Lecture 1 : Introduction

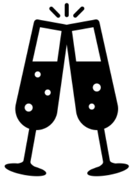
Subject Introduction

Slides were originally developed by Profs. Edward Lee and Sanjit Seshia, and subsequently updated by Profs. Gavin Buskes and Iman Shames.

Outline

- Subject overview
- Class timetable
- Embedded System Design schedule
- Assessment
    - Workshops
    - Project
    - Assignments
    - Exam
- Feedback

Welcome!



- Welcome to ELEN90066 Embedded System Design.
- This subject provides a practical introduction to the design of microprocessor-based electronic systems. The lectures and project work will expose students to the various stages in an engineering project (design, implementation, testing and documentation) and a range of embedded system concepts.

- Lecturer: Dr. Farhad Farokhi, Department of Electrical and Electronic Engineering (EEE).
- Office: Room 3.11, Third Floor, EEE building (193).
- Email: farhad.farokhi@unimelb.edu.au

Subject Overview

- Lectures, workshops, and project expose students to the various stages in an engineering project (design, implementation, testing, and documentation) and a wide range of embedded system concepts.
- Topics include:
  - Interfacing to the analogue world via analogue-to-digital and digital-to-analogue converters
  - Operating systems concepts, multi-tasking, resource management, and real-time issues
- This material will be complemented by exposure to standard software tools, including compilers and debuggers.

Intended Learning Outcomes

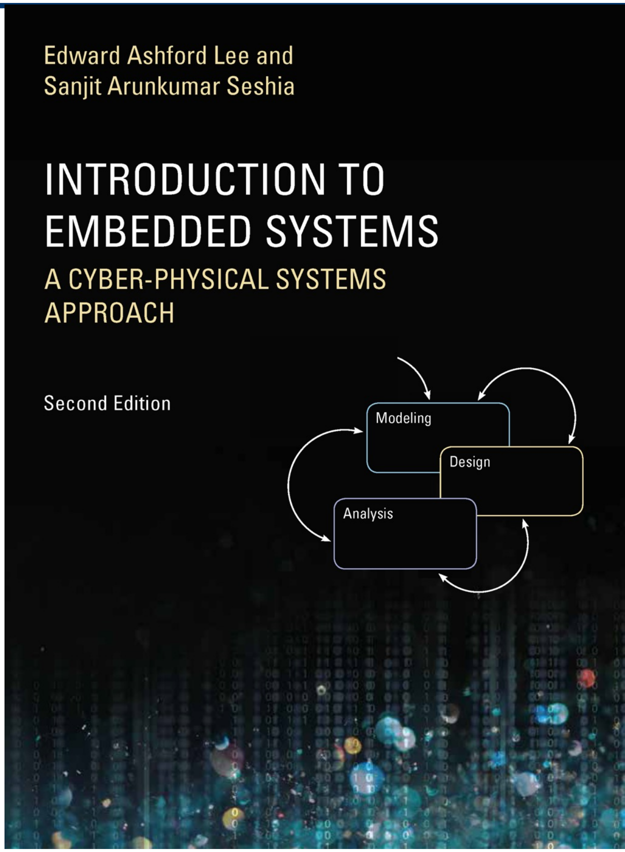
- On completing this subject, it is expected that the student be able to:
  - ❑ Design, build, and test the hardware components (microprocessor, bus, and peripheral interfacing) of an embedded system
  - ❑ Develop and test the low-level software components of an embedded system
  - ❑ Conduct a small embedded system design project

Lectures and Consultation times

- Lectures:
  - Monday: 16:15 – 17:15 Zoom
  - Wednesday: 10:00 – 11:00 Zoom
- Consultation Time:
  - Monday: 12:00 (noon) – 13:00 Zoom
- **CHECK THE LMS REGULARLY FOR EVERYTHING**
  - Slides and video recordings;
  - Updates and announcements;
  - Additional resources;
  - Workshop materials;
  - Exercises.
- **Expect emails with updates, reminders and feedback!**

