Embedded System Design Syllabus

ECEN 5613 – Fall 2017

Lectures: Monday Evenings, 6:00pm-8:30pm, HUMN 1B90 (new room and time)
Instructor: Professor McClure, Department of Electrical, Computer, and Energy Engineering
E-mail: Linden.McClure@Colorado.EDU
Instructor Office Hours: TBD, available by Skype and Zoom
Course Web Site: http://ecee.colorado.edu/~mcclurel/index.html   Lab: ECEE 1B28
TAs: Praveen Gnanasekaran, Sanjana Kalyanappagol, Sandeep Kumbargeri
TA Office Hours: weekdays, evenings, weekends, and by appointment
Office Hour Schedule:  http://ecee.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 microcontroller architecture and instruction set will be discussed, and a wirewrapped microcontroller board will be built and debugged by each student. The TI MSP432 (ARM Cortex-M4F) microcontroller will also be studied and used in lab assignments. The course will culminate with a significant final project which will extend the concepts covered 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 and skills in 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 include ECEN 2120/2350, ECEN 3100/3350, and ECEN 5813. 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, tablets, MP3 players, virtual reality systems, 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, drones/UAVs, spacecraft, boats, and cars all contain embedded processors. Late model cars may contain more than 65 embedded microprocessors, controlling such tasks as antilock braking, traction and stability control, climate control, engine control, entertainment system control, navigation, airbag deployment, etc. The Boeing 777 aircraft contains over 1,200 processors and more than 4 million lines of software! 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 (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. In recent years, 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 individual 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 "ESDF17" 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.

This course requires the use of the Zoom conferencing tool which is currently not accessible to users using assistive technology. If you use assistive technology to access the course material, please contact your faculty member immediately to discuss.

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, students will add additional hardware elements to their boards, and will develop the firmware to control this new hardware. Students will also get exposure to the ARM architecture and gain experience with using an ARM development board to develop code. During the final weeks of the course, students will focus on significant projects (perhaps with a more advanced processor), and will proceed through design, development, documentation, and presentation of their work. Although the course is scheduled for one evening each week, 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. Lecture may be cancelled for lab sessions.

**Week 1: August 28**
- Course overview, expectations, logistics, processes, syllabus, FAQ, and prerequisite material.
- 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 #1 topics. Design cycle, planning a development project, derivation of requirements, tradeoffs.
- SPLDs. 8051 instruction set, ASM51 assembler and Emily52 simulator. Code development process.
- Work on software elements of Lab #1 this week.

**Week 2: September 4**  
**Labor Day Holiday**
- CU Holiday, no lecture on September 4th. Work on software elements of Lab #1 this week.

**Week 3: September 11**  
**Finish Lab #1 this week**
- Lab #1 & #2 topics. Board layout considerations, signal integrity (noise, crosstalk, etc.), decoupling.
- Data sheets, PCB power delivery, voltage regulators. Thermal considerations, heat sinks. EMI, EMC.
- Oscillators and reset circuits. Microprocessor supervisory circuits, watchdog timers.
- Development and debugging strategies and techniques. Logic probes, voltmeters and oscilloscopes.
- Schematics and wiring diagrams, recommended practices, CAD tools. CU Honor Code.
- Parts kits. Introduction to Embedded Systems Laboratory, equipment, and soldering.

**Week 4: September 18**  
**Lab #1 submission due this week**
- Lab #2 topics. Core component circuitry (µP, ROM, RAM). CTP topic preferences.
- Interfacing different logic families, fanout, signal buffering, noise margins, pullups/pulldowns.
- Microcontroller peripherals interfacing. 8051 timing diagrams, program read, data read, data write.
- Debugging using logic analyzers, state and timing information.
- Timing requirements, propagation delay, setup, hold, rise/fall times, timing analysis. Clock skew.
- Memory maps, decoding logic, glue logic, programmable logic (PALs, FPGAs). Memory technology.
- Switch debouncing in hardware and firmware, keypad decoding. 8051 timers/counters.

**Week 5: September 25**  
**Finish Lab #2 required elements this week**
- ARM overview. KiCad? Work on current topics presentations? Student/professor meetings?

