CMPE 118[218]/L:
Introduction to Mechatronics

Gabriel Hugh Elkaim
Introduction to Mechatronics

- T-Th 9:50 – 11:25 AM, E2-280 Simularium
- Lab: JBE 111, 113, and 115, 24/7 access

- Instructor: Prof. Gabriel Hugh Elkaim
- TA’s: Pavlo Vlastos and Kyle Cordes
- Tutors: Jeremy Crowley, Sterling Dreyer, Denzel Mapp, Joshua Pena, Calvin Ryan, William VanHuyning, Jason Vranek, Milo Webster
Lab Sections

- Times when TA’s and Tutors are guaranteed to be in the labs:
- These will be posted as soon as we know them.
Textbooks


Optional Textbooks (good references):

- “Mechanical Devices for the Electronics Experimenter,” by Britt Rorobaugh
- “The Cartoon Guide to Computer Science” by Larry Gonick
# Syllabus/Readings

## UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE-118(218)/L: INTRODUCTION TO MECHATRONICS

## LECTURE/LAB CALENDAR

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Reading (to be completed before the lecture)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 late September</td>
<td>CKO Ch. 9, 10,11- Basic Electronics</td>
</tr>
<tr>
<td>BasicCircuitI, BasicCircuitII, BasicOpAmps</td>
<td></td>
</tr>
<tr>
<td>1 late September</td>
<td>CKO Ch. 1,2,3,4-Intro, microprocessors, C code</td>
</tr>
<tr>
<td>Microprocessors</td>
<td>CKO Ch. 31 – Troubleshooting</td>
</tr>
<tr>
<td>0 Course Introduction</td>
<td>CKO Ch. 5 – Event Driven Programming, event checkers, State Machines</td>
</tr>
<tr>
<td>0.5 ES Framework/HSM for Lab 0</td>
<td>CKO Ch. 6 – Software Design, abstraction, architecture, testing</td>
</tr>
<tr>
<td>1 Event Driven Programming, State Machines</td>
<td>CKO Ch. 13 – Sensors</td>
</tr>
<tr>
<td></td>
<td>H+H Section 15.02 = Light levels</td>
</tr>
<tr>
<td></td>
<td>15.05 – Hall effect + magnetic field</td>
</tr>
</tbody>
</table>
Personal Responsibility

UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE118(218)/L: INTRODUCTION TO MECHATRONICS

PERSONAL RESPONSIBILITY IN THE CLASSROOM

STUDENT RESPONSIBILITY:
Over several years of teaching, we (the instruction staff) have come to the conclusion that we need to give you (the students) a concrete set of expectations both in terms of what we expect from you, and what you can expect from us. While this document is certainly incomplete and will not be able to cover all possible scenarios, we hope that you will use it as a set of guidelines to properly set your expectations, and thus endeavor to meet them.

1) You are an adult, and we expect you to behave like one. Your success (or failure) is your own responsibility.
   a) It is your responsibility to learn the material covered in the class
   b) It is your responsibility to adequately prepare for the class, e.g., reading and homeworks.
   c) It is your responsibility to turn in assignments on time, in the right format, in the right place, and to verify that you did so (especially for electronic submissions).
Grading

COURSE: 67% Labs and Projects
33% Quizzes, and Exams

QUIZZES, EXAMS: 20% Midterm (Take home)
12% Quizzes (weekly)
1% Participation

LABS AND PROJECT: 8% Lab 0
8% Lab 1
8% Lab 2
8% Lab 3
35% Project
Class Website

- [https://classes.soe.ucsc.edu/cmpe118/Fall17/](https://classes.soe.ucsc.edu/cmpe118/Fall17/)

Welcome to CMPE-118 Introduction to Mechatronics.

If you are on the waitlist for the class, show up to the first lecture.

Fall 2015's Presentation: SLUGWARS: A Slime Awakens

In Fall 2015, the class has been tasked with building a droid that acted as a rebel ship trying to evade an imperial TIE fighter, avoiding the scouts, shooting the TIE fighters, and escaping through the hyperspace portal. SLUGWARS, invite. The final project was the subject of a KSBW News story here, and numerous photos from last year can be seen here.

Background
Piazza (1.2)

- https://piazza.com/ucsc/fall2017/cmpe118l/home

CMPE 118/L: Introduction to Mechatronics

Description

Mechatronics is the synergistic combination of mechanical engineering ("mech" for mechanics), electronic engineering ("tronics" for electronics), and software engineering. The purpose of this interdisciplinary engineering field is the study of automation from an engineering perspective and serves the purposes of controlling advanced hybrid systems such as production systems, synergy drives, planetary rovers, automotive subsystems such as anti-lock system, spin-assist and every day equipment such as autofocus cameras, video, hard disks, cd-players, washing machines, legomatics etc.

Mechatronics is centered on mechanics, electronics and computing which, combined, make possible the generation of simpler, more economical, reliable and versatile systems.

The word "mechatronics" was first coined by Mr. Tetsuro Morita, a senior engineer of a Japanese company, Yaskawa, in 1969. Mechatronics may alternatively be referred to as "electromechanical systems," or as "smart products."

