Developing Laboratory Based Electronics/Software Projects using a LEGO Mindstorms Programmable Brick

Donald Wilcher
MaDon
Introduction
One of the challenges confronted by EC (Electrical – Computer) engineering educators is developing electronics and software laboratory projects that will convey design methods used in industry to freshman students in a user-friendly manner. Based on the freshman student's background he or she may not be familiar with industrial design practices because of a variety of reasons: lack of cooperative or internship positions available to first year students or not exposed to electrical laboratory experiments in physics or circuit design and analysis. Therefore, it is the mandate of the EC engineering educator to devise innovate methods to illustrate engineering design techniques to the freshman ECE (Electrical-Computer)engineering student without overwhelming them with Calculus or Differential Equations. By using an innovative toy created by the LEGO™ company as a pedagogical tool for engineering instruction, sophisticated laboratory projects in electronics and software programming can be developed. A hands-on approach to learning digital/analog circuits, microcontrollers, I/O interfacing, and computer language programming are key educational elements needed by the EC engineering student within the undergraduate career. The LEGO programmable brick or P-Brick allows the freshman student to seamlessly learn these key areas in electrical engineering through play. The LEGO P-Brick allows a range of complexity to be design by the EC engineering educator for creating laboratory based electronics/software projects. This paper will discuss how these engineering laboratory projects can be develop using the design methods of Hatley/Pirbhai, structural analysis, and state machine development. Mechatronics based machines and devices will serve as the core pedagogical multidisciplinary tool of instruction for development of the LEGO P-Brick based projects.

Understanding an Electronic Control Unit Architecture: Hately/Pirbhai Method
The LEGO P-Brick although a child's play toy is a really a powerful 8 bit microcontroller based electronic control unit. It has a Hitachi H8/3929 series microcontroller capable of running at a clock rate between 10 and 16MHz [1]. The Hitachi H8/3929 microcontroller contains 16 Kbytes of on board ROM (read only memory) and 512 bytes of RAM (random access memory). Embedded within the ROM is an algorithm use for downloading the firmware, and displaying data on the LCD (liquid crystal display). In addition to the internal RAM, there is 32 Kbytes of external or off board RAM inside of the LEGO P-Brick as well. The internal RAM is used for storing the firmware of the P-Brick and the off board memory allocated for user programs. These internal circuit blocks allow the LEGO P-Brick to provide programmable control functions to the attached I/O wired to it. Therefore, the programmable brick is a small Electronic Control Unit (ECU) capable of providing unique instructional opportunities for freshman EC engineering students to investigate configurable hardware architectures. Configurable hardware in this context meaning the ability to add external I/O circuits to the programmable brick with embedded software to process and control these interfaces based on coded behavior. To assist the student in developing such ECU architectures, a design methodology must be learned.

Fundamentally, the first step in developing an ECU architecture is to create a design model. Most industries use models for purposes such as studying requirements for system design, examining feasibility and manufacturability of new products, and determining how to build an actual system [2]. The Hately/Pirbhai method allows architectural models to be built for the purpose of
capturing and managing system requirements. By establishing a framework of the target system, a roadmap for developing processes and providing a cue for what models to be built and how to build them is necessary [2]. Capturing requirements for the product development is basically defining features of the system and the control functions that will execute them. To document these functional objectives requires the designer to construct a paper model of the system. Figure 1 shows the Hately/Pirbhai System Architecture model for documenting functional requirements. The model consists of five blocks:

- Main Functions – Activities the core processor must do.
- User Interface - Parts of a system that interact with user.
- Input - Functions and subsystems that provide input data for the target system to use.
- Support Functions – Any function or subsystems that provide assistance to the rest of the system to keep it running.
- Output – Functions and subsystems that are controlled by the target system.

With the model built, features of the target system can be mapped into these five architectural categories. The EC engineering educator can instruct the student to develop a project book or word processor file with this architecture model serving as page one of their design documentation. The model in Systems Engineering terminology thereby becomes the Level 0 or the initial concept development stage for designing the target ECU. Immediately, the student should be able to see a direct link between this architectural model and LEGO P-Brick's physical I/O structure. Now the ECU lab project using a LEGO P-Brick can be design by the EC engineering educator with an intended instructional objective in mind.

The next step in constructing a LEGO P-Brick ECU Architecture lab project is to create a context diagram or Level 1 model of the intended system. A context diagram is a pictorial block drawing showing the required inputs and outputs that the ECU is to manage. As an illustrated example, a simple motor controller is the target system and instructional aid in discussing ECU architecture design for the student's lab project. Figure 2 shows a context diagram of the LEGO P-Brick motor controller. The LEGO P-Brick controller's function is to turn on a small dc motor using a SPST (single pole - single throw) switch and a similar switch to turn the electromechanical load off. The student would be required to develop this Level 1 design capturing the electrical I/O signals necessary for controls management of the motor. Therefore, this design model would be included with the Level 0 documentation discussed in the previous paragraph. As shown in Figure 2, the context diagram is a top level sketch of electrical I/O the LEGO P-Brick will manage.

