ECE 1161/2161
Embedded Computer System Design 2

Introduction

Wei Gao

Course Information

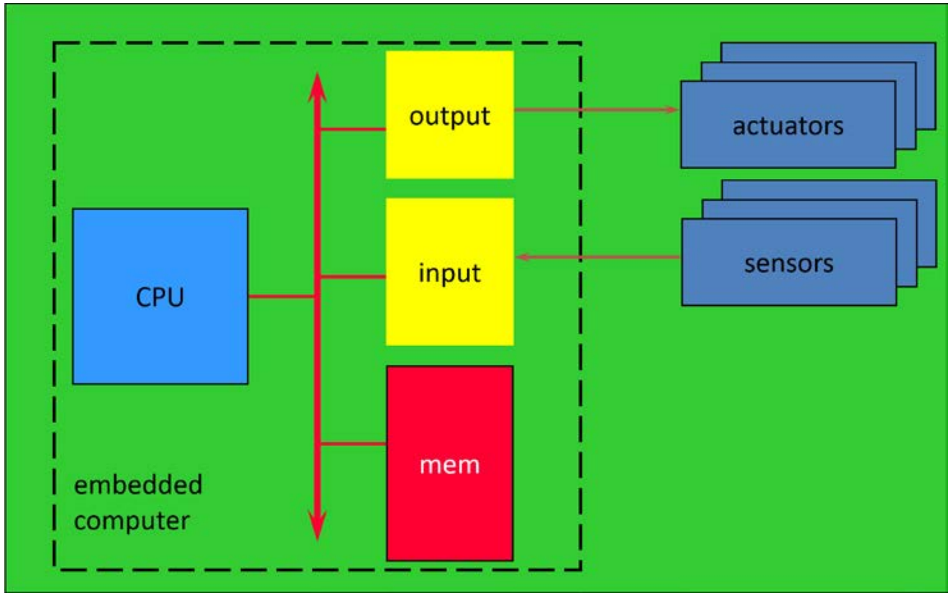
- Class time: 4:30pm – 5:45pm TuTh
- Instructor: Wei Gao, weigao@pitt.edu
  - Office: 1205 Benedum
  - Office hour: 3:30pm – 4:30pm TuTh
- TA: Ting Han, tih34@pitt.edu
  - TA office: 1237 Benedum
  - TA Office hour: TBD
- Schedule and lecture notes posted at instructor website
  - http://www.pitt.edu/~weigao/ece1161/spring2018/ece1161-sp18.html
- CourseWeb is used for posting announcements, grades and project feedback

Course Description

- Course description in catalog
  - Organized as a full term project carried out by student design groups. A complex embedded system will be designed, implemented and tested ~~using Altera and other CAD tools.~~ Grade will be based on project reviews and the final project report. Proper design process will be emphasized.

Course Description

- ECE1160/2160 Embedded Computer System Design 1
  - Common design methodology and concepts
  - Embedded system basics
    - Primary technologies of different system components
    - Design considerations: real-time, power, cost



Course Description

- ECE1161/2161 Embedded Computer System Design 2
  - Applying these basics to emerging application paradigms
  - Advanced design choices and technologies
  - Practical hands-on skills

Cyber-physical systems

Smart Cities and Communities

Virtual Reality

Course Description

- ECE1161/2161 Embedded Computer System Design 2
  - Advanced embedded computing architecture
  - Mobile cloud and edge computing
  - Advanced I/O interfaces
  - Batteryless sensing and communication
  - Energy harvesting
  - Cross-technology communication

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.

