AC 2011-2200: COMPUTER INTERFACE INNOVATIONS FOR AN ECE MOBILE ROBOTICS PLATFORM APPLICABLE TO K-12 AND UNIVERSITY STUDENTS

Alisa N. Gilmore, University of Nebraska - Lincoln

Alisa N. Gilmore, P.E. is a Senior Lecturer in the Department of Computer and Electronics Engineering at the University of Nebraska - Lincoln. Since 2006, she has served as Senior Staff for administering NSF grants in the ITEST and Discovery K-12 programs associated with using robotics in the K-12 arena to educate teachers and motivate student achievement in STEM. At the University, she has developed and taught courses in robotics, electrical circuits and telecommunications. Prior to coming to UNL, Ms. Gilmore worked in telecommunications and controls R&D and manufacturing. She has used her industry background to foster industrial partnerships at the University, and to develop courses and supervise students in projects that support educational robotics.

Mr. Jose M. Santos, University of Nebraska-Lincoln

Mr. Santos is an undergraduate student at the University of Nebraska-Lincoln (Omaha Campus) where he's currently earning a double-major in Computer Engineering and Mathematics. He also holds a Bachelor's Science degree in Electronics Engineering Technology (EET) from DeVry Institute of Technology (now DeVry University). He is the creator and lead software architect of the CEENBoT-API (Application Programming Interface) presently in use in various engineering curricula at the University of Nebraska.

Aaron Joseph Mills, Iowa State University

Aaron Mills graduated in 2010 from the University of Nebraska - Lincoln and the University of Nebraska at Omaha with degrees in Computer Engineering and Computer Science. During his undergraduate he was involved extensively in SPIRIT as a software developer for the CEENBoT platform. He is currently pursing a Master's degree in Computer Engineering at Iowa State University, but remains involved in SPIRIT as the primary developer of the CEENBoT Graphical Programming Interface. Aaron spends much of his freetime at Iowa State University as a collaborator with the electrical group on Team PrISUm, a student-run organization that designs, builds, and races solar powered electric vehicles. His interests lie primarily in the area of embedded system hardware and software development.

Computer Interface Innovations for an ECE Mobile Robotics Platform Applicable to K-12 and University Students

Since the 1990's, robots have been adopted into K-12 classrooms and a host of University programs to engage and motivate students in STEM achievement and to aid in teaching core STEM disciplines. The robots used in these efforts have ranged from commercially available robots such as those manufactured by Lego®, VEX Robotics, or Pioneer, to a host of custom developed robots, such as those developed by Electrical and Computer Engineering (ECE) departments at Oregon State University (TekBot Learning Platform®) and Montana State University (ECEbot) [5,6,14,16]. The selection of the particular robot platform is a significant decision in such efforts and is based on many factors, including the platform's cost, size, components, modularity, and programming options [7, 8]. This selection is determined in part by the users' ability to configure, control and program the robot for their own purposes. For young elementary students who wish to program robots to act autonomously, their ability to do so lies in being able to use the programming language the robot supports. On the other hand, for more advanced users and for users at the college level, students require greater challenges and programming options that advance with these challenges. Few robot platforms have succeeded in offering programming options that address needs and meet interests across the K-University (K-16) educational continuum, from young elementary to college-level robot programmers.

In a fashion similar to a number of ECE departments, the University of Nebraska-Lincoln's Department of Computer and Electronics Engineering (CEEN Dept.) developed a custom robot in 2008, naming it the CEENBoT™, for use throughout a 4-year degree program and in an NSF funded ITEST K-12 educational outreach project. This paper provides an overview of the faculty and student development work that has resulted in the creation of Computer Interface Innovations for the CEENBoT™ mobile robotics platform that make it applicable to a wide range of student programming abilities, from novice programmers (young children in K-12) to college level ECE programmers. This feature sets this robot platform apart from others in its potential to be utilized for educational efforts that grow with the user, from elementary school through University. This is of particular interest to University programs that adopt robots into their degree programs and extend robotics into K-12 STEM outreach.

This paper is organized into the following sections: Introduction to the CEENBoT™ Mobile Robotics Platform, The Need for Computer Programming Interfaces for K-16, followed by descriptions of the Application Programming Interface (API), the TI Calculator Interface, and the Graphical Programming Interface (GPI) created for the CEENBoT™. Field Tests and Future Plans are briefly discussed, followed by Concluding Remarks.

Introduction to the CEENBoT™ Mobile Robotics Platform

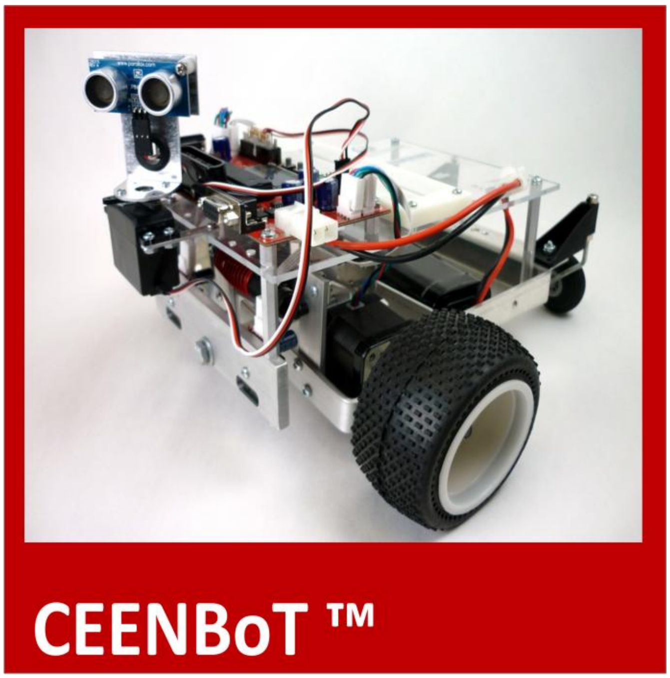
In 2008, the University of Nebraska-Lincoln department of Computer and Electronics Engineering created the CEENBoT™ robotics platform, which was designed to be applicable to K-12 outreach efforts and a continuum of courses in a 4-year Computer and Electronics Engineering program. The CEENBoT™ was designed to address educational needs at K-12 and University levels in terms of being a highly flexible, robust platform for project-based, hands-on learning with expandability for various microprocessors [1]. It has a wide range of applications developed for K-12 math and science standards, and an ongoing development for a grades 5-8 curriculum via an online interactive website [4]. It has a modifiable design consisting of off-the-shelf electronic hobby store components, instead of proprietary components as with commercially available robot kits.

