2013 21st Mediterranean Conference on Control & Automation (MED)
Platanias-Chania, Crete, Greece, June 25-28, 2013

The Engino Robotics Platform (ERP) Controller for Education*

Georgios A. Demetriou, *Member, IEEE*, Antonis Lambrou, Nikleia Eteokleous and Costas Sisamos

Abstract—**The Engino Robotics Platform Controller, which is presented in this paper, is a control box intended for primary and early secondary education students. It is used to teach basic control, robotics and technology based courses. Along with the controller a series of external sensors have been developed that can be directly connected to the controller. The controller and the sensors allow students to build robots and other automated or interactive systems using the Engino components.**

I. INTRODUCTION

The company Engino Toy Systems in collaboration with Frederick University has developed a new robotic system as an extension to the Engino construction toy. The system consists of 2 parts: the Engino Robotics Platform (ERP) box with its peripherals and the Engino Graphical Programming Interface (EGPI). The product is the result of three year's research project funded by the Research Promotion Foundation (RPF) of Cyprus. The main objective of the ERP system is to provide effective tool not only for teaching robotics, control systems and technology course but also to be integrated as an instructional tool within the teaching and learning process, aiming to achieve instructional goals. The ERP system is suitable to be used by pre-primary to higher education students.

Robots are increasingly becoming important in advanced applications for the military, industry, space, home and many others. This has increased the need for more robotics' engineers for manufacturing, research, development and education [1]. Robotics, and more specifically mobile robotics, is widely accepted as a multidisciplinary approach to combine and create knowledge in various fields as mechanical engineering, electrical engineering, control, computer science, communications, and even psychology or biology in some cases. This has significantly changed the nature of engineering and science education at all levels, from K-12 to graduate school. Most schools and universities have already started integrating robotics courses into their curricula.

One of the greatest challenges in teaching young people is that of capturing their interest and of nurturing a fascination for learning. Educational robots play a big part in the robotics field as we start nurturing the basic ideas of robotics from a young age. Using robots in education simplifies this by appealing to the young mind [1].

Figure 1. Engino Robotics Platform (ERP) Controller

The reported project has been motivated by the challenge to design and implement a stand-alone robotics controller for the company Engino Toy Systems. The central objective is that students can build robots and other automated systems using the Engino parts and the Engino Robotics Platform (ERP) controller to control them. The ERP is shown in Fig. 1.

This paper emphasizes on the description of the ERP controller and thus only describes the programming that can be achieved directly on the ERP box (manual programming). It does not describe the programming software in any detail. Using the Engino Graphical Programming Interface (EGPI) software is beyond the scope of this paper.

The rest of the paper is organized as follows: Section 2 describes the previous work that has been done in this field. Section 3 gives a detailed description of the ERP controller, its peripherals and functionality. Section 4 delineates some cases studies of Engino robots using the ERP, and finally Section 5 concludes the paper.

II. RELATED WORK

The idea of robotics integration in education has been around for longer than someone might think of. It has been more than 20 years that the literature refers to the use of robots and programming in education [2, 3]. However, the great revolution in the field of educational robotics has been achieved throughout the last decade, where robotics escaped

*Research supported by RPF (ΙΠΕ).
G. A. Demetriou is with the Computer Science and Computer Engineering Department of Frederick University, 3080 Lemesos, Cyprus (telephone: +357-25730975; fax: +357-25735001; e-mail: g.demetriou@frederick.ac.cy).
A. Lambrou is with the Computer Science and Computer Engineering Department of Frederick University, 3080 Lemesos, Cyprus (e-mail: com.la@frederick.ac.cy).
N. Eteokleous is with the Department of Education of Frederick University, 3080 Lemesos, Cyprus (e-mail: pre.en@frederick.ac.cy).
C. Sisamos is the founder and managing director of Engino Toy Systems, Lemesos, Cyprus (e-mail: csisamos@engino.com).

978-1-4799-0997-1/13/$31.00 ©2013 IEEE

the laboratory and made efforts to connect to education [4]. The robotics materials (building blogs/ bricks, sensors and motors) which are perceived as toys by the children, and the educational activities designed using the aforementioned materials; bring the students closer to technology and their relationship with it [4, 5]. Numerous research studies suggest that robotics integration for educational purposes is an effective teaching method; however more research needs to be conducted in order to provide the foundation for implementing the appropriate teaching practices and strategies for the design of such learning environments [5]. Finally, researchers argue that if robotics activities are appropriately designed and implemented have great potential to significantly improve and enhance the teaching and learning process [6, 7].