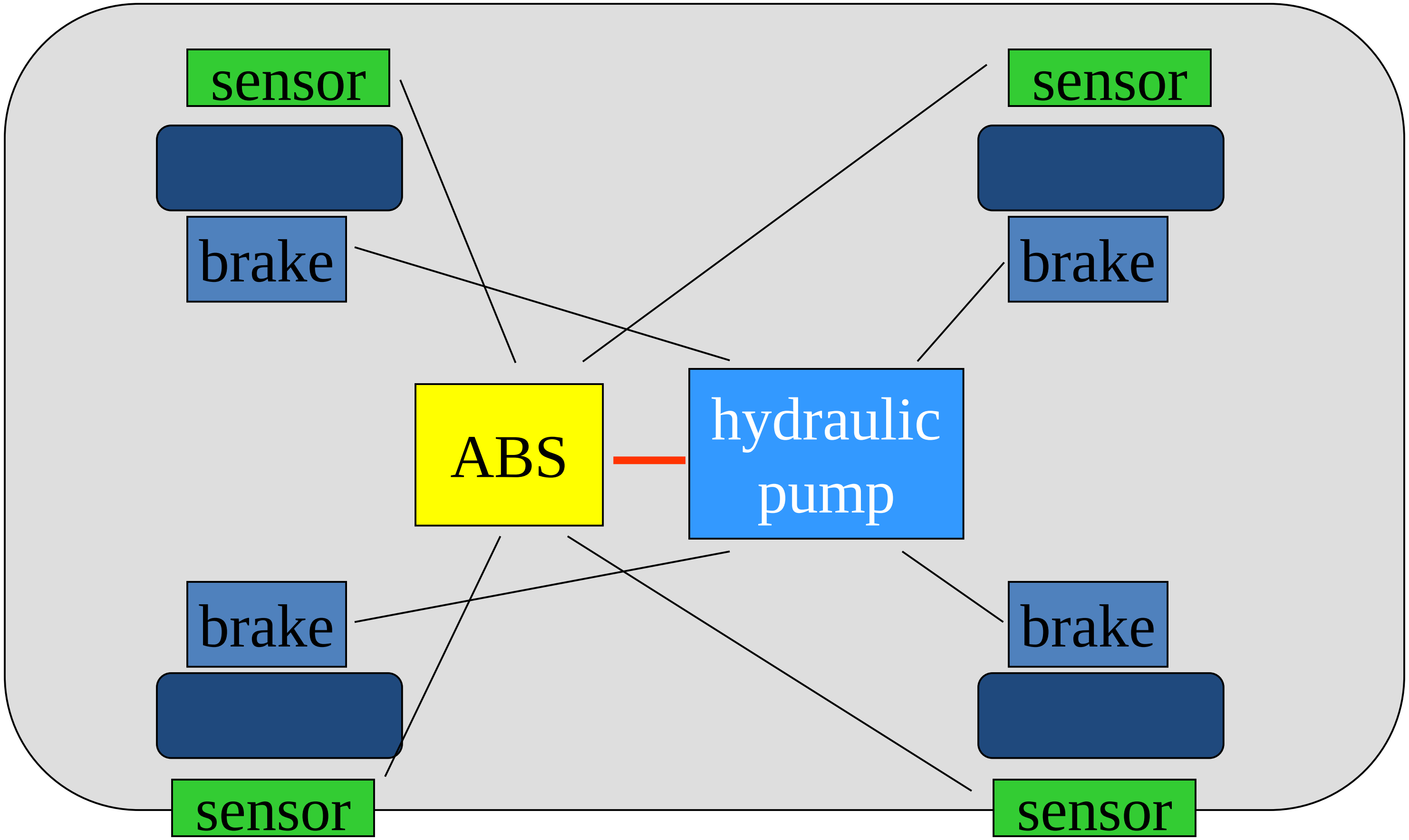
Examples of Embedded Systems

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

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

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…
- Future
  - Cars may drive by themselves??
  - Control your car by speaking out, or even your mind

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 2
  - Full 1080p video streaming: high throughput required

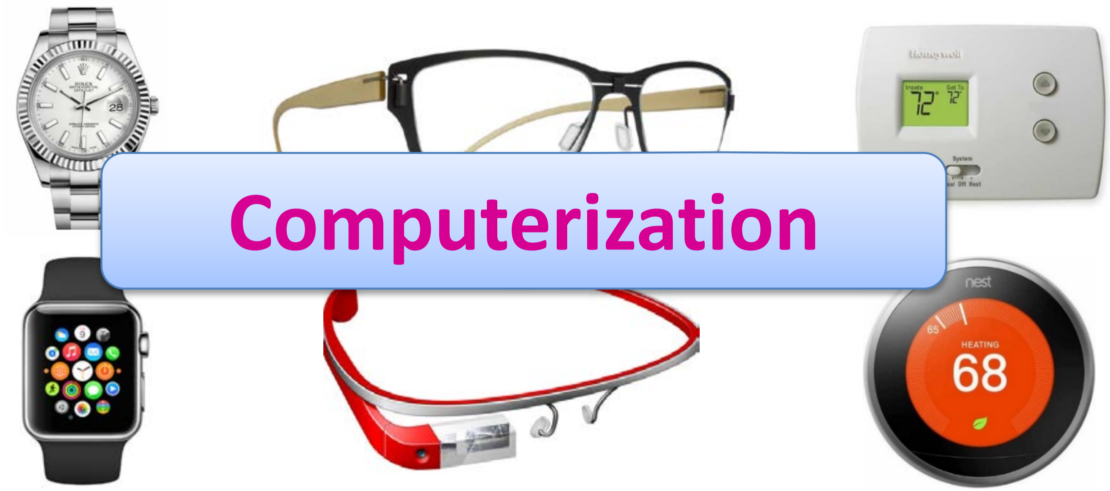
Why are those systems special?