**Week 6: October 2**  
**Lab #2 submission due this week**
- Lab #3 topics. Class eval. Intro to Atmel AT89C51RC2. C programming review?
- Serial communication, RS-232/485, line drivers/receivers, charge pumps, terminal emulation, USB.

**Week 7: October 9**
- Lab #3 topics. Cross-assemblers, cross-compilers, linkage editors, disassemblers, other tools.
- Intro to SDCC, makefiles, and IDEs (Code::Blocks, Eclipse).
- C variables, bit operations, pointers. Interrupts in C. Interfacing C and assembly.
- Software development, version control, coding standards, code reviews.

**Week 8: October 16**  
**Finish Lab #3 required elements this week**

**Week 9: October 23**  
**Lab #3 submission due this week**
- Lab #4 topics. EEPROMs and synchronous serial communication (I²C, SPI, etc.). LCDs.
**Week 10: October 30** *(Submit PDR presentations)*
- Final Project Design Review (PDR). Each project team presents development plan and milestones.

**Week 11: November 6** *(Finish Lab #4 required elements this week)*
- Guest Speaker? Student presentations. Work on final projects.
- Passive components. Designing with tolerances and margins.

**Week 12: November 13** *(Lab #4 submission due this week)*
- Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs).
- Code review exercise. Firmware design, main loop/interrupt driven designs, device drivers.
- Work on final projects. Debug session?

**Week 13: November 20** *(Fall Break Nov 20-24)*
- CU holiday, no lecture, class cancelled.

**Week 14: November 27**
- Guest Speaker? ARM architecture lecture? Informal evaluations? Student presentations?

**Week 15: December 4**
- Last lecture. Finish remaining student presentations, current events, emerging technologies.
- Semester wrap-up. Review of vocabulary. Course evaluations (on-line FCQs).

**Week 16: December 11**
- 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 after 9:30pm on this evening.

**Week 17: No Final Exam**

<table>
<thead>
<tr>
<th>Assignment Overview</th>
<th>Signature Due Dates</th>
<th>Submission Due Date</th>
<th>Cutoff Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab #1: Basic hardware, SPLD, assembly, simulator</td>
<td>9/13 – 9/20</td>
<td>9/21/2017</td>
<td>9/27</td>
</tr>
<tr>
<td>CTP: Current Topics Presentation – topic preferences</td>
<td></td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Lab #2: Decode logic, EPROM, timer ISRs</td>
<td>9/29 – 10/04</td>
<td>10/05/2017</td>
<td>10/11</td>
</tr>
<tr>
<td>Lab #3: SRAM, RS-232, assembly, intro to 8051 C</td>
<td>10/20 – 10/25</td>
<td>10/26/2017</td>
<td>11/08</td>
</tr>
<tr>
<td>PDR: PowerPoint Submission</td>
<td></td>
<td>10/29/2017</td>
<td></td>
</tr>
<tr>
<td>CTP: PowerPoint Submission</td>
<td></td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Lab #4: EEPROM, LCD, and C programming</td>
<td>11/10 – 11/15</td>
<td>11/16/2017</td>
<td>11/29</td>
</tr>
<tr>
<td>Final Project: Demo Presentation Submission</td>
<td></td>
<td>12/10/2017</td>
<td></td>
</tr>
<tr>
<td>Final Project/Lab #5: Student's choice</td>
<td>Demo 12/11</td>
<td>12/16/2017</td>
<td></td>
</tr>
<tr>
<td>Final Project report and other files/materials</td>
<td></td>
<td>12/16/2017</td>
<td>12/17</td>
</tr>
</tbody>
</table>

Lab assignments will be scored per the CU grading standards. Assignments which are not completed (signatures obtained) by the **Signature Due Date** or which are not submitted by the **Submission Due Date** will be late and will receive a late penalty deduction. The final project may not be submitted late. Assignments will not be accepted after their **Cutoff Date**; however, incomplete work will be considered so it is in the student’s best interest to submit by the cutoff date any work they have done. Late penalty deduction:
- Up to 1 day late: 2/3 letter grade [e.g. An ‘A’ is reduced to a ‘B+’, a ‘B+’ to a ‘B-’]
- 1-2 days late: 1/3 letter grade more (1 letter grade total) [e.g. An ‘A-’ is reduced to a ‘B-’]
- From 2 days late until the cutoff date: 1-1/3 letter grade [e.g. An ‘A-’ is reduced to a ‘C+’]

The date that the student submits the code they use for signoffs will be the date used to establish late penalties. Signatures will be considered done as of the date the software files were submitted for signoff. However, if the software files are further updated by the student, the signature date will be updated.
Course Requirements

- University policies on academic integrity (http://www.colorado.edu/policies/student-honor-code-policy) will be followed. **Students must understand the CU Honor Code.** 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 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.