There are other controllers/robotic systems that are currently used in education. Examples include the TTS Eggbox [8], the LEGO WeDo [9], the LEGO Mindstorms NXT [7, 10], and the VEX Robotics Design System [11]. The TTS Eggbox and the LEGO WeDo are simplified educational systems which are intended for children as young as 5 years old. These systems can be programmed to do simple tasks, such as turning motors or turning lights on and off. On the other hand, the LEGO Mindstorms NXT and the VEX Robotics Design Systems are advanced robotic systems which can be programmed to interact with the environment and follow a complex programing structures. Nevertheless, these systems are not intended for young children and they are primarily targeting students at the high school level.

The ERP system fills in the gap between basic programming and advanced programming. It allows students to program simple tasks as well as more complex tasks. The ERP offers three types of programming modes. Each programming mode has a different difficulty level ranging from the easiest to the most difficult.

III. The Engino Robotics Platform (ERP) Controller

The ERP system consists of the ERP controller, the external devices (sensors, lights, motors, etc.) and the programming software. The user can construct robots using the Engino parts and the ERP with its external devices. Programming the robots can be done in two ways: directly from the ERP controller (manual programming) or through the graphical interface (advanced programming).

The ERP can be programmed directly using the buttons that are on the controller casing (box). With this type of programming the user can program the robots manually to do simple operations such as turn on motors, turn on lights, play sound, etc. It does not allow more complex programming such as loops and other control structures. More complex programming can be done using the EGPI software. The software has two modes: graphical simulation (shown in Fig. 2) and graphical programming (shown in Fig. 3). The graphical simulation allows the user to program the robot or the graphical simulation of the ERP using a graphical illustration of the actual ERP box. The user can add virtual components (i.e. motors, sensors, lights) using “drag and drop” and program them the same way as they would be programmed on the actual ERP. This type of programming has the same limitations as with programming on the actual ERP box manually since it simulates the ERP operation. Programs can be simulated on the graphical simulation or can be downloaded on the actual ERP box and executed there for autonomous operation.

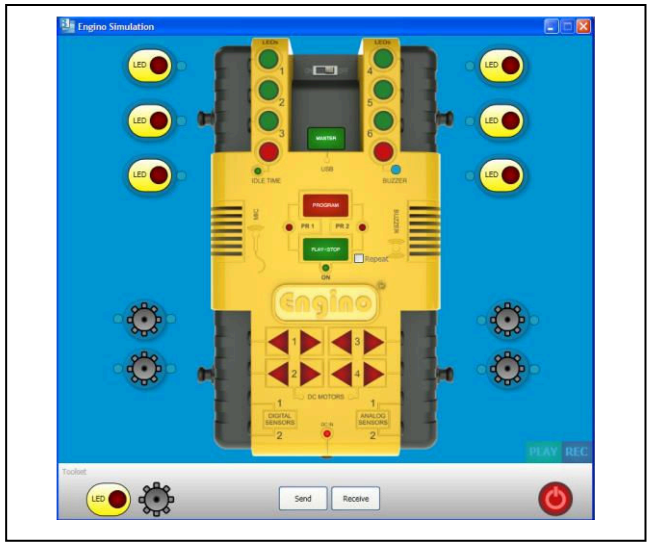

Figure 2. Simulation Programming Mode of the ERP

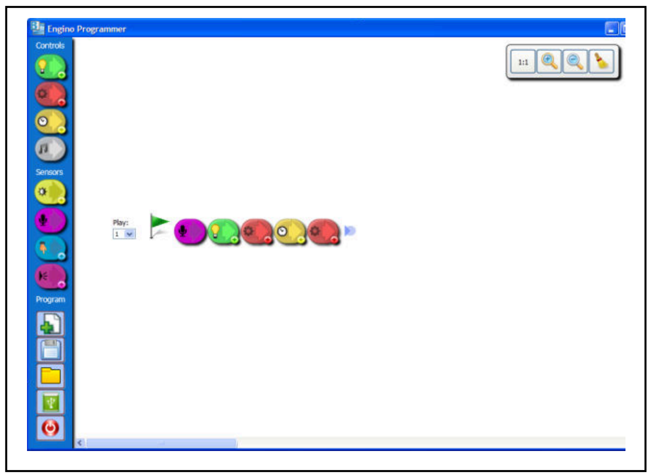

Figure 3. Graphical Programming Mode of the ERP

The other programming mode, using graphical programming, is done using the Engino programming language developed specifically for this application. It allows students to construct more complex programs to control the robots. Programs can be executed directly from a computer that is connected to the ERP box or they can be downloaded to the ERP box and executed from there. The latter allows for autonomous operation since the ERP should not be connected to a computer in order to operate.

