ECE4144: Intro to Embedded Systems

NYU TANDON SCHOOL OF ENGINEERING
Spring 2025

EE4144: Intro to Embedded Systems

- Instructor:
  - Professor Matthew Campisi
    - Office: 370 Jay, 8th floor
    - Email: mcampisi@nyu.edu
  - Teaching Assistant:
    - TBD

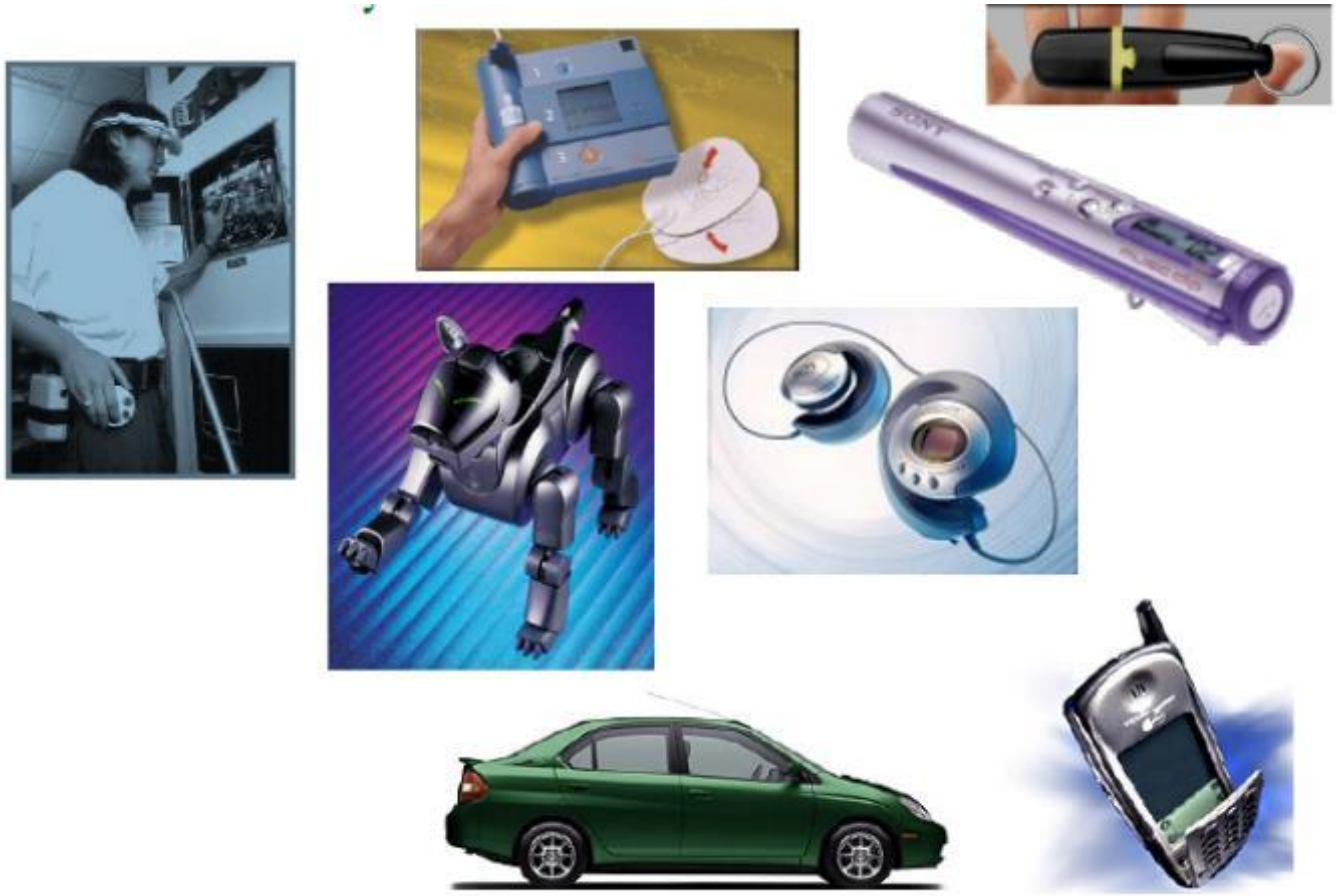
Logistics

- Class Meeting Times and Locations:
  - T/Th 10:00PM – 11:20PM (2MTC, RM:801)
- Textbook – No required Text, but lots of reference materials
- Quizzes:
  - 2 In Class Quizzes
- Embedded Challenge Spring 2025
  - TBD

Some Policies

- Grading Policy
  - 2 Quizzes 40% (20% each)
  - Homework/Labs 20%
  - Embedded Challenge S2025 40%

EMBEDDED SYSTEMS



EMBEDDED SYSTEMS

What are Embedded Systems?

