Seattle
Making Value for Society
122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Paper ID #12115

BYOE: Student Designed Advanced Laboratories for Embedded Computing Concepts, Hardware, and Design

Dr. Harry Courtney Powell, University of Virginia

Harry Powell is an Associate Professor of Electrical and Computer Engineering in the Charles L. Brown Department of Electrical and Computer Engineering at the University of Virginia. After receiving a Bachelor's Degree in Electrical Engineering in1978 he was an active research and design engineer, focusing on automation, embedded systems, remote control, and electronic/mechanical co-design techniques, holding 16 patents in these areas. Returning to academia, he earned a PhD in Electrical and Computer Engineering in 2011 at the University of Virginia. His current research interests include machine learning, embedded systems, electrical power systems, and engineering education.

Prof. Joanne Bechta Dugan, University of Virginia

Joanne Bechta Dugan is Professor of Electrical and Computer Engineering and the Director of the Computer Engineering Programs at the University of Virginia. Her research focuses on probabilistic assessment of the dependability of computer-based systems. She has developed the dynamic fault tree model, which extends the applicability of fault tree analysis to computer systems. Dugan holds a B.A. degree in Mathematics and Computer Science from La Salle University, and M.S. and PhD degrees in Electrical Engineering from Duke University.

BYOE: Experiential Based Coursework for Advanced Embedded Computing Concepts and Design

Presenter information:

Feel free to contact the presenter for help or ideas in using these systems in your own classroom:

Prof. Harry C. Powell, hcp7ad@virginia.edu
Department of Electrical and Computer Engineering
University of Virginia

Summary

This presentation demonstrates several experiments developed in two different half-courses that accompany the one-semester capstone class. These half-courses allow students to explore embedded system interfacing or higher-level embedded concepts while planning and investigating ideas for their capstone project.

Our objective in this presentation is to give other educators working with courses in Embedded Computing a description of our laboratory experiments that have been shown to provide students with a contextual basis for the relevance of the in-lab experience[1]. We will describe these experiments in a high level way and show how they build on earlier BYOE presentation material from the University of Virginia. In showing how we structured these experimental courses, along with the actual experiments, we provide insight on how individual institutions may apply them to their own courseware. We can supply these materials to other instutions, or assist them in fabricating their own. Hardware, example code and suggested experiments are available.

Pedagogical Context

The *Introduction to Embedded Computing* course has been a popular one within Electrical and Computer Engineering at the University of Virginia[2]. Our techniques of blending concepts from across the entire curriculum within the framework of experiential learning has gratifyingly led to demand for more coursework within this area. We endeavored to meet this demand with several exploratory laboratory based half-courses, giving students a range of topics to explore and additionally allowing us to assess the efficacy of future course development and overall direction in a small experimental fashion. A further benefit is that the structure of these courses provides valuable experience to the students in preparation for the completion of their capstone designs. In one course, students are asked to design a new experiment for the introductory embedded systems class; two such experiments are included in the presentation. In another course, students work at a higher level of abstraction to design and implement a simulated traffic system, with autonomous vehicles that obey traffic signals, stay on the road and maintain a safe following distance.

Design your own experiment

The "*Design Your Own Embedded Experiment"* was envisioned as providing a mechanism for developing future experiments for the evolving landscape of our embedded computing course offerings. The requirements for this course were straightforward. Students were to develop a header board that could be employed in future variations of our embedded computing courses that should be compatible with our existing hardware platform, the *MSP430 Launchpad*™ from Texas Instruments Inc[3]. This platform is very low in cost and is employed in our prerequisite courses as described elsewhere[2].

