Rapid Prototyping and Implementation of Student Designs Using Simulink and the LEGO MINDSTORMS NXT

J.C. Peyton Jones
Department of Electrical & Computer Engineering, Villanova University

Abstract

The use of embedded / mechatronic systems in teaching is being revolutionized by a) the advent of increasingly powerful yet low-cost computational devices and sensors, and b) by modern Automatic Code Generation tools which allow these devices to be programmed directly from high-level designs - without the difficulties traditionally associated with low level embedded system programming. This paper outlines new tools which enable the students to access the hardware capabilities of the 32-bit LEGO MINDSTORMS NXT brick from within the MATLAB / Simulink environment, and to automatically generate and cross-compile the necessary code for real time autonomous implementation. LEGO hardware I/O is represented in the Simulink design mode as blocks for accessing motors, encoders, push-buttons, ultrasound sensors, light sensors etc. 'External Mode' capability also enables users to communicate with the embedded target at run-time in order to tune controller parameters and/or monitor signals in a highly interactive manner. A series of tutorial examples are used to illustrate the educational opportunities afforded by this new toolchain.

1. Introduction

The use of embedded / mechatronic systems in teaching is being revolutionized by the advent of increasingly powerful computational devices and sensors, and by high-level Automatic Code Generation tools with which to program them. Low-cost yet highly capable computational hardware and sensors have evolved as a natural outgrowth of Moore's law, and the use of embedded computing devices has become nearly ubiquitous. High-profile examples include the Mars Rover and the fully autonomous vehicles of the DARPA Urban Challenge, but there are also many humbler examples encountered daily in the home. The family car, for example, has traditionally been regarded as a predominantly mechanical system, but today a typical car may incorporate more than 50 embedded microprocessors, communicating with each other over in-vehicle networks and optimizing the overall system behavior with respect to safety, performance, fuel economy and emissions.

Embedded systems also have a well established track record in education, particularly in the robotics area. Many researchers have found that student's motivation to learn increases significantly with hands-on robotics-based projects, [1-3]. Others have successfully used robotics as a unifying theme in introductory courses, [4-7], and still others have used robotics as way to attract women into Computer Science, [8]. The increase in embedded computational power, however, is also strongly correlated to programming complexity. This has motivated a move, both in academia and industry, away from low-level programming in assembly language, towards higher level programming in C, or still higher design environments such as MATLAB and Simulink. In particular, modern Automatic Code Generation or 'Rapid Prototyping' tools allow embedded target devices to be programmed directly from high-level system designs without the traditional difficulties associated with low level issues, fixed point computation, or C-language programming. These tools also have the potential to transform the use of robotics or other embedded applications in education, enabling students to undertake more complex and challenging problems while focusing on the high-level pedagogical goals rather than low level issues.

To this end, a number of researchers have sought to develop tools to integrate MATLAB and Simulink with LEGO MINDSTORMS NXT hardware for educational purposes. Behrens et al., for example developed the RWTH toolbox which wirelessly sends commands and receives data to/from a LEGO NXT platform [9]. This 'remote control' approach has the control algorithm running on the host PC with the robot acting primarily as a dumb sensor / actuator. It has been used successfully at Villanova University for a number of years as part of the freshman Introduction to Engineering course, but it is less appropriate for more advanced real-time applications. Though simple, the approach is limited to low bandwidth control applications due to the time varying delays which inevitably occur in the host-target communication channel. An alternative Simulink-based real-time tool was first developed by T. Chikamasa in the form of the Embedded Robot (ECRobot) coder [10], and this has been used by the author and others in earlier projects [11], [12]. However its function-call based architecture does not conform to the normal Simulink Real-Time Workshop (RTW) design process, and it imposes significant constraints on user designs.

