AC 2011-437: USE OF THE ARDUINO PLATFORM FOR A JUNIOR-LEVEL UNDERGRADUATE MICROPROCESSORS COURSE

Nathaniel Bird, Ohio Northern University

Nathaniel Bird is an Assistant Professor of Computer Science and Computer Engineering at Ohio Northern University. He received his doctorate from the University of Minnesota, Twin Cities. Research interests include computer vision, robotics, behavior analysis, and undergraduate pedagogy. Dr. Bird is a member of ACM, IEEE, and Tau Beta Pi.

Use of the Arduino Platform for a Junior-Level Undergraduate Microprocessors Course

Abstract

This paper outlines the benefits of incorporating the Arduino microcontroller board into a junior-level course on microprocessors for students majoring in electrical engineering and computer engineering. The Arduino is an open-source hardware platform that has recently gained a wide following among hobbyist and artist communities for its ease of use and the ability to build interactive projects with it quickly. A description of a microprocessors course that used these boards is provided. Results are presented demonstrating students' learning of microprocessors through their ability to develop projects of their own design. Survey results characterizing student enthusiasm for learning the material in this manner are also shown.

Introduction

The microprocessors course described in this paper is designed for students in electrical and computer engineering at the beginning of their junior year. A key aspect of this course is the heterogeneous nature of the students taking it–the computer engineering students have had many programming intensive courses, whereas the electrical engineering students have had one programming course in their freshman year, leading to a discrepancy in abilities. This paper describes how the Arduino platform was integrated into this situation.

The Arduino platform was found to be beneficial for several reasons. The first is student enthusiasm. The device is an open-source hardware platform that is programmed using an open-source programming environment, which resonates with many students. Development for this platform is something they can do on their own laptops, which is more inviting than developing on restricted software for a bolted-down piece of hardware in a locked lab.

The second main benefit is the platform's accessibility to the students. This course is designed for students from two different programs–electrical engineering majors with very little programming experience, and computer engineering majors with significant programming experience. The Arduino boards are advertised to be straight forward to use and program; they are designed for use by artists and hobbyists. The C++ like programming language used to program the board is far easier to grasp for students with little programming experience than the register-based assembly language used in previous iterations of this course. This frees the class from an endless struggle with syntax, and focuses attention instead on the projects at hand.

The third advantage is time. Arduino boards are inexpensive enough that it is reasonable for the students to each purchase their own board for the class. This allows them to use the boards at home as well as in the lab. The labs for this course are three hours per week,

for a ten week quarter. This totals to 30 hours at most, which is not enough time to build proficiency if the equipment can only be used in lab. Because the students have their own boards though, they can now use them at home and get more time with them over the duration of the course.

Finally, the fourth benefit of using the Arduino platform is it opens up the possibility of having the students do individual, quarter-long projects. This lets students investigate something that excites them personally, and invests them in the learning process. It also requires learning outside of the confines of the classroom and even the book, as most students decide to pursue projects dealing with complex behaviors and specialized components of their choosing.

Background on the Arduino

Arduino is a for-profit company that builds the self-entitled Arduino open-source hardware platform. What this means is that the hardware designs are open, as is all software used to program the boards. Third parties are welcome to build and distribute their own, but the name 'Arduino' cannot be used by third parties. A rather large enthusiast community has sprung up around the Arduino board, making it a popular platform for personal projects.

The Arduino boards incorporate an ATMega microcontroller, along with necessary electronics to connect it to a computer via USB for programming, and to help regulate its power and sanitize its inputs and outputs to a degree. There are several types of Arduino boards, and the Arduino Duemilanove with the ATMega328 microcontroller was the board chosen for the class. Since starting the class, however, the Duemilanove has been discontinued and replaced with the similar Arduino Uno board.

Arduino provides an open-source software development environment for programming the Arduino boards. The boards are programmed using a modified version of C++, in which some concepts esoteric to novice programmers are hidden. Straight forward functions are included for most common microcontroller tasks. The programming environment makes it very easy to write programs, and loading them onto the board is as easy as plugging in the USB cable and clicking the "upload" button.

Using the Arduino board shields students from some of the "nitty-gritty" detail of programming and interfacing many other microcontrollers. This is a strength of using this platform in an introductory class. In essence, by using a platform that automates away much of the "grungy" detail, the focus of the class can be on semantics instead of syntax. Students can pay more attention to the underlying meaning of what they are doing, instead of dealing with Baroque intricacies quite so much. This is an overall benefit to the course, as students who end up in jobs utilizing microcontrollers will have to learn a new system regardless, which should be easier when they have a firm understanding of the underlying concepts.