Subject Textbook

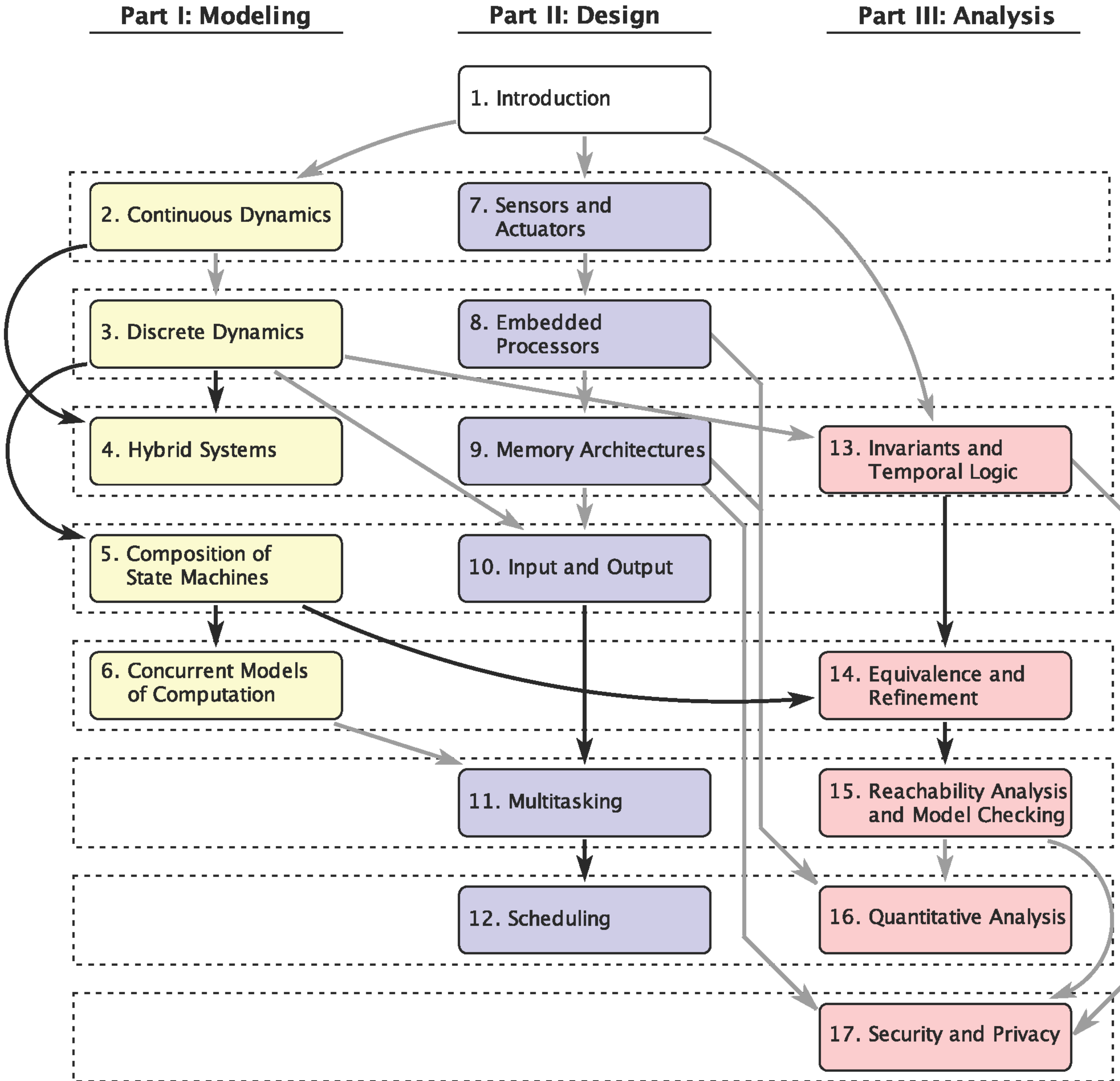
Edward A. Lee and Sanjit A. Seshia,
Introduction to Embedded Systems,
A Cyber-Physical Systems Approach,
Second Edition, MIT Press, 2017.
Available online at :
http://leeseshia.org/index.html



The textbook strives to identify and introduce the *durable intellectual ideas* of embedded systems as a technology and as a subject of study. The emphasis is on modeling, design, and analysis of cyber-physical systems, which integrate computing, networking, and physical processes.

