ECE 1175
Embedded Systems Design

Introduction

Wei Gao

Course Information

- Class time: 3:00pm – 4:15pm Mondays and Wednesdays
  - Lectures will be online in the first two weeks
  - Zoom link for lectures: https://pitt.zoom.us/j/95207016944
- Instructor: Wei Gao, weigao@pitt.edu
  - Office hour: 2:00pm – 3:00pm Mondays and Wednesdays
  - Zoom link for office hours: https://pitt.zoom.us/j/92599848819
- TA: Xiangyu Yin, eric.yin@pitt.edu
  - TA Office hour: TBD
- Course materials will be posted on Canvas
  - Each lecture note will be posted before class
  - For online lectures, lectures will be recorded and video recordings will also be available on Canvas
- Canvas will also be used for posting announcements and grades

Lab Information

- Lab hours: Monday 6:00pm – 8:50pm
  - 1211 B/C
- Labs will start on 1/24
  - Labs will be open after we have covered the basics of course contents
  - Every student should pick up one Raspberry Pi 4 kit from the TA before the first lab session
- Online labs
  - Zoom link: https://pitt.zoom.us/j/91374645635

Definition

- Embedded system: any device that includes a computer but is not itself a general-purpose computer.

- Application specific
  - The design is specialized and optimized for specific application
  - Don’t need all the general-purpose bells and whistles.