The Arduino website[1] is the main source of information on the Arduino project. It includes information about all of the different hardware boards as well as documentation of the programming environment. Many example Arduino applications are available as well. The site is well designed, easy to search, and is a tremendous resource for anyone interested.

Course Setup

The course described here was designed to be a practical, hands-on class and the focus was on understanding microprocessors, their components, and properties such that they could be used as components in larger systems. This is in contrast to some other microprocessor courses where the focus in on building a microprocessor,[2] not necessarily using them.

This course included a midterm and a final exam to test fundamental knowledge regarding microprocessor components and concepts, but the majority of the course work took the form of the weekly directed laboratory assignments as well as the quarter-long project.

As part of the course, students were required to purchase their own Arduino board and a few other parts. This is a departure from the way this course has been run in the past in our department, where the school maintained a set of microcontrollers that were only available during the laboratory sessions. However, due to the desire to incorporate a quarter-long project and potentially longer laboratory assignments, restricting the use of the Arduino boards to laboratory sessions was not feasible, so students were required to purchase their own boards. Costs were kept in line with previous iterations of the course by requiring a *much* cheaper textbook, however.

The following is the list of components the students were required to purchase for the course.

- Arduino Duemilanove board.
- USB cable to connect the board to a computer for programming.
- Bread board.
- 16x2 LCD display kit.
- Miscellaneous resistors and LEDs.

The following list is the items that, in hindsight, should have been required but were not.

- 7-segment LED display.
- Stepper motor.
- H-bridge IC.

---

[1]The Arduino project website is at http://www.arduino.cc

- Piezoelectric speaker.
- Miscellaneous buttons, transistors, potentiometers, photoresistors, and other parts.

Textbook

There are few books available about the Arduino platform. To some extent, this is for good reason–the Arduino website has all of the documentation for the project posted in an easily searchable manner. However, for a university course, it was felt that a book was necessary for reference. The textbook chosen for the course was Noble's "Programming Interactivity" book.[3] Unfortunately, this book did not turn out to be a good fit for the course. It was aimed at visual artists, an audience the class did not identify with. Also, only a few chapters were useful as it covered two interactive programming environments in addition to the Arduino.

At the time the course was designed, there were only two other possibilities for books. The first was Oxer and Bleming's "Practical Arduino" book.[4] This book contains a collection of how-to guides for small projects that are too large and material intensive for a weekly lab. It would serve as a great source of ideas for the quarter-long project, but would not work as a course reference book. The last book is Banzi's "Getting Started with Arduino".[1] This is a short, cheap book that serves purely as a reference for the Arduino board and programming environment. Banzi's book will likely be used next time this course is taught.

Laboratory Assignments

Nine laboratory sessions were conducted over the course of the quarter. The assignments themselves were designed to reflect and build upon the foundational concepts discussed in lecture. The remainder of this section briefly outlines the assignment for each laboratory. Note that because the students had their own Arduino boards and materials, they were able to continue to work at home if they got into trouble and ran out of time.

Lab 1: Hello world. This assignment focused on getting students used to the Arduino board and programming environment by attaching an LED to an output pin and making it blink.

Lab 2: Binary counter. This assignment had students implemented a binary counter. This reinforced binary number systems, multiple outputs, and how the internal clock worked.

Lab 3: 7-Segment LED display. For this assignment, a counter was implemented to count up from 0 to 9 on a 7-segment LED display.

Lab 4: Analog input. In this assignment, students read an analog voltage value from a potentiometer attached to an analog input pin, and then displayed a corresponding value between 0 and 9 on an attached 7-Segment LED display. This assignment helped cover the importance of writing good code so it can be reused.

Lab 5: Control of a stepper motor. This assignment used the Arduino board to control a stepper motor. A potentiometer attached to an analog input pin was used to control the rotation speed.

Lab 6: LCD display units. This assignment had the students implement a simple video game simulation of a claw game using a 16x2 LCD display. A potentiometer was used to move the claw from side to side and a button was used to make it drop and try to pick up a prize.

Lab 7: Serial communication between Arduino boards. This assignment was a group project in which the students had to pair up. One parter was responsible for programming a querying device that requested information and the other partner was responsible for programming a responding device.

Lab 8: Digital clock. For this assignment, students built a simple digital clock. Interrupts were used to set the time on the display. Bounce was an issue students had to deal with.

Lab 9: Music. In this assignment, students created the functions necessary to play a simple melody of their choice through a piezoelectric speaker attached to a digital output pin.

