AC 2012-4446: COMPUTER ENGINEERING CAPSTONE PROJECTS IN THE COMPUTER SCIENCE DEPARTMENT

Dr. Afsaneh Minaie, Utah Valley University

Afsaneh Minaie is a professor of computer science at Utah Valley University. Her research interests include gender issues in the academic sciences and engineering fields, embedded systems design, mobile computing, wireless sensor networks, and databases.

Mr. Ali Sanati-Mehrizy, Pennsylvania State University
Mr. Paymon Sanati-Mehrizy, University of Pennsylvania

Paymon Sanati-Mehrizy is currently a senior at the University of Pennsylvania, studying biology. Currently, his research interests consist of higher education curricula, including within the field of wireless sensor networking. After graduation, Sanati-Mehrizy plans to attend medical school.

Dr. Reza Sanati-Mehrizy, Utah Valley University

Reza Sanati-Mehrizy is a professor of the Computing Sciences Department at Utah Valley University, Orem, Utah. He received his M.S. and Ph.D. in computer science from the University of Oklahoma, Norman, Okla. His research focuses on diverse areas, such as database design, data structures, artificial intelligence, robotics, computer integrated manufacturing, data mining, data warehousing, and machine learning.

Computer Engineering Capstone Projects in the Computer Science Department

Abstract

As with many computer science and engineering programs, students of the computer engineering area of specialization in the computer science program at Utah Valley University (UVU) conclude their degree programs with a semester 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. This paper presents the details of sample projects that the students have done in this capstone course.

Background Information

Utah Valley University (UVU) is a state institution with more than 32,000 students. UVU is located in Utah County which has a population of over 430,000 residents, 78% of which are members of the Church of Jesus Christ of Latter Day Saints. The Computer Science department at UVU offers a Bachelor's Degree in Computer Science with four areas of specialization, which include Computer Science (traditional), Computer Engineering, Database Engineering and Computer Networking. The Bachelor of Science in Computer Science program was one of the first Bachelor of Science programs implemented at UVU in 1993. The program's goal has been to provide a quality program that meets accreditation standards while providing the students with a skill set that allows them to succeed in computing careers. The curriculum content for the Computer Science degree is based on the 2001 ACM Curriculum Report. The Computer Science degree at UVU was accredited by Accreditation Board for Engineering and Technology (ABET) in 2002 and currently has more than 800 students. Students in this program take core courses until the first semester of their junior year, when they begin choosing their electives for different specialization areas.

Computer Engineering Curriculum at UVU

The UVU computer engineering area of specialization relies on a core curriculum shared with the other areas of specializations within the computer science department. It consists of introductory courses in programming, discrete mathematics, data structures, operating systems, networking, computational theory, and computer architecture. Computer engineering students specialize with additional second-tier courses in electrical engineering, embedded systems, digital signal processing, digital systems design, as well as a capstone design course that requires an embedded design project. The embedded software, digital design, and capstone design courses give the computer engineering program its character. They integrate software and hardware design skills and prepare students to build modern digital systems from start to finish. In the embedded systems course, students learn to use microcontrollers and their interface effectively to build systems that controls physical devices. The digital design course teaches them to program algorithms into hardware. In the capstone design course, the students apply their skills to a product that we try to make similar to those of which industry engineers are currently working on. It is required for our computer engineering students to do an embedded systems project.

Embedded Systems

An embedded system is a computing machine that is generally a component of some larger product, and its purpose is narrowly focused on supporting that product. One can say that it is a computing machine dedicated to a specific purpose[1]. "Embedded systems cover a large range of computer systems from ultra-small computer-based devices to large systems monitoring and controlling complex processes[2]". Today, 99% of all computing systems belong to embedded systems[3]. Embedded systems are ubiquitous and have pervaded most aspects of modern life. Every week millions of tiny computer chips are manufactured and find their way into our everyday products. In the US, homes have an average of 30 to 40 microprocessors or microcontrollers each. Embedded Systems is one of the most dynamic, fast growing areas in industry. Embedded systems design addresses the challenges of hardware and software co-design[4].