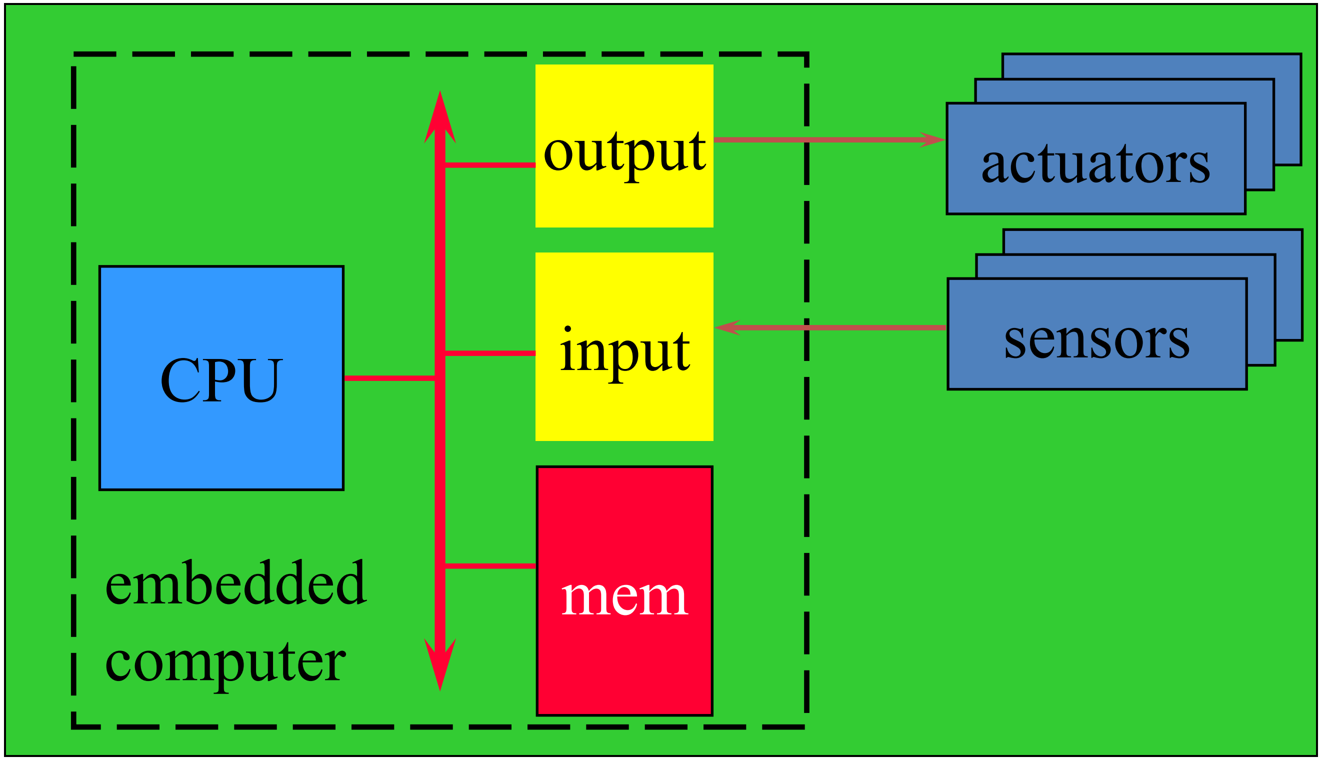
Examples of Embedded Systems

- Cellphone, Personal Digital Assistant (PDA)
- Printer.
- GPS.
- Automobile: engine, brakes, dash, etc.
- Digital camera.
- Household appliances: microwave, air conditioning
- Wrist watch.
- Implanted medical devices.
- and a lot more …

- Fact: > 95% of all microprocessors are used for embedded systems.

Embedding a Computer

- A computer-on-a-chip
  - a type of microprocessor emphasizing self-sufficiency and cost-effectiveness
  - has CPU, input/output, RAM, ROM, clock and etc on a single chip



Embedding a Computer

- More than simply a computer
- Functionality
  - Sensing
  - Communication
  - Control & actuation
- The appropriate I/O is important!

Early history

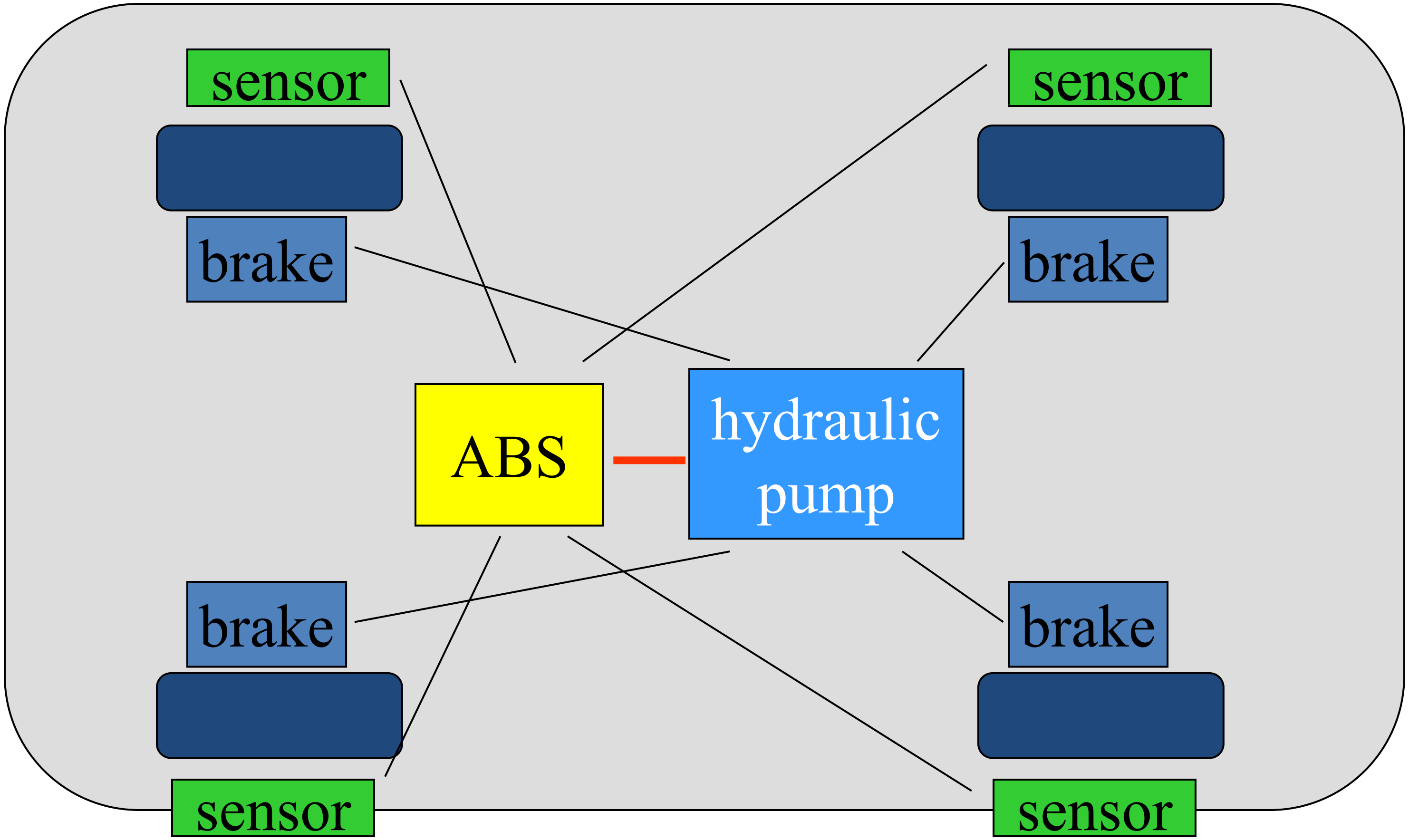
- Late 1940’s: MIT Whirlwind computer was designed for real-time operations.
  - Originally designed to control an aircraft simulator.
- First microprocessor was Intel 4004 in early 1970’s.
  - Predecessor of the famous Intel 8086
- HP-35 calculator used several chips to implement a microprocessor in 1972.