The final document development step require in the LEGO P-Brick Architecture lab is to create the electrical circuit blocks responsible for providing input/output signals. Figure 3 shows the Level 2 architecture design for the LEGO P-Brick motor controller with the electrical I/O circuit blocks. The objective behind the design is to show in detail the intended electrical components that will interface to the LEGO P-Brick. This design exercise will allow the student to select physical components that will interface to the LEGO P-Brick when performing the hands-on lab portion of the architectural design project. The Level 2 architecture design will also provide an engineering guide for the software development phase of the project as well. Although this example illustrates I/O interfacing using electrical components like switches and a motor, sophisticated digital switching and analog sensing along with LEDs or incandescent lamps can easily be substituted as well. The touch sensor used in the LEGO Mindstorms Robotics Invention
System (RIS) can easily be replaced with a DC electronic switch quite easily. The schematic circuit shown in Figure 4 can be breadboarded easily and then attached to the target input of the LEGO P-Brick. Using customized DC electronic switching circuits (hardware add-ons) should encourage the student to learn by inquiry [3]. The Electronics Learning Lab from Radio Shack is idea for building additional electronics I/O circuit blocks and controls projects for the freshman EC engineering student as well. The learning lab has two workbooks: analog and digital. In the pages of the two workbooks are source of practical electronic circuits that can be integrated into the ECU Architecture lab project quite easily. In addition to the workbooks, the learning lab includes a wealth of the analog and digital integrated circuits (IC), transistors (both bipolar and MOSFET[Metal Oxide Semiconductor Field Effect Transistor] devices), and passive components (resistors and capacitors). Pre-cut wire is included with the lab as to expedite the prototyping of an electronics I/O circuit. To utilize this prototyping station into the ECU Architecture Lab would consist of the EC engineering educator providing systems requirements to meet a particular functional objective thereby requiring the student to add an existing circuit from either the analog or digital workbook. This mini lab exercise will allow the student to research possible circuit blocks for creating flexible LEGO P-Brick/ECU architectures.

The key to creating an ECU architectural lab project using the Hately/Pirhbai method is interactivity. Another important criterion of an engineering education technology is the degree to which it encourages "interaction" between the learner (the student) and the ideas embodied by the engineering education technology [3]. Using the LEGO P-Brick is successful as a pedagogical tool because the student is engaged in expressing their systems ideas and knowledge directly with the material, evaluating their design concepts without the need for intermediary representations [3]. Using the Hately/Pirhbai method along with the LEGO P-Brick will allow the EC engineering educator to develop a laboratory experience that can meet some of the 13 objectives for engineering instructional laboratories discussed at an ABET January 2002 workshop [4]. Three objectives of the 13 discussed at the workshop that the systems architectural lab met the requirements of are Models, Experiment, and Design. By decomposing the architectural levels, the LEGO P-Brick/ECU lab allow the students to use a structural analysis technique for understanding system interaction to the physical world as represented by the ECU in its surrounding industrial environment.

**Electronics Learning and Sensor Labs**

The Electronics Learning Lab discussed earlier can provide a wealth of instructional projects in I/O interfacing to the LEGO P-Brick. Laboratory projects can be develop whereby additional electronic circuits can be use to solve a particular Proof Of Concept (POC) study [5]. For example, an analog to digital sensor (ADS) simulator can be built on an electronics learning lab breadboard to control a LEGO P-Brick based robot. A potentiometer can be use as a variable sensor hardware to an operational amplifier (op-amp) configure as a comparator circuit. When the input voltage generated by the potentiometer wired to the non-inverting pin of the op-amp (IC) is greater than an established reference voltage applied at the inverting pin of the IC, the output voltage swings to the positive rail of the circuit. A silicon diode interface circuit (SDIC) will convert the voltage to a suitable current level the LEGO P-Brick's input circuit will process and drive the Robot Digital Assistant's (RDA) mechanical mechanism [5]. Figure 5 shows the systems block diagram of the ADS simulator. The EC engineering educator can develop a laboratory
project whereby a POC design statement and a system block diagram is provided to the freshman student. With the functional requirements described in the POC design statement, the student will design the circuit, gather electronic components, pre-cut wires, and breadboard it using the electronics learning lab, validating the results of the simulator experiment. An additional laboratory project the EC engineering educator can create is to use the same requirements but prototype the simulator using an equivalent Sensor Lab. The Radio Shack Sensor Lab is a dedicated electronics learning prototyping tool whereby a workbook with a wealth of circuits are described in detail. A handful of integrated circuits, resistors, capacitors, several sensors, and pre-cut wire come packaged with the kit. The simulator project can be modified whereby the focus of interest is the performance of the ADS circuit. The LEGO P-Brick will be used as a data logger capturing the ADS signal received at one of its inputs. To analyze the data, Robolab™ software can be used to capture the results in either direct or download mode to a "xy" plot screen [6]. Again, the Hately/Pirhbai method and structural analysis should be strongly emphasized to the student as design tools for completing the assign laboratory project from the EC engineering educator.