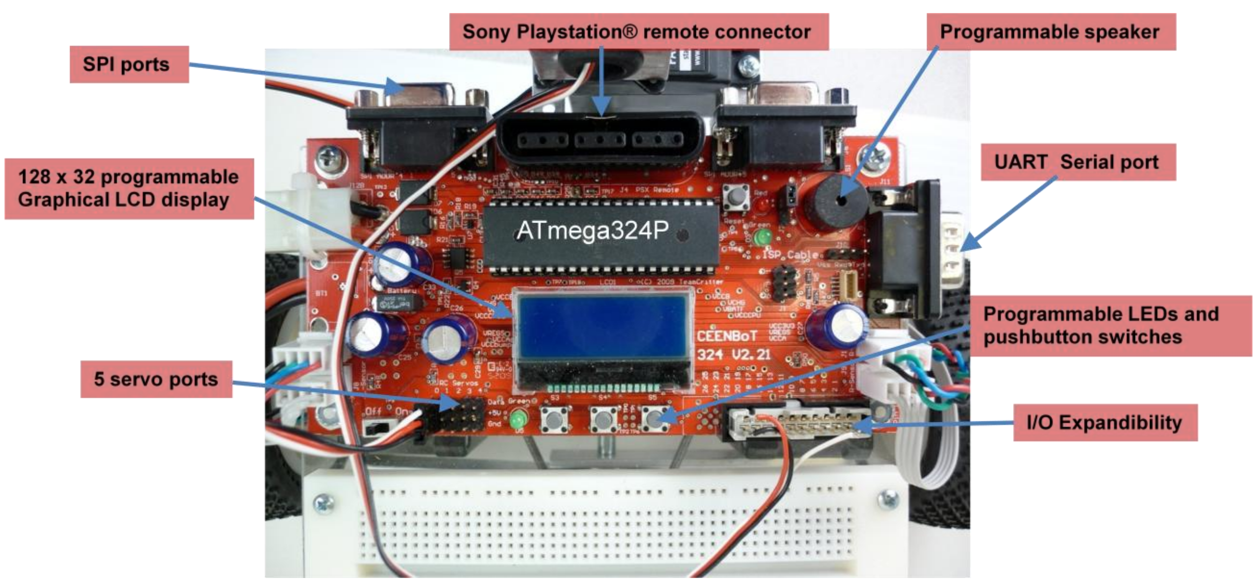
Since its creation the CEENBoT™ has been applied as a centerpiece for hands-on learning in the extensive Silicon Prairie Initiative for Robotics in Information Technology (SPIRIT) teacher professional development ITEST project (2006 - 2008). SPIRIT and the follow-on SPIRIT 2.0 Discovery K-12 project (2008 - 2012) have continued as a vibrant collaboration between faculty and staff at the University of Nebraska-Lincoln (UNL) department of Computer and Electronics Engineering, the University of Nebraska at Omaha (UNO) College of Education, and local Omaha public schools [1,4]. The SPIRIT project was shown to be highly successful for equipping K-12 teachers in STEM training, and to have promising initial results for increasing motivation in student STEM learning [1,3]. Thus, since its creation, the CEENBoT™ has proven an applicable platform for K-12 teacher training and engaging to students in the department's University courses and labs [2]. However, until recently, a missing element was a seamless method for programming the CEENBoT™. The original CEENBoT™ platform was designed by two CEEN undergraduate students, and it had the following attributes:

CEENBoT™ Attributes (6" by 8" footprint)

- Stepper motors for precision control
- Full independent wheel suspension for traversing uneven indoor or outdoor terrain
- Larger capacity, quick-change power supply
- Interchangeable rubber drive tires
- Remotely controllable using a Sony PlayStation® wired or wireless remote controller
- Large prototype board for projects and reliable quick connectors
- Serial-to-peripheral interface (SPI) to allow communication between multiple multiprocessors
- Flexible for K-16 educational applications to meet needs at multiple levels
- Platform will accommodate GPS, on-board video camera, robotic arm, and various sensors, wireless technologies, and microprocessor platforms
- Available in a number of configurations from unassembled kits to completed modules

With feedback from CEEN professors and K-12 field tests, in 2009, the original student-developed platform was re-designed into a more cost-effective, robust, energy efficient platform, with a single microprocessor design that allowed the robot to be programmed for the first time so users could create custom autonomous programs. The resulting CEENBoT™ version 2.2 included improved power management, longer battery life, and a streamlined microcontroller board [2]. Other upgrades included: a 128 x 32 programmable graphical LCD display, 5 servo motor control ports, I/O expandability for additional sensors, a programmable speaker, 3 programmable LEDs, and 3 programmable control switches. Photos of the current CEENBoT™ platform, and a close up of the CEENBoT™'s controller board are shown below:





The Need for Computer Programming Interfaces for K-16

In 2008, the newly created CEENBoT™ did not possess a straightforward method which enabled users to program and control it beyond the included firmware settings. The firmware included two modes: a remote motor control mode, where the robot could be remotely controlled via a Sony PlayStation® handheld controller and an obstacle avoidance or "bumpbot mode" that triggered escape movements when the robot's front infrared sensors were triggered. Clearly, for use beyond these applications, robust programming options were needed to meet the needs of the K-12 SPIRIT project, a GEAR-TECH-21 4-H project for which the CEENBoT™ was chosen as the centerpiece of a national scale-up grant, and for deeper University use [17]. The chart below shows a sampling of commonly used robot platforms and the programming languages they support. Where provided by the manufacturer, the intended user audience is also listed.

Robot Platforms	**Programming Options**	**Characteristics**
LEGO® MINDSTORMS® NXT, Age 12+ [11]	NXT Graphical Programming, based on National Instruments' Labview®	Modular LEGO building blocks & sensor components [10]
VEX Robotics, Middle School+	easyC, ROBOTC, MPLAB	Modular metal components for building mechanical variations, including arms [9]
TekBot® Learning Platform, ECE Level	AVR microcontroller C programming	Built from a kit of electronic components; ECE focus
iRobot Create	MSRS, Python	Roomba without the vacuum; RS-232 control
MIT Handyboard	Interactive C	A robot controller that needs a mobile body, often used with LEGO®
Parallax Boe-Bot®, Age 13+	PBASIC	Published K-12 Curriculum with student guide [12]
Scribbler Robot, optional Georgia Tech IPRE Fluke board [13]	Custom Graphical GUI programming, Python, C++	Standard Programming GUI for age 14+. Programmable with Python over bluetooth with Fluke board, used at the University level.

In order to equip the CEENBoT™ with a programming interface for the K-12 audience, leadership from the SPIRIT and 4-H projects met in the fall of 2008 to create a set of requirements for a CEENBoT™ Graphical Programming Interface (GPI) software that would be suitable for young children without programming experience and be able to support and extend the functionality of existing Nebraska 4-H robotics lessons. SPIRIT Senior Staff and CEEN faculty Alisa N. Gilmore served as the GPI Project Coordinator and was responsible for supervising a group of six undergraduate CEEN students recruited to work on creating this interface. The GPI work took place from November 2008 to August 2009, and in the end the group of students produced a user friendly, functional GPI that fulfilled the project's requirements. The GPI software was named CEENBoT™ Commander. This initial version went through further refinements from September 2009 to January 2011 based on feedback from the SPIRIT project team, under the technical guidance of one student from the original group, Aaron Mills, who has remained the primary developer of the GPI.