Early history

- Automobiles used microprocessor-based engine controllers starting in 1970’s.
  - Control fuel/air mixture, engine timing, etc.
  - Multiple modes of operation: warm-up, cruise, hill climbing, etc.
  - Provides lower emissions, better fuel efficiency, and better safety

Anti-Lock Brake System

- Pumps brakes to reduce skidding.
  - Real-time and safety



Automotive Systems

- A high-end car may have **100** microprocessors:
  - 4-bit microcontroller checks seat belt;
  - microcontrollers run dashboard devices;
  - 16/32-bit microprocessor controls engine;
  - Navigation;
  - Entertainment: DVD, audio, satellite radio…

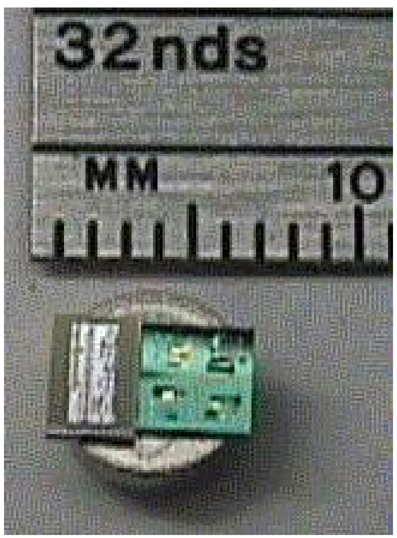
Other examples

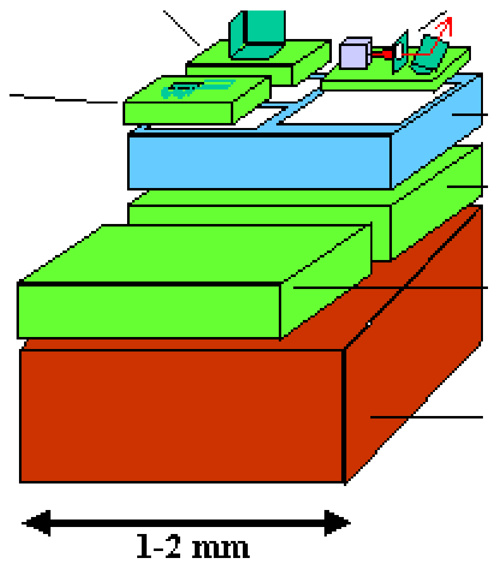
- Simple control
  - Front panel of microwave oven
  - Digital control of air conditioning
- Canon EOS 3 has three microprocessors.
  - 32-bit RISC CPU runs autofocus and eye control systems.

- Sony BRAVIA LCD TV has a standalone microprocessor for image processing
  - BRAVIA Engine
  - HD video streaming: high throughput required
  - AI embedding