The area of Embedded Systems Design has undergone tremendous growth in recent years. A major contributor of this growth has been the addition of networking technologies, database management systems, and operating systems to embedded systems. Embedded systems have application in many areas, including automotive/transportation, government/military, medical equipment, telecommunications, avionics/aeronautics, aerospace electronics, office automation, data-communication, industrial automation, and consumer electronics[5].

The design of embedded systems has been around for more than thirty years. However, the academic subject of embedded systems design is a relatively new subject. It is considered to be an interdisciplinary field combining areas such as computer science, electrical engineering, applied mathematics, and control theory[4].

The tremendous growth in embedded computing has given rise to a demand for engineers and computer scientists with experience in designing and implementing embedded systems. Embedded system design is currently not yet well represented in academic programs. Most computer engineering programs teach programming and design skills that are appropriate for a general-purpose computer operating under the control of a commercial operating system rather than for the more specialized embedded systems[6,7].

With the ongoing integration of Embedded Systems in the everyday environment, the demand for engineers schooled in the domain of these systems is ever present. This makes it essential for universities to lay down the foundations for understandings of the field of embedded systems.

In order to prepare our computer engineering students for the embedded systems design experience, we offer a senior level course on embedded system design. However, this single course on embedded systems design is not sufficient to teach the students the skills that they need. To satisfy the ABET requirements; our students are required to take a capstone design course. The focus of our computer engineering capstone design class has been the design of embedded systems. By requiring an embedded design project in our capstone course, our students are getting the experience that is needed in the area of embedded system design as well as the capstone design experience [2].

Sample Capstone Projects at UVU

The goal of projects in our Capstone Design course is to provide our students with a realistic embedded design experience and to teach them the tools and methodologies that can help them be successful. Our senior design course is structured as a collection of independent student projects. This course is offered every semester. The students in the Computer Engineering area of specialization take this course during their last semester. Students either can come up with an embedded project themselves or work on a project that is given to them by their advisors. Students write a proposal to define problems and identify solution approaches for their project and the hardware and software that is needed for their project. After several iterations, the advisor approves their project proposal. Then, they start working on their projects. Students are required to write weekly progress reports and meet with their advisor during a weekly scheduled time for each student. At the end of the semester, they turn in a final written report and a final presentation which is evaluated by several faculties from the department. The follow are sample Senior Design Projects which reflect common student projects[2].

Sample Project 1: Teleoperated Rover

The objective of this project was to design a remotely-controlled and highly mobile robot to allow an individual to remotely see and interact with others or the environment. The robot chosen for this project was the Lynxmotion Rover robot[1] (Figure 1). The robot chassis was made from heavy-duty anodized aluminum structural brackets and laser-cut lexan panels. It used four 12.0 V dc gear head motors and 4.75 "tires and wheels"[8].

Figure 1: The Rover[8]

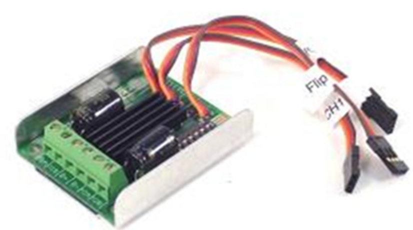

Figure 2: Sabertooth Motor Controller [8]

The motors were controlled by a Sabertooth 2x10 R/C motor controller (Figure 2) which is in turn connected to an SSC-32 servo controller (Figure 3), which received commands from a Synapse SNAP Proto Board.

Figure 3: SSC-32 Servo Controller[8]

The Synapse SNAP wireless system was chosen for this project. SNAP is an Internet-enabled, IEEE 802.15.4-based, auto forming, and multi-hop mesh network. The SNAP protocol can support up to 16 million nodes in a single network. Any node can talk to any other node that is in range and any node can talk indirectly to any other node via intermediate nodes. The nodes are all programmed over-the-air.

