

## **Capstone Projects in a Computer Engineering Program Using Arduino**

### **Dr. Afsaneh Minaie, Utah Valley University**

Afsaneh Minaie is a professor of Computer Engineering at Utah Valley University. She received her B.S., M.S., and Ph.D. all in Electrical Engineering from University of Oklahoma. Her research interests include gender issues in the academic sciences and engineering fields, Embedded Systems Design, Mobile Computing, Wireless Sensor Networks, Nanotechnology, Data Mining and Databases.

### **Dr. Reza Sanati-Mehrizy, Utah Valley University**

Reza Sanati-Mehrizy is a professor of Computer Science Department at Utah Valley University, Orem, Utah. He received his M.S. and Ph.D. in Computer Science from the University of Oklahoma, Norman, Oklahoma. His research focuses on diverse areas such as: Database Design, Data Structures, Artificial Intelligence, Robotics, Computer Aided Manufacturing, Data Mining, Data Warehousing, and Machine Learning.

# Capstone Projects Using Arduino Platform

## Abstract

Capstone courses play a crucial role in Computer Engineering curricula. The principle purpose of a Capstone project course is to offer a summative opportunity for graduating senior engineering students to apply their professional skills and knowledge in a single experience and prepare them for work in industry. Like many engineering programs, students at Utah Valley University complete their requirements for graduation with a semester long capstone design project course. The intention of this course is to apply competencies gained during their first three years toward the solution of an embedded design problem. This paper presents the details of sample projects that the students have done in this capstone course using Arduino prototyping platforms.

## Introduction

It is common practice to incorporate one or two semester long senior capstone design projects in undergraduate computer engineering courses. The goal of these capstone design project courses are to provide graduating students with the opportunity to validate understanding of the concepts they have learned during the course of their studies and for instructors to measure the achievement of established learning goals. As with many computer engineering programs, students of the computer engineering program at Utah Valley University complete their degree programs with a semester long capstone design experience. The intent is for students to utilize competencies developed in the first three years of the curriculum in the solution of an embedded design problem.

## Embedded Systems

An Embedded system is any mechanical or electrical system that is controlled by a computer working as part of an overall system<sup>6</sup>. Embedded systems is one of the most dynamic and fast growing areas in industry. These systems continue to provide the core for a wide range of applications, such as smart cards for mobile payment and Diabetic meters. Embedded systems are ubiquitous. The area of Embedded System design has undergone tremendous growth in recent years. A major contributor of this growth has been the addition of networking technologies, database management systems, operating systems, and security modeling<sup>7</sup>. The remarkable growth in embedded computing has given rise to a demand for engineers and computer scientists with experience in designing and implementing embedded systems. In order to prepare our computer engineering students for the embedded systems design experience, we offer two required courses on embedded system design. The first course in the embedded sequence is a junior level course which presents an introduction to the basic building-blocks and the underlying scientific principles of embedded system. This course covers both the hardware and software aspects of embedded processor architectures and assembly language programming. In this course, students use the Freescale 68HC12 microcontroller. The second embedded course presents the design of hardware and software required for embedded real-time systems. This course covers the types of real-time systems, fuzzy logic controllers, sensors, real-time operating

systems, embedded C programming skills, and wireless sensor networks. In the second embedded course, students do their project on Freescale HS12 microcontrollers.

Nowadays, the embedded systems design problems have become more complicated and complex, requiring creative thinking and skilled engineers to solve these problems. To better prepare our students in this area, the focus of our computer engineering capstone design course has been the design of embedded systems. By requiring an embedded design project in our capstone course, our students receive hand-on training in embedded systems that will enable them for careers after graduation<sup>7</sup>.

### Senior Design Project Course at Utah Valley University

Our Senior Design Project Course serves as a project-oriented capstone course for computer engineering majors. This required course emphasizes major hardware and software co-design. This course satisfies the ABET (Accreditation Board for Engineering and Technology) requirements for providing students with a significant hands-on design experience<sup>8</sup>. Our senior design course is structured as a collection of open-ended independent student projects which are mutually selected by the faculty supervisor and student. It is shown that this type of student-driven, open-ended project requires a great deal of instructor’s flexibility, deep familiarity with available components, and ready suggestions for potential projects. However, for instructors who are willing to take on the effort, a student-driven design project can provide significant experience for students in problem specification and engineering design. The typical design process experience includes problem definition and constraints, gathering information, concept generation, preliminary design, detail design, communication of results, and improvements<sup>12</sup>. Our capstone course is based on the Engineering Design Process which is outlined in Table 1<sup>10</sup>.