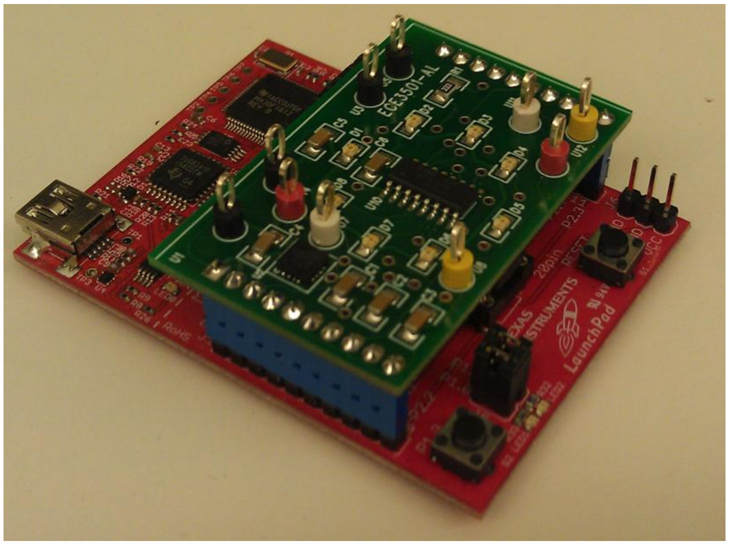

Figure 1 Typical Launchpad and header Board

A typical Launchpad and companion header board from our introductory course is shown in Figure 1. It is useful to note that this board was designed by one of the authors and is locally manufactured. Students were expected to design a compatible board of this relative complexity. A further requirement was that this board should be of a scope that an experiment could be configured that would illustrate concepts from embedded computing as well as at least one concept from across the rest of the ECE curriculum. Students were required to write software that could be used as the basis for an experiment of the general context and difficulty used in the introductory course, and to submit a proposal for write-up that could be used as the handout for this experiment in subsequent introductory course.

Another distinguishing feature of our experiments is to provide visibility of key signals, including not only the digital signals from the microcontroller, but the external signals as well. This is in keeping with our broad goal of understanding an experiment from several viewpoints, and keeps the concept of an embedded system as the sum of disparate components, each of which must be designed to function in concert with the others.

The CAD software employed was *MultiSim*™ and *UltiBoard* ™ from National Instruments Inc[4]. This allowed us to leverage existing student expertise, and is the software currently being employed in our new *Fundamentals of Electrical Engineering* courses.

We will demonstrate 2 of the student designs that are representative of the class results.

These experiments are:

- Dual stepper motor control for controlling an "Etch-a-Sketch"

- Industrial communications using RS485 networks

The dual stepper motor control experiment is designed to give students expertise in working with this important class of electro-mechanical component, as well as experience in writing software that can execute common commands used with such a device, i.e. ramping up and down in speed tracking multiple axes and stopping at a specified position or after a pre-determined number of steps.

The student designed board is shown in 2. This circuit board is designed with readily available commercial components and are of size formats that allow students with little soldering experience to assemble their own systems. In spite of the relative simplicity of the design, it is capable of a high step rates and can drive modest motors of NEMA size 23 or smaller. It is also capable of micro-stepping. In order to give the experiment a "fun factor" it was conceived as a mechanism for drawing simple shapes on an *"Etch-a-Sketch"*[5]. The students designed a simple gear reduction mechanism in our maker space and adapted it the knob shafts on the *"Etch-a Sketch"*, as well as the stepper motor. The design of the system was conceived to be flexible and allow a wide array of experiments to be conducted that required 2 axis control. Other experiments that might be contemplated include such devices as a mobile robot, pen-plotter, or combined with other modules to be used as a simple CNC device. Additionally it may be of interest to mechanical engineering departments for mechatronic experiments. We envision that there can be "paths" through this experiment that would be of interest to electrical engineering, computer engineering, computer science, or mechanical engineering depending on the depth of material provided to students in each scenario.