A. System Description

The ERP PCB is shown in Fig. 4. Fig. 5 shows the ERP in block diagram. The whole system functions around the microcontroller which is an ARM 32-bit MCU with flash memory [12]. The rest of the peripheral (analog and digital inputs, microphone, USB, LED’s and motors) are connected at the various ports of the MCU. On top of the MCU runs a Real Time Operating System (RTOS) [13] which allows multi-tasking operations such as controlling multiple outputs in parallel. The MCU flash memory is used to store programs in order for the system to run autonomously.

The ERP can communicate directly with a host computer through a build in USB connection. This connection has three purposes:

- Control the ERP through the EGPI
- Download programs to the ERP from the EGPI
- Upload programs to the EGPI from the ERP control box.

The ERP can facilitate up to four external sensors at the same time which can be connected to the box through the two analog inputs and two digital inputs. The two analog inputs are used for the Engino analog sensors and the digital inputs are used for the Engino digital sensors. Currently, only a light analog sensor, a contact digital sensor and a distance digital sensor are available.

The controller can control up to 4 DC Engino motors and 6 LEDs through its output ports.

Additionally, an on-board microphone and an on-board electromagnetic transducer (buzzer) are embedded into the ERP controller. Both of these devices are used for real-time interaction with sound.

Eighteen (18) buttons are connected to the MCU. The buttons can be seen clearly on the ERP box of Fig. 1. The following buttons are part of the ERP box:

- Eight (8) buttons are used to manually program the motors. With the buttons the user can program the motors to move in either direction.
- Six (6) buttons are used to manually program the LEDs. The LEDs can be turned on by pressing the button.
- One (1) button is used to place idle time within a program during manual programming.
- One (1) button is used to play the buzzer during manual programming.
- One (1) button, labeled *PROGRAM*, is used to start manual programming.
- One (1) button, labeled *PLAY-STOP*, is used to start (execute) and stop the program stored in the ERP's memory. If this button is pressed continually then the program goes into an infinite loop mode until the button is pressed again.

The ERP can store one program at a time, that can be either recorded manually or a program downloaded from the programming software.

B. *Peripherals*

1) *Sensors*

As mentioned above, the ERP system can receive feedback from 2 types of sensors: (1) analog sensors, and (2) digital sensors.

The touch sensor is a simple contact switch, which is turned on when pressed. The distance sensor uses an Infrared (IR) emitter and an IR receiver in order to identify if there is a reflector (obstacle) in front of the emitter at any time. When the receiver collects an adequate amount of IR light (i.e. there is an object in front that reflects enough IR light) the digital sensor is turned on. Engino is planning to develop more analog and digital sensors that can be used with the ERP system.

