and its successes. Thorough technical documentation is required throughout the project. Teams have robot assignments which are specifically scaffolded to be useful to them in their robot design. They also have a final technical presentation and a report over the entirety of the project. There are also a variety of formal tests of their robots throughout the semester as well as a final competition. The next five sections outline these different evaluations of performance in detail.

Performance Tests

Throughout the robot project, teams are evaluated on their robot's performance regularly. Each week, the teams are asked to complete one assigned task on the course during a performance test. The main goal of the performance tests is to ensure continual progress throughout the project especially on some of the more challenging tasks. As students approach the end of the project, they will have completed four different performance tests that cover all course objectives. This ensures that each team has demonstrated a baseline of performance in regards to each tasks.

Performance tests focus on a single task each week. Student can choose to develop new code each week or build on the code from the previous week. Teams who make their codes more modular tend to have an easier time in programming the code for each week.

The instructional teams use performance tests as a graded component as well. The performance tests have points assigned to each component of the test, and this is the weekly score for the team. The automated scoring from the course is used to determine the correct number of points to allocate for each run, and each team is allowed five official attempts per performance test. The team must be scored by a member of the instructional team on an official performance test run. The tests may only be completed during a class period in the presence of a GTA or instructor.

Additionally, there is a stretch bonus and early completion bonus. Both of these opportunities are to encourage teams to move beyond the weekly expectations and complete the assigned tasks in a timely fashion. The instructional team is able to evaluate how each team is doing and devote specific attention to teams that may be struggling. This ensures that the majority of teams are seeing success when they reach individual competition.

Individual Competition

Nearing the end of the project, all of the teams compete in individual competition. Instructional teams for each class facilitate a competition during a regular class meeting. This simulates the expectations of the final, public competition in a smaller setting a week prior to the final competition. Individual competition is also the first time the team is scored on an attempt of the entire course.

The individual competition has three rounds for teams to score points. For the first round, the course is chosen randomly. The second round is instructor’s choice, and the final round is the team’s choice. Even though each course is identical in design, there are slight differences that can affect autonomous robots. This often results in a tendency for teams to design around the intricacies of a specific course. The individual competition partially randomizes courses to prepare students for the randomized format of final competition. It also stresses on of the program’s core design practices, which is to find ways to account for variations.

Points are allocated to two different categories during the individual and final competitions. Each task is separated into primary and secondary task scoring. The primary task points count toward the run score for that robot and the grades of the competitors. The secondary points count only toward the run score for the purposes of the competition. For example, one of the tasks on the 2016 course was picking up and depositing cargo supplies. The primary points were associated with touching and controlling the supplies, meaning the sensors of the original cargo bin are disengaged. The secondary points were awarded if the cargo was deposited to the proper drop zone. In order to have a perfect run, all of the primary and secondary points must be awarded. The scoring is completely electronic and non-disputable for individual competition. After all of the individual competitions are complete, the winning team from each class is awarded bonus points, and the average for each team is used to create the bracket for final competition with time being the tiebreaker.

Final Competition

The design project culminates on a Saturday late in the semester where all teams compete during final competition. The courses are transferred from the classroom to a larger arena on campus by the teaching assistants the night prior to the competition, and the event is open to the public. Corporate sponsors of the program attend as well to judge the competition and distribute scholarships and awards.

Despite the added level of excitement, the final competition is very similar in expectations to the individual competition. The students are still limited in time between runs and time allowed on the course, and the scoring is identical. From the students' perspective, the greatest change is the number of runs on the course. The competition opens with three round robins. This is a chance for teams to adjust any sensor differences due to the new location and to have practice before the head to head competition. The winner of each round robin is announced after each round, and this process is used to award the top achieving teams in consistency.

The robotics tournament follows a single elimination bracket that is seeded with the individual competition scores and times. The top team (one of the four) on each course moves on to the next round. Automated scoring is still used with the same previously mentioned system. Members of the instructional team regulate the rounds and reinforce the scoring.

Once the competition has come to a close, the corporate sponsors award teams for most innovative, and best engineered. Awards are also given out to the highest scoring teams from the round robin runs, as well as the top four teams from bracket play. Following final competition, students finalize their documentation, and awards are given out for best documentation as well as for teams exhibiting "gracious professionalism".

Technical Writing and Documentation

Throughout the entire semester, each team is responsible for completing documentation of their robot project. The skillset of presenting one's work is one of the best practices gained by young engineering students. Teams are required to document the entire project in great detail, and they must reach several checkpoints along the way.

The documentation assignments include brainstorming (both as a team and individually), a team working agreement, a design schedule, mockup, drivetrain calculations, pseudocode, report outline, electrical documentation, technical drawing sets (made in SolidWorks), a final report, and an electronic notebook. Teams are also required to have a testing log for all testing done on the courses, meeting minutes, and meeting agendas. The online notebooks generally include all major documents, competition performance, videos, and photos of robot development. The final report, among other assignments, is worth a large portion of the grade for the project and allows students to utilize the technical writing skills from their first semester in the program.

