EECS 149/249A Project Overview

Projects are your opportunity to do something interesting with the skills you've learned. Unlike the labs, you have control over what you want to work on and which aspects of the engineering of that project you want to focus on. We're really excited for the project portion of the class and want you to create awesome things that will give you real-world experiences in embedded design and implementation.

Project Topics

An EECS149 project is very open-ended. Projects should be relevant to the class (where relevancy is judged by making use of at least three topics covered in lectures), and nearly all of them will have some embedded hardware component.

Good projects usually have:

- A good technical plan with realistic timelines.
- Effective use of models. Models could cover physical and/or cyber components, dynamic behavior of the system, safety and reliability, and timing.
- High quality software architecture: Modular, well-defined components with well-defined interfaces. Code should be well documented, with procedures clearly labeled with their functionality, with the names of variables and procedures carefully chosen to reflect their role, etc. Modularity implies, for example, that a software component interfacing a sensor is distinct and separable from another software component controlling an actuator.
- Good choices of hardware, considering cost, capability, reliability, and availability (watch out for long lead times on ordering parts!). Again, modular designs with clean architectures are better. Each component should have a clearly defined role and a well-chosen interface, and should be of the right scale and capability for the task at hand.
- Attribution. At every opportunity, when presenting your project or writing about it, you must credit your sources of inspiration and technology. Who wrote the sample code that you downloaded and modified? What is its copyright? What YouTube video inspired you? Who created the web page describing the project that you are emulating? Everything we do as engineers depends on what other people have done before us. It is extremely important to give credit to those people.
- Effective use of platform-based design and model-based methodologies you have learned in the class. Define clearly how you will use models for design, validation, analysis, and testing, and what software tools you used for modeling, simulation, and software development.

There are several research projects that have been put together by the professors which are described here. Past project ideas can be found here: (2015, 2016, 2017).

Project Teams
Groups for projects should have 3 or 4 students. Five-student groups are possible but need to be justified to the course staff in advance. It is expected that all students will be contributing equally to the project. All project work is due as an entire group (not individually).

Project Management
A public Github repository MUST be used for storing all project artifacts (code, hardware designs, posters, reports, etc.). Private repositories (shared with course staff) are possible, but need to be justified in advance.

Project Rubric
All projects will be evaluated on the basis of the quality of the design plan, documentation, and execution with particular attention on the methodology used. It is neither necessary nor sufficient to have a "cool demo" to get a good grade.

Project Item	**Percentage of Grade**
Proposal	5%
Milestone 1	5%
Milestone 2	5%
Project Poster	5%
Project Expo	5%
Project Presentation	10%
Final Project (Technical)	50%
- Difficulty	- 20%
- Integration	- 25% (Does it work? How well and how polished?)
- Breadth	- 5% (Does it use three course topics?)
Final project report	14%
Peer Evaluations	1% (Did you fill them out, **not** their results)

Project Proposals
Proposals cover the plan for the final project. Groups for the project should be finalized by this time. A Github repository should be created and a link to be shared with the course staff in the project proposals. We expect a one to two page proposal that includes the following topics:

- Goals
- Approach
- Resources
- Schedule
- Risks
- Link to the project Github repository

We will provide feedback about the project proposals in the following week. An example project proposal can be found here: Project Proposal Example

Milestone 1
Milestone 1 is the first check-in on how the project is going roughly a week and a half after the proposals. The expectations for milestone 1 are a one to two page document including:

- Architecture drawing
- Progress so far
    - Include links to Github
- Modifications to goals and project scope
- List of needed resources (code, parts, expertise, etc)
- Schedule for remaining time (with team member-task assignments)
- Identification of major risks

We also expect project groups to have scheduled a meeting with course staff to discuss their project. We will send a link with open meeting slots as Milestone 1 approaches.

Milestone 2
Milestone 2 is the final check-in on project status roughly two weeks before the final demos. The expectations are a one to two page document including:

- Progress so far
- Modifications to goals and project scope
- List of needed resources (code, parts, expertise, etc)
- Schedule for remaining time (with team member-task assignments)
- Identification of major risks

We also expect project groups to have scheduled a meeting with course staff to discuss their project. We will send a link with open meeting slots as Milestone 2 approaches.

Project Poster
The project poster will be used to explain your project and demo during the demo day. While text is necessary to explain things, do not include whole paragraphs of explanation on a poster. The goal is to succinctly explain your project so that a casual observer can understand it. Your poster should include:

- Project name and team members
- Project goals
- A diagram of the project architecture
- Other figures that help to explain or demonstrate the project

We will provide a template for the project poster.