As part of a National Science Foundation 'Transforming Undergraduate Education in STEM' (TUES) program award, a more user-friendly Villanova LEGO Real Time target (VU-LRT) blockset has been developed which integrates more seamlessly with the standard RTW design process [13-15]. The project was also sponsored by the MathWorks, and the VU-LRT blockset is available on the MATLAB Central file exchange site [16]. The toolbox has been downloaded over 1000 times, and it has been used in a number of educational projects across a range of different institutions. Recently, however, the MathWorks released their own LEGO MINDSTORMS NXT block library, [17]. Though functionally very similar to VU-LRT, the new product no longer requires the Real Time Toolbox since target support for the NXT, (and for several other low-cost hardware platforms), is now built in to the base MATLAB / Simulink product (version R2012a onwards). Furthermore, the new MathWorks NXT support includes 'External Mode' capability which enables users to communicate with their hardware in order to monitor and / or tune the process while their model code is executing in real time. This paper briefly introduces the new MathWorks NXT target, and the LEGO MINDSTORMS NXT hardware that it is designed to support. A number of laboratory exercises and classroom modules are also briefly outlined to illustrate both the capabilities of the new toolchain, and the educational opportunities that it affords.

2. The LEGO MINDSTORMS NXT: Hardware and Software Alternatives

Although much of this paper is concerned with Simulink-based rapid prototyping and automatic code generation, it is worth first considering the advances in hardware that have given rise to powerful, yet affordable, embedded target devices. The evolution of embedded computing devices is reflected in the wide variety of robot hardware platforms in current or recent use within the STEM community. Many of these devices, such as the Parralax BOE Bot [18], HandyBoard [19], ActivMedia [20], Arduino [21] and first generation LEGO RCX brick [22] for example, are based on 8-bit processors. Typically these machines run at clock speeds of 20 MHz or less, have 32 KB or less of RAM and are hard to program effectively because of finite word length and memory issues. Recently, however, 32-bit machines have become available at very similar cost to their 8-bit predecessors. These include the LEGO MINDSTORMS NXT [23], the Surveyor SRV1 [24], and the Korebot, [25]. The NXT, in particular, is a 32-bit machine with twice as much memory as the older RCX version (Fig. 1). It has Bluetooth communications which allow a master unit to communicate with up to three secondary NXT units, and features 4 input ports and 3 output ports. The output ports can be used to drive *dc* motors and incorporate encoder position feedback, and the input ports can be connected to an increasingly broad variety of sensors available from LEGO MINDSTORMS or from third party suppliers such as HiTechnic and Vernier Software & Technology. The electrical interface to these ports has been published [26], so it is also possible to have students design and integrate their own sensors and actuators. Users can also access the in-built push-buttons, the on-board loudspeaker, and other utilities such as a battery voltage monitor and a timer clock. Mechanically, the NXT integrates seamlessly into the LEGO Technic world, enabling students to implement both the physical as well as the software aspects of their designs.

As outlined in the introduction, advances in embedded hardware have also motivated advances in tools for software development, and the NXT can now be programmed in C, Java, Python, Ada, and variants of these and other languages. Higher level, more graphically oriented programming tools are also available, including NXT-G which is generally targeted at novice programmers, as well as more professional products such as LabView, and Simulink. The MATLAB / Simulink environment which is arguably the most pervasive in the STEM community, is already tightly integrated into the research activities and educational curriculum at Villanova University and other institutions, and Simulink was therefore chosen as the design environment for the project.

Figure 1. The LEGO MINDSTORMS NXT brick and associated peripherals

3. MathWorks Target Support for the LEGO MINDSTORMS NXT

The MathWorks Target Support for the LEGO MINDSTORMS NXT provides a block library and toolchain to enable target implementation of high-level Simulink designs on the NXT brick. The block library defines a high-level interface to NXT hardware for users, as well as the target-specific low-level code and cross-compilation details necessary to implement this in the final executable. The process is illustrated in Fig. 2. The start point is a user design in the form of a Simulink model. When the user selects 'Run on Target Hardware' from the tools menu, Simulink generates the corresponding C code, which is then automatically cross-compiled and linked into an executable, downloaded to the target, and run under a real time OSEK operating system developed for the NXT [27]. As shown in Fig. 2, it is also possible to include code for 'External Mode' communications which enables the user to interact with the NXT target at runtime, (as indicated by the dotted line in the figure). This feature, which was not available with earlier tools such as ECRobot or VU-LRT, is very useful for tuning model parameters during execution and for monitoring data. Further details, and an example, are given in Section 4.2.

