Lecture 1: Introduction to Embedded Systems
An *embedded system* is:

- a special-purpose computer designed to perform dedicated functions often with real-time constraints.
- a system embedded as part of a complete system.
- really any system that is not a PC...although PCs contain several embedded systems.
Five main categories:
- Digital signal processing
- Control
- Networking
- User interfacing
- Data storage

Most embedded systems perform more than one of these functions.
Popularity with consumers

>99% of new microprocessors are found in embedded systems.

Cnet’s Top 10 Must-haves

- Apple iPhone
- Apple iPod Nano
- Vudu
- Nintendo Wii
- Apple MacBook
- Sling Media Slingbox A/V
- Sony PSP
- Samsung LN-T4665F
- Sony Handycam HDR-CX7
- Shure SE110 Sound Isolating Earphones
Popularity with other industries

Automotive
   Air bag controllers
   Anti-lock brakes

Communications
   Satellite phones
   Cell phone base stations

Industrial
   Point-of-sale systems
   Robotics

Medical
   Life-support
   Medical testing

Military
   GPS
   Missile guidance
Why is it unique?

Computation is subject to physical constraints such as timing deadlines, memory restrictions, and power consumption requirements.
The traditional abstraction of separating software from the hardware and environment does not work.
Hardware, software, and control are integrally intertwined.
Designers must understand both hardware, software, and control.
"The Embedded Systems Design Challenge" by Henzinger and Sifakis continues this discussion.
Top-Down Design Process

- Analyze the problem
  * Specifications
  * Constraints

- High level design
  * Block diagrams
  * Data flow graphs

- Engineering design
  * Call graphs
  * Data structures
  * I/O interfaces

- Implementation
  * Hardware
  * Software

- Testing
  * Not done
  * Done

New requirements → Analyze the problem → High level design → Engineering design → Implementation → Testing → New requirements
Discover the requirements and constraints.

Requirements are general parameters that the system must satisfy.

Specifications are detailed, specific requirements.

Constraints are limitations under which the system must operate.
Embedded system design metrics

Nonrecurring engineering cost
Unit cost
Size & weight
Performance (accuracy, precision, resolution, response time, bandwidth)
Power
Flexibility, maintainability, reliability, testability, & compatibility
Time-to-prototype
Time-to-market
Correctness
Safety
Look & feel
High-Level Design Phase

Build a conceptual model of the hardware and software system.
Design broken into modules or subcomponents.
Estimate cost, schedule, and expected performance.
Develop a data flow graph for the system.
Data Flow Graph for a Motor Controller
Construct a preliminary design. This should include the hierarchical structure, basic I/O signals, shared data structures, and overall software scheme. Build mock-ups of mechanical parts and user software interface. 

*Call graphs* can be used to show how software and hardware interact.
Call Graph for a Motor Controller

- Timer
  - ADC routines
    - ADC
  - Actuator routines
    - Actuator
  - LCD routines
    - LCD
  - Keypad routines
    - Keypad
During this phase, the design is actually built. Implementation of subcomponents may actually be started during the earlier phases. Debugging embedded systems can be very difficult. Therefore, extensive use of hardware/software simulation and cosimulation is essential.
During this phase, we evaluate the performance. First, debug and validate the basic functions of the system. Next, evaluate and optimize various performance parameters such as execution speed, accuracy, and stability.
During this phase, we:

- Correct mistakes,
- Add new features,
- Optimize execution speed or program size,
- Port to new computers or operating systems, and
- Reconfigure the system to solve a similar problem.

Must be able to deal with changes in requirements or constraints.

Not actually another phase, but more loops through the entire cycle.
Basic Components of a Computer System
Memory-Mapped Computer System

- Processor
- RAM
- ROM

Bus

Memory Mapped I/O System

Input Devices

Output Devices

Input Signals

Output Signals
Isolated I/O Computer System

- Intel 486 processor
- Memory Control
- RAM
- MEMR, MEMW
- ROM

Isolated I/O System

I/O Control
- IOR, IOW

Input Devices
- Input Signals

Output Devices
- Output Signals
- Address, Data
Memory Read Cycle
DMA Read Cycle