CES 520 Embedded Systems (3 units)

Sonoma State University
Fall semester 2006
Tuesdays 6-8:50 pm
Salazar Hall room 2003

Instructor: Alan Bloom, 707-538-7115 (home), bloomal@sonoma.edu
Office hours: After class or by appointment

Prerequisites:

ES 230-231 Electronics I and Lab
CS 351 Computer Architecture

The student should be familiar with basic analog and digital circuits and components, computer architecture, binary arithmetic, programming in ANSI C, and the *concept* of assembly and machine-language programming.

Textbooks:

An Embedded Software Primer
David E. Simon
Addison-Wesley; ISBN 0-201-61569-X; © 1999, 15th printing (2005)

Embedded Systems Design using the Rabbit 3000 Microprocessor
Kamal Hyder and Bob Perrin
Newnes; ISBN: 0-7506-7872-0; © 2005, 1st printing

Course Description:

An embedded system may be defined as any system that contains a microprocessor but is not a computer. Today they are ubiquitous. Microprocessors are embedded in products ranging from microwave ovens to Boeing 767s. The challenges and methodologies applicable to designing embedded hardware and software differ in many respects from the ones faced by designers of conventional computers and non-embedded analog and digital systems.

The student will be introduced to some of those differences, how they affect system design issues, and tools and techniques for developing efficient and reliable systems. One difference is that hardware and software are tightly coupled in embedded systems. Often the same engineer designs both. Even if not, the software designer needs to have a good understanding of the hardware and vice versa.

In this course, the student's hardware and software skills will be brought together through hands-on applications while developing a real-world project. We will use a Rabbit Semiconductor development kit based on the Rabbit 3000, a medium-complexity microcontroller that includes most of the features found in typical devices intended for the

embedded market. Software will be developed in the C programming language which is extensively used in today's embedded systems. The labs are used as a series of building blocks that culminate in the final project.

Course Objectives:

The student will gain proficiency in techniques applicable to the design and development of hardware and software for embedded systems.

Policies:

Attendance: Since classes are three hours long, there is quite a bit of material in each session. A student who needs to miss a class should contact the instructor immediately to obtain assignments and course materials.

First day of class		August 22, 2006

Grading:		
Midterm exam	25%	October 17, 2006
Final exam	25%	December 12, 2006
Project	40%	Due December 5, 2006
Lab assignments	10%	
(Grades will be curved)		

Late work: Grade will be reduced 20% for each day late.

I will try to post class notes and graphics on the web the day following each class.

Lab Assignments:

Lab assignments are used as building blocks for the final project. Labs are to be done individually by each student. Some labs may require a functional demonstration to the instructor. Lab assignments involving writing code require the student to turn in the code for review. This may be done with paper printouts handed in at class or (preferably) in electronic form on the web or by email.

Project:

The project is to be done individually by each student. It will involve building a working prototype using the RCM3000 Development Kit. Any additional circuitry that the student adds to the development kit must be captured on a schematic. The student may use any schematic capture program of choice. Hand-drawn schematics are also acceptable, provided that they are neat and legible. All schematics must accurately reflect the hardware that the student has built. All software that is written must compile without warnings and run on the RCM3000 Development Kit. The software should follow good coding practices, and be well commented and easily understood by the instructor.

Course Outline:

Note: Portions of the course are subject to change to meet the needs of the students.

WEEK 1 August 22, 2006

- Introduction
    - Course mechanics
    - The field of embedded systems.
- Overview of embedded systems
- Lab 1: Install hardware and software. Run sample programs. (Due in 1 week)

WEEK 2 August 29, 2006

- Embedded Microcontrollers
    - Microcontroller features in more detail
    - Criteria for choosing a microcontroller
- The Rabbit 3000, architecture and features
- Other manufacturers of embedded microcontrollers

WEEK 3 September 5, 2006

- Software
    - Interrupts
    - Multitasking
        - Timing estimation
        - Interaction models
        - Execution strategies
    - The shared-data problem
    - Real-time scheduling
    - Principles of good coding for embedded systems
- Lab 2: Write a test program and library file. (Due in 1 week)

WEEK 4 September 12, 2006

- Software implementation issues
- Some serial interfaces in more detail
- Miscellaneous hardware
- Lab 3: Add EEPROM and temperature sensor hardware and software. (Due in 2 weeks)

WEEK 5 September 19, 2006

- Debugging
    - Design techniques to reduce bugs
    - Software tools
    - Hardware tools:
    - State Analysis and timing analysis

WEEK 6 September 26, 2006

- Driving Hardware
- Interfacing to the external world
- Interrupts, practical details
- Lab 4: Add relays and LEDs hardware and software. (Due in 2 weeks)

WEEKS 7/8 October 3/10 2006

- Real-time operating systems (RTOS)
    - Typical features of a RTOS
    - Architectures
    - Process vs thread
    - Interprocess communication (IPC) techniques
    - The priority inversion problem
    - POSIX
    - Memory management
    - Available real-time operating systems

WEEK 9 October 17, 2006

Mid-term exam

WEEK 10 October 24, 2006

- ASIC/FPGA design process
    - Simulation of the design at a high abstraction level
    - Generating HDL (Hardware Description Language)
    - Verification of design requirements
    - Logic systhesis
    - Verification of the equivalence of the resulting netlist to the HDL
    - Static timing analysis
    - Place and route.
    - Extraction models and netlist comparison
    - Simulation using test vectors.
    - Programming (FPGA) or fabrication (ASIC) of the part
    - Testing the part
- CPLD/PAL design

WEEK 11 October 31, 2006

-Analog interfacing
    - A/D converters
    - D/A converters
    - PWM
- Networks - begin this week and finish the topic next week

WEEK 12 November 7, 2006

- Networks
    - Definition of a network
    - Examples
    - Communication techniques
    - Field Area Networks
    - Fieldbuses
    - Physical networks
    - 7-layer OSI model
    - TCP/IP
    - Security issues
    - Network programming using Dynamic C
- Lab 5: Write program to serves a web page to display temp. and date/time. (due in 2 weeks)

WEEK 13 November 14, 2006

-High-level system design issues
    - Customer requirements
    - Hardware vs software tradeoff
    - Software language choice
    - Hardware choices
- Example system

REVIEW November 21, 2006

WEEK 14 November 28, 2006

- Testing
    - Timing analysis
    - Design for testability
    - Failing "safe"
    - Power-on self tests
    - Unit testing
    - Regression testing

WEEK 15 December 5, 2006

-SoC (System on a Chip)
    - On-chip networks (NoC)
    - IP libraries
    - Configurable processors
    - Testing SoC
- Formal models
- UML
- Formal verification methods