CEENBoT™ version 2.2 introduced in the Fall of 2009 allowed users to create custom programs on the CEENBoT™ for the first time by programming its central Atmel ATmega324P microprocessor in C. The new programming capability was trial run the fall semester of 2009 with a large degree of success in an Introduction to Mobile Robotics course developed and taught by Alisa Gilmore [2]. In order to simplify the user experience of programming in C on the CEENBoT™ and test labs for a new Mobile Robotics course, Professor Alisa Gilmore recruited CEEN undergraduate student, Jose Santos, to work on these issues during the summer of 2010. Out of his creative work, the CEENBoT™ Application Programming Interface (API) was born. Mr. Santos simultaneously worked to complete a project that enabled Texas Instrument (TI) graphing calculators to be used to program the CEENBoT™, a project that was begun as a senior capstone design project for a target audience of K-12. Jose has continued as the primary developer of the CEENBoT™ API.

In summary, the computer interface innovations developed for the CEENBoT™ mobile robotics platform for K-16 include: the Graphical Programming Interface (GPI), the Application Programming Interface (API), and the TI Graphing Calculator Interface. They are summarized in the chart below. Detailed descriptions of each follow.

CEENBoT™ Mobile Robotics Platform Programming Options	**Description**	**User Audience**
CEENBoT™ Commander Graphical Programming Interface (GPI)	A graphical drag and drop Integrated Development Environment that allows inexperienced programmers to link graphical programming elements together to control the CEENBoT™	K-12 (Elementary +)
TI Graphing Calculator Interface	An interface that allows commands on the TI graphing calculators to be used to program the CEENBoT™	K-12 (Middle School +)
CEENBoT™ Application Programming Interface (API)	An extensive suite of CEENBoT™ specific C-functions designed to simplify interaction with the CEENBoT™'s firmware and hardware	K-12 (Advanced) and University

The CEENBoT™ Application Programming Interface (API)

While the primary motivation for creating the CEENBoT™ API was to simplify the details needed to program the CEENBoT™, a secondary motivation was conceived by its student developer in an attempt to serve the need for compiling and uploading programs that had been created graphically by the CEENBoT™ GPI. Thus the CEENBoT™ API originated as a need for a killer application to be created to program the CEENBoT™ v2.21 platform with both a graphical language (GPI) and a sequential language (C).

The CEENBoT™ API allows a user to write programs in C in a manner that simplifies control of the CEENBoT™ platform. The CEENBoT™ API forms the primary core and foundational component that also enables other software technologies to write programs to the CEENBoT™, including, the GPI (The CEENBoT™ Commander) and the CEENBoT™ TI Interface. This idea is conveyed in Figure 1 below:

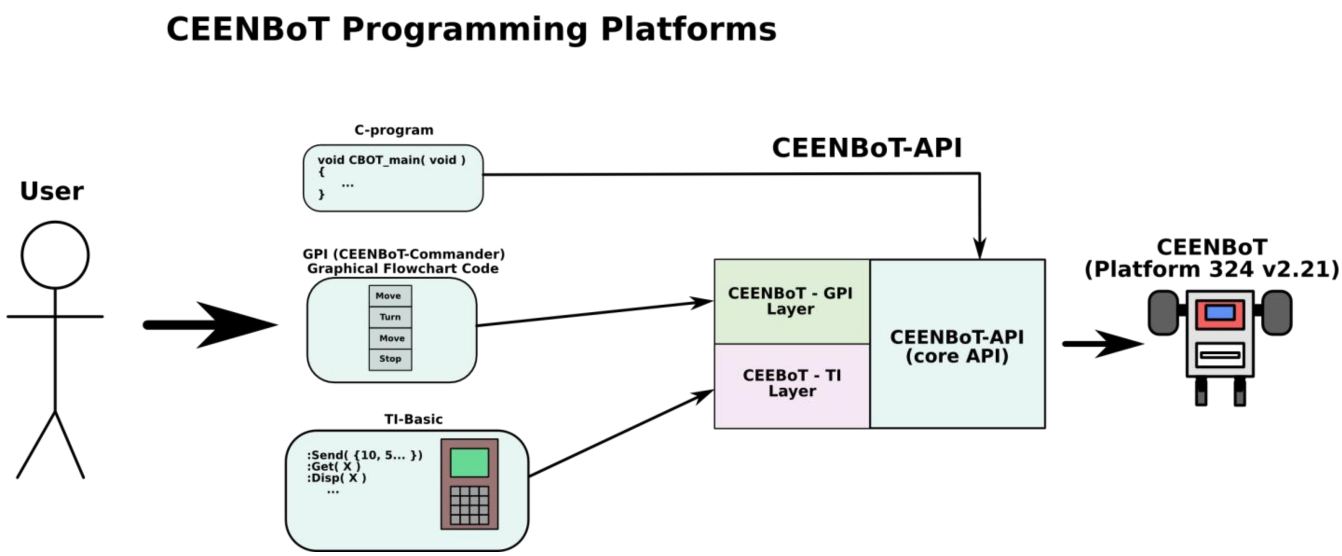


Figure 1

CEENBoT™ API is a static library that is used in conjunction with the C compiler (AVR- GCC) that targets the CEENBoT™'s AVR microcontroller architecture. It is essentially a collection of C functions pre-compiled into a single static library file.

The CEENBoT™ API exposes a rich set of functions that allows users to control and manipulate the CEENBoT™ in a simplified manner via well-documented function calls. The API functions allow various hardware resources available on the CEENBoT™ to be easily manipulated. Some of these resources include peripherals embedded on the microcontroller unit itself, such as control of I2C (or TWI), SPI, or UART. Or, the CEENBoT™'s on-board peripherals can be controlled, such as writing to the graphical LCD display, flashing LEDs, and driving stepper motors.

