

BME464L: Medical Instrument Design

Class Syllabus (Fall 2013, Palmeri)

- Instructor:** Dr. Mark Palmeri (mark.palmeri@duke.edu)
- Lab Master:** Matt Brown (matt.brown@duke.edu)
- Teaching Assistant:** Dongwoon Hyun (dh65@duke.edu)
- Lecture:** Tuesday and Thursday from 11:45 - 13:00 in 201 Hudson Hall
- Lab:** Thursday from 16:40 - 19:40 in P012 Teer
- Class Schedule:** All class activities are being managed in a Google Calendar that is linked to the class website listed below. XML, iCal and HTML links are available to make this calendar accessible in your calendar service of choice. Specific lecture and lab details, along with deliverable due dates will be posted on this calendar, and new due dates will be announced in lecture and by class [email](#).
- Class Website:** <http://mlp6.github.io/Medical-Instrument-Design/> This website is actually a git repository hosted and rendered by GitHub! Please be sure to check the website often for announcements, deliverable deadlines, project resource, and useful tips as your projects progress throughout the semester. Content changes on the website will be tracked through the commit history to the git repository used to manage class resources by Dr. Palmeri. Significant changes and additions to the website will be announced by class [email](#). [Sakai](#) will also be used to post grades.
- Course Objectives:** The course is a culmination of the knowledge and engineering skills that you have developed over the past 3 years. You will learn some new tools in design brainstorming, project management, and software version control. You are expected to present a coherent design plan and analysis to support the final project that you deliver at the end of the semester. Some specific course objectives include:
- Develop a plan for building an instrument to serve a particular need
 - Draw a schematic and block diagram of a proposed project
 - Present a project proposal to a group
 - Find electronic parts using specifications
 - Write and debug code for a microcontroller (Arduino or Beaglebone)
 - Design software to control temporal processes
 - Manage software with a version control system (git)
 - Build an electronic medical instrument
 - Calibrate an instrument
 - Create a test environment for an instrument
 - Present technical work to a group
 - Work in teams of four to complete a sizable project
 - Understand ethical issues related to medical instrumentation
 - Apply appropriate statistical methods during the design process

- Attendance:** Lecture / lab attendance and participation count for 10% of your class grade. It is very understandable that students will have to miss class / lab for job and school interviews, personal reasons, illness, etc. Absences will be considered *excused* if they are communicated to Dr. Palmeri at least 48 hours in advance or through submission of [Short Term Illness Form \(STIF\)](#) *before* class or lab. Unexcused absences will count against the attendance and participation component of your class grade.
- Office Hours:** Dr. Palmeri will have office hours by appointment (Monday and Wednesday are the best days of the week). Other days / times are always available by appointment, and there is an open-door policy active all semester (i.e., if my office door is open, then you are welcome to pop in). Dongwoon will also be available, as needed, to help with all aspects of your projects throughout the semester.
- Textbooks & References:** There are no required textbooks for this class. However, if you are looking for a good overview of microcontrollers, as presented in the context of the Arduino, then I recommend that you consider looking at Jeremy Blum's [Exploring Arduino](#). There are also many resources online, some of which are summarized on [BME354L: Introduction to Medical Instrumentation](#).
- Electronic Lab Notebooks (ELN):** LabArchives will be used for this course, with one shared notebook being used for each project group. Your ELN is one of your primary ways to demonstrate and document your entire design process, including brainstorming of design ideas, formal engineering analysis to justify your design choices, testing procedures, and performance analysis. Some guidelines for your ELN:
- Block diagrams of the device and submodules in the device should be presented.
 - Signal processing and software algorithm logic should be documented, and the associated code should be specifically referenced by the appropriate git repository URL and SHA1 hash(es).
 - Choice of electronic components should be supported by an analysis (when appropriate) that includes considerations such as power, bandwidth, SNR, etc. that could impact performance of your device.
 - Formal circuit schematics should be drawn and properly annotated for all electronic circuits.
 - Component datasheets, user manuals, etc. should be included.
 - Mechanical drawings, SolidWorks renderings, etc. should be included.
 - All measurements made during debugging and testing should be recorded with the appropriate precision, units, and estimates of uncertainty. These measurements should be accompanied by a motivation for making them, along with a summary of your interpretation of the data.
 - A copy of all reports and slides should be saved.
 - All software associated with your device, along with relevant documentation, should be maintained in your group's git repository. A tar / zip archive of your final software repository should be saved in your ELN.

We will review ELN content in lecture.

Distributed Version Control Software (git): Software management is a ubiquitous issue in any microcontroller engineering project, and this task becomes increasingly difficult during group development. Version control software has many benefits and uses in software development, including preservation of versions during the development process, the ability for multiple contributors and reviewers on a project, the ability to tag "releases" of code, and the ability to branch code into different functional branches. We will be using GitHub to centrally host our git repositories, with one repository existing for each project. Some guidelines for using your git repositories:

- **All** software additions, modifications, bug-fixes, etc. need to be done in your repository.
- The "Issues" feature of your repository should be used as a "to do" list of software-related items, including feature enhancements, and bugs that are discovered.
- There are several management models that we will review in class, with the major distinctions being individual user forks of the master repository, or a single mutli-write-user repository. We will review these different management models in class, after which your group should choose one and stick with it.
- Dr. Palmeri will only review code that is committed to your master repository or your personal fork of that repository.
- All of the commits associated with your repository are logged with your name and a timestamp, and these cannot be modified. Use descriptive commit messages so that your group members, Dr. Palmeri, Matt Brown and Dongwoon can figure out what you have done!! You should not need to email group members when you have performed a commit; your commit message(s) should speak for themselves.
- Code milestones should be properly tagged.
- Write software testing routines early in the development process so that anyone in your group or an outsider reviewing your code can be convinced that it is working as intended.
- Comment, comment, comment.
- Modular, modular, modular.
- Make commits small and logical; do them often!

We will review working with git repositories in lecture and lab.

Project Details: Project details, including budgets, part ordering procedures, etc. will be detailed in lecture with a separate handout.

SolidWorks and Pratt Student Machine Shop: Everyone will be receiving training in the 3D CAD software [SolidWorks](#). Additionally, everyone (not just some group members) needs to be certified to use the [Pratt Student Machine Shop](#). Please complete the online machine shop training and schedule a time for the safety exam with Steve Earp sometime in September!!

Grading: The following grading scheme is subject to change as the semester progresses.

Attendance & Participation	10%
ELN & GitHub Repository (weekly review)	30%
Progress Reports & Presentations	20%
Final Device (Function & Packaging)	15%
Final Report & Documentation	15%
Final Presentation & Poster	10%

**Duke
Community
Standard &
Academic
Honor:**

Engineering is inherently a collaborative field, and in this class, you are encouraged to work collaboratively on your projects. The work that you submit must be the product of your and your group's effort and understanding. All resources developed by another person or company, and used in your project, must be properly recognized.

All students are expected to adhere to all principles of the [Duke Community Standard](#). Violations of the Duke Community Standard will be referred immediately to the Office of Student Conduct.

Please do not hesitate to talk with Dr. Palmeri about any situations involving academic honor, especially if it is ambiguous what should be done.