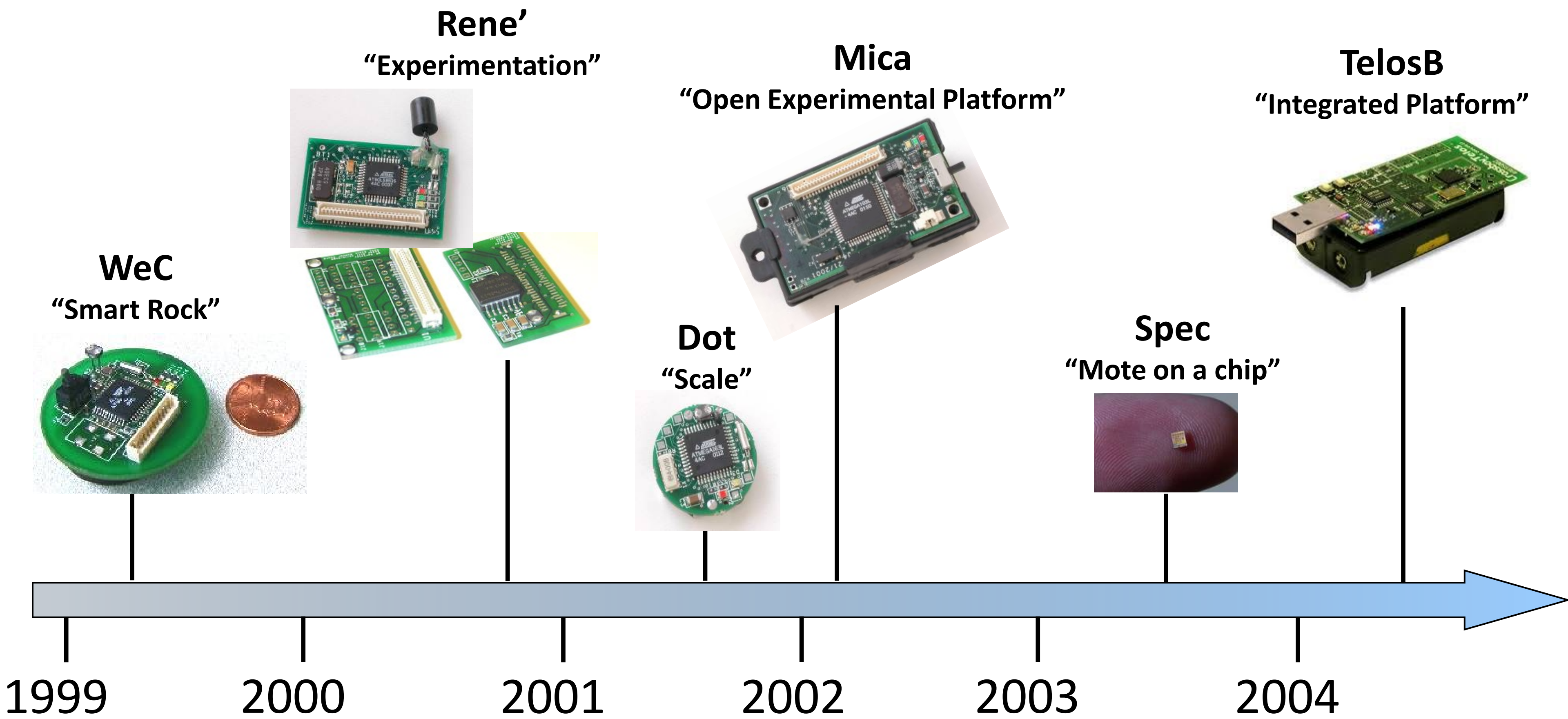
Typical Application: Wireless Sensor Motes

- sensors + microcontroller + radio
  - All feasible at very small scale
  - e.g., Berkeley Smart Dust



