CSE477 – Embedded Systems Design

- Welcome to CSE 477
  - Instructor: Gaetano Borriello
  - TAs: Mike Esler, Jeff Nichols
  - Industry Consultant: John Wedgwood
- Some basics
  - what is a system?
  - what is digital system design?
- Objectives of this class
  - embedded systems: hardware and software
  - interfacing and communication
  - projects: product concept to prototype, focus on invisible computing
- Class administration and logistics

What is a system (in our case, mostly digital)?

- A collection of components
  - that perform a specific function
  - judiciously chosen to meet some constraints
    - cost, size, power consumption, safety
  - communicates with its environment
    - human interaction
    - communication with other systems over wired/wireless networks
- One person's system is another's component
  - no universal categories of scope/size
  - subsystems need to be encapsulated
- How is it documented?
  - how much does one have to know about the internals to use it?
  - how easily can it be altered/re-configured? along what dimensions?

What is digital system design?

- Encompasses all computing systems
  - general-purpose
  - embedded or task-specific
- Involves many digital (and mixed digital/analog) technologies
  - programmable components (e.g., PLDs and FPGAs)
  - processors
  - interfaces to analog world (e.g., A/D, D/A, special transducers)
  - input/output devices (e.g., buttons, pressure sensors, etc.)
  - communication links to environment (wired and wireless)
- To create a device capable of solving a problem or performing a task that is useful to someone

Trends in digital system design

- Forces
  - cost (cheaper), size (smaller), weight (lighter), power (lower)
  - time-to-market (shorter)
  - upgradeability (in-the-field)
  - recyclability (reusable parts)
  - ubiquity (anywhere, everywhere, and highly task-specific)
  - standardization of interfaces (leverage)
- Effects
  - increased use of high-level languages
  - high-level specifications
  - automatic synthesis tools (hardware and software compilers)
  - programmable hardware (quick to prototype, reconfigurable)

Examples of embedded systems

Programmable hardware

- Programmable logic devices (e.g., PLDs, FPGAs)
- Microprocessors and microcontrollers
  - integration of surrounding logic onto processor chip
    - interrupt subsystem (priorities and vectoring)
    - timer co-processor (measures real-time in parallel w/ processor)
  - faster instruction cycle time
    - replace custom hardware that was previously required
    - parallelism in software
  - integrated memory (RAM, ROM, EPROM, cache, flash)
    - less parts but sensitivity to program/data size
  - processor cores with on-chip supporting logic/circuits
    - custom logic optimized to specific application
    - task-specific sensors and actuators (e.g., MEMS)
    - even application specific instruction sets (e.g., DSP processors)

CSE 477

- Capstone design course
  - serves to tie together the curriculum with an intensive design experience
- For computer engineering
  - programming, data structures, operating systems
  - electronics, logic design, computer design
  - communication skills (oral and written), documentation of designs
  - group effort (2 persons /group, 2-3 groups/theme) and interaction with users (customers)
- Project experience
  - connecting thread through the discipline
  - invaluable opportunity to add to student portfolio
  - just what employers/graduate schools want to hear about

Course rationale

- Assignments and exams
  - reinforce concepts presented in lecture
  - serve to create infrastructure for possible use in projects
  - opportunity to evaluate individual creativity and understanding
  - gain familiarity with laboratory equipment and software
- Embedded system design project
  - wide variety of possibilities in a chosen domain
  - design reviews of all projects (learn from others' experience)
  - must be possible to complete in 10 short weeks
  - presentation/documentation (this qtr coordinated with ENGR333D)
    - in-class presentations
    - web-based documentation

ENGR333D

- Special section of ENGR333
- Combining the subject matter of ENGR333 and CSE477 leads to efficiencies for students and instructors
- Expect it to be a more rewarding experience and lead to better on-line documentation
- Basic model:
  - work individually for ENGR333
  - combine work of group members for CSE477
  - permits revision and iteration

Project scope

- Groups of two or three students will construct a self-contained mostly-digital subsystem (i.e., a small embedded system)
- Projects within a set of themes
  - investigate cross-project communication issues
  - tie projects together into a story (for video later in the year)
- Characteristics
  - interfacing multiple components, based around a PC or PDA
  - analog/digital interfacing
  - reactive, real-time interactions
  - wired or wireless communication between components/other systems
  - exploit high-degree of integration
  - effective use of microcontrollers and SDKs

Project organization

- Hierarchical organization
  - 10 groups of 2 students
  - 2-3 groups within one theme
- Overall project domain
  - task-specific devices connected to web services
  - low-overhead (if not invisible) user interfaces
- Themes for this quarter
  - home automation
  - body networks
  - location-aware systems
  - advanced user interfaces

Course outline

- Introduction
- Microprocessors and microcontrollers
- Interfacing techniques
- Communication methods
- Software issues in embedded systems
- Case studies from industry guest lecturers
- Project design reviews and presentations
- Project demonstrations on 10 December

Course schedule

- First half
  - lectures
  - laboratory assignments
  - midterm exam
  - definition and initial design of project
- Second half
  - construction and debugging of project
  - design reviews and in-class presentations
  - project documentation on web
- Group meetings
  - 2nd week: project selection
  - 4th week: project planning
  - every week thereafter: theme meetings

Background (prerequisite) material

- Logic design (de-emphasized this quarter, more on this next qtr)
  - combinational logic
  - sequential logic
  - control/data-path
- Computer architecture
  - assembly language programming
  - computer organization
  - memory hierarchy
  - interrupt mechanisms
- Programming skills (strongly emphasized this quarter)
  - basic programming
  - data structures
  - facility with programming (C, Java)

Refreshers

- Courses
  - CSE341 – Programming Languages
  - CSE370 – Introduction to Logic Design
  - CSE378 – Machine Organization / Assembly Language Programming
  - CSE451 – Introduction to Operating Systems
  - CSE461 - Networks
- Find your textbooks and notes from these courses
  - review chapters and lecture notes as topics come up
  - review written assignments and any projects

Project ideas

- 3-D ultrasonic mouse
- MIDI flute synthesizer
- CCD camera
- Hand-held logic analyzer
- Mobile robots
- Talking toaster
- Some examples from the past two years . . .
- Some ideas for this quarter . . .
  - meet Thursday 4:30-6:30 to review a video and discuss
  - pizza/drinks will be provided

Goals for CSE477

- Lots of fun getting projects done
  - cool devices
  - amaze friends, family, and Ed Lazowska
- Lots of learning in the process
  - great way to complete your undergraduate career
  - killer interview material
- Produce a great new video
  - highlight the program and department
  - demonstrate that education can be at the leading edge
  - demonstrate the immediate relevance of our educational programs
  - help us draw more industrial support
  - win another video award (two previous videos have)