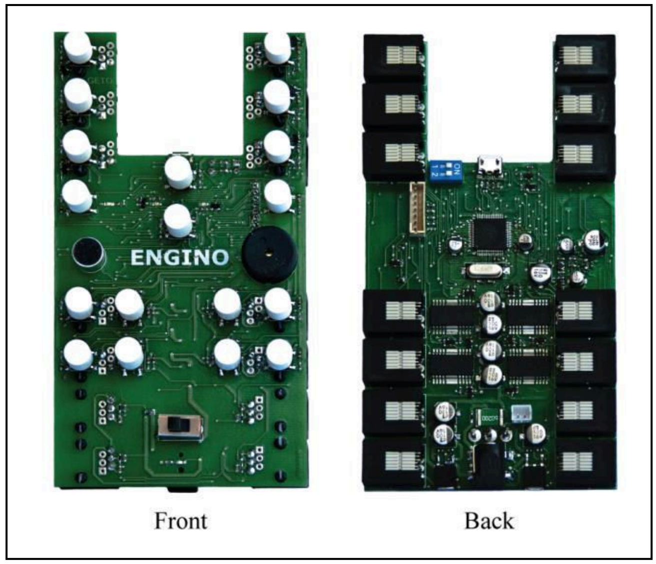

Figure 4. Engino Robotics Platform Controller PCB

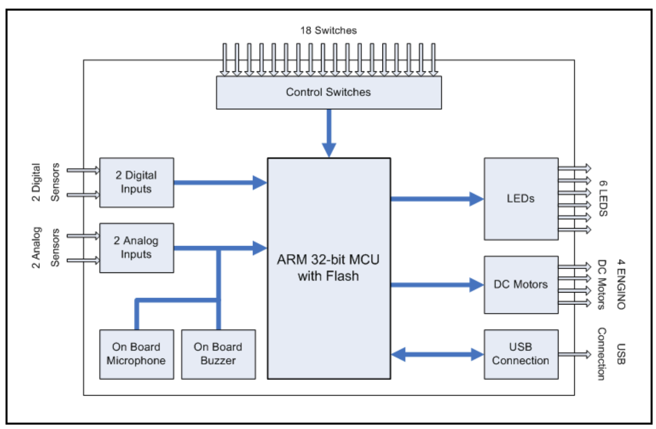

Figure 5. ERP Block Diagram

The light sensor is a photo sensor that detects the presence of light. It sends voltage to the analog input of the ERP, which is then converted to a 12-bit digital value by the ADC channel of the micro-processor. The 12-bit value represents the amount of light that is received by the sensor.

2) *Motors*

The ERP has four 9-Volt outputs that are connected to the micro-processor. These outputs can be used to control the ERP motors. Each output line is connected to an H-Bridge, such that the processor can control the motors to move forwards, backwards, or switch off.

3) LEDs

The ERP facilitates six 3-Volt outputs in order to be able to control LED functionality. Each LED output is connected directly to the micro-processor in order to be able to switch the LEDs on or off, by using software instructions.

C. Functionality

As mentioned above, the ERP functions around a RTOS. There is an event listener for each possible user interaction with the device (i.e. for each button on the device) that will fire up a corresponding action. An action executes a task (i.e. turning a motor) for as long as the button (or switch) that fired the action is on. Once the switch is turned off, the action stops. A small software library has been developed in C which includes all the actions of the device and all possible inputs. This library is used together with the RTOS. In parallel of the control processes, there are other processes running on the device, such as the USB Human Interface Device (HID) driver, which allows controlling the device in real-time from a host computer. The RTOS includes a simple round-robin scheduler for running all the processes. The interrupts between these processes are done in a micro-seconds scale and thus all processes appear to run in parallel.

Apart from the real-time interactions of the user with the control, there is also a programming mode and a play mode on the device. The programming mode allows the user to record every interaction and store it in the embedded flash memory of the device. The play mode can repeat the corresponding actions at any point in the future. During the programming mode, any idle time is not recorded. If the users want to have idle time in their program, they can use the idle switch on the device. The idle switch (or delay time) is used as an event that has no action and it can be recorded in memory.

Due to the real-time element of the control and the interactions from the user, the machine states are non-deterministic. Thus, the ERP is a non-deterministic finite state machine. In other words, there are many states that can be followed from each possible state of the control, and these states depend on the actions of the user. Because of the memory constrains, the user actions have a limitation when they are recorded in the programming mode. That is, the number of times an action can be recorded during the programming mode is a finite number which is chosen according to the memory size of the device. Nevertheless, this number is large and adequate for the applications of the ERP.

IV. Case Studies

The ERP's main objective is to be used as tool for teaching robotics, control systems and technology course to primary and secondary level students as well as to be integrated as educational tool within the teaching and learning process to achieve instructional goals. It allows students (users) to automate robots that they can build using the traditional Engino components.

To automate robots the user must be able to program the ERP. As described earlier, programming the ERP can be done in two different ways:

- Manually: using the buttons that are on the ERP box.
- Through the EGPI: The user can use either the simulated box or the graphical programming interface to create programs and then download them on the ERP.

In this section two case studies will be presented: (1) Manual and Simulated programming of a simple automated system, (2) Programming with the Graphical User Interface environment for more advanced operations with the sensors.

A. Manual and Simulated programming of a simple automated system

This is a simple automated system that shows an example of a system that can be programmed by young students (primary school). It is an example that emulates the behavior of a traffic arrow used on many roads around the world. Students are familiar with this example as most of them have already seen it operate in the real world. It is a simple arrow that lights up to point on a specific direction. It does not use sensors or motors. It is simple enough so it can be programmed manually or through the simulation.