Figure 4: SNAP RF Engine (Node)[10]

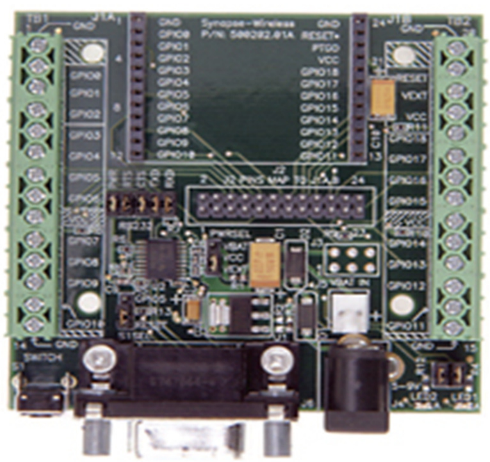

Figure 5: ProtoBoard SN171 SNAP Wireless System[10]

The Proto Board (Figure 5) communicated with a Synapse Bridge device using an 802.15.4 wireless mesh network. The Bridge device was then connected to a computer, where users, utilizing a graphical user interface (Figure 6) and a streaming video feed from an attached camera (Figure 5), can remotely control the robot rover[9].

Figure 5: The Camera [9]

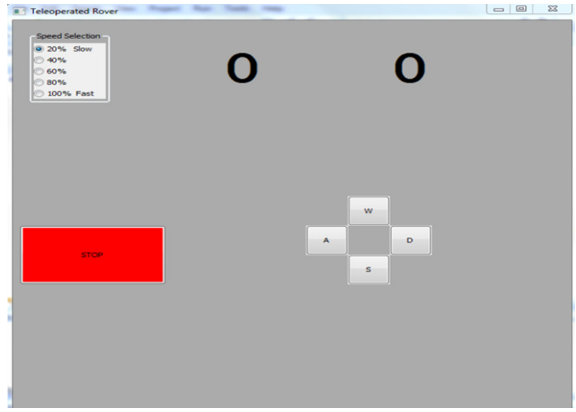

Figure 6: Graphical User Interface [9]

The interface for the SNAP network is called Portal. Portal is a program that allows the user to see and communicate with all of the nodes on the network. Each node can be programmed wirelessly through Portal almost instantaneously. Portal itself can also be used as a standalone node on the network [7]. Portal contains a Python editor in which users can develop applications that can then be directly loaded into nodes.

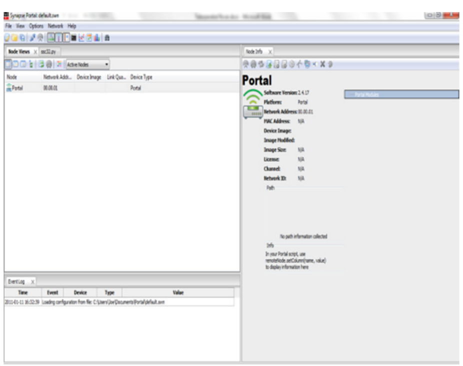

Figure 7: Portal[9]

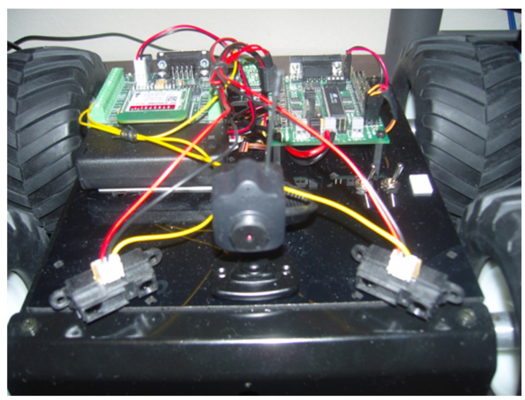

Figure 8: Teleoperated Rover Front View[9]

The heart of each SNAP node is the RF Engine (Figure 4). These engines are small low power transceiver modules that run at 2.4 GHz. Each RF Engine is loaded with the SNAP network operating system. The RF100 SNAP Engines with external amplifiers were used in this project. These RF Engines have a line of sight of up to three miles. Python was used as the language for programming of this project. Three different programs were created for this project: a GUI interface for remote control, a Proto Board program for the remote control, and a Proto Board program for autonomous mode. Figure 8 shows a picture of the Teleoperated Rover's front view and Figure 9 shows inside of the Teleoperated Rover.