From a user-perspective, the build and compilation details shown in Fig. 2 are largely transparent, and target specific features are encapsulated within the user-friendly LEGO MINDSTORMS NXT block library shown in Fig. 3. The user can drag and drop any of these blocks into their design in order to access any of the NXT's in-built features or attached sensors and actuators. A summary of these blocks and their function is as follows:

- Battery – outputs the current battery voltage
- Timer – outputs the time in ms since the model execution began
- Button Button – outputs a 1 if the specified NXT push button is pressed, else 0
- Light Sensor – outputs a measure of the light received by the light sensor
- Sound Sensor – outputs a measure of the sound intensity received by the sound sensor
- Touch Sensor – outputs a 1 if pressed, else 0
- Ultrasonic Sensor – outputs the distance to the closest object in view by the sonar sensor
- Acceleration Sensor – Outputs acceleration data in three axes (x, y and z)
- Gyro Sensor – outputs the rotation rate of the sensor
- Encoder – outputs the number of encoder pulses received as the motor rotates
- DC Motor – sets the applied motor voltage as a percentage of battery volts
- Speaker – sets the frequency and duration of tones driving the internal loudspeaker
- Send via Bluetooth – enables the model to communicate with other Bluetooth devices
- Receive via Bluetooth – enables the model to communicate with other Bluetooth devices

Note that the Bluetooth interface blocks provide a means for transmitting and receiving data between another Bluetooth device and the NXT during runtime. This is a useful feature for communicating between two NXT robots for example, although it is not as flexible as the external mode feature described above.