1) Description of the traffic arrow operation

Six (6) LEDs are used for this example. The LEDs switch ON in sequence to indicate the direction that the arrow points to. More specifically, the operations are: LED 1 turns on for 1 second, then LED 2 turns on for 1 second, then LED 3 turns on for 1 second, then LEDs 4 and 5 turn on for 1 second, and finally LED 6 turns on for 1 second. As soon as all LEDs are turned on, the system remains idle (LEDs ON) for 2 seconds. The process loops to the beginning.

2) Programming the traffic arrow operation

Manual Programming: In manual programming, the user is able to use the buttons on the ERP to record a sequence of instructions. To turn the ERP into programming mode, the PROGRAM button is pressed for 2 seconds. An indicator LED will start blinking once the device is set into programming mode. Every user's action is then recorded and stored into the embedded memory of the device. For the case study of the traffic arrow operation, the user will simply execute the following instructions: Press PROGRAM button for 2 seconds. Press button 1 (to switch on LED 1) for 1 second. Press button 2 (to switch on LED 2) for one second. Press button 3 for one second. Press buttons 4 and 5 simultaneously for 1 second to switch on LEDs 4 and 5. Press button 6 for 1 second. Press IDLE button for 2 seconds. Press PROGRAM button to finish recording. Press PLAY button to play the recorded sequence. Note here that the PLAY button can be pressed again to stop the sequence, or long-pressed to enter into repeat mode (in order to keep repeating the recorded sequence until the play button is pressed again).

Simulated Programming: In simulated programming the user can use the simulation interface that is provided on the host computer instead of using the buttons of the ERP. The simulation is a visual representation of the ERP with its corresponding buttons. The user can use the simulation while the device is connected with the USB driver, or when the device is not present. Either the device is present or not, the user can program the simulation in the same way as they would do on the ERP and store the sequence of instructions on the host computer. In the case where the user has to press multiple buttons at the same time, there is a selection area for each button and a MASTER Button that will press all selected buttons on the screen at the same time. The user can load and send the program to the device at a later stage, when the ERP is connected to the USB driver. Once the program is sent to the device, the user can press the PLAY button on the ERP to start executing the program. An advantage of the simulated programming is that the program can be stored on any computer and can be transferred to any other computer through the Internet or any other storage device.

B. Programming of a mobile robot using the graphical programming interface

Fig. 6 shows the robot that was used for our secondary case study. This mobile robot imitates the behavior of a cockroach. This example is more complex. It shows one of the applications where ERP can be applied as educational tool by secondary education students. This more complex application uses sensors and motors for its operation.

1) Mission description of the Cockroach Mobile Robot

This robot's mission is to emulate the behavior of a cockroach. The robot will remain inactive until the light dims (simulating how the cockroaches hide when the light is on). As soon as the light switches on the robot will start moving forward. When it detects an object at the front, it will stop, back up and turn right. It will then move forward again. When the robot backs up, the LEDs at the front of the robot will be ON. The robot will continue this operation until the lights turn on again. As soon as the lights turn on, the robot will move forward and stop (simulating how cockroaches run to hide as soon as the lights turn on).

2) Specifications of the Cockroach Mobile Robot

The cockroach Mobile Robot Uses 4 motors to control the 4 wheels, a contact sensor at the front to detect obstacles, a light sensor to detect the presence of light and 2 LEDs for interaction. Instead of the contact sensor one can use the IR distance sensor to detect objects before it makes contact. But since this mobile robot simulates the operation of a cockroach, it was better to use the contact sensor as cockroach makes contact with objects before they change directions.

3) Programming the Cockroach Mobile Robot

Since this is a robot that uses feedback from the sensors, it was more appropriate to use the graphical programming interface for its programming. Manual and Simulated programming are not possible in this case. Fig. 7 shows the complete program of this mobile robot.

V. CONCLUSION

The new robotic controller (the ERP), can be used in education for teaching young students the programming principles of robotics. The controller can be used to program custom made robots in manual, simulated, and graphical modes. Moreover, the ERP fills the gap between young students and advanced students in education, in contrast to other robotic controllers that already exist in educational systems. In the future, our aim is to improve the ERP in both hardware and software aspects, by introducing wireless technology, more advanced programming structures, and more programming control modules.