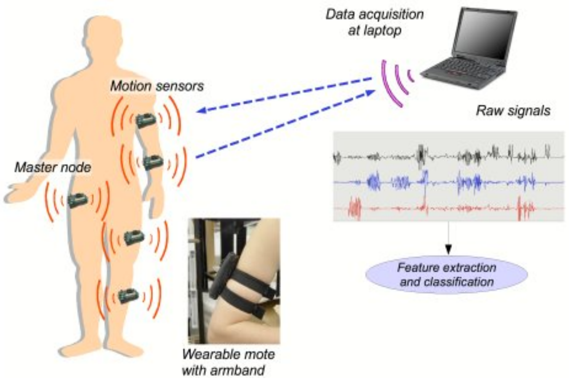


Sensor Motes Timeline



Typical Applications of Sensor Networks

Structural Health Monitoring

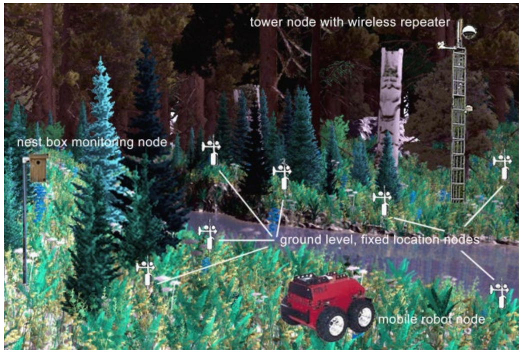


Health Care

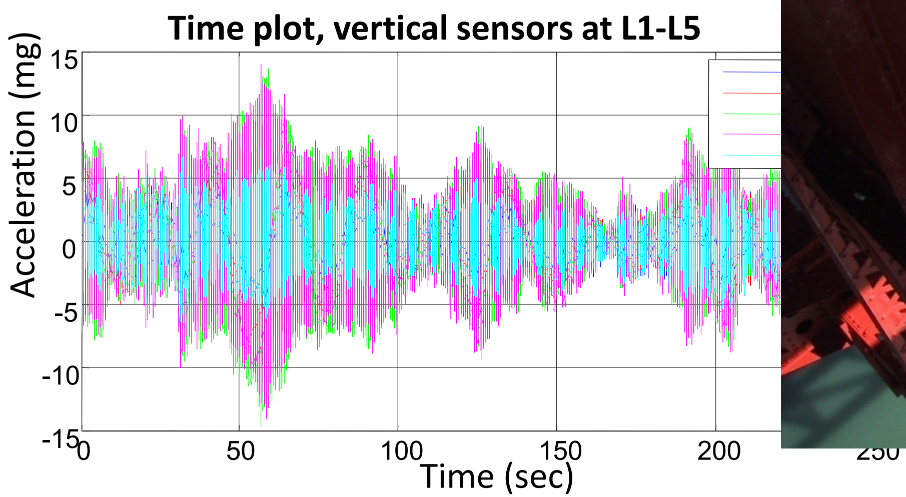
1. Smart sensors massively distributed and embedded in environments
2. Self-organized wireless network by communicating with each other
3. Real-time environment monitoring and control