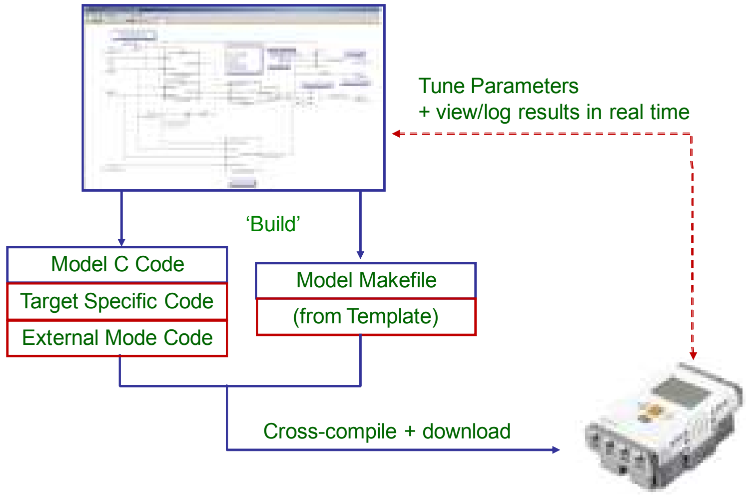

Figure 2. The Rapid Prototyping Process

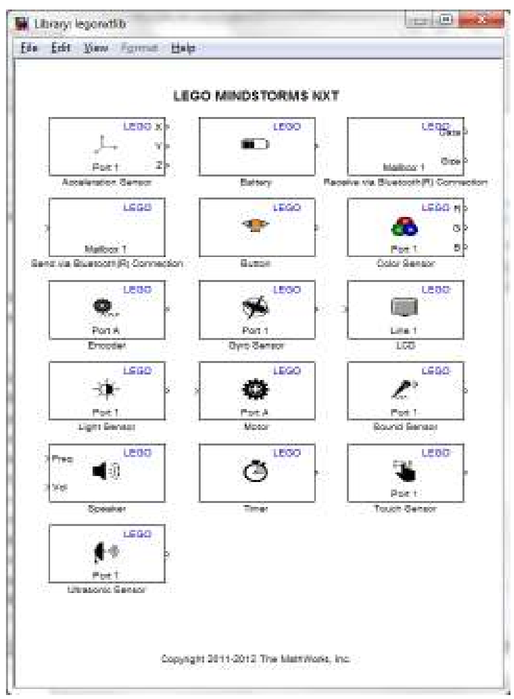

Figure 3. The LEGO MINDSTORMS NXT Block Library

4. Tutorial guides and examples

4.1 *Getting Started*

A new 'getting started' tutorial guide and set of examples has been developed to introduce students to the new MathWorks NXT Real-Time Target. These examples, (which are based on the original VU-LRT user manual), build on one-another as new techniques and concepts are introduced. The first example, illustrated in Fig.4, is intended to verify the correct functioning of all the tools and the successful transformation of a simple model into a real-time executable. The model therefore uses only the built-in button-press and loudspeaker functionality of the NXT, and is designed so that the loudspeaker emits a tone whenever the button is pressed. When the user initiates the run-on-target hardware process, the model is automatically transformed into a C program which is seamlessly cross-compiled and downloaded to the target over a USB cable. The downloaded executable then automatically starts executing on the NXT. Correct execution of the model is easily verified by confirming that a tone is sounded whenever the 'Enter' button is pressed.

The second example, illustrated in Fig. 5, again uses the built-in loudspeaker, but uses sensor inputs from the integrated LEGO motor / rotary encoders in order to modulate the loudspeaker volume and frequency about some constant pre-specified values. When executing in real time, the user can therefore adjust the emitted tone amplitude and frequency by turning a small wheel or dial attached to each motor/encoder device.