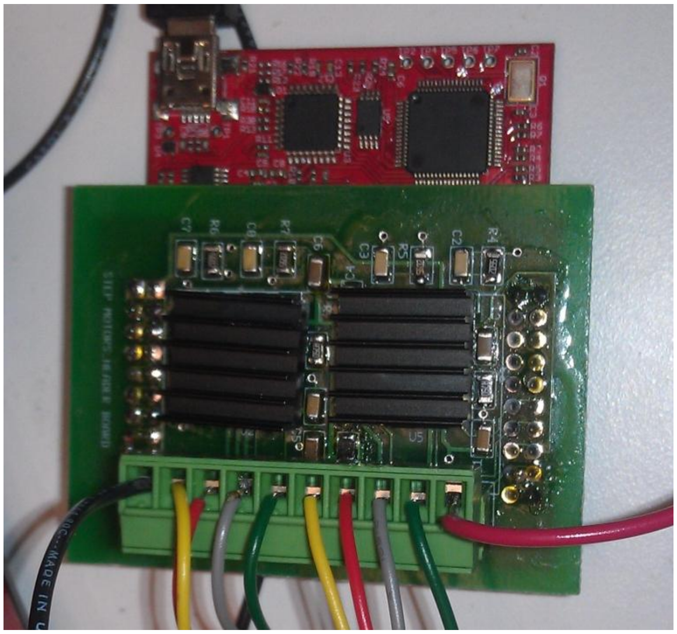

Figure 2 Dual Stepper Control

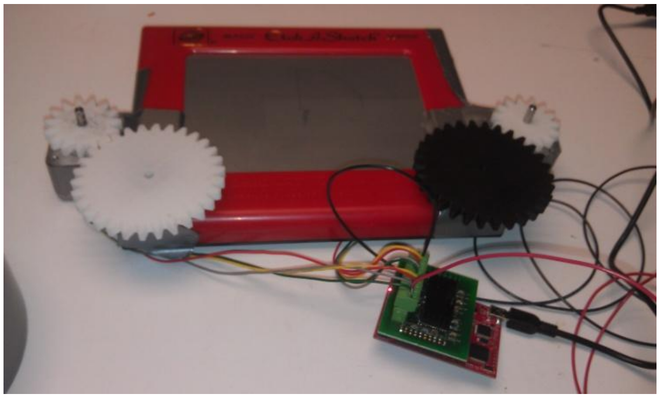

Figure 3 Complete Assembly

The industrial communications experiment was conceived of as one in which the students could gain experience in working with real-world factory communications systems, that frequently employ the RS485 differential transmission scheme. Additionally, the students included a mechanism for optionally varying the value of the line termination resistors. This allows the experiment to be used to convey concepts from E&M fields as well as traditional computer engineering material.

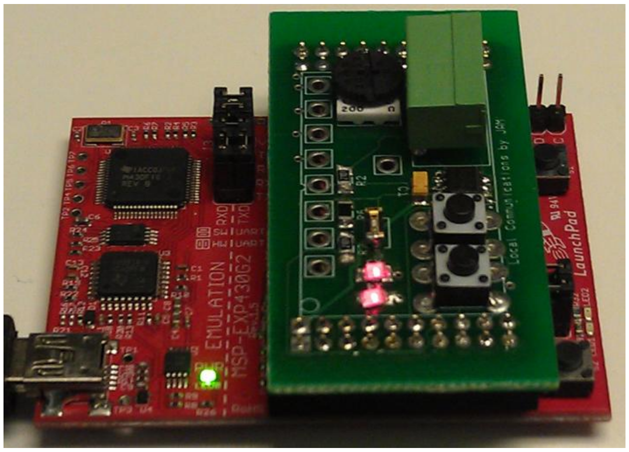

Figure 4 RS485 Interface

The student-designed board for this experiment is shown in Figure 4 . This board is designed for a bi-directional interface, and parallel connection of up to as many as 128 units. This is an accepted standard and is used in a diverse assortment of applications and protocols such as *Modbus*, *Profibus*, vehicular communications, and certain types of SCSI drive interfaces[6]. Thus it may find applications in experiments for mechanical engineering as well as electrical and computer engineering. Additionally, the students wrote a basic low-level library that allows communications from one board to the next to be easily accomplished. Manchester phase encoding was employed as the lowest level of encoding information which allows students to re-use code from some earlier experiments in our *Embedded Systems* course.