Fire Monitoring

Habitat Monitoring



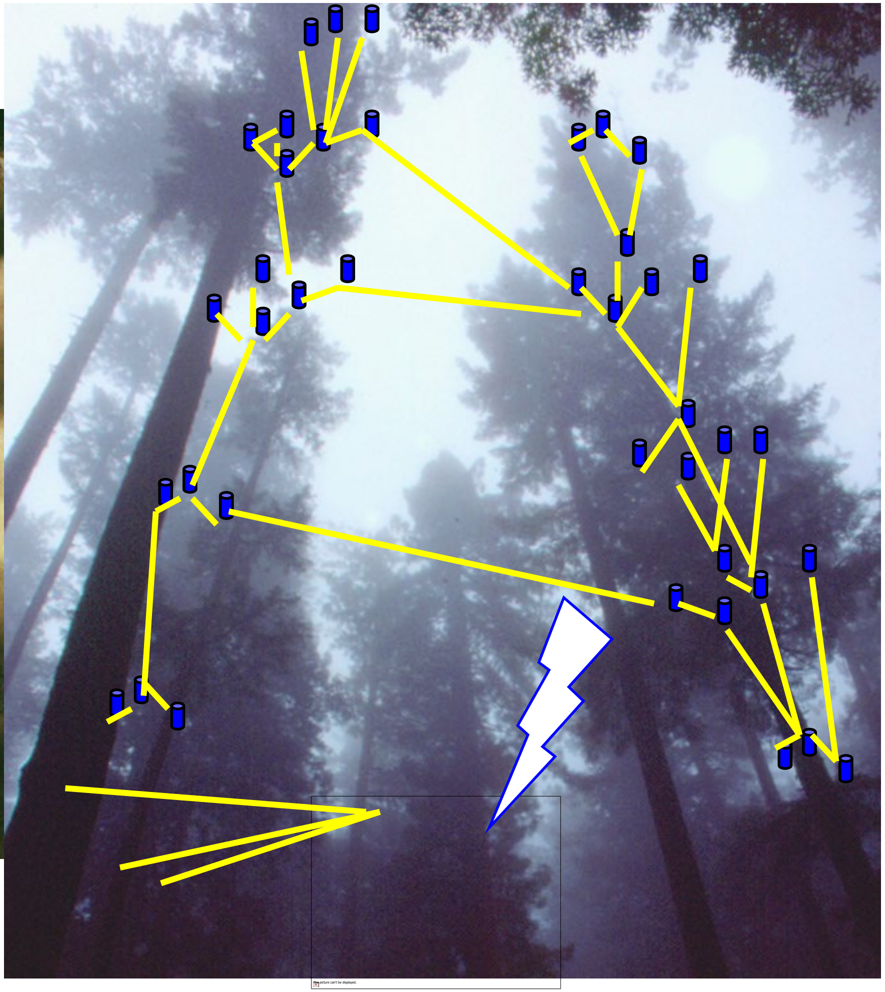
Structural Health Monitoring



Source: David Culler's MobiHoc'05 Keynote

Habitat Monitoring – Redwood Trees

- To understand complex interactions of tree growth and environment



Source: David Culler's MobiHoc'05 Keynote

Smartphones – Blurring Boundary

- Specialized microprocessor
  - Somehow much more powerful

Samsung Galaxy S8

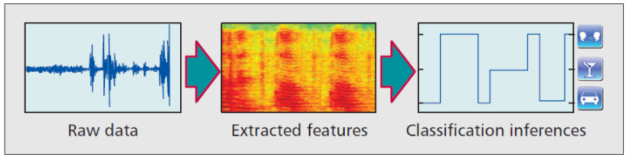
2.3 GHz w/ 8 cores

iPhone X

2.4 GHz w/ 6 cores

- Controls radio, sensors, display, speaker, …
- Multimedia contents and applications

Typical Applications using Smartphones



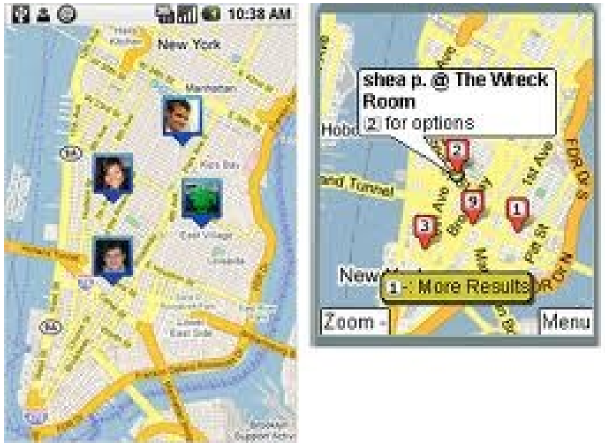
Sensing human behaviors

Demo from Dartmouth

1. **Sensing capabilities of smartphones**
   - **Mobile sensing**
2. **Mobile Internet access and user interaction**
   - **Mobile social media**
3. **Location-based services**