State Machine/Software Development and Reactive Controls

Traditional flow charts are used to define the logic and organization of how a computer program will work within an embedded controller or desktop workstation. The concern that software developers grappled with on flow charts are the lack of circuit block I/O definition and interfacing with the target intelligent system. State machines provide a graphical representation of the circuit block I/O interface functions within the embedded environment's software layer [5]. By using state machines, reactive controls can embedded into a mechatronics based device or system using software. The LEGO P-Brick, RCX (robot command explorer) compatible software, and a basic circuit block I/O interface can be used to demonstrate to the freshman EC engineering student reactive controls design and state machine/software development methods. The reactive methodology is a behavioral action characterized by triggered events of external conditions (i.e., sensor readings) or internal conditions (program execution status) [3]. State machines are pictorial reactive control models that show triggered events based on conditions related to environmental physical stimulus. Once a desired behavior of the mechatronics based system, say a robot, has been identified, a state machine can be created to capture the reactive control graphically. To write software would be a matter of converting each event and state to their equivalent programming command code language. An illustration of this pedagogy tool in reactive controls development can be demonstrated using a LEGO P-Brick robot known as V-Bot (voltage robot) [5]. The laboratory project administer by the ECE Educator to the freshman EC engineering student is to build a robot with the ability to produce a +5VDC output source upon the detection of its’ arm being "slap". Figure 6 shows the systems block diagram of V-Bot. One of the requirements of the laboratory project is for the student to develop the system block diagram in Figure 6 keeping with the engineering practice of building a documentation project book. The student will then be instructed to draw a state machine for the reactive controls with the following functional requirements:

- With the robot at rest, motors A, C, and B are off.
- By releasing the switch attached to the robot’s arm, motor C will turn on for 1 second.
- After 1 second elapses, motor A will turn on for 4 seconds.
• After 4 seconds elapses, motor C should reverse direction for 1 second.
• After 1 second elapses, motor B will turn on for 20 seconds.
• After 20 seconds elapses, motors A, C, and B are off.

To draw a state machine, a set of symbols is used to capture the states and transitions of the LEGO P-Brick's embedded behavior. Circles or ovals represent states with arrows defining transitions. The state machine for the given functional requirement example of the V-Bot laboratory project is shown in Figure 7. The next step of this project is to convert the state machine into compatible RCX software that will be embedded into the LEGO P-Brick. The LEGO Mindstorms RIS comes with the RCX code software or equivalent controls programming software like Not Quite C (NQC™) or Robolab™ can be use to embedded the reactive controls behavior into the P-Brick. With the code embedded into the programmable brick, the software can be validated. Writing the code either in RCX code, NQC or Robolab™ or appropriate computer programming language is the choice of the EC engineering educator.

Summary

This paper provided example laboratory projects demonstrating engineering design methods in electronics and software development using the LEGO programmable brick. The objective of this paper is to illustrate the ease in which the P-Brick can be use to stimulate logical thought processes for the freshman EC engineering student in the design, build, and test of mechatronics based architectures using Hately/Pirhbai and structural analysis methods used in industry. By using the LEGO P-Brick, advanced concepts of assembly language programming, microcontrollers, and circuit analysis can seamlessly be integrated into the student's EC engineering curriculum. It is hoped that these laboratory based projects will excite the EC engineering educator to explore developing other dynamic experiments in embedded controller applications like mechatronics based Real Time Systems (RTS), robotics based instrumentation, and intelligent machines using the LEGO P-Brick. Future work in this area of engineering instructional laboratories development consists of the author developing the previously mentioned laboratory projects via hands-on research in these specialized technical areas. The outcome of this research & development effort in EC laboratory projects will be a McGraw-Hill book written by the author to be published in the fall of 2003.

References


Biography
DONALD WILCHER is a technical author, engineering educator, research engineer, and owner of MaDon: an Engineering & Technology Curriculum Development and Consulting company that specializes in Robotics, HW/SW tools, and Engineering Laboratory Instruction Development. He has 17 years of Automotive industry experience in Electronics HW, SW test & development, and systems engineering.
Figure 1 Hately/Pirhbai System Architecture Model

Figure 2 Context Diagram (Level 1) of the LEGO P-Brick Motor Controller
Figure 4 Level 2 Architecture Design of the LEGO P-Brick Motor Controller
Figure 3  Level 2 Architecture Design for the LEGO P-Brick Motor Controller

Figure 4 Homebrew Digital Sensor using basic SPST electrical switch and 1K resistor
Figure 5 Systems Block diagram of the ADS Simulator

Figure 6 Systems Block diagram of V-Bot
Figure 7 State machine for V-Bot software development