Engineering Design	
<b>Requirement Analysis</b>	Identify the problem and constraints Define goals and criteria
<b>Functional Analysis</b>	Research and gather data
<b>Design Synthesis</b>	Brainstorm: Develop Possible Solutions Analyze potential solutions Model and test candidates Select a promising solution Build a Prototype Test and evaluate prototype Implement Communication of Results
<b>System Analysis and Control</b>	Improve: Review and redesign as needed

Table 1: Engineering Design Process<sup>10</sup>

Our Capstone course is offered every semester. The students in the Computer Engineering program take this course during their last semester. Students have the option of selecting their own embedded project or to work on a project that is given to them by their advisors. During the first week of the semester, students write a proposal to define problems and identify solution approaches for their project in addition to identifying the hardware and software that is needed for their project. After several iterations, the advisor approves their project. The faculty adviser will meet with each student individually on a weekly basis at a regularly scheduled, mutually agreeable time. These meetings are considered mandatory for the students. Occasional conflicts are inevitable, but the students need to understand that a portion of their grade for participation is based on attendance at the weekly meetings. At each meeting, issues associated with the project will be discussed and a status report will be provided by the student to the advisor. Students will keep a daily journal/work log detailing the work that was done, how much time was spent that day, and any technical details that might be needed for later reference. The faculty advisor keeps notes of each meeting as well as action items to be accomplished for the next meeting. Reviewing the log sheet from the previous meeting is a great way for the faculty to prepare for the upcoming one, and provides further evidence to the student of the meeting's importance. At the end of the semester, students turn in a final written report and final presentation which is evaluated by several faculty members from the department<sup>9</sup>.

## **Arduino**

Arduino is an open-source electronics prototyping platform based on easy-to-use hardware and software which is available at low-cost from multiple suppliers. Open-source hardware is an emerging concept similar to free and open software. Design files, schematics, board-layout, source code, and HDL modules can be shared and licensed so others can use, study, make and derive other projects from it. Both hardware and software of Arduino is open source. A worldwide community of professionals, students, hobbyists, artists, and programmers have gathered around Arduino open-source platform, and their contributions have added up to a large amount of accessible knowledge that can be of great help to students. Arduino was originally conceived by a team at the design Institute in Ivrea, Italy for use by a large non-technical audience of artists, designers, and architects<sup>11</sup>. The first Arduino was introduced in 2005. Sixteen versions of the Arduino hardware have been commercially produced to date. Some of the versions are: Arduino UNO, Arduino Leonardo, Arduino Mega, Arduino Nano, Arduino Due, and LilyPad Arduino. Arduino and Arduino-compatible boards use printed circuit expansion boards called "shields", which plug into the normally supplied Arduino pin headers. Shields can provide motor controls, GPS, Ethernet, LCD, etc<sup>5</sup>.

### **Advantages of Arduino<sup>5</sup>:**

- Inexpensive – Arduino boards are relatively inexpensive which makes it a good candidate to be used in student projects.
- Cross platform – The Arduino IDE runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.
- Open source and extensible software - The Arduino software is published as open source tool, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the jump from Arduino to the AVR C programming language on which it's based on. Similarly, you can add AVR-C code directly into your Arduino programs.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons Attribution Share – Alike license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

Engineering students quickly learn how to use the Arduino, and how to interface it with other hardware components with relative ease<sup>13</sup>. Majority of our students are nontraditional students that have to work and have family. These students usually pay for their own project parts so that they work on their project at home. Recently, a number of these students choose to use Arduino prototyping platform for their senior design project. The reason is that it is easy to use and an enormous amount of information is available on them for students to learn about. Another important reason is that it is attractive for these students is its low cost. This paper presents the details of two projects that our computer engineering students have done using Arduino boards in their capstone course.