- Anything that uses a microprocessor but isn't a general-purpose computer
  - Smartphones
  - Set-top boxes
  - Televisions
  - Video Games
  - Refrigerators
  - Cars
  - Planes
  - Elevators
  - Remote Controls
  - Alarm Systems
- The user "sees" a smart (***special-purpose***) system as opposed to the computer inside the systems
  - "how does it do that?"
  - "it has a computer inside of it"
  - "oh, BTW, it does not or cannot run Windows or MacOS"
- The end-user typically does not or cannot modify or upgrade the internals

What about this course?

- **Hardware**
  - I/O, peripherals: sensors/actuators, memory, busses, devices, control logic, interfacing hardware to software
- **Software**
  - Lots of C and assembly, device drivers, low-level real-time issues, scheduling
  - Interrupts
- **Software/Hardware interactions**
  - Where is the best place to put functionality – hardware or software?
  - What are the costs:
    - Performance
    - Memory requirements (RAM and/or FLASH ROM)
- **Integration of hardware and software**
  - Programming, logic design, architecture
  - Algorithms, mathematics and *common sense*

Careers in Embedded?

- Automotive systems
  - Perhaps designing and developing “drive-by-wire” systems
  - Self-driving vehicles
- Telecommunications
- Medical Devices
- Consumer electronics
  - Cellular phones, MP3 devices, integrated cellular/tablet
  - Set-top box and HDTV
  - Home and Internet appliances/ IOT
    - Your refrigerator will be on the internet more than you are!
- Defense and weapons systems
- Process control
  - Gasoline processing, chemical refinement
- Automated manufacturing
  - Supervisory Control and Data Acquisition (SCADA)
- Space communications
  - Satellite communications

Goals of the Course

- **High-level Goals**
  - Understand the scientific principles and concepts behind embedded systems
  - Obtain hands-on experience in programming embedded systems

By the end of the course, you should be able to

- Understand the “big ideas” in embedded systems
- Obtain direct hands-on experience on both hardware and software elements commonly used in embedded systems design
- Understand the basics of embedded system application concepts such as signal processing and feedback control
- Understand and be able to discuss and communicate intelligently about:
  - Embedded processor architecture and programming

The Big Ideas

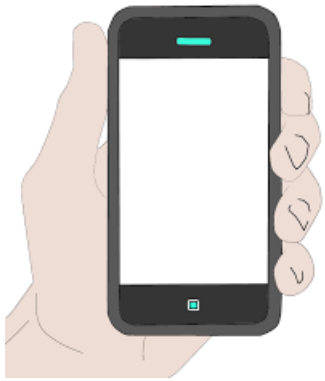
- HW/SW Boundary
- Non processor centric view of architecture
- Bowels of the operating software
  - Specifically, basic real-time operation with interrupts
  - Concurrency
- Real-world design
  - Performance vs. cost tradeoffs
- Analyzability
  - How do you “know” that your drive-by-wire system will function correctly?
- Application-level techniques
  - Signal processing, control theory

What is an Embedded system?

- An **embedded system** is a computer **system** with a dedicated function within a larger mechanical or electrical **system**, often with real-time computing constraints. It is **embedded**as part of a complete device often including hardware and mechanical parts. **Embedded systems** control many devices in common use today.

- Typically dedicated software (may be user customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general purpose display device