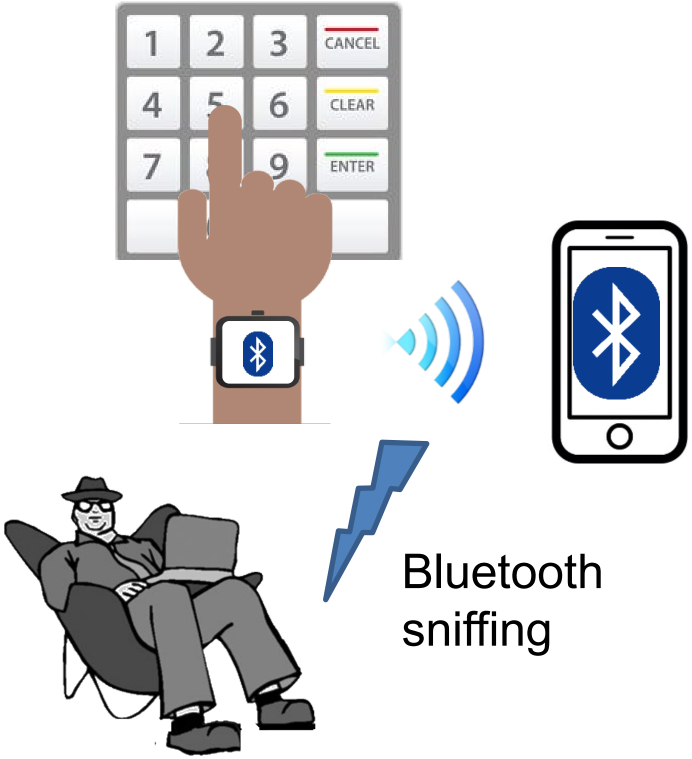
Mobile social networks

Yelp on smartphones



Wearables

- Smartwatches & wristbands



- New opportunities and challenges

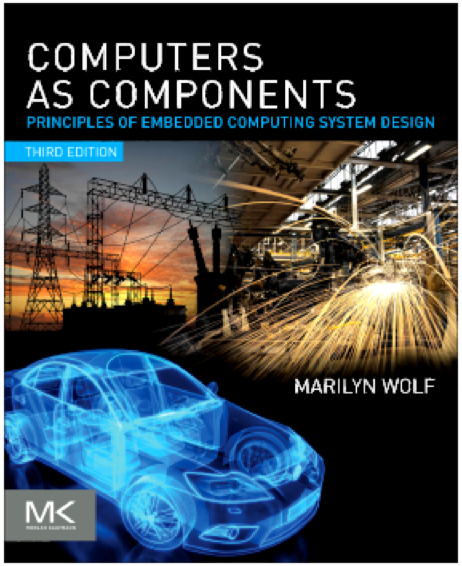
Why are those systems special?

- Application specific
  - Specialize and optimize the design for specific application
  - *Not* a general-purpose computer.
    - Don't need all the bells and whistles, e.g., hard drive, monitor, keyboard…
- Have to worry about both hardware and software
- Have to worry about non-functional constraints
  - Real-time
  - Memory footprint
  - Power
  - Reliability and safety
  - Cost

Just functionally working is NOT enough!

Goal of This Course

- The common principles of embedded systems design
  - Instead of a specific type of microcontroller
  - Emphasize aspects that are distinct to embedded systems
- Recommended Textbook:
  - *Computers as Components: Principles of Embedded Computing System Design*, 3rd edition
  - Marilyn Wolf, Georgia Tech
  - Available online: https://pitt.primo.exlibrisgroup.com/permalink/01PITT_INST/g3767l/alma9998575204906236
- Experimentation platform
  - The up-to-date microcontroller platform with strong capabilities



Goal of This Course

- References
  - Introduction to Embedded Systems: A Cyber-Physical System Approach
    - Ashford Lee and Sanjit Seshia
    - http://leeseshia.org/releases/LeeSeshia_DigitalV1_06.pdf
  - Embedded Systems: A Contemporary Design Tool
    - James Peckol
    - Wiley, 2011

What will you learn from this course?

- Hardware/software co-design
  - Integrating individual component designs towards a **holistic system**
  - Systematic views and rationales
- Hardware
  - Embedded system components: microprocessors, I/O, caches, memory, bus
- Software
  - Programming and optimization for optimal performance and low power
  - Real-Time Scheduling Analysis
  - Operating System Basics
  - Real-Time Operating Systems (RTOS)
- Practical system programming
  - Embedded operating system: Raspberry Pi

Skillsets & Prerequisites

- What skills do you need?
  - Basic understanding of analog and digital circuits
    - Logic gates, Boolean algebra, circuit building
  - Basic understanding about computers
    - Concept of processor, register, memory
  - Basic programming skills with sufficient proficiency with C and/or Python
- What skills will you get?
  - How to think and design systems as a whole piece
  - Hands-on system building skills
    - Operating digital I/O with heterogeneous data inputs and outputs
    - Practical programming over embedded platforms
    - Practicing buzz-word techniques over real system platforms

What will you NOT learn from this course?

- Hardware chip design
  - ECE 1192/2192: Introduction to VLSI Design
  - ECE 2162: Computer Architecture
- Communication and Networking
  - ECE 1150: Introduction to Computer Networks
  - ECE 1472: Analog Communication Systems
- Mobile application development
  - Fancy UI, graphics optimization, user experience…

Grading

- Homework assignments (4): 15%
  - HW1: 3%, HW2 – HW4: 4% each
  - One-week turnaround time
  - 2-4 short questions
- Lab assignments (5): 30% - 6% each
  - Two-week turnaround time
  - Each lab corresponds to a specific topic covered in lectures
  - Provide opportunities to practice hands-on skills
- Midterm exam: 25%
- Final exam: 25%
- Participation 5%

Course policy

- Academic integrity
  - Must be your OWN work
  - No collaboration for homework/lab assignments
- Lab policy
  - Results must be checked by TA
- Exam policy
  - Closed-book, No discussion, No make-up exams
  - Online proctoring
- Class policy
  - Attend each lecture