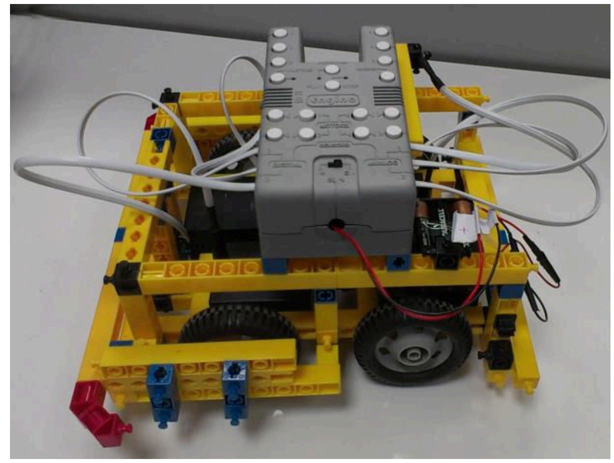

Figure 6. Cockroach Robot

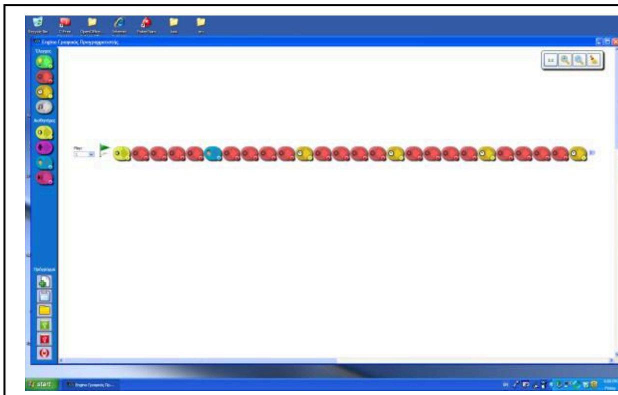

Figure 7. Graphical program for the Cockroach Robot

ACKNOWLEDGMENT

This project was funded by the Cyprus Research Promotion Foundation (RPF).

REFERENCES

[1] G. A. Demetriou, "Mobile Robotics in Education and Research," in Mobile Robots Current Trends, Z. Gacovski, Ed.Rijeka, Croatia: InTech, 2011, pp. 27-48.

[2] O. Miglino, H.H. Lund and M. Cardaci, M. "Robotics as an Educational Tool". *Journal of Interactive Learning Research.* Charlottesville, VA: AACE, 1999, 10(1), 25-47

[3] S. Papert. "Mindstorms: Children, Computers, and Powerful Ideas". NY, New York: Basic Books, 1980.

[4] J.M. Chambers and M. Carbonaro. "Designing, Developing, and Implementing a Course on LEGO Robotics for Technology Teacher Education". *Journal of Technology and Teacher Education*, Norfolk, VA: AACE, 2003, 11 (2), 209-241.

[5] D. Williams, Y. Ma, and L. Prejean. "A Preliminary Study Exploring the Use of Fictional Narrative in Robotics Activities". *Journal of Computers in Mathematics and Science Teaching,* . Chesapeake, VA: AACE, 2010, *29*(1), 51-71

[6] A. Bauerle and M. Gallagher. "Toying With Technology: Bridging the Gap Between Education and Engineering". In C. Crawford et al. (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference*. Chesapeake, VA: AACE, 2003, pp. 3538-3541.

[7] S. Papert. "Mindstorms: children, computers, and powerful ideas"(2nd ed.). New York, NY: BasicBooks, 1993.
[8] Egg-Box User Guide. (4th edition). [On-line]. Available: http://www.tts-group.co.uk/_RMVirtual/Media/Downloads/EggBox_UserGuide.pdf [July 13, 2012].
[9] Lego Education WeDo. [On-line]. Available: http://www.legoeducation.us/eng/categories/products/elementary/lego-education-wedo [July 12, 2012].
[10] LEGO Education. [On-line]. Available from: http://www.legoeducation.com [May 7, 2012].
[11] VEX Robotics Design System. [On-line]. Available: http://www.vexrobotics.com [July, 10, 2012].
[12] Medium-density performance line ARM-based 32-bit MCU with 64 or
[13] 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 communication interfaces. [On-line]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00161566.pdf [May 21, 2012].
[14] RTX Real-Time Operating System. [On-line]. Available: http://www.keil.com/rl-arm/kernel.asp [June 1, 2012].