

Embedded Systems - Shape The World

Dr. Jonathan W. Valvano, University of Texas, Austin

Dr. Jonathan Valvano is a professor in the Department of Electrical and Computer Engineering at The University of Texas at Austin and holds the Engineering Foundation Centennial Teaching Fellowship in Electrical Engineering. He received his S.B. and S.M. in Electrical and Computer Engineering from MIT in 1977 and his Ph.D. in 1981 from the joint Harvard-MIT program in Medical Engineering and Medical Physics. He joined the faculty at The University of Texas at Austin in 1981 and has 32 years of experience in teaching and research. He has received numerous teaching awards and authored five widely-used textbooks on embedded microcomputer systems. He has co-founded a successful medical device company called Admittance Technologies. His research involves integrated analog/digital processing, low-power design, medical instrumentation, and real-time systems.

Dr. Jonathan W. Valvano, University of Texas, Austin

Dr. Jonathan Valvano is a professor in the Department of Electrical and Computer Engineering at The University of Texas at Austin and holds the Engineering Foundation Centennial Teaching Fellowship in Electrical Engineering. He received his S.B. and S.M. in Electrical and Computer Engineering from MIT in 1977 and his Ph.D. in 1981 from the joint Harvard-MIT program in Medical Engineering and Medical Physics. He joined the faculty at The University of Texas at Austin in 1981 and has 32 years of experience in teaching and research. He has received numerous teaching awards and authored five widely-used textbooks on embedded microcomputer systems. He has co-founded a successful medical device company called Admittance Technologies. His research involves integrated analog/digital processing, low-power design, medical instrumentation, and real-time systems.

Embedded Systems - Shape The World

Jonathan W. Valvano¹, Ramesh Yerraballi¹², Chad J. Fulton³, Chinmaya Dattathri¹ ¹Electrical and Computer Engineering ²Biomedical Engineering ³Center for Teaching and Learning University of Texas at Austin

Abstract

We have designed, implemented and deployed a Massive Open Online Class (MOOC) with a substantial lab component within the edX platform. If MOOCs are truly going to transform the education, then they must be able to deliver laboratory classes. This offering goes a long way in unraveling the perceived complexities in delivering a laboratory experience to tens of thousands of students from all around the globe. We believe the techniques developed in this class will significantly transform the MOOC environment. Effective education requires students to learn by doing. In the traditional academic setting this active learning is delivered in a lab format. A number of important factors have combined that allow a lab class like this to be taught at this time. First, we have significant support from industrial partners ARM Inc and Texas Instruments. Second, the massive growth of embedded microcontrollers has made the availability of lost-cost development platforms feasible. Third, we have assembled a team with the passion, patience, and experience of delivering quality lab experiences to large classes. Fourth, on-line tools now exist that allow students to interact and support each other. We used edX for the delivery of videos, interactive animations, text, and guizzes. We used Piazza for discussion boards and Zyante for a programming reference. We partnered with element-14, Digikey, and Mouser to make sure the lab kit is available and low-cost. In this paper we will present best practices, successes and limitations of teaching a substantial lab across the globe.

Background

An *embedded system* combines mechanical, electrical, and chemical components along with a computer, hidden inside, to perform a single dedicated purpose. There are more computers on this planet than there are people, and most of these computers are single-chip microcontrollers that are the brains of an embedded system. Embedded systems are a ubiquitous component of our everyday lives. We interact with hundreds of tiny computers every day that are embedded into our houses, our cars, our bridges, our toys, and our work. As our world has become more complex, so have the capabilities of the microcontrollers embedded into our devices. Therefore the world needs a trained workforce to develop and manage products based on embedded microcontrollers.

Pedagogy

The overall educational objective of this class is to allow students to discover how the computer interacts with its environment. It provides hands-on experiences of how an embedded system could be used to solve problems. The focus of this introductory course is understanding and analysis rather than design. It takes an effective approach to learning new techniques by doing them. We feel we have solved the dilemma in learning a laboratory-based topic like embedded systems where there is a tremendous volume of details that first must be learned before hardware

and software systems can be designed. The approach taken in this course is to learn by doing in a *bottom-up* fashion. One of the advantages of a bottom-up approach to learning is that the student begins by mastering simple concepts. Once the student truly understands simple concepts, he or she can then embark on the creative process of design, which involves putting the pieces together to create a more complex system. True creativity is needed to solve complex problems using effective combinations of simple components. Embedded systems afford an effective platform to teach new engineers how to program for three reasons. First, there is no operating system. Thus, in a bottom-up fashion the student can see, write, and understand all software running on a system that actually does something. Second, embedded systems involve real input/output that is easy for the student to touch, hear, and see. Third, embedded systems are employed in many every-day products, motivating students to see firsthand, how engineering processes can be applied in the real world.