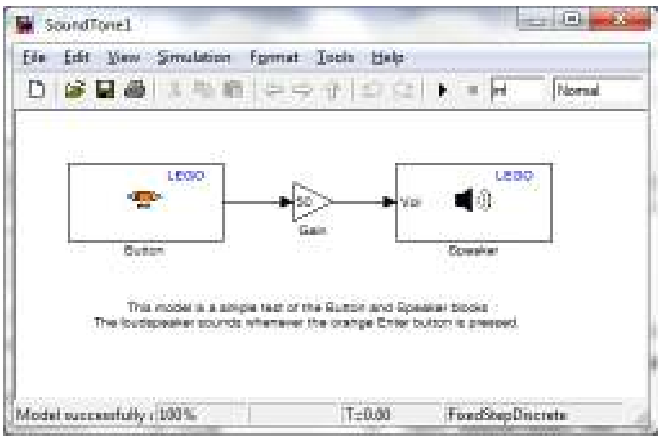

Figure 4. Sound tone on button press

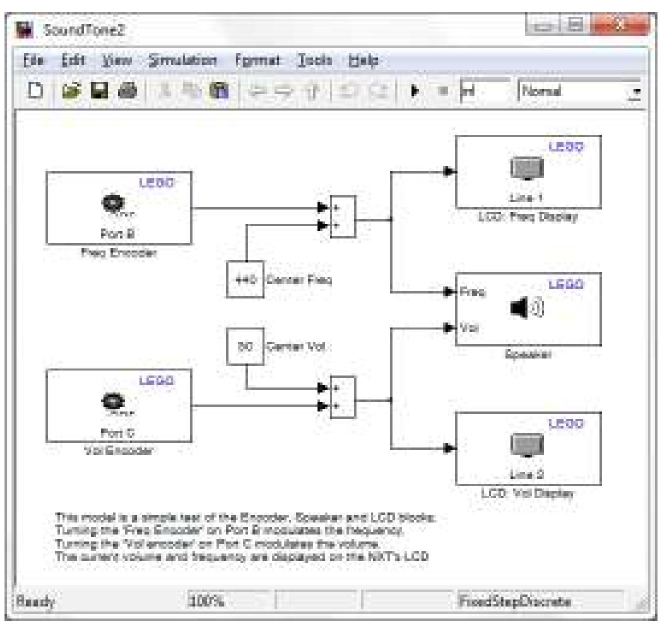

Figure 5. Loudspeaker volume and amplitude modulation

4.2 *Using External Mode*

Simulink's 'External Mode' feature allows users on the host computer to monitor real-time signals, and to modify or 'tune' model parameter values while the compiled code is actually executing on the NXT brick. This provides a valuable and highly interactive insight into the real-time internal workings of the model, and of its behavior as parameter values are changed. The third example provides a simple illustration of these features. The NXT is first 'paired' with the PC host machine so that the two can communicate over Bluetooth during real time code execution. The source model is also configured to select External Mode operation. The effect of this selection is for additional code to be inserted during the automatic code generation phase in order to manage the real time communication process. This communication link is then automatically established when the code starts executing on the NXT target. During execution, if the user makes any changes to the parameters of the original Simulink model, such as changing the Center Frequency value in Fig. 5, for example, then these changes are automatically transmitted to the NXT target, with immediate effect. Students can therefore tune and calibrate their designs in a highly interactive fashion. The External Mode communication link also allows data to be logged from the Target and displayed using one of the Simulink 'Sink' blocks. In the design of Fig. 6, for example, a digital Display block is used to display the current value of the volume variable, and a Scope block is used to display the time history of the 'frequency' signal. Students can also save this data to the MATLAB workspace for post-processing and report writing.

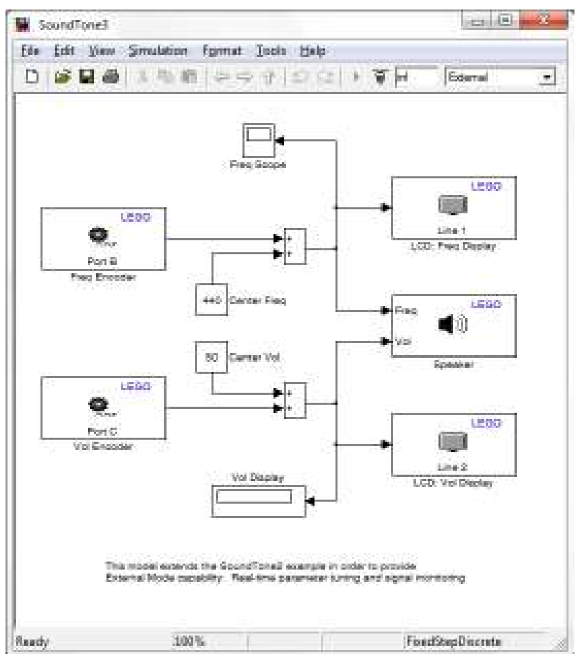

Figure 6. Using External Mode to log real time data to Scope and Display blocks

4.3 *A Simple Line-following Robot with Proportional Feedback*

In the next example, the idea (from the previous exercise) of modulating a base value based on sensor input, is used in a simple feedback control configuration to modulate the left and right wheel speeds of a bi-wheeled robot according to the light sensed by a light sensor. The aim is to track a black line marked on a light background floor. If the received light is less than some pre-defined constant setpoint value, the robot should turn rightwards. Conversely, if the received light is greater than the setpoint the robot should turn to the left. The robot should therefore track the edge of the line, with black to the left, and white to the right.

An initial design is shown in Fig. 7, where the students have to configure the signs of the various summation junctions in order to achieve correct operation. Students then explore the effect of the feedback gain and base speed on the robot's tracking performance and stability. Further exercises implement the same algorithm using embedded MATLAB function blocks, as well as enhancements to this algorithm using a finite state machine structure. In general, these blocks, which allow block behavior to be defined directly in MATLAB code, provide a powerful way to specify complex system behaviors.