The students also developed a basic protocol for communications and automatic board address assignment. Thus this experiment might be employed in a more software-centric course in communications protocols, i.e. in computer science, by abstracting away the lower level routines. Additionally, with the ability to control line terminations, the experimental hardware could be employed as part of classroom experiments in transmission line theory, attaching a much-needed practical reality to that course work, which is all too frequently a 'math only" course.

The scope of work demonstrated by the students is exemplified in the schematic of the RS485 interface of Figure 5.

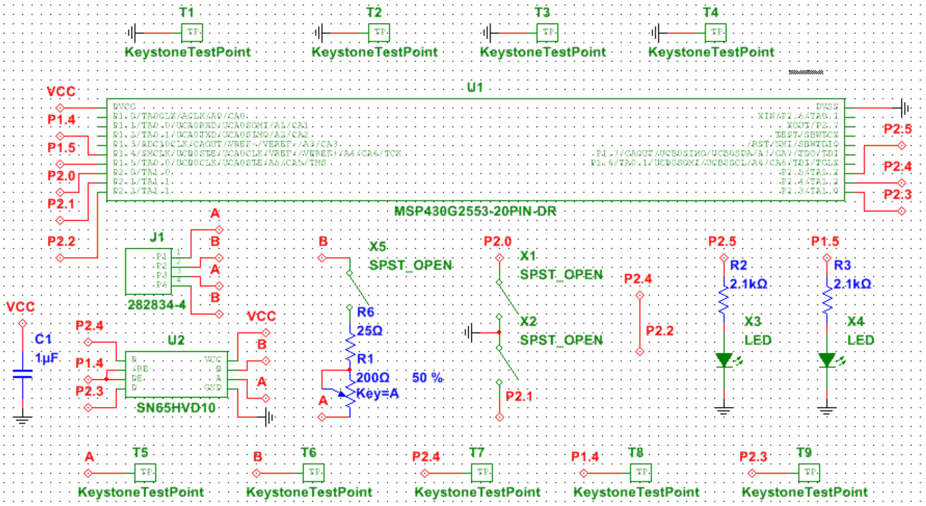

Figure 5 RS485 interface schematic

Real-Time Concepts

The second course, "*Real Time Concepts*" uses the *iRobot Create,* Figure 7, controlled by a *myRIO,* Figure 6, to investigate higher level issues. The *myRIO* is programmed in LabVIEW (which the students also learn in this class) to control a set of *iRobots* to simulate a traffic management system. Individual robots play the role of vehicles that interact with smart traffic signals (also controlled by *myRIO*) via sensors and equipped with video cameras. No data messages are passed, rather the robots and traffic light communicate visually and with other sensors

The specs of the project were simple: a set of robots would behave autonomously, stay in their lanes, maintain a safe following distance and obey traffic lights. No communication with or between vehicles was permitted. Each vehicle had to "see" the color of the traffic light, "sense" the lines on the "road" and stop if it came too close to another vehicle (IR sensors were used to detect other vehicles). An example of this experiment is shown inFigure 8.