This course is intended for beginning college students with some knowledge of electricity as would have been taught in an introductory college physics class. Secondly, it is expected students will have some basic knowledge of programming and logic design. No specific language will be assumed as prior knowledge but this class could be taken as their second programming class.

Learning objectives of the course

Although the students are engaged with a fun and rewarding lab experience, the educational pedagogy is centered on fundamental learning objectives. After the successful conclusion of this class students should be able to understand the basic components of a computer, write assembly and C language programs that perform I/O functions and implement simple data structures, manipulate numbers in multiple formats, and understand how software uses global memory to store permanent information and the stack to store temporary information. Our goal is for students to learn these concepts:

- 0) Understanding how the computer stores and manipulates data,
- 1) The understanding of embedded systems using modular design and abstraction,
- 2) Assembly language and C programming: considering both function and style,
- 3) Understanding how the computer executes instructions,
- 4) The strategic use of memory,
- 5) Debugging and verification using a simulator and on the real microcontroller
- 6) How input/output using switches, LEDs, DACs, ADCs, and serial ports,
- 7) The implementation of an I/O driver, multithreaded programming,
- 8) Understanding how local variables and parameters work,
- 9) Analog to digital conversion (ADC), periodic sampling,
- 10) Simple motors (e.g., stepper motor and DC motor control),
- 11) Digital to analog conversion (DAC), used to make simple sounds,
- 12) Design and implementation of elementary data structures.

Laboratory Kit

Active learning requires a platform for the student to learn by doing. Figure 1 shows the components of the lab kit. There are two difficulties with a physical lab kit deployed in a worldwide open classroom environment. The first problem is *availability of components*. We partnered with companies and distributors six months in advance of the course launch to guarantee

availability. The companies wanted us to specify the number of students who would buy the kit. In this regard, we were very lucky. We estimated 2000 people would register for the class and 1000 would buy the kit. In turns out Texas Instruments produced 10,000 microcontroller boards just in case. Much to our surprise 40,000 people registered and we estimate 11,000 purchased the kit. The second solution to the problem of availability was to have three world-wide distributors (element-14, Mouser, and Digi-Key). Working with these distributors, we created one-click landing pages for students to buy the kit. Furthermore, for each component in the kit (other than the microcontroller board), we had three or more possible parts. The third solution was to design the course with flexible deadlines and pathways. Each lab had a simulation and a real-board requirement. Students who were waiting for the parts to travel around the globe could proceed with the labs in simulation, and then go back and finish the real-board lab once the parts arrived.



Figure 1. The kit includes a microcontroller board, a solderless breadboard, wires, switches, LEDs, audio jack, slide pot, and some resistors.

The second problem is *cost*. The first solution to cost was to create a learning environment out of low-cost components. ARM provided a no-cost integrated development environment (IDE). The Texas Instruments Tiva LaunchPad costs only \$13, and ships to most countries around the world. In the US and Europe the kit costs about \$33 plus shipping. The second solution was to provide local buying options for each region (Europe, Asia, South America, Oceania, and Africa). In Asia the local suppliers offered lower prices than the world-wide distributors. The third solution was to get a grant for free kits. Because each lab has a simulation and a real-board component, we could identify students who were engaged in the class, yet couldn't get a kit (completed the labs in simulation but not on the real board). We gave away kits to 50 students. Because we used

a powerful microcontroller connected to a personal computer, we were able to build into the lab environment a no-cost voltmeter and oscilloscope.

Laboratory Assessments

In a typical embedded system lab, the student combines mechanical and electrical components interfacing them to a microcontroller to create a system. The student writes software that is loaded onto a microcontroller which then performs a specific and dedicated task. To get a grade the student demonstrates the lab solution to the teaching assistant. There are three tasks the TA performs: first the TA must **control** the process by asking questions or requesting the solutions perform appropriate tasks, second the TA must **observe** the actions and reactions, and third the TA must **judge** whether the solution achieved the desired outcome. We have captured these three TA-tasks by developing a suite of software plug-ins that run inside the compiler-debugger and additional software that resides in the microcontroller itself. These software modules perform the control, observation and judgment to certify the student has completed the lab.