Technical presentation

The final expectation of the team for the cornerstone design project is to provide a technical presentation. Each team is responsible for preparing a ten minute presentation over their actions for the semester. The target is for students to present their decision making strategies, and how those decisions impacted overall performance. The presentation is a great opportunity to discuss future improvements or changes the team wishes they could have made, all the while presenting their successes for the semester.

From an educational standpoint, the technical presentation simulates a task that is common to real world engineering careers. Finding success in a technical field is dependent on the engineer clearly presenting his/her work to others. The technical presentation in a first year design project allows this skill to be targeted and practiced early in engineering curriculum so that students are better prepared for real life scenarios.

Public outreach

A large element of the cornerstone design project is showcasing student designs for the public. This has historically been accomplished for the program via the aforementioned final competition which is free and open to the general public. This event, like many elements of the program has been refined and improved over the years. Most of the development has been towards improving how the public interfaces with the program. Beyond pamphlets that provide audience members with background information about the program, the structure of the event includes bracketed play, the progress of which is displayed on large projectors. Further aiding the audience in following the events is the RPS display which shows live video feeds of each competing team and overlays information about a team’s score.

Competition brackets

The competition brackets show the status of the competition overall, which while varying from year to year, usually encompasses between 70 and 84 teams of students. The software running the brackets allows the brackets to be zoomed in on, to follow the competition’s progression. It also has a secondary display mode that shows live updating scores of competing teams when a two minute competitive run is underway. This is done with software written in C#. The software communicates with the RPS which provides scoring information to the bracket software via its communications with the software that operates and controls each course. Using this, the software is able to provide a live progress bar showing the score of each team as it competes.

RPS Display

The RPS displays provide additional information to the competition brackets. They provide a top down live video feed of the course that each team is competing on. On top of this feed a tracking box is placed around each robot that follows it as it moves around the course and helps audience members quickly gauge where each competing robot is. At the bottom of this video feed is a ribbon of overlaid information, an example of which can be seen below in Figure 5. As can be seen in the figure, the team designation is displayed as well as the time spent on the current run. In the middle of the ribbon a graphical representation of each task is displayed along with a corresponding color that represents the robot's status in completing that event.