Users can take advantage of the API library's extensive set of functions (over 220) to write embedded programs that control the CEENBoT™ without the need for intimate knowledge of the its electronics or firmware. This allows the user to focus on actions, while the API handles the details. A user's program links with the API static library and uses the AVR-GCC compiler to generate a HEX file that is uploaded (or flashed) into the CEENBoT™ 's microcontroller's memory. This idea is illustrated in Figure 2 below:

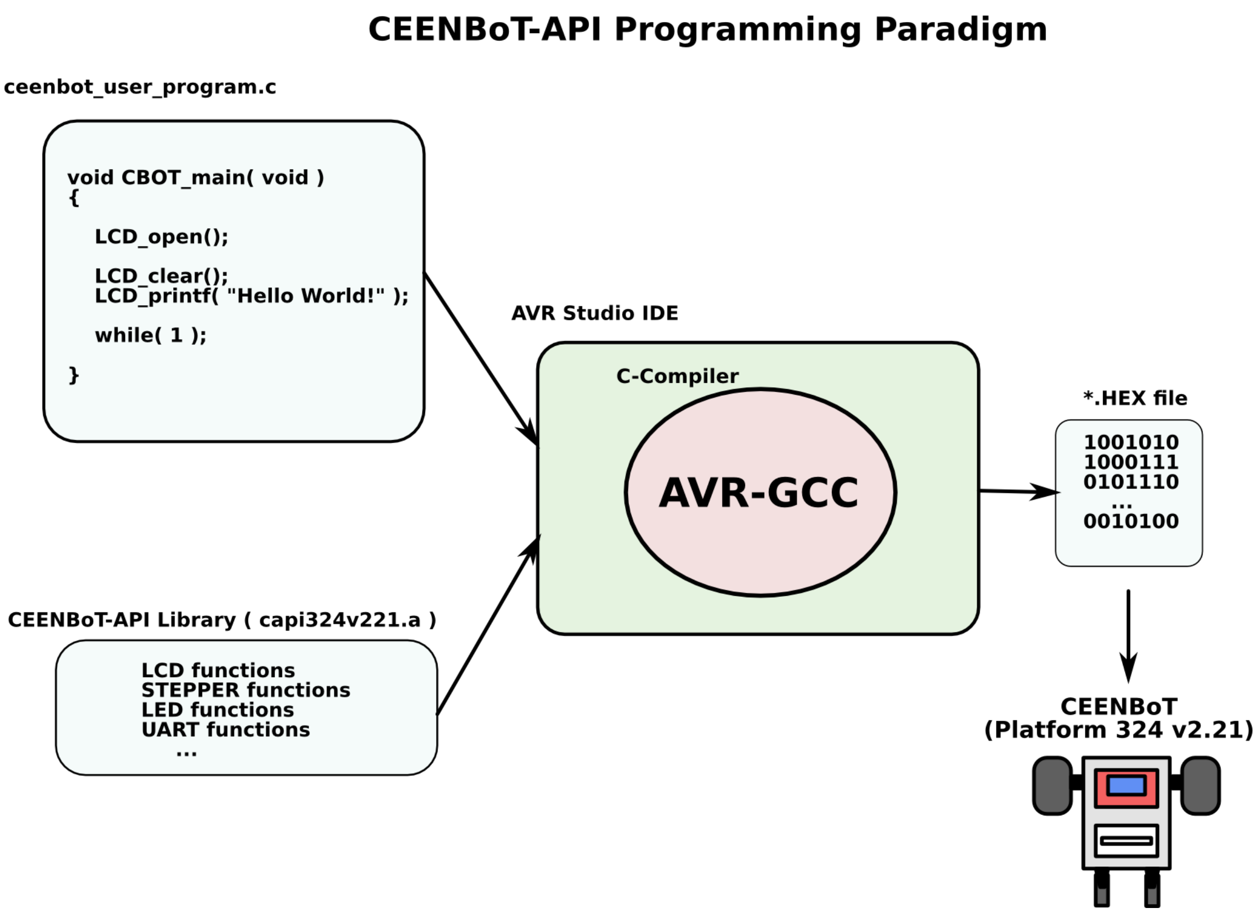


Figure 2

Presently, CEENBoT™ programming in C is done using the AVR Studio IDE (Integrated Development Environment) which is made freely available by ATMEL. The functions in the CEENBoT™ API library are grouped into functionally-related units called modules. Each module is in charge or acquiring the necessary resources (such as memory, I/O port pins, and peripherals) to achieve a task or control a particular peripheral device. The current functional modules available through the CEENBoT™ API include those listed below, and more are being developed:

ADC – Provides supporting functions for using the onboard Analog to Digital Converter (ADC) peripheral.

ISR – Provides supporting functions for declaring interrupt service routines (ISRs) which may also be used by other modules or user defined.

LED – Provides supporting functions for writing to the on-board LEDs.

LCD – Provides supporting functions for writing to the on-board graphical LCD display.

PSXC – Provides supporting functions for communicating with a Sony PlayStation2® (PS2) type controller using the on-board PS2 controller connector.

SPI – Provides supporting functions for using the serial peripheral interface (SPI) on the microcontroller unit.

STEPPER – Provides supporting functions for controlling the CEENBoT™'s stepper motors.

SWATCH – Provides supporting functions for using the stopwatch module, which can be used to measure time in units of 10us/tick.

TINY – Provides supporting functions for peripherals under direct control of a secondary supporting microcontroller unit, which on the CEENBoT™ is the ATtiny48. The TINY is used to acquire the state of on-board push-button switches, attached RC servos, and acquire the state of on-board Infrared sensors on the CEENBoT™.

TMRSRVC – Provides supporting functions for millisecond accurate timing services.

UART – Provides supporting functions for using the on-board UARTS in asynchronous mode.

USONIC – Provides supporting functions for using the Ping Ultrasonic sensor by Parallax, an optional peripheral used on the CEENBoT™ for Mobile Robotics courses.

Figure 3 illustrates the modular breakdown of the CEENBoT™ API and its subsystem modules. Note that while this figure illustrates the modular organization of the API, the entire API itself is encapsulated into a single static library file.

The advantage leveraged by the CEENBoT™ API can be conveyed by considering that the STEPPER subsystem module alone encompasses close to 2000 lines of code. A considerable amount of work would be required if the user is expected to do this work alone by writing similar code 'bare-metal' style to control the CEENBoT™'s motors. The CEENBoT™ API allows users to program without having to directly manage all the intricate details of the CEENBoT™'s electronics.