However, every system is unique – there are always exceptions

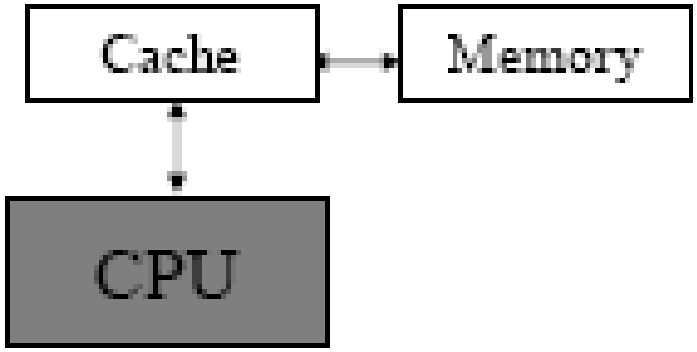


CPU: An All-Too-Common View of Computing

- Measured by:
  - Performance

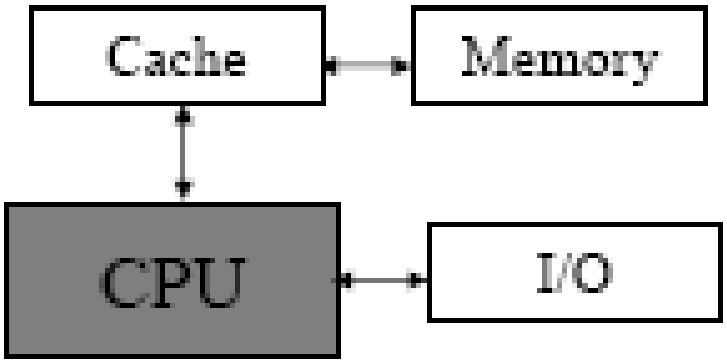
An Advanced Computer Engineer's View

- Measured by: Performance
  - Compilers matter too…



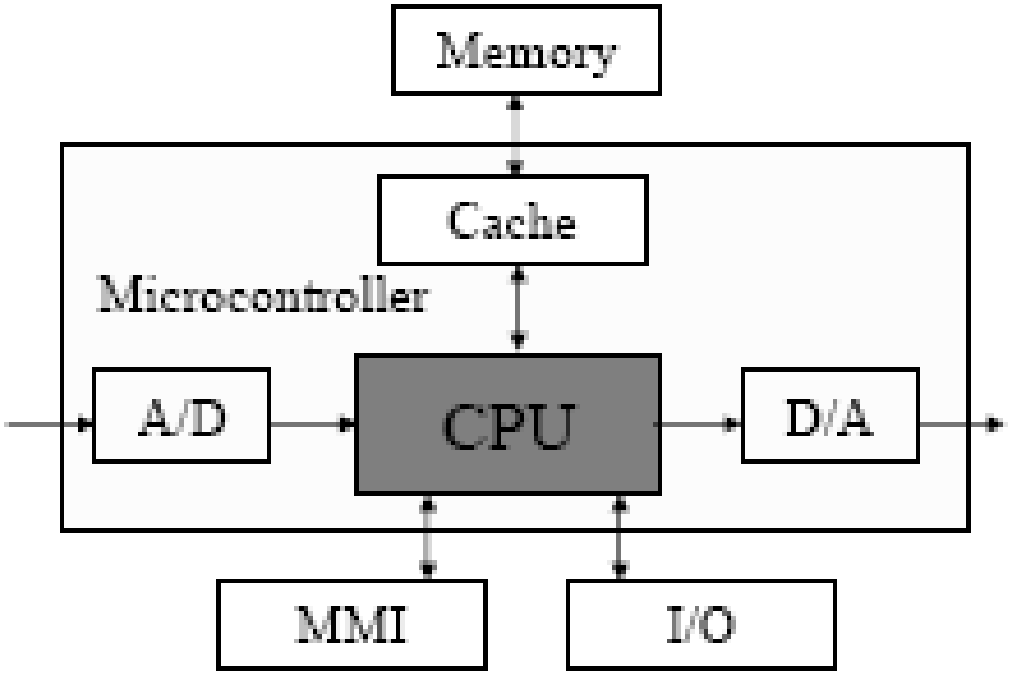
An Enlightened Computer Engineer's View

- Measured by: Performance, Cost
  - Compilers & OS matters



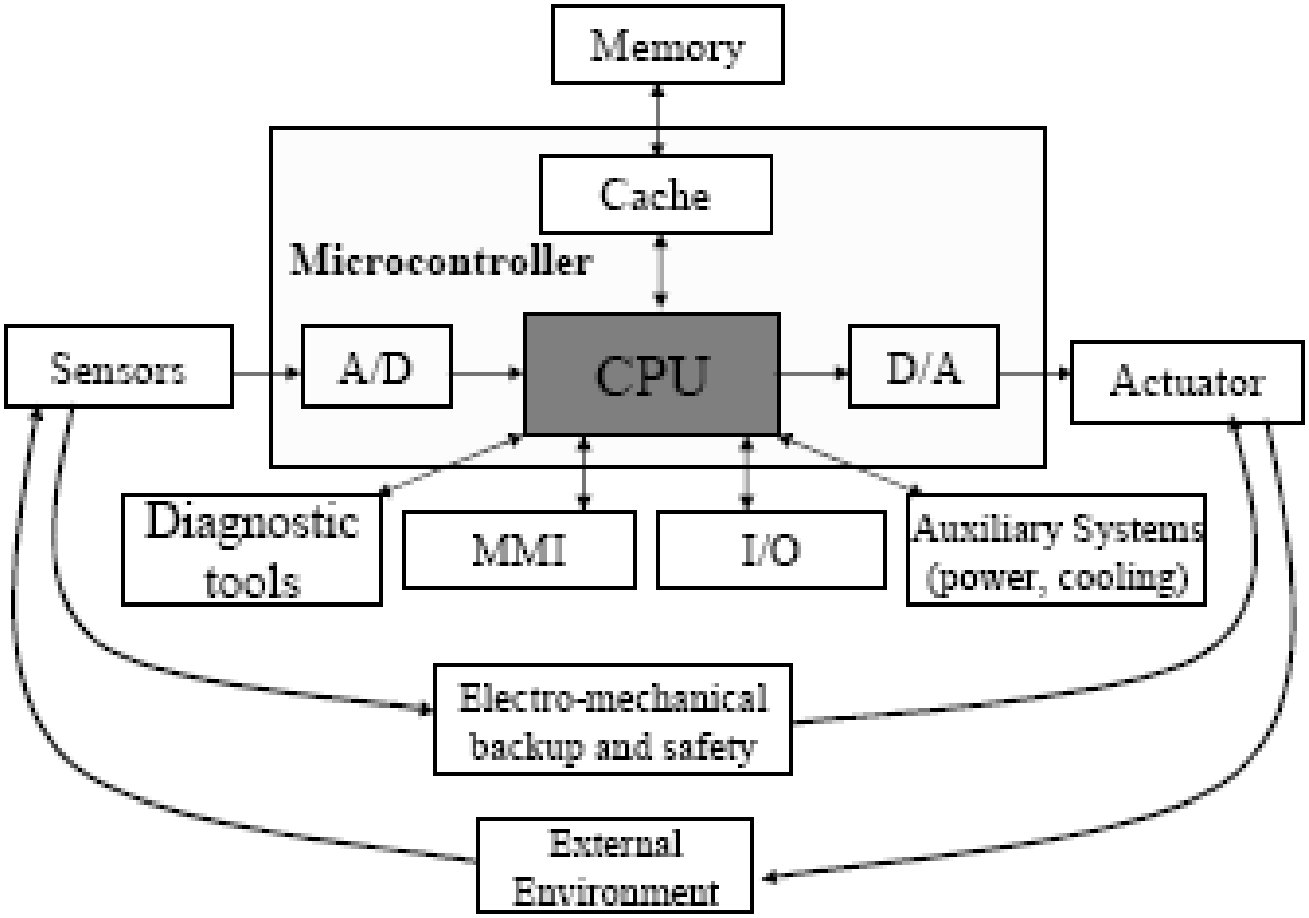
An Embedded Computer Designer's View

- Measured by: Cost, I/O connections, Memory Size, Performance



An Embedded Control System Designer's View

- Measured by: Cost, Time to Market, Cost, Functionality, Cost & Cost



A Customer's View

- Reduced Cost
- Increased Functionality
- Improved Performance
- Increased Overall Dependability

Why are Embedded Systems Different?

Four General Categories of Embedded Systems

1. General Computing
   - Applications similar to desktop computing, but in an embedded package
   - Video games, set-top boxes, wearable computers, automatic tellers
   - Tablets, Phablets

2. Control Systems
   - Closed loop feedback control of real-time system
   - Vehicle engines, chemical processes, nuclear power, flight control

3. Signal Processing
   - Computations involving large data streams
   - Radar, Sonar, Video compression

4. Communication & Networking
   - Switching and information transmission
   - Telephone system, Internet
   - Wireless everything

Typical Embedded System Constraints

- Small Size, Low Weight
  - Handheld electronics
  - Transportation applications weight costs money

- Low Power
  - Battery power for 8+ hours (laptops often last only 2 hours)