Roadmap

The three threads are designed to be read concurrently



Schedule

Week Beginning Monday	Week number	Lecture 1	Lecture 2	Workshop (remote)	Workshop (on campus)	Assignment
July-25	1	Introduction	Cyber-Physical Systems	n/a	n/a	n/a
August-1	2	Sensors and Actuators	Embedded Processors	n/a	n/a	n/a
August-8	3	Embedded Processors	Memory Architecture	Sensor Modelling	Sensor Modelling	n/a
August-15	4	Input/Output	Continuous Dynamics	Introduction to ROS	Embedded Development Tools	**Assignment 1 due August-19**
August-22	5	Discrete Dynamics	Discrete Dynamics	Modelling Environment	Interfacing Sensors	n/a
August-29	6	Concurrent Composition	Hierarchical State Machines	Simulating Robot	Kabuki Navigation in C	n/a
September-5	7	Synchronous-Reactive Models	Multi-Tasking	Implementing FSMs (Stateflow)	Kabuki Hill Climb in C	**Assignment 2 due September-9**
September-12	8	Scheduling	Scheduling	Controlling Virtual Robot	Model-based Kabuki Navigation and Hill Climb	n/a
September-19	9	Invariants and Temporal Logic	Invariants and Temporal Logic	Robot Design	Robot Design	n/a
September-26						
October-3	10	Equivalence and Refinement	Reachability and Model Checking	Robot Design	Robot Design	**Assignment 3 due October-7**
October-10	11	Reachability and Model Checking	Quantitative Analysis	Robot Design	Robot Design	n/a
October-17	12	Security and Privacy	Review and Exam Information	**Final Assessment**	**Final Assessment**	n/a
October-24	SWOT Vac					

Assessments

- Final examination (60% of total assessment)
  - **Exam is a hurdle!** You MUST pass the exam in order to pass the subject.
- Project final report and outcome (20%)
  - Team project report submitted on first day of exam period.
- Workshop work (10%: 5% pre-lab; 5% in-class tasks)
  - 6 assessed workshops with in-class tasks.
  - **Pre-labs MUST be done individually** and submitted in LMS.
  - Separate projects for online and on-campus students!
- Assignments (10%)
  - 3 **individual** assignments for **individual** feedback.

Workshops

- Workshops (10% of total assessment)
  - 3-hour workshops, working in teams of 2 and 3
  - Workshops commence on **Week 3** (LMS for schedule)

- Workshops 1-6: Learning the tools
  - Workshop classes progressively expose you to tools for designing, modelling and analysing embedded systems.
  - Workshops require **INDIVIDUAL** pre-lab work (MUST be submitted via LMS **before 8am** on **workshop day**).
  - Workshops have in-class tasks to be checked off by demonstrators as you go to get the points for them.

Workshops

- Workshops 7-9: Completing the project
  - (Re-)Design, experiment with and optimise your solution to the project.
  - Catching up on tasks from Workshops 1-6 that you did not get to complete in time.
  - Document your design and testing procedure.
    - There are NO pre-lab exercises and NO in-class assessments for these workshops.

- Workshop 10: The Grand Finale
  - Demonstration of final design and competition!

Workshops Groups

- Workshop groups consist of **at most THREE people**.
- Groups are to be **self-enrolled** via the LMS.
  - Go to the “People” link in the left-hand menu and then “Groups”

  - You MUST join a group in the same workshop session as you.
  - Check your timetable to confirm your workshop session number.
- *Don’t worry!* If you can’t find a group before your first workshop, you will be randomly assigned to one during the session.

Final Project Report

- Report (20% of total assessment)
  - Completed in teams.
  - Consider the final report a smaller, “practice” version of a capstone report.
  - Document background theory, design process, testing, and outcome.
  - Submit through LMS. One submission per team.
  - Further details will be posted on LMS.

The main objective of workshops is to realise theoretical concepts introduced in lectures for practical applications.

The Project

- Program a real-life embedded system, a Kobuki robot platform, that utilizes multiple sensors in order to successfully navigate an obstacle course and achieve an objective, including
  - Fixed obstacles
  - Ramps
  - Cliff edges

- Online workshops–virtual Kobuki will require downloading MATLAB (plus some add-ons) and a Virtual Machine.
- On-campus workshops–programming a physical Kobuki in either C or via a model-based approach using LabVIEW.