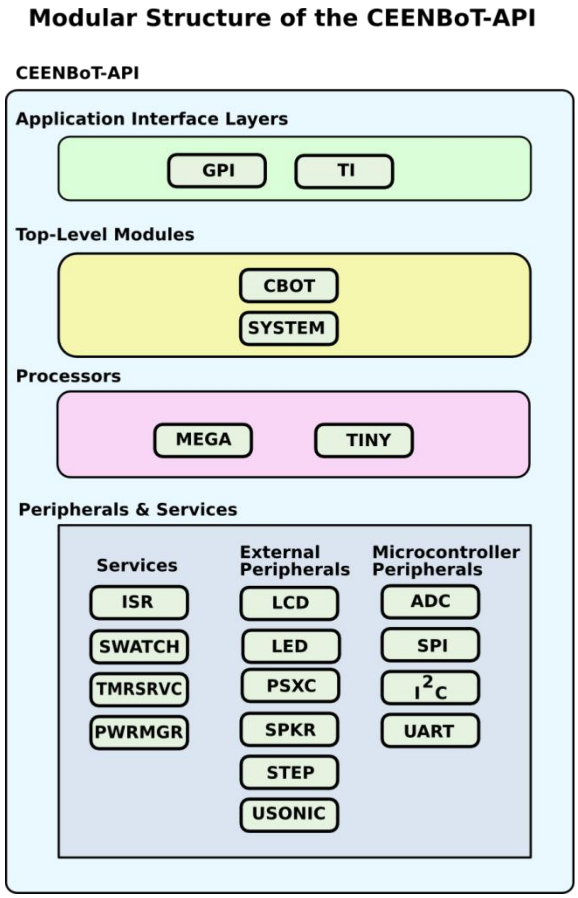


Figure 3

The functions of the CEENBoT™ API have been well documented to invite and entice users who would engage in C programming to explore the CEENBoT™. Available documentation includes a "Getting Started" guide, along with a more in-depth 158 page "Programmer's Reference Manual" that contains descriptions of all available functions and code examples. These resources allow ease of implementation in the classroom and immediate exploration. In addition, the API serves as the foundation for the other CEENBoT™ programming technologies including the CEENBoT™ Commander (GPI) and the TI Calculator Interface, which provide additional means to entice K-12 users (as well as instructors) into the world of STEM courses.

The CEENBoT™ Texas Instruments® (TI) Calculator Interface

A secondary software technology made possible by the CEENBoT ™ API is the CEENBoT™ TI Calculator Interface. The CEENBoT™ TI software interface consists of a thin software layer that sits on top of the CEENBoT™ API. It allows a user to connect a number of Texas Instruments graphing calculators to the CEENBoT™ with the appropriate interfacing hardware. The TI interface allows users to write programs on their TI calculators using TI-BASIC (an interpreted programming language used in nearly all Texas Instruments calculator models) in order to control the CEENBoT™ and have it perform various tasks, just as they would if they were writing programs with the CEENBoT™ API using the C programming language. The TI interface provides, yet, another option that invites exploration of the CEENBoT™ robotics platform in an open-ended and intuitive manner for the K-12 audience.

Documentation for the TI Calculator Interface is available in the form of a User's Guide and Command Reference manual. Graphing calculator models supported include the TI-82, TI-83, TI-84, TI-85, TI-86, and TI-89. The TI calculator of choice plugs into the TI/CEENBoT™ Adapter Board via a TI-communication Link Cable that connects it to the CEENBoT™, as shown in Figure 4 below.

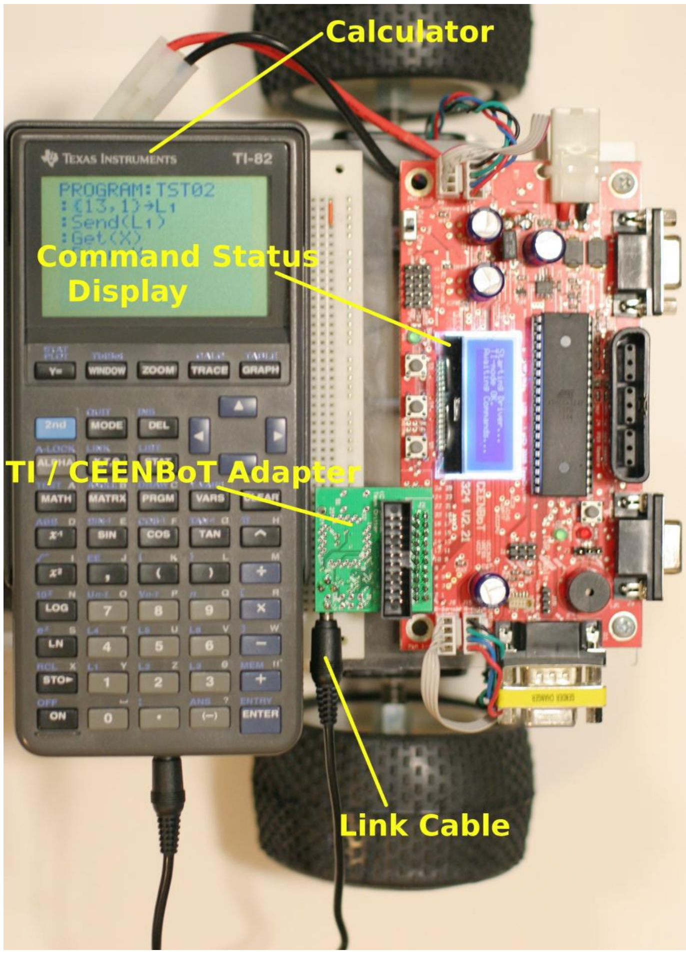


Figure 4

The CEENBoT™ Commander Graphical Programming Interface (GPI)

Graphical programming languages are gaining more interest in a variety of fields and industries. These languages can lower the barriers for entry for those who are not familiar with traditional text-based programming languages, such as C, and allow them to develop programs more quickly and with less training. The high-level elements in a graphical programming language are especially useful for abstracting complicated data transformations. This abstraction encourages the programmer to focus on developing the end application rather than getting mired in, for instance, hardware-level communication issues.

Thus, a graphical programming interface, or GPI, seems an ideal tool to get primary- and secondary-school students interested in math, science, and engineering. In particular, the goal of the GPI developed to empower this targeted audience with the ability to program a robot with no background in either programming or robotics. It is a tool designed by an educational institute for use in other educational institutes.

The Graphical Programming Interface (CEENBoT Commander) is a GUI application program that runs on a PC or MAC which allows users to write programs for the CEENBoT™ robotics platform using a graphically-oriented, intuitive user interface as an alternative to writing programs in the C programming language. The GPI internally generates code that is CEENBoT™ API compliant. This CEENBoT™ GPI code consists of a thin software layer that sits on top of the CEENBoT™ API.

The GPI project began in the fall of 2008 as an objective of both the SPIRIT project and the 4-H GEAR-TECH-21 project for which the CEENBoT™ would serve as the robotics platform for K-12 teachers, 4-H volunteers, and students. The objective was to simply create a simple graphical programming language interface for users with no programming experience to write programs to control the CEENBoT™.

After several options, including Arduino, were considered as platform for GPI development, Java was selected as the project language with the intent that a single application could run with minimal changes on both Windows and Macintosh operating systems.

The user interface of the GPI is designed with the intent that the flow of the program can be interpreted without much effort on the part of the user. In other words, the programs that can be built with it should be self-documenting. Tools are dragged and dropped from a list into a work area and then configured using simple input controls, as depicted in Figure 5 below.