### **First Project: Blulink Smart Key**

The objective of this project was to provide the convenience of an automotive “smart key” system in an aftermarket device for owners of vehicles in which these systems did not come pre-installed. While existing systems identify authorized keys using embedded RFID devices, the approach for this project was to further improve the convenience by using the Bluetooth connection from a smart phone as the authorized key for the system - effectively eliminating the need to carry any additional fobs or remotes besides their Bluetooth capable cell phone. Figure 1 depicts the installation of the Remote Start System<sup>3</sup>.

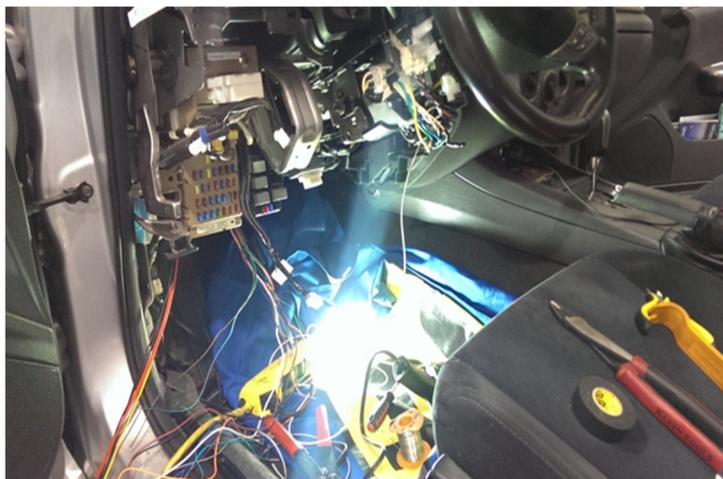


Figure 1 – Installation of Remote Start System<sup>3</sup>

The project utilized an Arduino Uno microcontroller (Figure 3) in combination with a Bluetooth Low-Energy shield (Figure 4) as a proof of concept of the desired functionality. Using this setup also required designing various circuits on a breadboard to allow interfacing with the existing computer systems inside a vehicle to safely start and operate the vehicle as if the physical key were in the ignition<sup>3</sup>.



Figure 3 – SunFounder UNO R3<sup>1</sup>

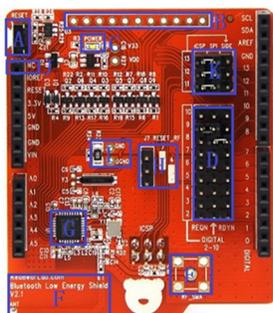


Figure 4 – ReadBear Labs BLE shield<sup>2</sup>

A demo fixture containing various LEDs, switches, and relays were used to simulate actual operation and represent the components inside the vehicle. The system operates as follows: upon

entering the vehicle, this system responds in the same manner as a traditional smart key system. Pushing the start button while depressing the brake will start the engine, while pushing the button without the brake depressed will energize only the accessory circuitry of the vehicle. Figure 5 shows a high level overview of the system that the student designed<sup>3</sup>.



Figure 5 – High Level System Overview<sup>3</sup>

Additional features include: smart door locks: the system is able to determine the user's proximity based on the RSSI of the Bluetooth connection - automatically unlocking the doors upon approaching the vehicle, and locking the doors as they walk away from the vehicle, remote start capabilities: allowing the user to start their vehicle with the push of a button from a smartphone application when in range. The student developed an Android application that is able to establish a connection and communicates with another device via Bluetooth<sup>3</sup>.

This project was successful and the student commented after finishing his project that “the completion of this project has allowed me to experience a simplified version of the design process from start to finish. Starting with the project proposal, to the research and development phases, and finally to the documentation and presentation of the completed product. By carefully documenting the whole process I was able to get an idea of the importance of and the relative time required to complete each stage of the development process. As per the requirements of the project, I also gained experience working with embedded system development. This included selecting an appropriate microcontroller for my project, learning how to program it, and designing circuits/writing code to interface with and control other components. This project required the design of several custom circuits for connecting the push button and interfacing with

the relays so they would operate as desired. Android development was an area that I had very little experience with starting this project. After building an application that is able establish a connection and communicate with another device via Bluetooth, I feel pretty confident in my abilities to design other types of applications as well. Overall, I found developing the mobile application to be one of the more difficult, but also more “fun” parts of this project.”<sup>3</sup>