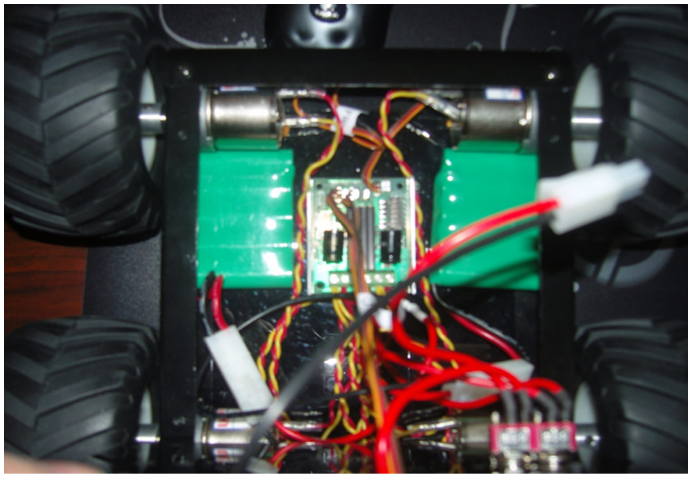

Figure 9: Teleoperated Rover Inside[9]

The student responsible for this project was incredibly satisfied with his capstone project, stating that "I had a lot of fun with this project. Not only did I learn a lot, but I had a good time doing it. I feel that this project was a real success.[9]"

Sample Project 2: Autonomous Quad-Copter UAV

The goal of this project was to build an Autonomous Quad-Copter UAV (Unmanned Aerial Vehicle). The Autonomous Quad-Copter UAV is a self-controlled aircraft using four motors/propellers and inertia-measuring sensors. Quad-Copters are notable for having no moving parts other than their motors, which makes them simple and robust. ArduCopter is the Arduino-based quad-copter UAV[10] (Figure 10). The ArduCopter is an open-source hardware (OSHW) project contributed to by many individuals. OSHW is an emerging concept similar to Free and Open Software (FOSS). The concept of OSHW allows sharing of design files, schematics, board-layouts, source codes, and HDL modules. An OSHW project allows others to study and learn and derive other projects from the originals. The Quad-Copter is a derivation of ArduCopter, which is an OSHW[12].

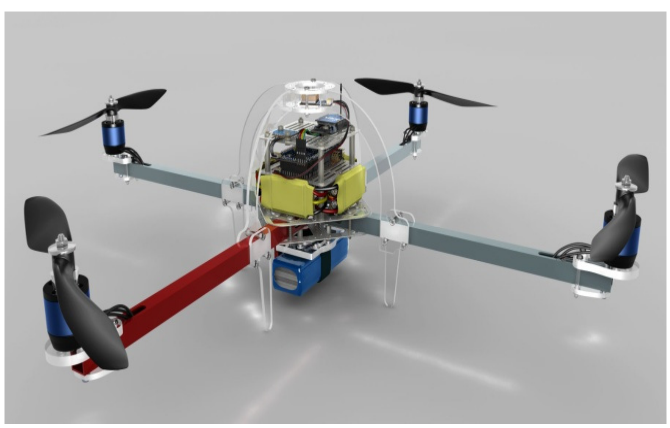

Figure 10: ArduCopter[11]

ArduCopter Platform Feature List[11]

- “6 Degree of Freedom IMU stabilized control
- Gyro stabilized flight mode enabling acrobatics (loops and barrel rolls)
- GPS for position hold
- Magnetometer for heading determination
- Barometer for altitude hold
- IR sensor integration for obstacle avoidance
- Sonar sensor for automated takeoff and landing capability
- Automated waypoint navigation
- Motor control using low cost standard PWM Electronics Speed Controllers (ESC's)
- On board flight telemetery data storage
- Mounted camera stabilization capability
- Wireless command & telemetry for long distance communication
- Capability to fly in "+", "x", quad, hexa and octo configurations
- Battery level detection
- User configurable LED flight pattern
- Capability to use any R/C receiver
- ArduCopter Configuration and Ground Control Software
- Realtime graphs of flight data
- GUI for configuration of PID and other flight parameters”