Figure 2 shows a screen shot of the learning platform in simulation. We build this platform around Keil uVision from ARM. We wrote a set of simulation tools, which are dynamic linked libraries (DLLs) derived from ARM Application Notes 154 and 196. ARM provides the editor, compiler, and instruction set simulator for the Cortex M microcontroller, and we added I/O device simulation, external device simulation, and automatic lab grading.



Figure 2. Learning environment in simulation mode.

We also built a real board grading platform around Keil uVision from ARM. The real-board environment includes a DLL that we wrote. We derived this DLL from ARM Application Note 204. The real-board grader also has software embedded alongside the student code, and a standalone application that processes serial port data sent from the microcontroller to the PC. Figure 3 shows a screen shot of the same lab on the real board. We created this environment with a voltmeter, oscilloscope and an automatic lab grader. The grader instructs the student to push individual switches and then uses the oscilloscope to test the validity of the output of the student solution.



Figure 3. Learning environment on the real board.

Demographics

Table 1 illustrates that this class was indeed world-wide. There were 39,990 students, and demographic data was optional at the time of registration. 1261 students paid \$50 for an ID-verified certificate from edX. We used Piazza for the forum, and Figure 4 shows three concentrations USA, Europe and India.

Asia	27.4%
Europe	26.1%
North America	25.0%
South America	10.3%
Africa	4.5%
Middle East	2.7%
Southeast Asia	1.7%
Oceania	1.3%
Central America	1.0%

Age	Count
10 to 19	1232
20 to 29	16835
30 to 39	3389
40 to 49	2156
50 to 59	926
60+	383

Male	24851
Female	2769

Table 1. Breakdown of the students by region male/female and age, self-reported.



Figure 4. Distribution of students using the forum, self-reported.

Challenges of a World-wide Class

There are many challenges of delivering a class to a world-wide audience. We saw about 5000 students working on the labs, but only 3500 students registered on the forum. In order to run a lab class at this scale, it was important for students to interact on the forum. There are cultural barriers for getting students to seek help on a forum.

The course was delivered completely in English. Videos have subtitles, but we did receive numerous requests to translate the material into other languages. We had neither the money nor expertise required for translation.

We did require students have administrator rights on the personal computer to install the Keil uVision and DLLs onto their local computer. Because of the requirement to use Keil uVision, we did require Windows OS. There were some requests for Linux solutions, which we could not

accommodate. We also noticed many students had PC more than 10 years old, requiring us to rebuild the DLLs without using new instructions.

The biggest problem we faced was internet speed. The course included hundreds of videos, all recorded in high definition. Many students were unable to stream the videos in real time. To solve this problem we created a table of video links that allowed students to download videos offline. In addition each video had two sources, because some countries block or limit access to YouTube.

Conclusions

In conclusion, we successfully produced a MOOC with a substantial lab component, delivered to a global audience. A MOOC Lab needs the ability to control, observe, and assess. Embedded microcontrollers provide these three requirements. There are two major limitations of this lab environment and grading engine. The first is lack of creativity. Because of the need to automatically grade, students could not creatively solve a more interesting and more difficult problem and get credit. For example in the face to face class, we allow students to build a music player on top of their digital piano. An effective learning environment needs to ask more openended problems and allow more creative solutions. We did have an open-ended video game lab at the end of the class, but we had no mechanism to grade it. The second limitation is the quantitative grading assessments. Basically we employed black-box testing, providing known inputs and measured the resulting outputs of the student solution. An effective educational experience will require some qualitative assessments of the style and structure of the solution.

Peer interactions are critical. We encouraged students to team up in groups of 2 or 3, but the MOOC platform requires a lot more infrastructure and support for managing tens of thousands of teams. We did experience students were extremely engaging in the forums and provided accurate and timely answers to each other's questions.

Industry collaboration is essential to deploy a class such as this. We started about 18 months before deployment and received a rich and substantial support from a variety of sources. In addition it took serious commitments from the university to provide funding, technology, contacts, and management support.

In summary, we are convinced this MOOC course provides education to a world-wide audience, but until we provide for open-ended thinking and qualitative assessments, we are not ready for college credit.