- An electronic copy (and optionally a hard 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.

- 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 if necessary 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.

- **Students must regularly maintain a backup of their files (schematics, source code, reports) on a physically separate backup medium (like a networked drive, USB drive key, USB hard drive, etc.) so that access to their files is retained even if their primary computer or storage drive suffers a failure.**
Academic Integrity - Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the academic integrity policy of the institution. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access, clicker fraud, resubmission, and aiding academic dishonesty. Credit must be clearly given for code or 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. All incidents of academic misconduct will be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found responsible of violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code Council as well as academic sanctions from the faculty member. Additional information regarding the academic integrity policy can be found at http://honorcode.colorado.edu.

Academic Accommodations

If you qualify for accommodations because of a disability, please submit to your professor a letter from Disability Services in a timely manner so that your needs may be addressed. Disability Services (303.492.8671, dsinfo@colorado.edu) determines accommodations based on documented disabilities. If you have a temporary medical condition or injury, see Temporary medical conditions under Quick Links at the Disability Services website (http://www.colorado.edu/disabilityservices/) and discuss your needs with your professor. This course requires the use of the Zoom conferencing tool which is currently not accessible to users using assistive technology. If you use assistive technology to access the course material, please contact your faculty member immediately to discuss.

Religious Holidays

Every effort will be made to reasonably and fairly deal with students who have serious religious observances that conflict with mandatory lectures, scheduled exams, assignments, etc. Please notify your professor well in advance, so that there is time to make adequate arrangements. See policy details at http://www.colorado.edu/policies/observance-religious-holidays-and-absences-classes-andor-exams

Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesies and sensitivity are especially important with respect to individuals and topics dealing with differences of race, color, culture, religion, creed, politics, veteran’s status, sexual orientation, gender, gender identity and gender expression, age, disability, 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. For more information, see the policies on classroom behavior and the student code. See policies at http://www.colorado.edu/policies/student-classroom-and-course-related-behavior and at http://www.colorado.edu/osc/sites/default/files/attached-files/studentconductcode_15-16.pdf

Discrimination and Harassment

The University of Colorado Boulder (CU-Boulder) is committed to maintaining a positive learning, working, and living environment. CU-Boulder will not tolerate acts of sexual misconduct, discrimination, harassment or related retaliation against or by any employee or student. CU’s Sexual Misconduct Policy prohibits sexual assault, sexual exploitation, sexual harassment, intimate partner abuse (dating or domestic violence), stalking or related retaliation. CU-Boulder’s Discrimination and Harassment Policy prohibits discrimination, harassment or related retaliation based on race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been subject to misconduct under either policy should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding sexual misconduct, discrimination, harassment or related retaliation can be found at the OIEC website (http://www.colorado.edu/institutionalequity/).
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:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Superior, outstanding</td>
<td>15%</td>
</tr>
<tr>
<td>A-</td>
<td></td>
<td>20-25%</td>
</tr>
<tr>
<td>B+</td>
<td>Above average</td>
<td>20-25%</td>
</tr>
<tr>
<td>B-</td>
<td></td>
<td>28-38%</td>
</tr>
<tr>
<td>C+</td>
<td>Average, has adequately met course requirements</td>
<td>0-10%</td>
</tr>
<tr>
<td>C-</td>
<td></td>
<td>7%</td>
</tr>
<tr>
<td>D+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-</td>
<td>Minimum passing grade</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Fail, has not met course requirements</td>
<td></td>
</tr>
</tbody>
</table>

The normal CU grading standards as shown below will be applied to this class. See the following site for more information: [http://ecee.colorado.edu/~mcclurel/grading.html](http://ecee.colorado.edu/~mcclurel/grading.html)

A     Superior, outstanding
A-    
B+    Above average
B-    
C+    Average, has adequately met course requirements
C-    
D+    
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.