As can be seen from the above list, this UAV is a complex system, and the exposure of undergraduate students to complex system is very beneficial to their education. This project was not built using an ArduCopter kit. The Quad-Copter was built by purchasing the parts separately and designing some of the module. Figure 11 shows a sensor board that is called ArduPilotMega IMU Shield board has an accelerometer, Gyros (triple axis), a data logger, and a pressure sensor. Figure 12 depicts the microcontroller board for the UAV and it is called ArduPilot Mega and Figure 13 shows a picture of the Autonomous Quad-Copter UAV that was built by our student.

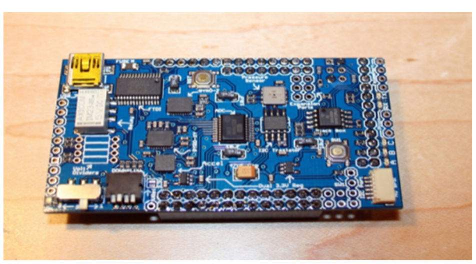

Figure 11: ArduPilotMega IMU Shield board[12]

Figure 12: ArduPilot Mega (Based on a 16MHz Atmega1280 processor) [12];

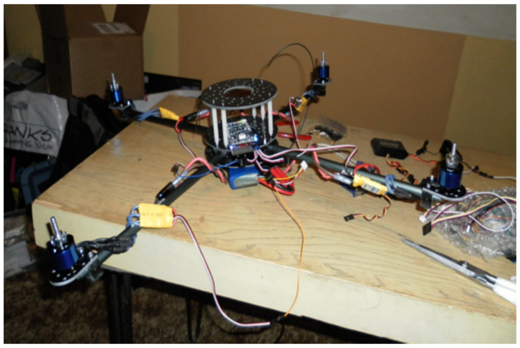

Figure 13: Autonomous Quad-Copter UAV[12]

The Quad-Copter functioned properly; however, it did not have the thrust to fly higher than a few inches. The reason for it was that the kit was not purchased and the student bought parts separately, and some of the parts were heavier than the kits parts. This student commented after finishing his project that "even though my Quad-Copter did not fly, I had a lot of fun with this project and it was an excellent learning experience[12]."

Sample Project 3: Speech Synthesis and Voice Recognition for Robot Communication

This project involved the utilization of speech synthesis and voice recognition to control robots. Two robots were used in this project one referred to as Robonova (Figure 14) and the other as Boe-Bot (Figure 15). The Boe-Bot robot is a simple two-wheeled robot. The brain of this robot is BASIC Stamp II microcontroller, and it can be programmed in BASIC language. The Robonova is a humanoid robot that can be programed for user defined movements. The Robonova robot uses 16 servos. The Robonova robot can be programmed using BASIC programming language. The brain of the ROBONOVA-1 is the ATMEL ATMega128 processor microcontroller. The ATMega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture.[13]

Figure 14: The Robonova Robot[13]

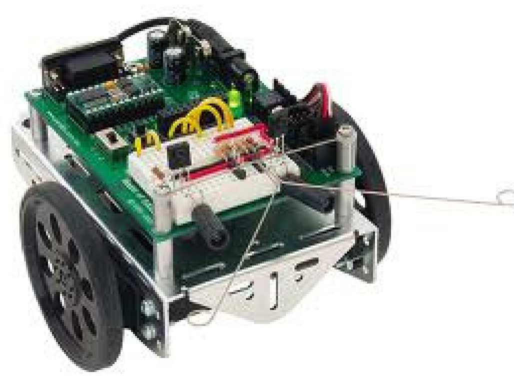

Figure 15: The Boe-Bot Robot[13]

A SpeakJet chip, which is a completely self-contained, single chip voice and complex sound synthesizer, was interfaced with the Boe-Bot robot. This chip was used to synthesize a human voice to generate words and phrases to control the Robonova robot. The voice that was generated from the SpeakJet chip was truly synthetic sound because it didn't use any recordings of a human voice; however, the sound was generated from mixing sound of the oscillators inside of the chip. SpeakJet chip was placed on the education board and was controlled by BASIC Stamp microcontroller. BASIC Stamp II sends the series of commands through TX pin into RCX in of SpeakJet to generate the speech and sound[13].