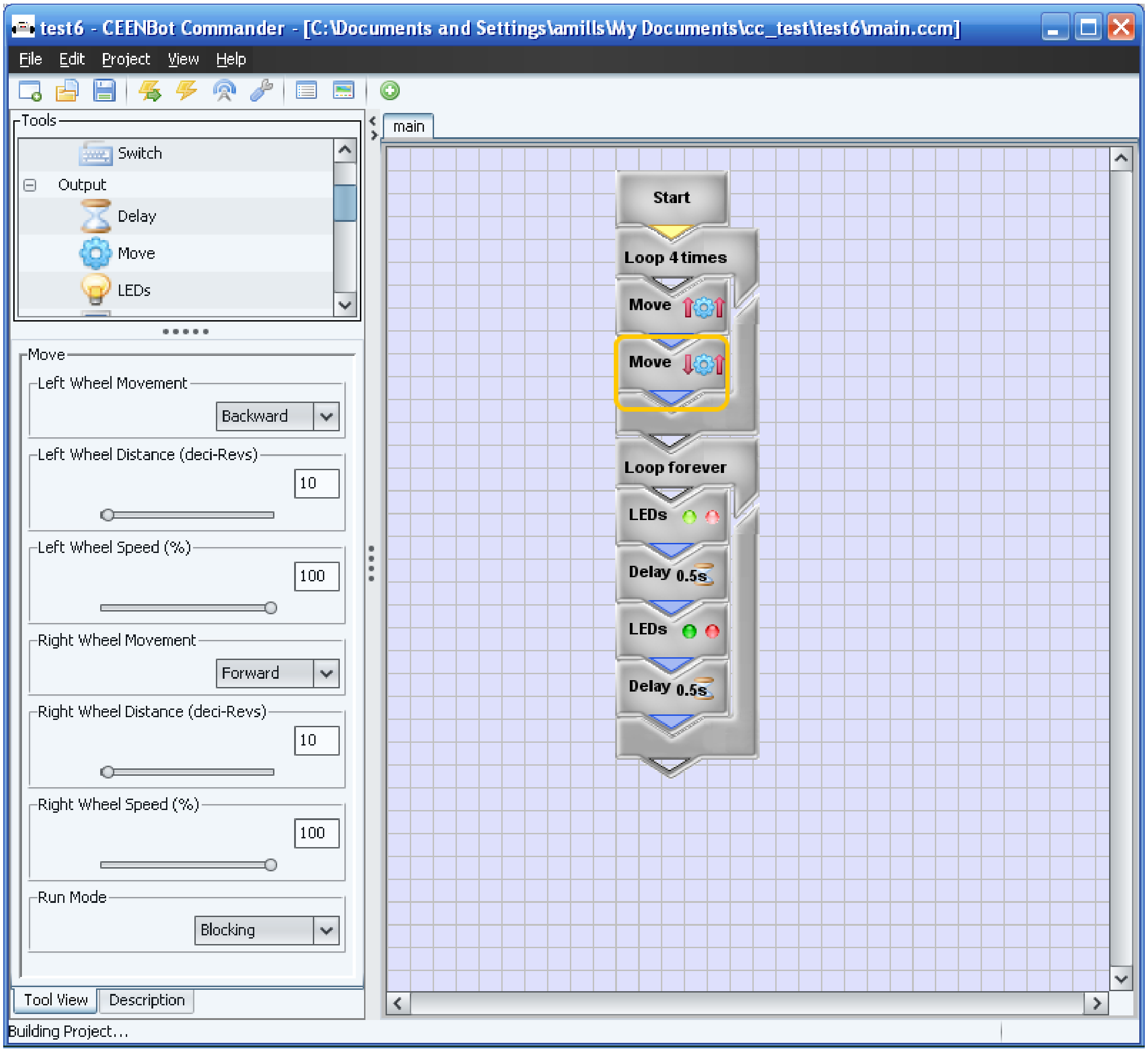


Figure 5

The following is an abbreviated list of the currently available programming tools and their descriptions:

Tool	Description
BumpLeft	Reads the state of the left infrared bump sensor into a variable.
BumpRight	Reads the state of the right infrared bump sensor into a variable.

Switch	Reads the state of a selected switch into a variable.
Delay	Pauses program execution for a select amount of time.
Move	Sets the desired distance and velocity for the left and right wheels, as well as the run mode (e.g. blocking vs. non-blocking)
LEDs	Sets the state of selected LEDs.
Display	Prints text or variables onto the onboard LCD.
Branch	Allows decision-making based on some condition.
Loop	Allows repeating blocks of code either a fixed number of times, or based on a condition.
Math	Allows basic math functions to be performed on variables.
Goto	Causes the program flow to jump into another loaded module, and then return after it has completed execution.

For purposes of encapsulation and code-reuse, the GPI allows users to develop a series of independent modules or sub-programs which can then be loaded into the project. These modules are stored on disk as XML, which is both human-readable and conducive to the nested nature of program flow control methods. When it comes time to build the project, each onscreen tool or component is translated into C code based upon its current configuration, and then this code is compiled as normal against the CEENBoT™ API library. The final result is a HEX file that can be flashed onto the target platform using the GPI itself or external software.

An example program created using the GPI is shown in Figure 5. Program execution begins at the Start tool and travels downward.