The only major area the Arduino platform was found to be lacking in was clock-based interrupts. A laboratory session built around timer interrupts was not possible due to the Arduino environment's lack of integrated access to the timers in the ATMega microcontroller. The Arduino environment takes all of them over to implement pulse-width-modulation output on select pins.
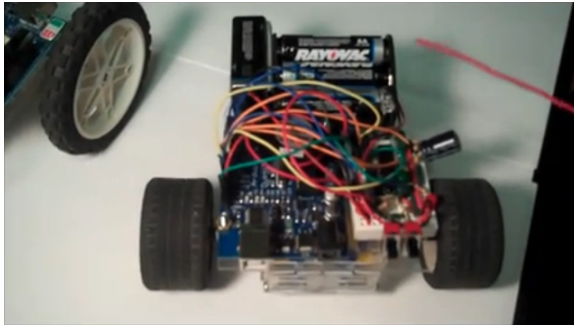
Quarter-Long Project

As part of this course, students were to develop a programmed, interactive device over the course of the quarter. A scope sufficient for this length of time was required from the students. They were allowed to work in small groups, but many decided to work alone as well. The use of any microcontroller for the project was allowed, but a microcontroller was required and use of the Arduino was strongly encouraged.

There were several components on which the project was graded, described as follows.

Project. The purpose of the project itself was to develop and build a device of the student's own design, and to get it working. To get a passing grade, the project needed to result in the creation of a physical device. The device needed to have a programming aspect, and had to be interactive.
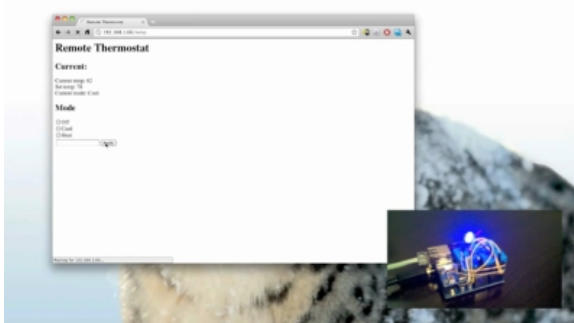
The students were free to choose their own topics for their projects and were encouraged to be be creative and to build something that interests them personally. The students were requested to develop a project that explores their hobbies, interests, and reasons for

(a)   (b)

(c)   (d)

Figure 1: Example student quarter-long projects. (a) Laser-tag playing robots; (b) automated fish tank monitor; (c) web-enabled thermostat; and (c) music-playing cloak.

continuing in their program for as long as they have. As this course was junior-level, the students had more than the requisite circuits experience to build their devices.

Many unique and interesting projects were developed and few of the most outstanding were the following. See Figure 1 for images.

- One group of two students built a pair of wheeled robots that played laser tag with each other. The robots were controlled by television remote controls, and played a sad song when a hit was registered.
- One student built a device to automate lighting, heating, and feeding of his tropical fish aquarium.

- One group of two students built a device to control a thermostat from a remote web page.
- One student created a cloak that played music only when walking.

Written reports. Communication about the project status was performed throughout the quarter using written reports.

The first written report was the proposal, due two weeks into the quarter, which in a paragraph outlined the project idea and how it would be addressed. This was useful because it got the students thinking about the project early and gave the instructor a chance to provide feedback about the general scope before much work had been done.

The second written report was due halfway through the quarter, and was meant to show that the projects were about half finished. This essentially took the form of an abbreviated final report. Having a mid-project report helped to keep the students on track, and provided an early warning for projects that were encountering problems. Feedback was provided to help steer projects that were encountering problems toward a successful resolution.

Finally, the third written report was the final project report due at the end of the quarter. This roughly 7–10 page report outlined the project, its inspiration, design, and results. Class time was devoted to guide the students in how to craft a technical paper about a project like this. Writing technically about a project of their own design was a good experience for the students.

Live presentation. In the last two class periods of the quarter, students took turns delivering Powerpoint presentations to the rest of the class regarding their projects and answering questions. Due to the number of projects, the presentations had to be kept to about five minutes each, but the exercise was worthwhile because it gave the students experience presenting, and sometimes defending, their ideas to a large group.