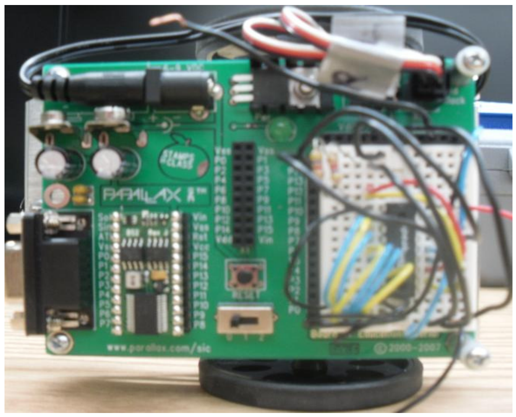

Figure 16: Boe-Bot Robot with SpeakJet chip[13]

In order to give the Robonova robot the ability to recognize and react to Boe-Bot's voice command, a VRbot module was connected to it. The VRbot module used UART interface to communicate with MR-C3024 servo controller board in the Robonova robot.

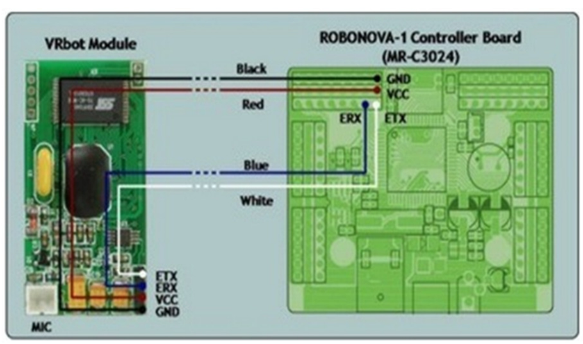

Figure 17: Interfacing VRbot to Robonova-I [13]

This project was successful and the student's comment after finishing his project was positive.

Educational Benefits and Assessment

The students in the Computer Engineering area of specialization in the Computer Science department at UVU are required to take the Senior Design Project course (EENG 4800). The senior design projects are completed in a semester. The purpose of this course is to emphasize major hardware and software design identification and completion. A suitable topic for the design project will be mutually selected by the faculty supervisor and the student. The objectives of this course are:

- Identify relevant topics from previous courses and then applying them to their project.
- Identify and specify design requirements from general problem descriptions.
- Communicate design ideas and information.
- Demonstrate creative thinking.
- Develop information gathering skills
- Demonstrate oral and written communication skills.

The faculty adviser will meet with each student individually on a weekly basis at a regularly scheduled, mutually agreeable time. At each meeting, issues associated with the project will be discussed and a status report will be provided to the adviser. 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. Faculty adviser 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. Weekly schedule is given in Table 1.

Week	Activity
1	Submission of project proposal
2	Finalization and acceptance of project proposal by faculty adviser
3	Project design, phase 1 (review of literature, consultation with adviser, acquisition of project resources)
4	Project design, phase 1 (review of literature, consultation with adviser, acquisition of project resources)
5	Project design, phase 2 (develop design specification)
6	Project design, phase 2 (develop design specification)
7	Project execution
8	Project execution
9	Project execution
10	Project execution
11	Project execution
12	Write research report
13	Write research report and develop oral presentation
14	Submit written project report to faculty adviser
15	45-minute oral presentation of project results

Table 1; Weekly Schedule

In the last week of the semester, students present their projects to the department's faculty and students. Each faculty member in attendance will evaluate the oral presentation using the following rubric, with the maximum number of points for each area of assessment as follows:

Quality of verbal communication	30 points
Quality and utilization of visual aides	20
Quality of project demonstration	20
Quality of responses to faculty questions	20
Relevance to student's area of specialization	10
Total possible points	100