Assignments

- <u>Assignments (10% of total assessment)</u>
  - Three assignments, equally weighted
  - <u>Assignments **MUST** be done **individually**</u>
  - Assignment 1 due end Week 4
  - Assignment 2 due end Week 7
  - Assignment 3 due end Week 10
  - Submission (and feedback) is through LMS

Final Exam

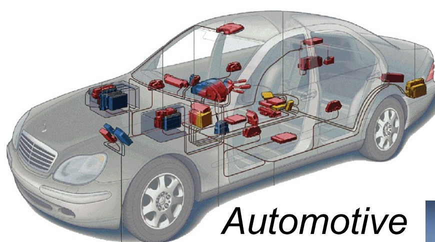
- [Final Exam (60%)]
  - Final exam is online, open-book.
  - Final exam is a **HURDLE** requirement. YOU WILL FAIL THE SUBJECT IF YOU FAIL THE FINAL EXAM.
  - Past exams prior to 2019 cover different material!

Questions?

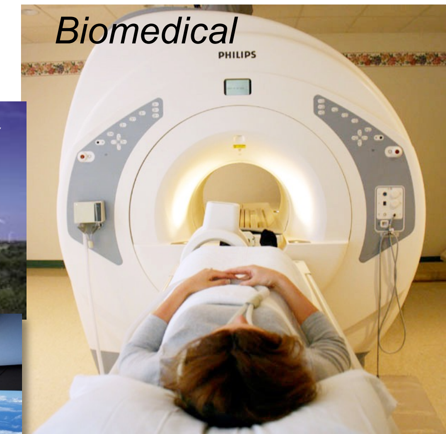
Focus on Cyber-Physical Systems
Full of Contradictory Requirements

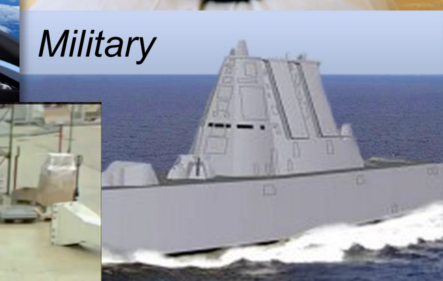
It's not just information technology anymore:

- Cyber + Physical
- Computation + Dynamics
- Security + Safety

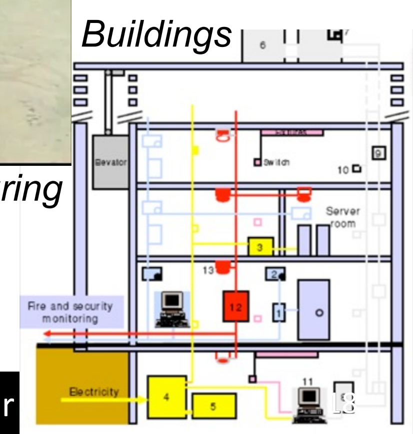


Automotive





Manufacturing



Contradictions:

- Adaptability vs. Repeatability
- High connectivity vs. Security and Privacy
- High performance vs. Low Energy
- Asynchrony vs. Coordination/Cooperation
- Scalability vs. Reliability and Predictability
- Laws and Regulations vs. Technical Possibilities
- Economies of scale (cloud) vs. Locality (fog)
- Open vs. Proprietary
- Algorithms vs. Dynamics

Innovation:

Cyber-physical systems require new engineering methods and models to address these contradictions.