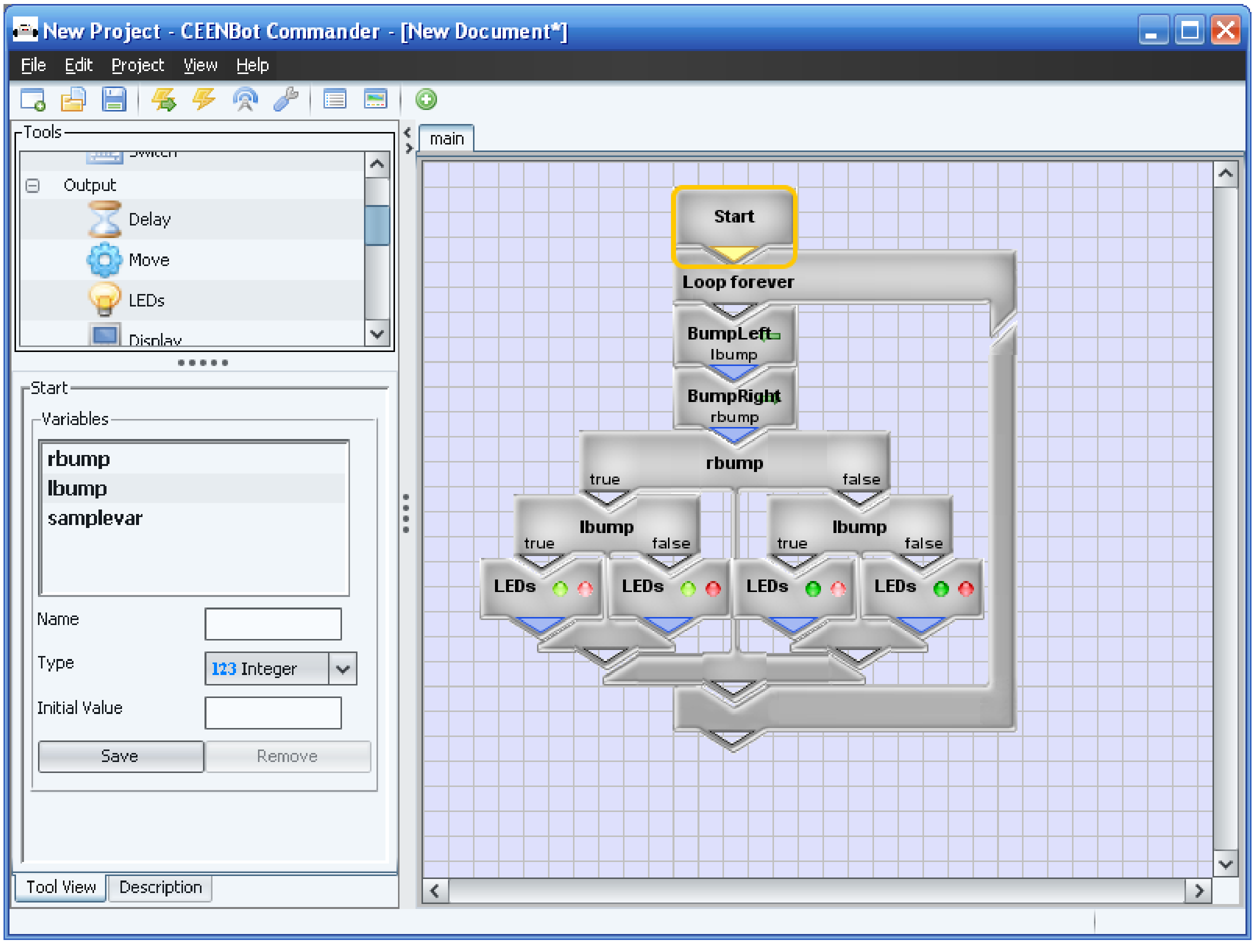


Figure 6

In pseudocode, the program expresses the following:

```
} while( true ){
        bool lbump = leftIRSensorObstructed();
        bool rbump = rightIRSensorObstructed();
        if ( rbump ){
                if( lbump ){ setGreenLED();  setRedLED(); }
                      else {setGreenLED(); clearRedLED(); }                    }
        else {
                if( rbump ){ clearGreenLED(); setRedLED();}
                      else { clearGreenLED(); clearRedLED(); }
              }                                                                }
```

Even more simply, if we consider Boolean variables {BumpLeft, BumpRight} as inputs and variables {GreenLed, RedLed} as outputs, the program expresses the set of equations:

RedLed = BumpLeft

GreenLed = BumpRight

It is also possible to preview the automatically generated C code inside the GPI. This was mainly included as a debug mechanism for developers, but may later be expanded upon as a way to simultaneously introduce users to traditional text-based languages, such as C. The C code for the previous example is shown in Figure 7. This is just one example of how GPI developers need to continue to work closely with students and educators to get a clear idea of how they hope to use the software in school curricula. Looking forward, many other exciting tools and features are being planned. One which will probably come up in the near future is scheduling and the capability for multi-threaded programs, something that should be neatly facilitated by the existing chainlike form of user programs.