### Second Project: Smart Homes with Smart Phones

The objective of this project was to create a home automation system that can turn lights on and off with an android phone and to get data from sensors and forward it to an android app over Bluetooth low energy wireless communication protocol. The following components are used in this project: an Arduino UNO microcontroller, a RedBearLab’s BLE shield, an easyVR 2.0 voice recognition shield, four RFduino’s, an android phone and a windows 8 PC for programming. The different software packages that were utilized to program the various components were Android studio, Arduino IDE v1.0.6 for the Arduino UNO and v1.6.0 for the RFduino’s. The system overview of the project is shown in Figure 6. A model home was designed and built. Top view of the Model Home is given in Figure 7. The model home is portable and it is 2 ft. long x 2 ft. wide x 1 ft. high made from building materials. There are outlet boxes in each room and are wired to provide 120 V<sub>AC</sub> . There are incandescent lights mounted to the top of each room and wired to an eight channel relay board. There is a power outlet box mounted to the under the home in order to mount the Arduino UNO, BLE shield, and easyVR voice recognition module<sup>4</sup>.

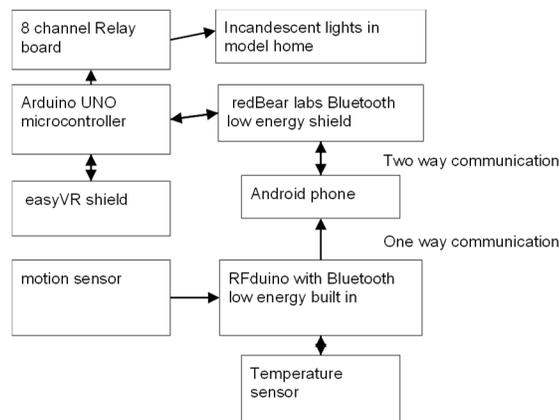


Figure 6: High Level System Overview<sup>4</sup>



Figure 7: Top View of the Model Home<sup>4</sup>

The Main board or the brains of the operation is the Arduino UNO R3. It has an Atmel ATmega328 microprocessor embedded in it. The relay board is controlled by the Arduino UNO. The wiring diagram is shown in Figure 8<sup>4</sup>.

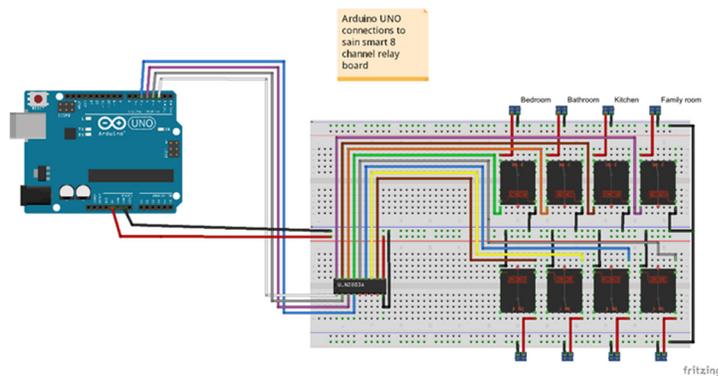
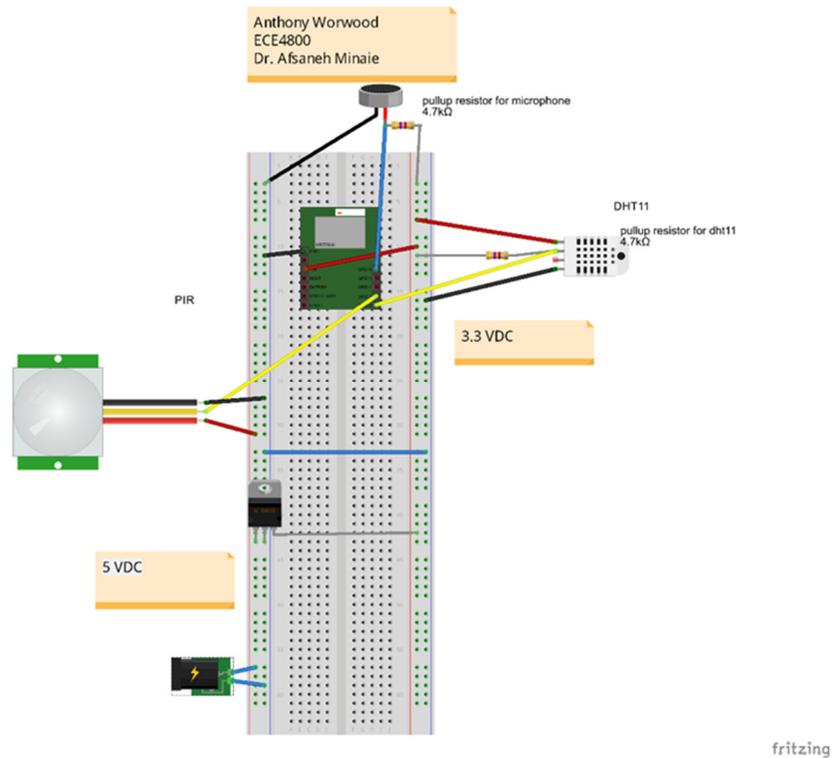