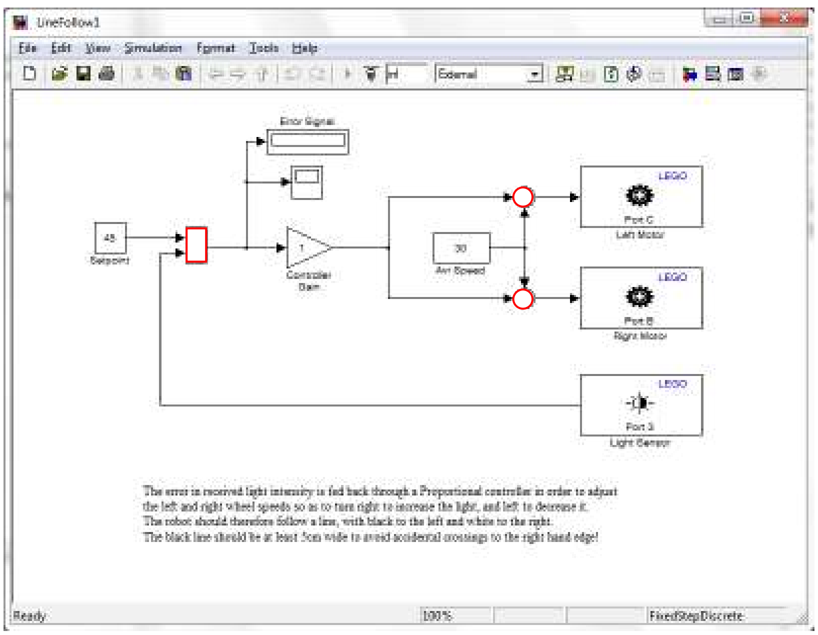

Figure 7. Initial line-following robot feedback control structure

5. Conclusion

Advances in low-cost target hardware, and automatic code generating software tools provide an opportunity for educators to engage students in challenging embedded system applications without the traditional difficulties associated with low-level hardware programming. In particular, the new MathWorks target support package for the LEGO MINDSTORMS NXT which supersedes the initial (and very successful) VU-LRT toolbox, is helping to bring these capabilities to an even wider audience. Several simple examples drawn from a set of introductory laboratory exercises have been presented. These examples demonstrate the ease with which user designs can be implemented in real time hardware on the NXT, and the way in which External Mode capability can be used to interact with the target in real time.

Several more advanced laboratory modules have also been developed, but due to space restrictions, are not reported here. In a senior-level Machine Learning course, for example, students implement a genetic algorithm on the LEGO NXT to enable a spider-like robot to learn how to walk [15]. In another example, taken from a graduate level Digital Control course, a digital speed control system, implemented on the NXT, is used to challenge student preconceptions about the effect of digitizing a simple analog control system. Formal evaluation of these initiatives is underway, though limited by the relatively small class sizes at Villanova University. It is hoped to report on these results in the near future.

Acknowledgements

Support for this project from the National Science Foundation (DUE No. 0837637 [13]), and from the MathWorks Inc is gratefully acknowledged. This work is neither endorsed nor maintained by the LEGO group. MINDSTORMS, NXT, Technic and LEGO are trademarks of the LEGO group.