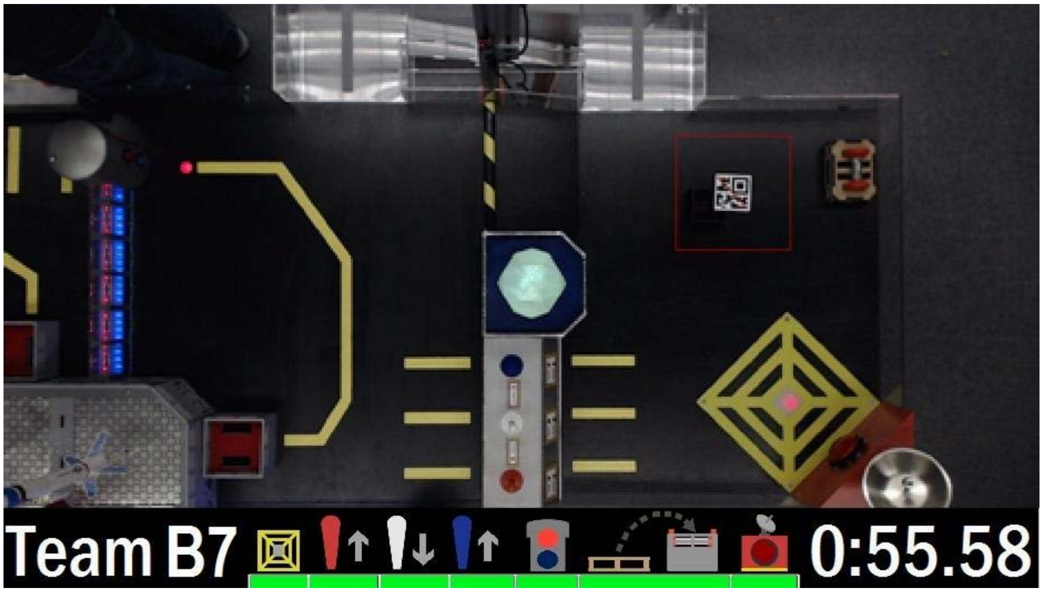

Figure 5: User interface of RPS system

Conclusion

The Ohio State University's College of Engineering honors first-year engineering program has made immense strides in maximizing student growth. Beginning with classroom content, the program has utilized large teaching teams to maximize the benefits of the inverted classroom approach, allowing fast paced learning for students with minimal burden. This pacing is crucial in providing students with the hands-on background in engineering needed to tackle an autonomous robotics project. The project is designed to be not only challenging but incredibly engaging and realistic. This is accomplished through several developed technologies. Of primary importance is an interactive robotics course based on a real world scenario and containing several tasks of varying complexity for student robots to finish. This is supported not only by modular hardware and software, but also an optical positioning system. To interact with the course students are provided an in-house controller, a store of materials, and access to automated fabrication processes such as 3D printing and laser cutting. These materials provide resources to students that give creative freedom while also being under the control of the program. Students work with the materials and the course to develop their autonomous robots in teams of four. They achieve weekly goals via performance tests that lead to completion of the entire course, first in individual competition and then in a public final competition event. The public event not only provides students a rewarding opportunity to present their work, but also provides opportunity for public outreach for engineering. In all, the program fully encompasses many of the most important aspects of engineering education in today's world. Students leave the program not only accomplished in their endeavors, but endowed with a level of hands-on experience rarely seen so soon in an engineer's college career.

References

1. Gateway Engineering Education Coalition. (n.d.). Retrieved February 12, 2017, from http://www.gatewaycoalition.org/index.aspx
2. Robotics - Autonomous Waste Disposal. (n.d.). Retrieved February 12, 2017, from http://drexel.edu/engineering/resources/prospective-students/undergraduate/freshman-engineering-experience/freshman-engineering-courses/ENGR102/robotics
3. Pavlic, T., & Dutta, P., & Hoffmann, M., & Radigan, J., & Beams, J., & Justen, E., & Demel, J.T., & Freuler, R.J. (2003, June), *Experiences With A Comprehensive Freshman Hands On Course Designing, Building, And Testing Small Autonomous Robots,* Paper presented at 2003 ASEE Annual Conference & Exposition, Nashville, Tennessee. https://peer.asee.org/11882
4. Vernier, M. A., & Wensing, P. M., & Morin, C. E., & Phillips, A., & Rice, B., & Wegman, K. R., & Hartle, C., & Clingan, P. A., & Kecskemety, K. M., & Freuler, R. J. (2014, June), *Design of a Full-Featured Robot Controller for Use in a First-Year Robotics Design Project,* Paper presented at 2014 ASEE Annual Conference & Exposition, Indianapolis, Indiana. https://peer.asee.org/20260
5. Newstetter, W. C., & Svinicki, M. D. (2014). Learning Theories for Engineering Education Practice. In *Cambridge handbook of engineering education research*. Cambridge University Press.
6. Morin, B., & Kecskemety, K. M., & Harper, K. A., & Clingan, P. A. (2013, June), *The Inverted Classroom in a First-Year Engineering Course,* Paper presented at 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia. https://peer.asee.org/22605
7. Harper, K. A., & Zierden, H. C., & Wegman, K. R., & Kajfez, R. L., & Kecskemety, K. M. (2015, June), *Teaching Assistant Professional Development Through Design: Why They Participate and How They Benefit,* Paper presented at 2015 ASEE Annual Conference & Exposition, Seattle, Washington. 10.18260/p.24806
8. Fentiman, A., & Gustafson, R. J., & Merrill, J., & Demel, J., & Freuler, R. (2001, June), *Developing And Implementing Hands On Laboratory Exercises And Design Projects For First Year Engineering Students,* Paper presented at 2001 Annual Conference & Exposition, Albuquerque, New Mexico. https://peer.asee.org/9104
9. Frank, D. J., & Witt, K. J., & Hartle, C., & Enders, J. J., & Beiring, V., & Freuler, R. J. (2016, June), *A Low-Cost Robot Positioning System for a First-Year Engineering Cornerstone Design Project,* Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26355
10. Cohen, W. F., & Enders, J. J., & Kolotka, K. L., & Freuler, R. J., & Grzybowski, D. M. (2016, June), *MAKER: Applications of 3-D Printing and Laser Cutting in the Development of Autonomous Robotics,* Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.25605
11. Whitfield, C.A., & Freuler, R.J. & Allam, Y., & Riter, E.A. (2011, August) "An Overview of Highly Successful First-year Engineering Cornerstone Design Projects", *Proceedings of the 2011 International Conference on Engineering Education,* ICEE-2011, Belfast, Northern Ireland.
12. McGourty, J., & De Meuse, K. P. (2001). The Team Developer: An Assessment and Skill Building Program. New York: John Wiley & Sons, Inc.
13. Schulz, M.H., & Danish, E.J., & Leonhardt, T.W., & Jackson, J.W., & Frank, D.J., & Freuler, R.J. (2017, June), "*Modular System of Networked Embedded Components for a First-Year Engineering Cornerstone Design Project*", Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio.