

# Embedded System Design

## ECEN 4613/5613 – Fall 2009

Lectures: Wednesday Evenings, 5:30pm-8:00pm, ECEE 1B28

Instructor: Professor McClure, Department of Electrical and Computer Engineering

E-mail and Office: **Linden.McClure@Colorado.EDU**, ECEE 2B30A (inside 2B37), 970-898-0946

Instructor Office Hours: Wednesdays 8:00pm-8:30pm and by appointment; alternating Saturdays 12-3pm

Course Web Site: <http://ece.colorado.edu/~mcclurel/index.html> ECEE 2B37 Lab Phone: 303-492-4946

TAs: [Brandon.Gilles@Colorado.EDU](mailto:Brandon.Gilles@Colorado.EDU) . [Riley.Pack@Colorado.EDU](mailto:Riley.Pack@Colorado.EDU)

TA Office Hours: weekdays, evenings, weekends, and by appointment

Office Hour Schedule: [http://ece.colorado.edu/~mcclurel/office\\_hours.html](http://ece.colorado.edu/~mcclurel/office_hours.html)

## Course Description

In this class, the fundamentals of embedded system hardware and firmware design will be explored. Issues such as embedded processor selection, hardware/firmware partitioning, glue logic, circuit design, circuit layout, circuit debugging, development tools, firmware architecture, firmware design, and firmware debugging will be discussed. The Intel 8051, a very popular microcontroller family, will be studied. The architecture and instruction set of the microcontroller will be discussed, and a wirewrapped microcontroller board will be built and debugged by each student. The course will culminate with a significant final project which will extend the base microcontroller board completed earlier in the course. Learning may be supplemented with periodic guest lectures by embedded systems engineers from industry. Depending on the interests of the students, other topics may be covered.

## Required Background

Knowledge of microprocessor architecture and assembly language, microprocessor peripherals, digital design, and the C programming language is a prerequisite for this course. The corresponding CU-Boulder courses are ECEN 2120, ECEN 3100, and CSCI 1300. Although not listed as formal prerequisites, circuits/electronics (ECEN 3250) and computer organization (ECEN 4593) are highly recommended. An understanding of compilers, assemblers, linkers, operating systems, analog design, diodes, transistors, and electromagnetic fields and waves will be useful. Refer to the course FAQ for more information.

## Course Context

Embedded systems are involved in almost every facet of modern life. Cell phones, MP3 players, pagers, PDAs, energy conversion systems, answering machines, microwave ovens, televisions, VCRs/DVRs, CD/DVD players, video game consoles, GPS devices, network routers, fax machines, cameras, music synthesizers, planes, spacecraft, boats, and cars all contain embedded processors. Late model cars may contain as many as 65 embedded microprocessors, controlling such tasks as antilock braking, climate control, engine control, audio system control, airbag deployment, etc. Logic analyzers and digital storage oscilloscopes utilize embedded processors to support real-time operation. Even PCs, which are designed around powerful CPUs, contain embedded systems. Storage drives (floppy, hard disk, solid state, CD-RW, DVD+RW, Blu-ray), and external peripherals such as printers, scanners, and other SCSI, SAS, SATA, USB, or IEEE 1394 devices all contain embedded processors. During 2007, microprocessor manufacturers sold on the order of 100 million processors for use as computer CPUs. In comparison, during the same time frame, microprocessor manufacturers sold more than **3 billion** embedded processors, primarily consisting of 32-bit, 16-bit, 8-bit, and 4-bit devices. The tremendous number of applications for embedded computing has given rise to high demand for engineers with experience in designing and implementing embedded systems. This course will give students hands-on experience and opportunities for experimentation in this exciting field.

## Course Mechanics

This course is meant to be a hands-on type course, giving students a chance to hear and read about embedded system topics, and then put those concepts to work by developing and debugging embedded system hardware and firmware. Student participation in active discussions of the course topics will be expected. Lecture periods will include a short break sometime in the middle. The course grade will be based on class attendance and participation, lab assignments, presentations, quizzes, teamwork, and an embedded system term project. Four structured lab assignments will be given. Lectures will be closely integrated with the lab assignments and will be organized to provide students with the information necessary to successfully complete each assignment. Students may work independently or in groups of up to three on the term project. Team members will be expected to share the workload equally. Various homework assignments will be given to guide students through the course material, but most of these will be optional. The instructor and TAs will be available to help students during office hours, by appointment, and by e-mail. Students with questions should send e-mail to the TAs and the instructor to ensure the quickest response time. All e-mail correspondence related to this class should include the text "**ESDF09**" and a specific subject as the subject line of the message, so that e-mail may be filtered automatically. Course information and documents will be available on the course web site, which will be updated throughout the semester.