E Pluribus Unum: Out of Many, One

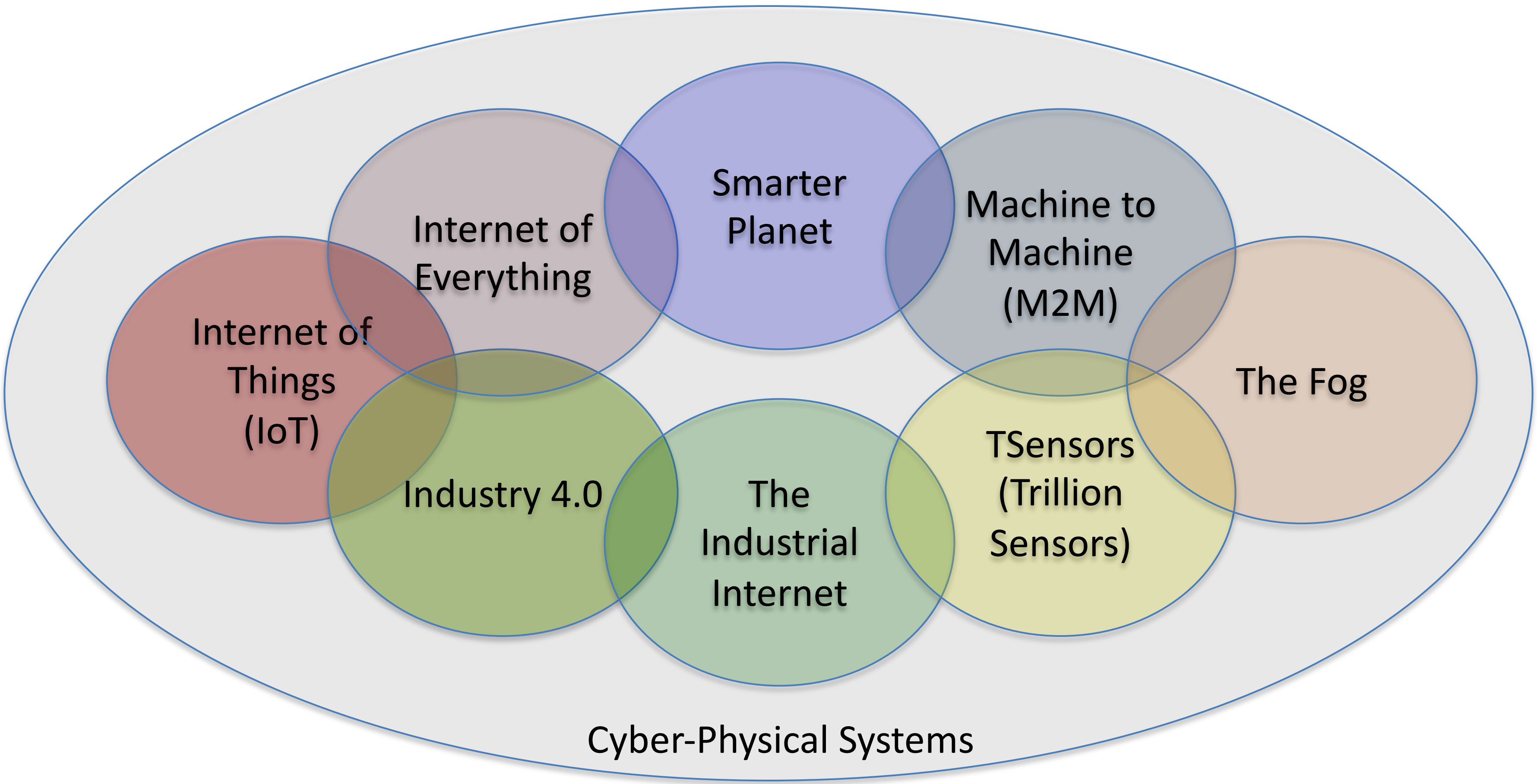


Diagram adapted from Edward Lee

We Need Models of Systems/Environment

What are models?

- Representation of physical behavior by mathematical means.
- Model dominant behavior of system and incorporate real data, ignore other insignificant phenomena.
- Simplified model yields physical insight, allowing engineer to design system to operate in specified manner.
- Approximations neglect small but complicating phenomena.
- After basic insight has been gained, model can be refined (*if it is judged worthwhile to expend the engineering effort to do so*), to account for some of the previously neglected phenomena.
- There is no exact model; only useful and useless ones.

Properties of a System

- The conditions under which the system design is “correct” are precisely characterized — stability/convergence.
- Computational aspects of the solution are understood — computational complexity/timeliness.
- Graceful degradation of performance of the system for the given solution under disturbances that are unaccounted for — robustness.
- A solution with the above properties can be reliably implemented in the real world — designed using right abstractions.
- We do not look for silver bullets! One particular solution does not solve all problems, however, a systematic approach always yields a solution for given problems.