Figure 6 *myRIO*

Figure 7 *iRobot Create*

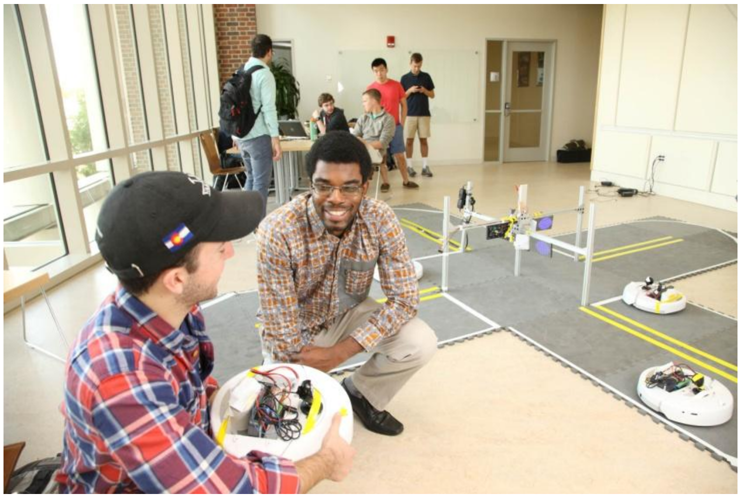

Figure 8

Separate groups developed algorithms to obey the traffic signals, stay inside the lines, and maintain safe following distances. One other group optimized the image processing code to detect the traffic light color while the last group designed, implemented and built the traffic light and “road”. The groups met regularly to refine their separate functions and then to integrate them into a single system. The integration required the specification and analysis of multiple interacting state machines to ensure that all conditions are specified and handled properly.

The materials required to complete this experiment are:

- National Instruments myRIO[7]
- *iRobot Create*[8]
- Any commonly available USB camera
- Floor tiles that may be marked to simulate roads
- LED display for simulating traffic signals[9]
- Miscellaneous plastic pipe and connectors available locally from hardware stores

These materials are all readily available, and of modest cost. We can supply course materials, sample code, and courseware assistance to other institutions.

Summary and Conclusions

These 2 courses were first offered on an experimental basis in the Fall 2014 semester. We found the students to be very interested and engaged and all expressed positive impressions of their learning experience. Additionally, we gained insight into what the students were taking away from their introductory coursework.

We envision these courses as emerging as part of an on-going incremental development in our embedded computing curriculum. For example, one of the outcomes of the “*Design Your Own Embedded Experiment”* is a new sequence of experiments that will be merged into our existing course enabling us to broaden our options for this course and perhaps develop new coursework that may be more focused for students from other majors such as Computer Science or Mechanical Engineering. The “*Real Time Concepts*” course enables us to improve and modify our course work to introduce the students to new programming paradigms and hardware abstractions, which we envision as valuable additions to our program.

These experiments will mesh well with embedded computing curricula and will allow educators to exploit the physical hardware in a variety of course scenarios. We further expect that the process of ongoing experiment development will allow curricula at our own - and other - universities to stay fresh. Further we believe that working with these experiments will foster a community of educators that can contribute to this important field.

References

1. Harry C. Powell, Joanne Bechta Dugan. Embedded and Cyber-physical Computing Courseware as the Nexus of Curriculum Reform in Electrical and Computer Engineering. Proceedings of ASEE- Se Section. MAcon Ga.; 2014.

2. Dr. Harry Powell, Prof. Joanne Bechta Dugan. Embedded computing reinforces and integrates concepts across ECE curriculum. Proceedings of the Annual Conference of the ASEE, 2014. Indianapolis, Indiana; 2014.

3. Ultra-Low-Power MSP430 Microcontroller LaunchPad Kits [Internet]. [cited 2014 Dec 1]. Available from: http://www.ti.com/ww/en/launchpad/launchpads-msp430.html#tabs

4. NI Ultiboard – Printed Circuit Board Layout and Routing - National Instruments [Internet]. [cited 2014 Dec 1]. Available from: http://www.ni.com/ultiboard/

5. Welcome to Ohio Art | Ohio Art [Internet]. [cited 2015 Jan 31]. Available from: http://www.ohioart.com/

6. RS-485 - Wikipedia, the free encyclopedia [Internet]. [cited 2014 Dec 1]. Available from: http://en.wikipedia.org/wiki/RS-485

7. NI myRIO - National Instruments [Internet]. [cited 2014 Jan 4]. Available from: http://www.ni.com/myrio/

8. iRobot: iRobot Create® Programmable Robot [Internet]. [cited 2015 Mar 12]. Available from: http://store.irobot.com/education-research-robots/irobot-create-programmable-robot/family.jsp?categoryId=2591511&s=A-ProductAge&gclid=CLr78f_Fo8QCFeXm7AodZjsA3g

9. Adafruit RGB Matrix HAT + RTC for Raspberry Pi - Mini Kit ID: 2345 - $24.95 : Adafruit Industries, Unique & fun DIY electronics and kits [Internet]. [cited 2015 Mar 12]. Available from: http://www.adafruit.com/products/2345