## Course Organization

The course has several goals. First, it will expose students to the field of embedded systems, and will provide a knowledge foundation which will enable students to pursue subsequent courses in real-time embedded systems software and computer design. Students will become familiar with the associated technical vocabulary and will learn about potential career opportunities in the field of embedded system design. Second, students will have the opportunity to develop an embedded system from the ground up, starting with electronic components and data sheets, and progressing through construction of hardware and implementation of firmware. This will provide students with an opportunity to gain a thorough understanding of the phases of embedded system development and familiarity with hardware and software development and debugging tools. Third, students will be given the opportunity to develop design skills, through well-bounded design assignments as well as open-ended design assignments. Fourth, students will have the opportunity to learn how information gained in multiple other core engineering classes comes together to be applied to real-world design. Fifth, students will be given an opportunity to experience embedded system design in a manner similar to that practiced in industry, and will gain knowledge beneficial for obtaining a job in this field.

The course will be structured around several key lab assignments and the final project. During the first part of the course, students will focus primarily on basic embedded system concepts, and will develop a basic hardware platform consisting of an 8051 microcontroller family derivative and supporting circuitry. At the same time, students will become exposed to the 8051 instruction set, and learn how to use a cross assembler and simulator to develop code. During the middle of the course, students will focus more on firmware concepts, and will develop code in assembly and C to control the basic hardware. In addition, during this period students will add additional hardware elements to their boards, and will develop the firmware to control this new hardware. During the final weeks of the course, students will focus on significant projects, and will proceed through design, development, documentation, and presentation of their work. Although the course is scheduled for Wednesday evenings, lectures may not be given on all of these days during the semester; instead, one to three class periods may be used to provide students with additional time to work on their development assignments. In order to give students perspective from multiple viewpoints, class discussions on several topics will be pursued. Guest speakers may discuss embedded systems topics during the semester.

# Tentative Syllabus

Note: The following syllabus is tentative, and is provided to give insight into the types of topics to be discussed during the semester. However, not all topics will be discussed in the order given or on the dates shown. **Adjustments will be made as the course progresses.**

## **Week 1: August 26**

- Course overview, expectations, logistics, processes, syllabus, FAQ, and prerequisite material.
- Embedded systems descriptions, definitions, and vocabulary. Design Engineer's notebook.
- Design considerations and requirements, processor selection and tradeoffs.
- Overview of board development process, wire wrapping vs. soldering.
- Microprocessor/microcontroller architectures and instruction sets, 8051 architecture, busses.
- Lab access, Buff OneCards (access to 2B37 and east ECE wing door), computer accounts.

## **Week 2: September 2**

- Design cycle, planning a development project, derivation of requirements, tradeoffs.
- 8051 instruction set, ASM51 assembler and Emily52 simulator. Code development process.
- Examples of assembly code, discussion of mnemonics, calculation of execution time.
- Device programmers, EPROM emulators, Intel hex records and Motorola S-records. SPLDs.
- Schematics and wiring diagrams, recommended practices, CAD tools. Parts kits.

## **Week 3: September 9** *(Finish Lab #1 this week)*

- Board layout considerations, signal integrity (noise, crosstalk, etc.), decoupling, techniques.
- Manufacturing and test engineering, PCB design, ground and power planes, EMI, EMC.
- Data sheets, power supplies, voltage regulators. Thermal considerations, heat sinks.
- Oscillators and reset circuits. Microprocessor supervisory circuits, watchdog timers.
- Development and debugging strategies and techniques. Logic probes, voltmeters and oscilloscopes.
- Parts kits. Introduction to Embedded Systems Laboratory, equipment, and soldering.

## **Week 4: September 16** *(Lab #1 submission due date)*

- Designing with tolerances and margins, part variations and substitutions, reliability/part count.
- Interfacing different logic families, fanout, signal buffering, noise margins, pullups/pulldowns.
- Microcontroller peripherals, selection and interfacing. Core component circuitry ( $\mu$ P, ROM, RAM).
- 8051 timing diagrams, program read, data read, data write.
- Debugging using logic analyzers, state and timing information.
- Port pin structure. Controlling port pins in asm. User interface design, human factors. Driving LEDs.

## **Week 5: September 23**

- No lecture. SMT Lab?? Work on Lab #2 and current topics ideas.

## **Week 6: September 30** *(Finish Lab #2 this week)*

- Timing requirements, propagation delay, setup, hold, rise/fall times, timing analysis. Clock skew.
- Memory selection and interface, SRAM, NVRAM, DRAM, EPROM, EEPROM, Flash.
- Memory maps, decoding logic, glue logic, programmable logic (PALs, FPGAs).
- Switch debouncing in hardware and firmware, keypad decoding. 8051 timers/counters.
- Interrupts and Interrupt Service Routines (ISRs). Estimating code bandwidth requirements.
- Class eval? Intro to Atmel AT89C51RC2.

## **Week 7: October 7** *(Lab #2 submission due date)*

- Serial communication, RS-232/485, line drivers/receivers, charge pumps, terminal emulation, USB.
- Cross-assemblers, cross-compilers, linkage editors, disassemblers, other software tools.
- High level software in embedded systems. Embedded cross compilers, operating systems, software development tools. Intro to SDCC and MICRO-C, Emily52 simulator, makefiles, and Eclipse IDE.

- Monitors, in-circuit emulators, debuggers, monitors, software engineering, debugging using software.
- C variables, bit operations, pointers. Interrupts in C. Interfacing C and assembly.

### **Week 8: October 14**

- Software development, version control, coding standards, code reviews. Work on Lab #3.

### **Week 9: October 21** (Finish Lab #3 this week)

- EEPROMs and synchronous serial communication (I<sup>2</sup>C, SPI, etc.).
- LCDs.

### **Week 10: October 28** (Lab #3 submission due date)

- Final Project Design Review (PDR). Each project team presents development plan and milestones.

### **Week 11: November 4**

- Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs).
- Motor control, stepper motors, DC motors, PWM, H-Bridges. Case study: hard disk drive.
- Firmware design, main loop designs, interrupt driven firmware, device drivers.
- Jump tables, POST, memory testing, Little/big endian issues, math functionality.

### **Week 12: November 11** (Finish Lab #4 this week)

- Guest Speaker? **Student current topic presentations?** Work on final projects.

### **Week 13: November 18** (Lab #4 submission due date)

- Guest Speaker? **Student current topic presentations?** Work on final projects. Informal evaluations?

### **Week 14: November 25** (Fall Break)

- No lecture, class cancelled.

### **Week 15: December 2**

- Last lecture, topics TBD. Current events, emerging technologies, migrating C code.
- Semester wrap-up. Review of vocabulary. Course evaluations (FCQs).
- Early student demo

### **Week 16: December 9**

- Last Class. Final project presentations. Students will demonstrate their final projects to the class in the embedded systems laboratory. In order to complete all presentations, it may be necessary to extend the class period until 9:00pm on this evening.

### **Week 17: No Final Exam**

Lab Assignment Overview	Signature Due Date	Submission Due Date
Lab #1: Basic hardware, SPLD, 8051 assembly, simulator.	9/15/2009	5:30pm 9/16/2009
Lab #2: Decode logic, EPROM, basic I/O, timer ISRs and assembly.	10/06/2009	5:30pm 10/07/2009
Lab #3: SRAM, RS-232, debug monitor, assembly, intro to 8051 C.	10/27/2009	5:30pm 10/28/2009
Lab #4: EEPROM, LCD, and C programming.	11/17/2009	5:30pm 11/18/2009
Final Project/Lab #5: Student's choice.	12/09/2009	12/12/2009

Lab assignments will be scored per the CU grading standards. Assignments which are not completed (signatures obtained) by the **Signature Due Date** will be late and will receive a late penalty deduction of 1 to 10 points. Assignments which are not submitted by the **Submission Due Date** will receive additional deductions. The final project may not be submitted late.