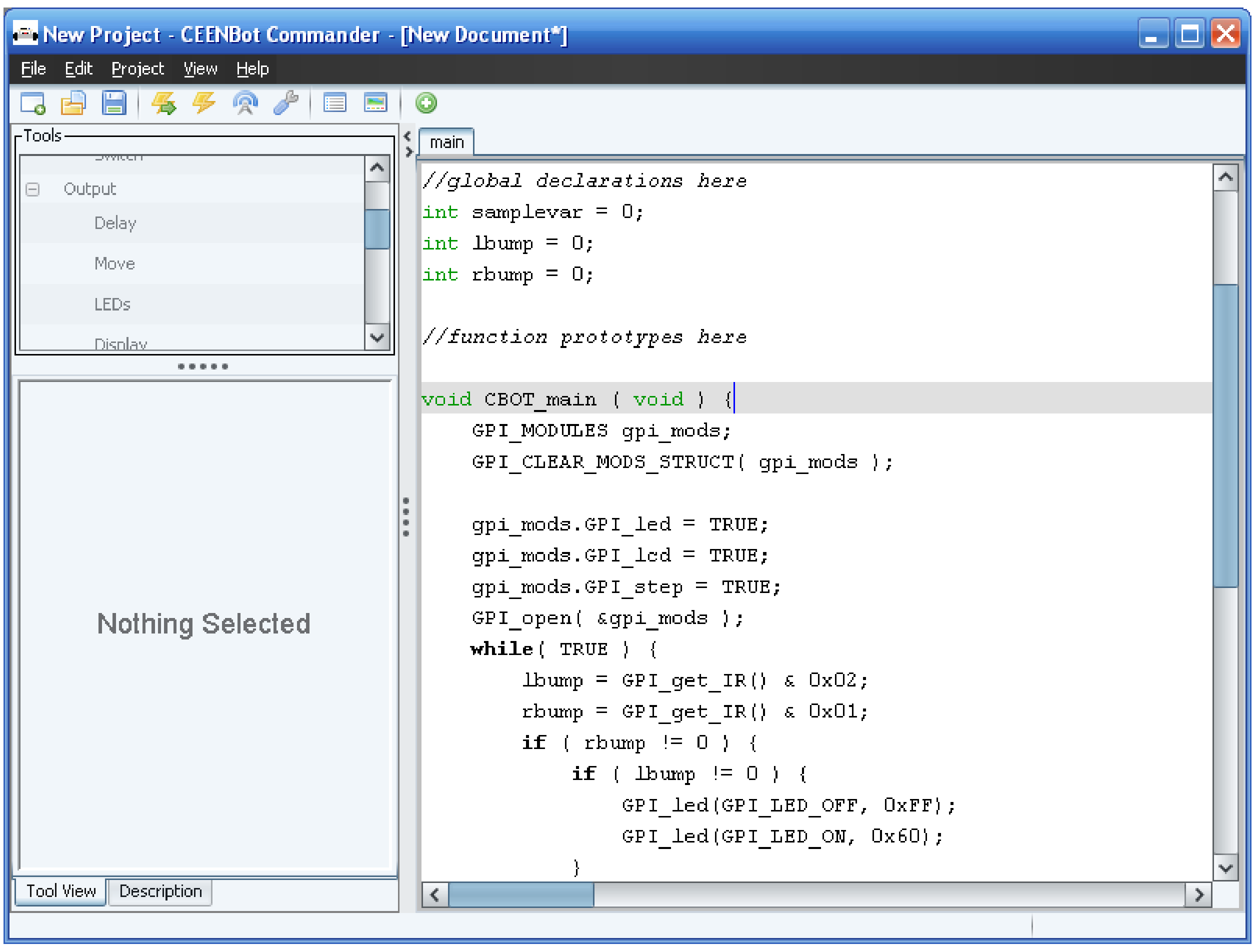


Figure 7

Field Tests and Future Plans

The beta versions of the CEENBoT™ API, GPI and TI Calculator Interface were all deployed into the hands of college and K-12 teacher and student users in the fall and early spring semesters of 2010-2011 for feedback purposes. They were received with very positive results. At the University level, the CEENBoT™ API was used as a foundational element for lab programming exercises in a new 4 credit hour Mobile Robotics course created and taught by Alisa N. Gilmore in the fall of 2010. The course was comprised of 13 senior and 2 junior students and focused on the implementation of reactive and behavioral-based robotics using the CEENBoT™ platform. API functions were used as building elements that allowed students to write embedded programs and integrate and control a variety of sensors for AI mobile robotics applications. The feedback from a detailed course survey was very positive. Over 78% of the students in the course agreed or strongly agreed that the API was a means for learning concepts on the syllabus, 71% felt it provided a source of motivation or increased their interest level in the class, and, for 71%, the API provided a sense of personal engagement on the assignments. The API was also introduced into the introductory CEEN 1030 course taught by Roger Sash. In this course, all CEEN students build a CEENBoT™ and take it with them for applications in follow-on courses. The CEENBoT™ API exposed these students to embedded system concepts and basic C programming as students in this class had never taken a programming course, or were concurrently taking their first programming course in Java. The students were able to program the CEENBoT™ in several lab exercises using CEENBoT™ API functions with much ease, and were generally very excited to be able to program their robot which they constructed earlier in this course.

The CEENBoT™ GPI and TI Programming Interface were also introduced to the sustained SPIRIT project K-12 teacher Saturday workshops during the fall 2010. The reception for the user group of approximately 80 K-12 teachers was extremely enthusiastic, even when testing early versions of the software with some bugs present, for which they provided valuable feedback. The teachers were presented with hands-on exercises and told the software wasn't perfect, but to comment on needed enhancements. As a result of the teachers' overwhelmingly positive response, both of these innovations were included as competition categories for their student teams in the 2nd Annual Nebraska Robotics Expo [15], which includes the CEENBoT™ Showcase, in its third year sustained from the SPIRIT project. The Showcase was held on February 19, 2011, including over 200 K-12 student participants in a number of events that involved the CEENBoT™. Included for the first time this year was the Autonomous Maze event in which student teams could choose to use the API, GPI or TI interface to program their robots to navigate tasks in a given course. The Autonomous maze was a success with Elementary, Middle school and High school teams competing and successfully completing the tasks, using all three technologies (Elementary teams even chose to use C-programming with the API).

These innovations will continue to be utilized in a progressive sequence of CEEN University courses (API), K-12 outreach (API, GPI and TI interface), and an upcoming roll-out of the 4-H

GEAR-TECH-21 project (GPI). The CEENBoT™ and API are also being tested at collaborating ECE departments.

Concluding Remarks

This paper provided an overview of Computer Interface Innovations developed at the University of Nebraska-Lincoln for the custom CEENBoT™ robotics platform which makes this platform applicable to a wide range of student interest and ability levels from K-12 to upper level University students. Few robot platforms possesses a range of programming options that allow access to novice programmers such as young elementary children up to embedded systems programmers such as upper level engineering majors. These Computer Interface Innovations make this possible. The potential impact for early recruiting of STEM talent from the youth pool increases dramatically using a platform that can grow with students as they advance in skills and ultimately follow them to a degree program.

Bibliography

1. Gilmore, A., Chen, B., and Grandgenett, N. "Using Robotics to Equip K-12 Teachers: The Silicon Prairie Initiative for Robotics in Information Technology", ASEE 2009
2. Gilmore, A., Detloff, H. "Assessing Senior Student Experiences with a Novel Mobile Robotics Learning Platform in a Computer and Electronics Engineering Program", ASEE 2010
3. Grandgenett, N.F., Chen, B., Timms, M. (2008). Project Report: The Silicon Prairie Initiative for Robotics in Information Technology (SPIRIT). A final evaluation report for the National Science Foundation Project #0525111, October 31, 2008. Accessed on January 21, 2009 from http://www.ceen.unomaha.edu/TekBots/SPIRIT2/Reports/.
4. Grandgenett, N.F., Chen, B., Ostler, E., Timms, M. (2008). Project Report: The Silicon Prairie Initiative for Robotics in Information Technology 2.0 (SPIRIT 2.0). An evaluation report for the National Science Foundation Project #0733228, December 20, 2008. Accessed on January 21, 2009 from http://www.ceen.unomaha.edu/TekBots/SPIRIT2/Reports/.
5. Heer, R.L., Traylor, T.T., Fiez, T.S. "Enhancing the Freshman and Sophomore ECE Student Experience Using a Platform for Learning. IEEE Transactions on Education", 46(4), November 2003.
6. Smith, A., Heer D., Traylor, T., Fiez, T.S. "A Custom Microcontroller System used as a platform for learning in ECE", ASEE 2004
7. Liu, Y. "From Handy Board to VEX: The Evolution of a Junior-Level Robotics Laboratory Course", ASEE 2009.
8. Berry, C. "Mobile Robotics: A Tool for Application-Based Integration of Multidisciplinary Undergraduate Concepts and Research", ASEE 2010
9. VEX Robotics Programming Software, accessed 1/18/11: http://www.vexrobotics.com/products/programming/

10. LEGO® MINDSTORMS® NXT Programming, accessed 1/19/11: http://mindstorms.lego.com/en-US/default.aspx?icmp=COFranchiseUSMINDSTORMS
11. LEGO® Parents Website on Starting Age for Products, accessed 3/4/11: http://parents.lego.com/en-gb/products/products.aspx?id=older
12. Parallax Boe-Bot® Robot, accessed 1/19/11: http://www.parallax.com/tabid/411/Default.aspx
13. Institute for Personal Robotics in Education (IPRE), accessed 1/19/11: http://wiki.roboteducation.org/Main_Page
14. Oregon State University Programmable TekBot® Systems, accessed 1/19/11: http://eecs.oregonstate.edu/education/products/programmable_systems.php
15. The Nebraska Robotics Expo 2011, CEENBoT™ Robotics Showcase, accessed 3/10/11: http://www.ceen.unl.edu/CEENBoT/Showcase/
16. Maher, R. C., Becker, J., Sharp, T., Peterson, J., and Towle, B.A. "Development and Implementation of a Robot-based Freshman Engineering Course", Proceedings of the ASEE Annual Conference and Exposition 2006
17. GEAR-TECH-21: Geospatial and Robotics Technologies for the 21st Century, accessed 3/10/11: http://4hset.unl.edu/4hdrupal/