- 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!

Cyber-Physical Systems

- Physical objects -> digital objects



  - Wearable computing
  - Innovative Sensing
  - Low-power networking
- Interconnection -> Internet of Things

Smart Building

- Occupancy sensing and monitoring
  - Camera, infrared, ultrasound, etc
  - Adjustment based on user needs
- Remote and intelligent control
  - Lighting, HVAC, sound
  - Custom and zonal control
- Information infrastructure
  - Ubiquitous display and feedback
  - Emergency evacuation

- Honeywell's vision: https://www.youtube.com/watch?v=kQ3CJdwP3fY

Smart Cities and Communities

- What is a smart city?
  - https://www.youtube.com/watch?v=vpSLICKnjPc
  - Public safety
    - Gunshot detection: https://www.youtube.com/watch?v=f8jkApBTGd4
  - City surveillance and planning
    - Traffic monitoring and control
    - Air quality and noise monitoring
    - Array of Things in Chicago: https://www.youtube.com/watch?v=pFL5QNwgs6A

Intelligent Transportation System

- Autonomous driving
  - Road sensing
    - Traffic detection, pedestrian detection
  - AI decision and control
    - Following and avoidance

- Communication
  - Vehicle to road side
  - Vehicle to vehicle
  - Toyota's vision: https://www.youtube.com/watch?v=uwLE3csyDAc

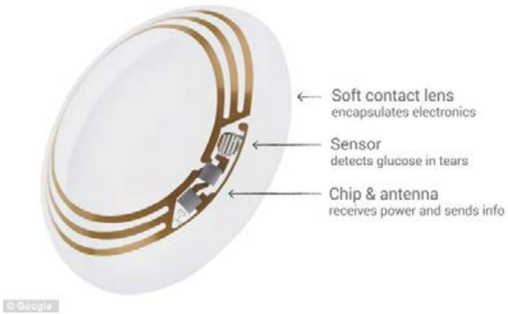
Virtual Reality

- Immersive experience
- Sensing is the key!
  - Headset
    - Gyroscope + accelerometer
    - Eye gaze tracking: https://www.youtube.com/watch?v=ImgfCFk8qy0
    - Emotion sensing: https://www.youtube.com/watch?v=2aXnfxH-anA
  - Hand controllers
    - Motion tracking with accelerometers
  - More controllers…

Smart Health

- Digital fitness tracking

- Tele medicine and diagnostics



- Surgery assistance
  - Smart brain surgery system: https://www.youtube.com/watch?v=QOafVIkLgyk

Course Organization

- *<u>Goal:</u>* design and implement your own embedded system idea as a semester-long project
- Part 1: classroom lectures
  - Key enabling technologies in the above application paradigms
    - Computing architecture
    - Communication
    - Sensing
    - Low-power design
  - Necessary background for project designs
- Part 2: Project milestone presentations & discussions
  - Your projects are expected to focus on the paradigms presented in Part 1
  - Keep your progress on track
  - Details to be discussed in the next class

What will you learn from this course?

- Most recent technical advances in emerging embedded computer systems
  - Integrated computation, communication, sensing and control
  - Revolutionary ideas and designs
- Hands-on experiences working with modern embedded platforms
  - Custom hardware prototyping
  - Familiarity and experiences with new hardware devices
  - Adoption of machine learning, AI and signal processing software
- Experiences in collaborative project development and management
- ***Optional:*** use this course as alternative of your senior design

What will you NOT learn from this course?

- Mathematics
  - We focus on hands-on system development skills
- Chip design
  - ECE 1192/2192: Introduction to VLSI Design
  - ECE 2162/3162: Computer Architecture
- Operating system
  - COE 1550: Introduction to Operating Systems
- Mobile application development
  - Fancy UI, graphics optimization, user experience…

Grading

- Based on your performance in course project
  - Project proposal: 10%
  - 4 interim project milestones: 15% each
  - Project final presentation & report: 25%
  - Class participation: 5%
- Project: 2-3 students per group
  - Each student will be graded individually
  - Each team member needs to undertake an equitable portion of workload
  - Details in next class

Course policy

- Academic integrity
  - Your project must be your OWN work
    - Your original project idea + development work
  - Online code repository: open-source only
  - No collaboration between teams
- Project policy
  - Be serious about taking this course: Late drop will affect your project teammates!
  - Clearly identify the contribution of each group member
- Class policy
  - No laptops in class
  - Attend each lecture
    - Active involvement in in-class project discussions is mandatory

Next Class

- Overview of course projects