University of California at Berkeley
Mechanical Engineering

ME 135: Microprocessor Based Design of Mechanical Systems
Spring 2008

Course Description
ME135 covers software design and implementation methodologies suited to the control of complex mechanical systems. The design methodology allows for the operational description of mechanical systems in a way that can be presented to semi-technical personnel as well as serve as a basis for software development. Implementation is based on LabVIEW, a graphical data-flow computing language, and LabWindows CVI (NI C compiler). Implementation methodology is presented with software portability a primary emphasis. Students work in teams to design and implement solutions to problems of increasing complexity using prototype lab equipment, including a design project for which they must formulate their own objectives.

Topics Covered
LabVIEW and C programming principles; task and state design methodology; principles of real time computing; real time implementation issues; feedback control basics; operator interface.

Course Objectives
By the conclusion of the course, students should be able to:
- Assess the relative difficulty of a problem
- Outline a solution to it
- Estimate the resources required to solve the problem
- Develop and document a design for the control software
- Identify critical safety issues
- Implement a prototype solution
- Test and evaluate the solution
- Work as part of a team to accomplish these goals

Basis for Grading
- 8 Lab exercises
- Progress reports
- Final project proposal
- Midterm milestone presentation
- Final project and presentation

Lab
The lab is the heart of this class. Most of the individual work to be done revolves around lab projects and the lectures are keyed to the labs.

The lab is located in room 2170 Etcheverry and is available at all hours with a cardkey (get the card key from room 5104 Etcheverry; key-ordering hours are limited). **Please respect the**

security in the lab: we have been able to run it in this open-shop manner because people are careful to keep the door locked and keep an eye on things.

The lab contains a variety of mechanical systems and computers with the appropriate interfaces for controlling them. The computers are new dual-core Dell's running Windows and the LabVIEW real-time operating system ETS.

What Can Be Used – Outside Materials and Academic Honesty

You can use any material that you find, anyplace you find it **as long as you reference it**. If you are using code or solutions that from somewhere else (the web, another student, other groups, previous work, etc.) you are expected to carry the solution well beyond the point attained by the work you are referencing. Your total effort should be the same but if you can use "found work" you will be able to explore advanced techniques.

Lab Exercises

There are eight lab exercises of one or two weeks each. These exercises are designed for learning LabVIEW and C and using LabVIEW to acquire data from and control mechanical hardware (mostly motor-driven systems). The exercises represent increasing levels of complexity, requiring more sophisticated software techniques for later exercises.

Each exercise requires a brief report that consists of two components. The first, a quick user's guide, should allow anyone to operate the interface after reading it. The second, a detailed power user manual, should explain key design decisions and describe the code so that someone experienced with LabVIEW can modify it if necessary. **Brevity is valued in all reports and the more intuitive the interface and code, the less documentation is needed!**

Tentative schedule:
1. Learning to use LabVIEW (GUI development) (3 weeks)
2. Digital data acquisition (1 week)
3. Motor control and robot kinematics (3 weeks)
4. Network communication (1 week)
5. Image processing (1 week)
6. Final project (5 weeks)

Group Work

All of the lab assignments are done by groups of three students each. Under exceptional circumstances students groups of two or four can be formed. Grading will be mostly by group, but peer evaluations will be a part of the grade.

The major reasons for group work are that most of this kind of work in industry is done in groups, the learning is richer in groups, and the lab resources are limited.

Group dynamics can sometimes be tricky. Don't wait – the earlier these problems are addressed, the better. If problems do arise that can not be resolved, feel free to come and talk to us.

Final Project

The project needs to demonstrate use of real time control software. There is no requirement any hardware be built, so it is perfectly acceptable to use existing hardware in the lab. You are also free to build or modify hardware for the project, but keep in mind that building hardware is a very time-consuming activity and the grade will be based mainly on the software control system design and implementation.

Finding the right level of complexity in a project is important. If it is too simple, getting a top grade is difficult even if the project is carried out well. Likewise, choosing something too complex makes it difficult to make enough progress to demonstrate your understanding. I am happy to discuss project plans with you. The final project proposal should document what you want to do and describe the test specifications. The performance specification and associated testing will be given strong weight in grading. If the original specification is not met, the project performance must be justified accordingly.

The final project includes a presentation and demonstration of the product to myself and your colleagues in the class. There is no separate grade for the presentation, but the presentation is considered as part of the overall project grade.

Midterm Milestone Presentation

Students must give a formal presentation of their project progress to the instructor partway through the term. Every student is expected to present at least part of the project, and this presentation will be graded on a group as well as an individual basis.

Grading

Lab Exercises	40%
Midterm Milestone	10%
Final Project	50%

Texts

The recommended text is: *Control Software for Mechanical Systems*, Auslander, Ridgely, & Ringgenberg, Prentice-Hall, 2002, ISBN 0-13-786302-0. It is a useful introduction to the TransRunC scheduling software, which we will use for a couple exercises and you may choose to use for the final project.

In terms of LabVIEW, the best references are the various tutorials and articles available from National Instruments website at http://www.ni.com .

Many C tutorials are also available online. The book many reference as the bible for C is: The C Programming Language by Brian W. Kernighan and Dennis M. Ritchie. They are the inventors of the language.

Instructors

George Anwar
6169 Etcheverry Hall
Office Hours: TuTh 3:30-5 PM
ganwar@integratedmotions.com

GSI

Tim Suen
2170 Etcheverry Hall
Office Hours: M 4-6 PM, Th 5-6 PM
twsuen@berkeley.edu

Course Web Site

http://www.me.berkeley.edu/me135/

Submit Assignments

me135hw@gmail.com

Acknowledgements

Several companies have donated equipment, supplies and software to this lab. Our thanks goes to HP, Intel, National Semiconductor, Microsoft, IBM, and National Instruments.