Bibliography

1. L. Greenwald, and D. Artz, “Teaching artificial intelligence with low cost robots,” In Accessible hands-on artificial intelligence and robotics education, ed. L. Greenwald, Z. Dodds, A. Howard, S. Tejada, and J. Weinberg, pp. 35-41. Technical Report SS-04-01. Menlo Park, CA: AAAI Press, (2004).
2. S. Coradeschi and J. Malec “How to make a challenging AI course enjoyable using the RoboCup soccer simulation system, in RoboCup-98: Robot soccer world cup II: Lecture notes in artificial intelligence, vol. 1604, pp.120-124, ed. M. Asada and H. Kitano. Berlin: Springer, (1999).
3. M. Goldweber, et al. “The use of robots in the undergraduate curriculum: Experience reports,” Panel at 32nd SIGCSE Technical Symposium on Computer Science Education, Charlotte, North Carolina..
4. F. Klassner, “Robotics as a Unifying Theme for Computing Curriculum 2001”, National Science Foundation DUE #0088884, (2001).
5. D. Miller, “Walking Before Running: Filling the Freshman Engineering Gap by Building Mobile Stiquito(TM) Robots”, National Science Foundation DUE #0088158 (2001).
6. N. Chao, “A Low Cost Hands-On Laboratory Experience for Introductory Engineering Students”, National Science Foundation DUE #0125583 (2001).
7. L. Fairchild, “Robots in an Introductory Survey Course in Computer Science”, National Science Foundation DUE #0087963 (2001).
8. N. McNulty, “Understanding Technology through Robots and Multimedia”, National Science Foundation DUE #0088370 (2001).
9. A. Behrens, et al. “MATLAB meets LEGO Mindstorms- A freshman introduction course into practical engineering”, IEEE Trans. on Education, Vol.53, No.2, (2010), 306-317
10. T. Chikamasa, “Embedded coder robot NXT instruction manual”, www.mathworks.com/ matlabcentral/fileexchange/13399/, 2009.

11. McNinch, L. C., Soltan, R. A., Muske, K. R., Ashrafiuon, H., Peyton-Jones, J. C. “An Experimental Mobile Robot Platform for Autonomous Systems Research and Education”, Proceedings of the 17th IASTED International Conference on Robotics and Applications, (2009): 412-418
12. McNinch, L. C., Soltan, R. A., Muske, K. R., Ashrafiuon, H., Peyton-Jones, J. C. “Application of a Coordinated Trajectory Planning and Real-time Obstacle Avoidance Algorithm”. Proceedings of the 2010 American Control Conference, June 30-Jul 2, Baltimore MD, (2010).
13. J.C. Peyton Jones, F. Klassner, S. Kulkarni, C. Nataraj, “Introducing undergraduates to complex systems through rapid-prototyping of low-cost, networked mobile robots”. National Science Foundation DUE No. 0837637.
14. Peyton Jones, J.C., McArthur, C., Young, T. The VU-LEGO Real Time Target: Taking Student Designs to Implementation. Proceedings of the ASEE 2011 Annual Conference & Exposition, June 26-29 2011, Vancouver, Canada. (2011).
15. F. Klassner, K. Lehmer, J.C. Peyton Jones, Genetic Algorithms with LEGO Mindstorms and MATLAB, 25th International Florida Artificial Intelligence Research Society Conference (2012).
16. VU-LEGO Real Time Target, website, http://www.mathworks.com/matlabcentral/fileexchange/29857-vu-lego-real-time-target
17. MathWorks Support for LEGO MINDSTORMS NXT, website: http://www.mathworks.com/academia/lego-mindstorms-nxt-software/
18. Parallax Inc, website: http://www.parallax.com/
19. The Handy Board, website: http://www.handyboard.com/
20. Mobile Robots Inc, website: http://www.mobilerobots.com/
21. Arduino, website: http://www.arduino.cc/
22. LEGO MINDSTORMS RCX, website: http://www.lego.com/
23. LEGO MINDSTORMS NXT, website: http://www.lego.com/
24. Surveyor SRV-1, website, http://www.surveyor.com/SRV_info.html
25. Korebot II, website, http://www.k-team.com/mobile-robotics-products/korebot-ii
26. LEGO MINDSTORMS Developer kit, website, http://mindstorms.lego.com/en-us/support/files/default.aspx
27. NXT OSEK/JSP, website, http://lejos-osek.sourceforge.net/