CSE 466 – Software for Embedded Systems

Instructor:
- Gaetano Borriello
  - CSE 572, Hours: W 10:30-11:30, Th 11:00-12:00
  - gaetano@cs.washington.edu

Teaching Assistants:
- Waylon Brunette
  - CSE 003, Hours TTh 2:30-4:30
  - wrb@cs.washington.edu
- Vaishnavi Sannidhanam
  - vaishu@cs.washington.edu
  - CSE 003, Hours TTh 2:30-4:30

CSE 466 – Software for Embedded Systems

Class Meeting Times and Location:
- Lectures: MGH 231, MWF 9:30-10:20
- Lab: CSE 003, T – Section A, 2:30-5:20
  - Section B, 2:30-5:20

Exams
- I: Friday, 5 Nov, MGH 231, 9:30-10:20
- II: Friday, 10 Dec, MGH 231, 9:30-10:20
- Final demo: Wednesday, 15 Dec, CSE Atrium, 8:30-10:20

Embedded systems

Definitions
- A device not independently programmable by the user.
- Specialized computing devices that are not deployed as general purpose computers.
- A specialized computer system which is dedicated to a specific task.
- An embedded system is preprogrammed to perform a narrow range of functions with minimal end user or operator intervention.

What it is made of
- Embedded systems range in size from a single processing board to systems with operating systems.
- A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function.
- Embedded systems can be part of a larger system or product, as in the case of an anti-lock braking system in a car.
- A specialized computer system that is part of a larger system or machine.
- Typically, an embedded system is based on a single microcontrol processor with the programs stored in ROM.
- Some embedded systems includes an operating system, but many are so small and specialized that the entire logic can be implemented as a single program.

Examples
- Virtually all appliances that have a digital interface – watches, microwaves, VCRs, cars – utilize embedded systems.
- A computer system dedicated to controlling some non-computing hardware, like a washing machine, a car engine or a missile.
- Examples of embedded systems are medical equipment and manufacturing equipment.
- While most consumers aren’t aware that they exist, they are extremely common, ranging from industrial systems to VCRs and many net devices.

Embedded system – from the web

What is an embedded system?

Different than a desktop system
- Fixed or semi-fixed functionality (not user programmable)
- Different human interfaces than screen, keyboard, mouse, audio
- Usually has sensors and actuators for interface to physical world
- May have stringent real-time requirements

It may:
- Replace discrete logic circuits
- Replace analog circuits
- Provide feature implementation path
- Make maintenance easier
- Protect intellectual property
- Improve mechanical performance

What do these differences imply?

Less emphasis on
- Graphical user interface
- Dynamic linking and loading
- Virtual memory, protection modes
- Disks and file systems
- Processes

More emphasis on
- Real-time support, interrupts (very small OS, if we’re lucky)
- Tasks (threads)
- Task communication primitives
- General-purpose input/output
- Analog-digital/digital-analog converters
- Timers
- Event capture
- Pulse-width modulation
- Built-in communication protocols
What is an embedded system? (cont’d)

- Figures of merit for embedded systems
  - Reliability – it should never crash
  - Safety – controls things that move and can harm/kill a person
  - Power consumption – may run on limited power supply
  - Cost – engineering cost, manufacturing cost, schedule tradeoffs
  - Product life cycle – maintainability, upgradeability, serviceability
  - Performance – real-time requirements, power budget

Capacity

- Assume:
  - 8 MHz processor @ one instruction/cycle
  - Assume fan runs between 30Hz and 60Hz
  - Assume 256ms period on speed control PWM, with 1ms resolution.

- What percent of the available cycles are used for the temperature controller?
  - (total instructions in one second) / (8MInstr/sec)

- How much RAM do you need?

- How much ROM?

Example: a temperature controller

Resource analysis of temp controller

Class logistics – see course web

- Class structure
- Business matters
- Grading
- Syllabus
- What we’ll be doing

Class structure

- Lecture
  - Closely linked to laboratory assignments
  - Cover main concepts, introduced laboratory problems
- Lab
  - Implementation of two projects
  - Lab reports due prior with 30 minutes of start of next lab section
- Exams
  - Two, based on lecture, lab, and reading
- Final demo
  - During scheduled final time – participation required
- Reading and source material
  - Some assigned, most you’ll find on your own
Business Matters

- Lecture slides will be online after class (links in several places)
- Get the CoursePak for CSE466 ($24.75, Communications B-042)
- Bring a $200 personal check to the first lab to check out a kit
- Random lab partner assignments, changed mid-quarter
- Sign up for CSE466 mailing list

Grading

- Lab reports:
  - Demonstration(s) required
  - Brief answers to questions embedded in assignment
  - Sometimes hand-in code
  - Do with your partner
- Distribution:
  - Labs: 40%
  - Exams: 30% (5 Nov and 10 Dec)
  - Demo: 10%
  - Class Participation: 20%

CSE466 Lab Projects

- Two multi-week projects
  - Four lab assignments each
  - Different lab partners
- First project
  - Familiarize with microcontroller
  - Learn how to interface various devices
  - Testing and debugging
  - Basic communication between chips and between chip and PC
- Second project
  - Wireless communication
  - Embedded operating system
  - Real-time issues
  - Testing and debugging
  - Emergent behavior of a collection of devices

CSE466 Lab Projects (cont’d)

- Project 1 – USB device
  - Platform: ATmega16 AVR microcontroller
  - Accelerometer and push-button used to control a room light
  - Connects sensor and actuator to PC through USB port
- Ball Lightning
  - Roll a ball to control a light (dimmer)
  - Accelerometer senses movement of ball – tilting
  - Push button activates sensing
  - Eventually would be wireless USB device to home PC
  - One or more in each room

CSE466 Lab Projects (cont’d)

- Project 2 – Ad hoc wireless network (“flock”)
  - Platform: UC Berkeley wireless sensor nodes (UCB “motes”)
  - Sound generation coordinated with neighbors and time of day
  - Emergent behavior between different nodes
- Flock-II
  - Install in Allen Center atrium for pleasing auditory display
  - Switch between bird songs, crickets, water sounds