# Embedded System Design Syllabus

## ECEN 5613 – Fall 2018

Lectures: Monday Evenings, 6:00pm-8:30pm, ECCR 118 and ECEE 1B28
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 and Slack
Course Web Site: http://ecee.colorado.edu/~mcclurel/index.html Lab: ECEE 1B28
TAs: Tristan Lennertz, Kiran Hegde, 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.

- Processor selection
- Power delivery, decoupling
- Clocks and resets
- Assembly programming
- Embedded C programming
- Incremental development (HW, SW)
- Test equipment / instrumentation (oscopes, multimeters, logic analyzers)
- Debugging techniques
- Data sheets
- Bus cycles, transaction types, timing diagrams, timing analysis
- Memory maps, chip select logic
- Serial interfaces (RS-232, I2C, SPI)
- I/O port pin driver circuits
- In-circuit programming
- Device drivers
- Interrupts and ISR's
- Memory mapped I/O
- Data conversion (DAC's, ADC's)
- Design reviews
- Design trade-offs
- Entrepreneurship
- Passive components

Topics such as embedded processor selection, glue logic, circuit design, circuit layout, circuit debugging, development tools, firmware architecture, firmware design, and firmware debugging will be discussed. The architecture and instruction sets of at least two microcontrollers will be studied, a microcontroller board with peripherals will be built and debugged by each student, and an ARM-based development board will be used for hardware interfacing and firmware development. Students will develop embedded software in C and assembly. 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 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. **This course serves as an excellent preparatory course for the other courses in the Embedded Systems Engineering program, and provides students with key skills that are important for job interviews.** This core course also counts for the Embedded Systems Certificate. See the course web site for more information.

Popular aspects of this class include:

- In-depth understanding of processor hardware and firmware fundamentals, including low level device drivers
- Experience in individually developing and debugging a hardware and firmware platform
- Development of laboratory skills, including test equipment, soldering, and prototyping
- Key knowledge and skills that are beneficial for job interviews
- Great foundation for other courses in the embedded systems engineering program
- Self-confidence building, students develop significant personal capabilities during the semester
- Professor has significant industry experience that is incorporated into the course
- Flexible final project, student can pursue a project of their interest independently or in a team
- No final exam

## 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 "**ESDF18**" 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 order to prepare for that type of work 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 processor instruction sets, and learn how to use a cross assembler and simulator to develop code. Students will also get exposure to the ARM architecture and gain experience with using an ARM development board to develop code. During the middle of the course, students will focus more on firmware concepts, and will develop code in 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. 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 27*

- 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, busses.
- Lab #1 topics. Design cycle, planning a development project, derivation of requirements, tradeoffs.
- SPLDs. Assembler and simulator. Code development process.
- Work on software elements of Lab #1 this week.

### *Week 2: September 3* *(Labor Day Holiday, Finish Lab #1 Part 1 elements this week)*

- CU Holiday, no lecture on September 3rd. Work on homework and Part 1 Elements of Lab #1.

### *Week 3: September 10*

- 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 17* *(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. 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.
- 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. Timers/counters.

### *Week 5: September 24* *(Finish Lab #2 Part 1 Elements this week)*

- ARM overview. KiCad? Work on current topics presentations? Student/professor meetings?
- 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 6: October 1* *(Lab #2 submission due this week)*

- Lab #3 topics. Cross-assemblers, cross-compilers, SDCC, makefiles, and IDEs
- C variables, bit operations, pointers. Interrupts in C. Interfacing C and assembly.
- Software development, version control, coding standards, code reviews.

### *Week 7: October 8*

- TBD. No lecture? Debug session? Work on Lab #3 and final project.

### *Week 8: October 15* *(Finish Lab #3 Part 1 Elements this week)*

- TBD. Student/professor meetings? No lecture? Debug session? Work on Lab #3 and final project.

### *Week 9: October 22* *(Lab #3 submission due this week)*

- Lab #4 topics. EEPROMs and synchronous serial communication ($I^2C$, SPI, etc.). LCDs.

### Week 10: October 29 *(Submit PDR presentations, Finish Lab #4 Part 1 Elements this week)*

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

### Week 11: November 5 *(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 12: November 12

- Guest Speaker? Work on final projects.
- Passive components. Designing with tolerances and margins.

### Week 13: November 19 *(Fall Break)*

- CU holiday, no lecture, class cancelled.

### Week 14: November 26

- TBD. Work on final projects. Student/professor meetings.

### Week 15: December 3 *(Submit final project demo)*

- Final project presentations TBD. Course evaluations (on-line FCQs).

### Week 16: December 10

- Last Class. Final project reports, presentations, semester wrap-up. Review of vocabulary.

### Week 17: No Final Exam

| **Assignment Overview** | **Signature Due Dates** | **Submission Due Date** | **Cutoff Date** |
|---|---|---|---|
| Lab #1: Basic hardware, SPLD, assembly, simulator | 9/07 – 9/21 | 9/22/2018 | 9/29 |
| Lab #2: Decode logic, NVSRAM, timer ISRs, RS-232 | 9/28 – 10/05 | 10/06/2018 | 10/13 |
| Lab #3: SRAM, UART, assembly, intro to embedded C | 10/19 – 10/26 | 10/27/2018 | 11/03 |
| PDR: PowerPoint Submission | | 10/28/2018 | |
| CTP: PowerPoint Submission | | TBD | |
| Lab #4: EEPROM, LCD, and C programming | 11/02 – 11/09 | 11/10/2018 | 11/17 |
| Final Project: Demo Presentation Submission | | 12/02/2018 | |
| Final Project/Lab #5: Student's choice | Demo 12/03 | 12/10/2018 | |
| Final Project report and other files/materials | | 12/10/2018 | 12/14 |

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 one day late: up to one letter grade [e.g. An 'A-' is reduced to a 'B-']
- From two days late until the cutoff date: up to two letter grades

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 courtesy 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:

| | |
|---|---|
| 15% | Lab #1 and Lab #2 |
| 20-25% | Lab #3 |
| 20-25% | Lab #4 |
| 28-38% | Final Project (including PDR) |
| 0-10% | Quizzes/Assignments, Lab Practical, Student Current Topics Presentations |
| 7% | Class Participation/Attendance/Punctuality, 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://ecee.colorado.edu/~mcclurel/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 may be available 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.