These evaluations of the oral presentation will be averaged and will determine the number of points allocated in the 20-point "Quality of oral presentation category" in the final grade. Students registered for this course will also take a program assessment test. The test currently administered is the Major Assessment Test for Computer Science (MFAT). The project final grade is calculated on a 100 point scale, with the maximum number of points for each area of assessment as follows:

Quality and rigor of research and research paper	25 points
Quality of other deliverables (hardware design, integration, management plan, code, etc.)	15
Degree to which project fulfills requirements specified in project proposal	20
Level of effort	10
Progress reports received on time	10
Quality of oral presentation	20
Total possible points:	100

The funding for these projects has been supported through grant proposals written by the faculty member running the course. There are several benefits associated with these projects, including[4]:

- The technical knowledge gained in putting to use their prior background in computer science and electrical engineering coursework.
- Using concepts from classes in Embedded Systems Design, Electronics, Networking, Digital Design, Circuit Theory, Data Structures, Computer Architecture, and Programming classes.

- Knowledge gained from Hardware and Software interfacing and integration.
- Use of programming languages such as C, C#, Assembly, Verilog, and JAVA.
- Learning microcontroller's architectures.
- Using new IDEs.
- Learning how to define system requirements, partition the design into subcomponents, design, build, test, and verify that the system requirements have been met.
- Developing project management skills.
- Developing written and oral communication skills and other professional skills.
- Designing and defending a solution to a real world problem.

This course has consistently rated highly in end of the semester student evaluations. Overall, students have positive comments about this course and are benefitted greatly from this experience.

Summary and Concluding Remarks

This paper discusses senior design projects completed by senior-year students in the undergraduate computer engineering area of specialization in the computer science department. This senior design course is structured as a collection of independent student projects. In order to ensure they have a successful senior design project, our students work rigorously to employ the technical expertise and theoretical knowledge gained during years of study. As a result, this opportunity for self-directly learning is high rewarding, as each student must design, build, and troubleshoot their fully functional embedded project. Throughout their final semester, these students develop the ability to debug, seek, and find information necessary to both understand and reverse-engineer poorly written documentation. Overall, both student feedback and final presentations indicate that these students have pride in their project accomplishments and have gained confidence in their engineering abilities.

References:

1. Sass, Ron and Andrew G. Schmidt, "*Embedded System Design with Platform FPGAs*", Elsevier, 2010.

2. Crnkovic, Ivica, "*Component-based Software Engineering for Embedded Systems*", ICSE 05, May 15-21, 2005, St. Louis, Missouri.

3. Roadmaps on Selected topics in Embedded Systems Design from the EU IST ARTIST project, http://www.artist-embedded.org/Roadmaps/index.html .

4. Minaie, Afsaneh and Reza Sanati-Mehrizy, "*Embedded Systems Capstone Projects in the Computer Engineering Area of Specialization within the Computer Science Department*", Proceedings of 2009 American Society for Engineering Education (ASEE).

5. Haberman, Bruria and Mark Trakhtenbrot, "*An Undergraduate Program in Embedded Systems Engineering", Proceedings of the 18th Conference on Software Engineering Education & Training* (CSEET'05), 2005.

6. IEEE Computer Society, www.computer.org/education/cc2001, retrieved on January 10, 2009.

7. Jackson, David and Paul Caspi, "*Embedded Systems Education: Future Directions, Initiatives, and Cooperation*", SIGBED Review, Volume 2, Number 4, October 2005.

8. http://www.lynxmotion.com/c-119-auton-combo-kit.aspx, Retrieved on January 3, 2012.

9. Walker, Joseph, "*Teleoperated Rover*", EENG 4800, Final Report, April 2010.

10. Synapse, "SNAP Reference Manual", 2010.

11. Ardu-Copter, http://code.google.com/p/arducopter/ , Accessed on Jan 10, 2012.

12. Harward, Nathan, "*Autonomous Quad-Copter UAV*", EENG 4800, Final Report, April 2011.

13. Rabpornpra, Piyapoom, "*Speech Synthesis and Voice Recognition for Robot Communication*", EENG 4800, Final Report, April 2011.