General Information

Welcome to Mechatronics, video lectures to watch before first class
Welcome to Mechatronics, video lectures to watch before first class

Welcome to the Fall 2016 CMPE-118/218S, Intro to Mechatronics.

I hope you have all had a good, productive, and restful summer.

As you may or may not yet realize, this class will be the most work you have put into anything in your life, to date. Also some of the most fun.
The class is overloaded with work, and we are always behind and running.
I will go over this in detail at the first lecture.

In order to try to give you a bit more time to get comfortable with the material, and not falling behind in the lecture, we want you to review your basic electronics and microcontrollers in the following linked lectures. This is an essential part of the class, and we expect you to have watched these carefully before we start.

Please read (carefully) the class information and syllabus and note the reading:
https://classes.soe.ucsc.edu/cmpe118/Fall16/LectureNotes/CMPE118_info.pdf
https://classes.soe.ucsc.edu/cmpe118/Fall16/LectureNotes/CMPE118_Syllabus.pdf

Before the class starts, watch the following:
Lecture (1.1) - (1.2): Microprocessors (Very relevant material for your first lab and partial review of CE12 and CE121)
Lecture (2.1), (2.2), (2.3): Basic Electronics I, II, and Basic OpAmps.

Yes there will be a quiz. Yes you will be doing things the first day. Get ready to hit the ground running fast, we will see you in a few weeks.

You can download all the videos here:
1.1: https://classes.soe.ucsc.edu/cmpe118/Fall16/Videos/Lecture_1_MicroControllers_Part1.mp4
1.2: https://classes.soe.ucsc.edu/cmpe118/Fall16/Videos/Lecture_1_MicroControllers_Part2.mp4
2.1: https://classes.soe.ucsc.edu/cmpe118/Fall16/Videos/CMPE118_BasicCircuits1.mp4
2.2: https://classes.soe.ucsc.edu/cmpe118/Fall16/Videos/CMPE118_BasicCircuits2.mp4
2.3: https://classes.soe.ucsc.edu/cmpe118/Fall16/Videos/CMPE118_BasicCircuits3.mp4
Lab Work (and what is expected)

- You will spend 20+ hours/week in the lab outside of the lecture
- This will go up for the last five weeks of class during the project
- “Prep work” is much more time efficient than trying to learn it on the fly
- Read (do NOT skim) everything we tell you
- Twice (maybe three times)
Lab Work (tips and hints)

- Do not attempt to “divide and conquer”
  - It is never faster
  - You never learn the parts you don’t do
  - The work is shoddy

- Proper preplanning prevents p***-poor performance
  - Pay attention to the prelabs

- You will make mistakes: it is expected
Lab Rules of Conduct (DO NOTS)

- No food or drink in lab
  - Eat your food in Jacks Lounge
  - Good for breaks at white board
- Outer doors are for emergency use only
  - Do NOT under any circumstances let people in from the outside
- **Clean up after yourselves**
  - You are sharing the lab with others
Academic Integrity

• Presenting someone else’s work as your own.
• Do NOT do it, not worth it
Questions?
Event-Driven Programming and State Machines

Gabriel Hugh Elkaim
Traditional Program Structures

START

1. If/else

2. Do/while

END

```c
main()
{
    while (1) {
        /* Everything happens in the ISR */
    }
}
```

GUT
Programming Embedded Systems (1.2)

Asynchronous — any input can happen at any time.

Simultaneous — inputs & outputs.

Sequence of inputs & outputs are unknown, unknowable, reorderable.

No end, no exit.
Programming Embedded Systems (2.2)

Inputs - Sensors (lots of them)

User Inputs - push buttons, keyboard, comms link

Outputs - update a display, turn something on/off (motors), in some a change anything physical in the world.

RTOS - Real-Time Operating System
Events and Services Framework (1.4)

Formalized methodology - concept Framework

Excellent Method for Implementing event driven programs (State Machines)

**Emphasize Design First**

Make it clear how to define our lower level entities. Make debugging much easier.
Events and Services Framework (2.4)

Rule #1

All tasks break down into **two fundamental**
classes:

1. **Event Detection**
2. **Service**

*Fast, atomic, non-blocking***

---

Which `!IsTimeExpired()`? 😵
What is an Event?

Some detectable change

open

1-3 mm

5-50

DEBOUNCE
What Happens with Noise?
Add Hysteresis

- On
- High Threshold
- Signal Plus Noise
- Low Threshold
- Off

Graph showing the effect of hysteresis with thresholds and signal levels.
Events and Services Framework (3.4)

Corrected to state:

Keep event detectors and service functions as short as possible.

Make them non-blocking.
Events and Services Framework (4.4)

Complete Embedded Program Structure

```plaintext
Initialize all hardware/software
while (1) {
    Test for Events () — round robin fashion
    Service Events () — slave machine
}
```
State Machines (1.4)

Description of an abstract machine:

At any point in time, it can be in ONE (and only one) of a fixed number of defined states.

Next state in a progression depends only on the current state and the inputs (events).

Idealized instantaneous transition from one state to next.
State Machines (2.4)

Useful tool for describing the behavior of an event-driven program.

Allows you to explore the behavior BEFORE you implement the code.

Natural fit into the event-driven software framework.