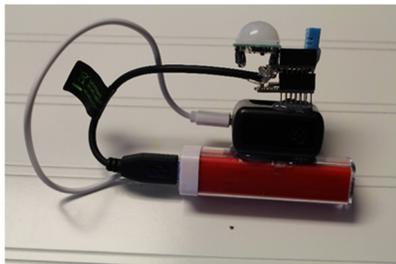
Figure 8: Wiring Diagram of Relay Board<sup>4</sup>

The RFduino's are made by SparkFun and use a 16 MHz cortex ARM processor and have Bluetooth LE 4.0 build in. They will be outfitted with temperature sensors and motion sensors to relay information back to the main Arduino. Figure 9 shows the wiring diagram for an RFduino module which consists of an RFduino, a temperature sensor, and a motion sensor. Figure 10 depicts the finished product of the RFduino Modules<sup>4</sup>.



**Figure 9: Wiring Diagram of RFduino Module<sup>4</sup>**

The RFduino modules were programmed to send packets over Bluetooth Low Energy of the data for the temperature sensors as well as the data from the motion sensors. The Android app displayed the temperature and motion data on the user interface as well as turning the lights on if motion is detected<sup>4</sup>.



**Figure 10: RFduino Module<sup>4</sup>**

The last feature of the project was to control the lights using voice commands using easy VR shield. The GUI for the Android app is shown in Figure 11<sup>4</sup>.