Tips to do well in Embedded

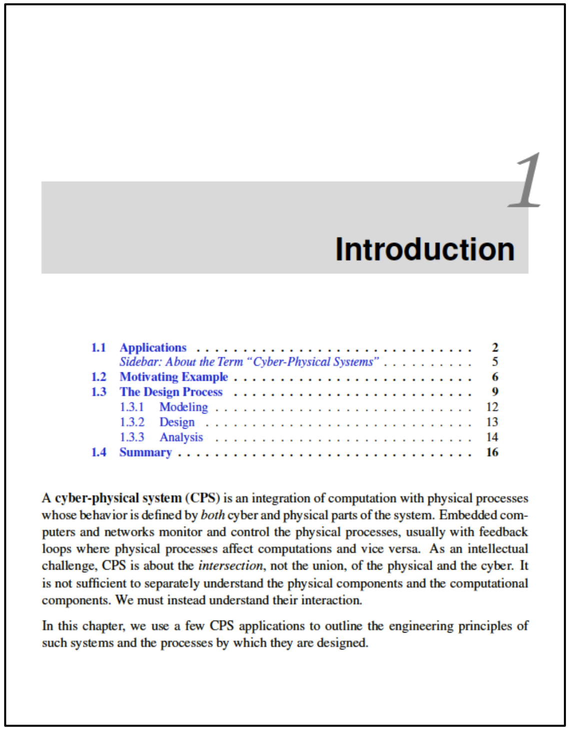
- **Stay current** with material taught throughout semester. Do **NOT** leave reading the textbook to the night before the exam!
- **Go over lecture material before it is presented** and then **reinforce material after it is presented**. Concepts learnt in lectures will be reinforced in workshops via real applications.
- **Read the textbook** for a more thorough explanation of what is covered in lectures.
- Use **feedback from graded assignments and workshops as well as un-graded quizzes** to assess your knowledge and guide your study.
- **Try problems yourself before asking how** to solve them.

Resources

- Lecture slides
- Recorded lectures
- Worked examples
- Subject textbook and other references
- Notes from workshop classes
- Assignment/quiz feedback
- Demonstrators in workshop classes
- Consultation sessions

Things to do …

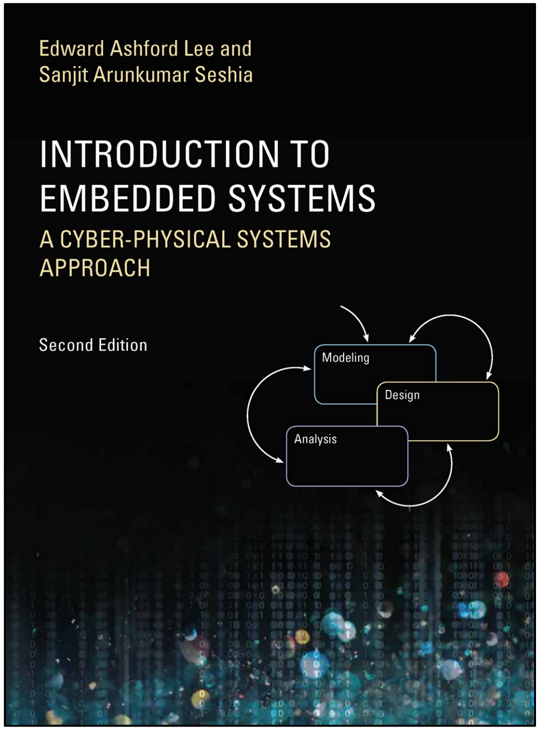
- Download the textbook and **read Chapter 1**
- **Sign up to a group** in your workshop class
  - Don't worry if you don't know anyone –you will be assigned a group in your first workshop class



1 Introduction

A **cyber-physical system** (CPS) is an integration of computation with physical processes whose behavior is defined by *both* cyber and physical parts of the system. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. As an intellectual challenge, CPS is about the *intersection*, not the union, of the physical and the cyber. It is not sufficient to separately understand the physical components and the computational components. We must instead understand their interaction.

In this chapter, we use a few CPS applications to outline the engineering principles of such systems and the processes by which they are designed.



Next Lecture

- Cyber-Physical Systems