## Course Requirements

- Students are expected to keep up with the course material. If you get confused or start to fall behind, attend office hours or schedule an appointment with the professor or TA as soon as possible. The goal of the course is to allow you to learn the material, and not to stress you out. However, the longer you are confused, the more material you miss, so try to stay on top of things. It is fine to ask lots of questions, as long as you are putting in the effort to learn the material.
- It is the student's responsibility to obtain materials handed out in a lecture which the student missed.
- You are responsible for any damage or missing equipment resulting from your negligence.
- Treat all lab equipment with care, as it is expensive. If the equipment is damaged, we may not be able to afford replacements. No equipment may be removed from the lab.
- All homework and reports must be legibly written or typed. Sloppy work will receive deductions.
- All programming code must be well structured/commented. Code must be robust (error handling). Code and comment quality will be evaluated as part of each assignment's grading.
- Schematics must be well drawn and should follow the guidelines to be presented in class.
- When requesting help from the TA or the instructor, students must present a complete and accurate schematic of their circuitry. Update schematics as you add or change circuitry.
- A hard copy and an electronic copy of each final project report including schematics and source code will be submitted for grading and will become the property of the instructor.
- Students are expected to complete assignments on time. Lab assignments will be accepted late, but the grade earned on the assignment will be reduced. Since each lab depends on the results from the previous labs, students should be careful not to fall behind. Students are responsible for getting the TA or instructor to sign off on their lab work prior to the due date. Due to limited lab station availability, it is wise to plan ahead. Consider scheduling an appointment with the TA.
- University policies on academic integrity (<http://www.colorado.edu/academics/honorcode/>) will be followed. Cheating and plagiarism will not be tolerated. Credit must be clearly given for code or hardware designs legally borrowed from others. Submission of project work performed previously or concurrently for a different course constitutes cheating, if instructor consent is not obtained prior to submission. **When in doubt, ask the instructor for clarification.**
- Students are expected to maintain a design engineer's notebook. This notebook should contain class notes, lab notes, designs, and references. This notebook must be legible and should be written in ink.
- Students are expected to participate in class discussions of course topics. In addition, students are expected to assist other students in understanding course material and assignments. Students who are experts in a particular area of embedded systems may choose to give a short presentation to the class as part of their class participation grade.
- In lieu of a required text, students should expect to spend some amount of money to purchase supplies for the class, including hardware, tools, integrated circuits, discrete components, and other parts for the final project.
- Students will be expected to obtain data sheets from the course web site or various manufacturers' web sites, and print them out at CU, their place of work, or at home.
- If you must miss a lecture, please let the instructor know in advance, if possible.

## Academic Accommodations

Students with disabilities who qualify for academic accommodations must provide a letter from Disability Services (DS) and discuss specific needs with the professor as soon as possible, preferably during the first two weeks of class. DS determines accommodations based on documented disabilities (303-492-8671, Willard 322). Other accommodations may be made in accordance with CU policies (<http://www.colorado.edu/policies/index.html>). It is the student's responsibility to notify the instructor of anticipated conflicts as early in the semester as possible so that there is adequate time to make necessary arrangements. If you have a temporary medical condition or injury, see the guidelines at <http://www.colorado.edu/disabilityservices/go.cgi?select=temporary.html>. Disability Services' letters for students with disabilities indicate legally mandated reasonable accommodations. The syllabus statements and answers to Frequently Asked Questions can be found at <http://www.colorado.edu/disabilityservices>.

## Religious Observances

Campus policy regarding religious observances requires that faculty make every effort to reasonably and fairly deal with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, notify the professor of anticipated conflicts as early in the semester as possible so that there is adequate time to make necessary arrangements. See policy details at [http://www.colorado.edu/policies/fac\\_relig.html](http://www.colorado.edu/policies/fac_relig.html)

## Honor Code

All students of the University of Colorado at Boulder are responsible for knowing and adhering to the academic integrity policy of this institution. Violations of this policy may include: cheating, plagiarism, aid of academic dishonesty, fabrication, lying, bribery, and threatening behavior. All incidents of academic misconduct shall be reported to the Honor Code Council ([honor@colorado.edu](mailto:honor@colorado.edu); 303-725-2273). Students who are found to be in violation of the academic integrity policy will be subject to both academic sanctions from the faculty member and non-academic sanctions (including but not limited to university probation, suspension, or expulsion). Additional information on the Honor Code can be found at <http://www.colorado.edu/policies/honor.html> and at <http://www.colorado.edu/academics/honorcode/>

## Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Students who fail to adhere to behavioral standards may be subject to discipline. Faculty have the professional responsibility to treat students with understanding, dignity and respect, to guide classroom discussion and to set reasonable limits on the manner in which students express opinions. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, culture, religion, politics, sexual orientation, gender, gender variance, and nationalities. Class rosters are provided to the instructor with the student's legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records. See policies at <http://www.colorado.edu/policies/classbehavior.html> and at [http://www.colorado.edu/studentaffairs/judicialaffairs/code.html#student\\_code](http://www.colorado.edu/studentaffairs/judicialaffairs/code.html#student_code)

## Discrimination and Harassment

The University of Colorado at Boulder policies on Discrimination and Harassment, Sexual Harassment, and Amorous Relationships apply to all students, staff and faculty. Any student, staff or faculty member who believes s/he has been the subject of discrimination or harassment based upon race, color, national origin, sex, age, disability, religion, sexual orientation, or veteran status should contact the Office of Discrimination and Harassment (ODH) at 303-492-2127 or the Office of Judicial Affairs at 303-492-5550. Information about the ODH and the campus resources available to assist individuals regarding discrimination or harassment can be obtained at: <http://www.colorado.edu/odh>, <http://www.colorado.edu/policies/discrimination.html>, and <http://www.cu.edu/policies/Personnel/sexharass.html>

## Grading

Expectations for students will be high. Student performance in this class will be compared to student performance across ECE undergraduate and graduate classes. A grade of 'A' will be reserved for students who have delivered outstanding work and who have clearly demonstrated a superior mastery of the course material. The majority of each student's course grade will be determined by the quality of the hardware and firmware assignments and the final project completed by the student during the semester. The rough weighting of each course element is shown below:

5%	Lab #1
10%	Lab #2
20%	Lab #3
20%	Lab #4
25%	Final Project
13%	Quizzes/Assignments, Lab Practical, PDR, Student Current Topics Presentations
7%	Class Participation, Attitude, Teamwork, Effort/Subjective

The normal CU grading standards as shown below will be applied to this class. See the following site for more information: <http://ece.colorado.edu/~mclclure/grading.html>

A	Superior, outstanding
A-	
B+	
B	Above average
B-	
C+	
C	Average, has adequately met course requirements
C-	
D+	
D	Below average
D-	Minimum passing grade
F	Fail, has not met course requirements

## References

The course will be taught using technical application notes, data sheets, and technical articles. For those students who desire additional references, a list is provided below. In addition, a tremendous amount of useful information can be found on the Internet. Documentation and links to useful web sites will be available on the course web site. A copy of the following books will be on reserve for the class in the Engineering Library on the CU-Boulder campus.

- "Debugging Embedded Microprocessor Systems" by Stuart R. Ball. Publisher: Butterworth-Heinemann. ISBN 0-7506-9990-6. Copyright 1998.
- "Embedded Microprocessor Systems: Real World Design" by Stuart R. Ball. Publisher: Butterworth-Heinemann. ISBN 0-7506-9791-1. Copyright 1996. Third edition ISBN 0-7506-7534-9. Copyright 2002.
- "Embedded Systems Design" by Steve Heath. Publisher: Butterworth-Heinemann. ISBN 0-7506-3237-2. Copyright 1997.
- "The Art of Designing Embedded Systems" by Jack G. Ganssle. Publisher: Butterworth-Heinemann. ISBN 0-7506-9869-1. Copyright 1999.
- "The Art of Programming Embedded Systems" by Jack G. Ganssle. Publisher: Academic Press. ISBN 0-12274880-8. Copyright 1992.
- "The Circuit Designer's Companion" by Tim Williams. Publisher: Butterworth-Heinemann. ISBN 0-7506-1756-X. Copyright 1991.
- "Programming Embedded Systems in C and C++" by Michael Barr. Publisher: O'Reilly & Associates, Inc. ISBN 1-56592-354-5. Copyright 1999.
- "An Embedded Software Primer" by David E. Simon. Publisher: Addison-Wesley. ISBN 0-201-61569-X. Copyright 1999.
- "Programming in C" by Stephen Kochan. Publisher: Hayden Books/Macmillan Computer Book Publishing Division. ISBN 0-672-48420-X. Copyright 1988.