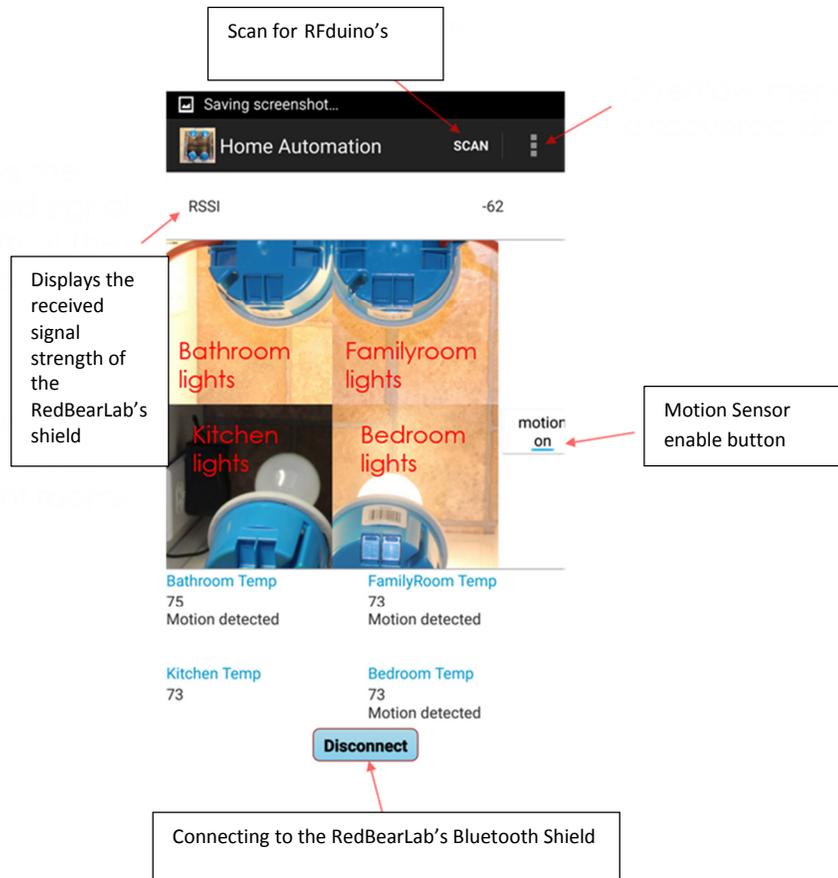


Figure 11: Android APP GUI<sup>4</sup>

This project was successful and the student commented that “there was a lot of learning that had to be done in order to complete this project. I started out not knowing anything about Arduino or Android app development and now I feel pretty confident in using them.”<sup>4</sup>

## Summary and Conclusion

Senior capstone design courses remain an engaging aspect of undergraduate computer engineering education and fulfill many requirements set forth by the Accreditation Board for Engineering and Technology (ABET). This paper presents senior design projects recently conducted utilizing Arduino boards. Our senior design course is structured as a collection of independent student projects. As our students are required to design, build, and troubleshoot a fully functional embedded project, they find this course both challenging and rewarding. These projects give the students the chance to use their technical expertise and knowledge gained during years of study, and they work very hard to have a working project by the end of the semester, often engaging in self-directed learning. In particular, a number of our nontraditional students choose Arduino prototyping boards for their projects. They develop the ability to debug, seek and find information they need, and the ability to understand and reverse-engineer poorly

written documentation. The students' feedback and their final project presentation indicate that they have pride in their project accomplishments and have gained confidence in their engineering abilities.

## References:

1. UNO R3 Control Board Product Page, <http://www.sunfounder.com/index.php?c=show&id=50&model=UNO%20R3> , accessed on January 31, 2016.
2. RedBearLab, BLE Shield Product Page, <http://redbearlab.com/bleshield/> , accessed on January 31, 2106.
3. Wheelhouse, Ian, ECE 4800, Utah Valley University, Final Report, April 2015.
4. Worwood, Anthony, "Smart Homes with Smart Phones", Final Report, April 2015.
5. Arduino website, <https://www.arduino.cc/> , Accessed on 1-31-2016.
6. Barrett, Steven and Daniel J. Pack, "Embedded Systems Design and Applications with the 68HC12 and HCS12" Prentice Hall, 2005.
7. Minaie, Afsaneh and Reza Sanati-Mehrzy, "Embedded Systems Capstone Projects in the Computer Engineering Area of Specialization within the Computer Science Department", Proceedings of American Society for Engineering Education, 2009.
8. ABET, Inc.: Criteria for Accrediting Engineering Programs; [http://abet.org/uploadedFiles/Accreditation/Accreditation\\_Process/Accreditation\\_Documents/Current/eac-criteria-2012-2013.pdf](http://abet.org/uploadedFiles/Accreditation/Accreditation_Process/Accreditation_Documents/Current/eac-criteria-2012-2013.pdf) , 2013.
9. Minaie, Afsaneh, "Wireless Sensor Network Projects in a Computer Engineering Program", Proceedings of American Society for Engineering Education, 2014.
10. Prairie, Michael, and Ronald Lessard, "Introducing Systems Engineering Concepts in a Senior Capstone Design Course", Proceedings of American Society for Engineering Education, 2012.
11. Huang, Brian, "Open-Source Hardware – Microcontrollers and Physics Education- Integrating DIY Sensors and Data Acquisition with Arduino", Proceedings of 122nd ASEE Annual Conference and Exposition, 2015.
12. Dieter, George, and Linda Schmidt, "Engineering Design", Fourth Edition, McGraw-Hill, 2009.
13. Hopkins, Mark, and Alexander Kibbe, "Open-Source Hardware in Controls Education", 121st ASEE Annual Conference & Exposition, 2014.