  - Limited cooling may limit power even if AC power available

- Harsh environment
  - Heat, vibration, shock
  - Power fluctuations, RF interference, lightning
  - Water, corrosion, physical abuse
- Safety critical operation
  - Must function correctly
  - Must not function incorrectly
- Extreme cost sensitivity
  - $0.05 adds up over 1,000,000 units

Embedded System Design World-View

A complex set of tradeoffs:

- Optimize for **more than just speed**
- Consider **more than just the computer**
- Take into account **more than just initial product design**

Multi-Discipline

- Electronic Hardware
- Software
- Mechanical Hardware
- Control Algorithms
- Humans
- Society/Institutions

X

Multi-Phase

- Requirements
- Design
- Manufacturing
- Deployment
- Logistics
- Retirement

X

Multi-Objective

- Dependability
- Affordability
- Safety
- Security
- Scalability
- Timeliness

Embedded System Designer Skill Set

- Appreciation for multidisciplinary nature of design
  - Both hardware & software skills
  - Understanding of engineering beyond digital logic
  - Ability to take project from specification through production
- Communication and Teamwork skills
  - Work with other disciplines, manufacturing, marketing
  - Work with customers to understand the real problem being solved
  - Make a good presentation; even better – write “trade rag” articles
- And, by the way, technical skills too…..
  - Low-level: Microcontrollers, FPGA/ASIC, assembly language, A/D, D/A
  - High-Level: Object oriented design, C/C++, Real Time Operating Systems
  - Meta-level: Creative solutions to highly constrained problems
  - Likely in the future: **U**nified **M**odeling **L**anguage, embedded networks