Demonstration video. In addition to the written presentation and live presentation, students were required to create a one to two minute "Youtube style" demonstration video of their project. The idea here was to leave the students with a quick demonstration that they could use whenever they wanted to quickly demonstrate their project to someone. Small digital video cameras were available for students to check out if needed.[2]

Student Feedback

During the last week of class, a short survey was distributed to the students to complete during lecture. A Likert scale from 1 (strongly disagree) to 5 (strongly agree) was used for

---

[2]The website containing the videos created by the students demonstrating their projects is available for viewing at http://natebird.com/teaching/microprocessors_projects_2010-2011.html

Table 1: Student Survey Results, Likert Scale Questions

Question	Mean	Std. Dev.
Before taking this course, I was very interested in the topic of microprocessors.	3.8	1.0
After taking this course, I am very interested in the topic of microprocessors.	4.3	0.8
My overall enthusiasm for this course is high.	4.1	0.9
Having my own microcontroller board that I could take home and work with on my own computer was more useful than having microcontroller boards that had to stay in the school lab.	4.9	0.3
The Arduino board was straight forward to program.	4.6	0.6
The labs helped my learning.	4.6	0.6
I enjoyed the labs.	4.3	0.8
The quarter-long project helped me learn.	4.3	0.8
I enjoyed the quarter-long project.	4.0	0.6
I like the fact that I was free to choose my topic for the quarter-long project.	4.8	0.4

Table 2: Student Survey Results, Yes/No Question

Question	Yes	No
Would you recommend that we use the Arduino boards for this course in the future?	100%	0%

some questions in this survey, the results of which are shown in Table 1. A Yes/No question was asked as well to determine the students' overall thoughts on continuing with the Arduino boards in the future, the results of which are shown in Table 2.

There are several interesting messages that can be found in these results. First, enthusiasm for the topic of microprocessors was high before and after the class, indicating students did not lose interest in the subject. Second, the Arduino platform was well liked by students, as were the laboratory assignments and quarter-long project that employed it. Third, both the laboratory assignments and quarter-long project were perceived as educational. Fourth, students liked having their own microcontroller board. Finally, students unanimously agreed that the Arduino should be used in the future. On these direct questions, there was very little negative about the course.

Table 3: Student Survey Results, Yes/No Question

#	Student Comment
1	Support for those with little understanding of programming logic and syntaxes.
2	I really liked the fact that we had our own equipment to use and keep.
3	Some labs were more difficult due to EE non-programming background.
4	Each part of course went a bit slow in terms of information flow. (e.g. more in-depth programming review over less time, more progress in learning about Arduino/electronics instead.)
5	Very worthwhile. Simple to use but very practical.
6	The Arduino was a good choice, easy to learn and use.
7	I loved the Arduino boards. The ONU Micromouse team is now using an Arduino as their microcontroller because of this course.
8	Having uniform kits would probably be an improvement.
9	The boards were easy to use yet effective. They support many options and can do a lot of different things. Might want to go to a larger board with more pins next.

The survey also requested open feedback from the students, where they could write anything they desired to be known. The student responses are shown in Table 3. There were a couple students who found the programming difficult, one of whom specifically cited the discrepancy in programming background between the electrical engineering students and the computer engineering students in the class. Despite how much easier it is to program the Arduino than other microcontrollers that are programmed in an assembly language, some students still felt that the programming was difficult.

Other students added that they found the Arduino easy to use, and liked the choice. One reiterated that they liked having their own board. Finally, one student mentioned that a uniform lab kit would be better than the mix of equipment students got off of the shopping lists, which is almost certainly true.

Reflection and Conclusion

There were several benefits overall to using the Arduino in the junior-level microprocessors course presented here. Student enthusiasm for using the board was quite high. It was accessible and straight-forward to to students without much programming experience. By requiring all of the students in the class to get their own board, it opened up the possibility for a quarter-long project that provides more hands-on experience than can happen in only the laboratory sessions.

The quarter-long projects were very successful. The students were all able to build something they were excited about. It made them invested in the course, and many of the students took to it by creating unique and interesting projects. For administration, tracking project progress with several written reports throughout the semester allowed most problems to be caught early. The vast majority of students finished the quarter with a working project.

There are several changes that are warranted to the course, however. A different book about the Arduino platform will have to be used in the future. Also, it somewhat difficult as an instructor to cover the theoretical material without a book designed for that. Another, more theoretical book would be a useful addition in the future. Finally, uniform kits should be built for the students to purchase next time. The shopping list worked to an extent, but laboratory sessions would be smoother if everyone had the exact same components.

In closing, using the Arduino board in a junior-level microprocessors course was a success. Students were enthusiastic about it, it was straight-forward to use, it promoted hands-on experience outside of class, and using it made it possible to incorporate a quarter-long project into the course. Overall, the Arduino was a good fit and a very good educational tool.

Acknowledgements

References

[1] Massimo Banzi. *Getting Started with Arduino*. Make, 2008.

[2] Jae Wook Jeon. A microprocessor course: Designing and implementing personal microcomputers. *IEEE Transactions on Education*, 43(4):426–433, 2000.

[3] Joshua Noble. *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*. O'Reilly Media, 2009.

[4] Jonathan Oxer and Hugh Blemings. *Practical Arduino: Cool Projects for Open Source Hardware*. Apress, 2009.