Project Expo
The project expo is your demonstration of the project during demo day. We do not expect your project to be fully functioning, but we do expect some demonstrable functionality that allows you to clearly communicate the intent and direction of your project.

Project Presentations & Final Demos
The final presentation will be your primary method (along with your report) to communicate the scope, design, and functionality of your project. We will specify the exact grading of the presentation in the coming weeks. This presentation will occur during the week of final exams.

You should also perform a final demonstration of the project during your presentation.

Project Report
The final project report will be a significant portion of the final project grade (along with the demos). The goal of the project report is to explain the design and implementation of your project as well as to perform a simple evaluation of its capabilities.

- Recommended length: four pages, two columns, 10 point font.
- Illustrate algorithms and models with diagrams, include pictures of your hardware, screenshots, etc.
- Keep the report short and precise; avoid lengthy and verbose descriptions!
- Project reports should articulate how the project involves three or more of the key concepts in the course.
- Also include high-level implication of the project. What could the project (hypothetically) enable or grow into if future engineers built systems that used the work you've already done?

We will provide a template for the final report.

Funding
Each project group will have the use of up to $200 for various hardware and materials necessary for their project. Purchasing will be done by the GSIs. Purchase requests can be made by filling out **https://goo.gl/forms/lUqUz0bFgYkdplEi1**, and GSIs will respond with confirmation of purchase. Additional resources are possible, but should be arranged with course staff in advance.

Tools
Groups may use any hardware that we have used in lab for their projects, for example one or more Buckler boards or Kobuki robots. Groups may also use any other hardware that is available to them.

Any software tools or libraries are acceptable for use in the project as long as their use is properly cited. You won't get credit for the work that someone else did, but hopefully building on top of previous work will allow you to do something even better.

Hardware

- Buckler (We have many on hand)
    - Additional on-board capabilities
        - Light, Temp, Humidity, Pressure
        - SD Card
        - Wireless (proprietary or BLE)
    - Grove sensors (We have some on hand)
    - Qwiic sensors (with a grove adapter)
- Kobuki (We have some on hand)
- Other boards
    - Microcontroller boards
        - STM32 Discovery
        - TI Launchpad
        - Particle
        - ESP32 Thing
        - Hail (We have some on hand)
    - Linux boards
        - Raspberry Pi
        - Beaglebone Black (We have some on hand)
- Hardware Vendors
    - Sparkfun
    - Adafruit
    - Seeed

Software

- Embedded libraries
    - Buckler repo (Example code and libraries for Buckler)
    - nRF SDK (Already used for Buckler)

    - STM Standard Peripheral Libraries
    - ARM CMSIS
- Embedded Operating Systems
    - Mbed
    - Contiki
    - Riot
    - Tock
- Software libraries
    - Ptolemy II
    - Yakindu
    - OpenCV
    - SciPy
    - Node-RED
    - Tensorflow
    - MQTT
    - Grafana
    - InfluxDB
    - Timescale

Workshops and Makerspaces

In COE there are multiple different options for students (any UC affiliate really) to obtain access to labs and makerspaces where there are a wide variety of tools available for hardware design and fabrication. Most of these labs require a paid membership on top of required training that can be completed at regularly scheduled times. Some of these spaces are free. If you have specific equipment needs, talk to the course staff about it and we should be able to point you in the right direction.

- Cory Student Workshop (CSW)
    - Location: Cory Hall, Room 297
    - Cost: FREE
    - Training: https://supernode.berkeley.edu/about/#cory-student-workshop
- Chenming Hu Innovation Lab (Supernode)
    - Location: Cory Hall, Room 246
    - Cost: FREE
    - Training: https://supernode.berkeley.edu/training/
- CITRIS Invention Lab
    - Location: Sutardja Dai Hall, Room 141
    - Cost: $75 per user, $225 for research users. No group options (but maybe you can negotiate with the lab manager)
    - Training: See Training here https://invent.citris-uc.org/resources.html

    - Note: you might look into the Maker Pass option
- Jacobs Institute for Design Innovation
    - Location: Jacobs Hall
    - Cost: See Maker Pass
    - Training: See Training here http://jacobsinstitute.berkeley.edu/our-space/makerpass/get-maker-pass/

Getting Help

Each project will be assigned a mentor. The mentor will be a domain expert in the area that the project covers and can be asked domain-specific questions. General questions (such as debugging help) should be directed at the GSIs rather than the project mentors.

You can see the office hour schedule for the GSIs and request appointments with them on this form

Timeline

- October 19 - Project Proposal Due
- October 30 - Milestone 1
- November 13 - Milestone 2
- December 11 - Project Presentations and Final Demos
- December 11 - Project expo
- December 11 - Peer Evaluations Due
- December 14 - Project Report Due