Master Course Syllabus

School of Engineering and Computer Science
Washington State University Vancouver

CS 466
Embedded Systems

3 Semester Hours
(2 lecture hours, 3 laboratory hours)

Catalog Description

Design and development of real-time and dedicated software systems with an introduction to sensors and actuators.

Prerequisite Courses

CS 360 – Systems Programming

Prerequisite Topics

- Proficiency with the C programming language
- Basic digital logic design
- Microprocessor systems and interfacing
- Operating systems concepts including concurrent programming, process synchronization and memory management.

Measured Course Outcomes

Students taking this course will:

1. Choose an appropriate scheduling algorithm and use it to schedule tasks in a real-time operating system, meeting the real time constraints of the system *(contributes to performance criterion* **E-1***)*.
2. Use synchronization primitives to control communications between tasks with different priorities in a real-time operating system *(contributes to performance criterion* **A-2***)*.
3. Design, implement and debug an embedded or real-time software program, which controls external devices and interprets data from external sensors *(contributes to performance criterion* **E-2** *and* **B-5***)*.
4. Deliver a well-organized, logical oral presentation, including good explanations when questioned *(contributes to performance criterion* ***G-2****)*.

Required Textbooks

David Simon, An Embedded Software Primer, Addison-Wesley, 1999, ISBN 0-201-61569-X

Jean J. Labrosse, MicroC/OS-II - The Real-Time Kernel, R&D Books (Miller Freeman), ISBN 0-87930-543-6

Reference Material

Embedded Systems Journal

Major Topics Covered in the Course

1. Review of microprocessor system architectures
2. Hardware architectures of embedded systems
3. Software architecture of embedded systems
4. Real-time operating systems (RTOS)
5. Networking in embedded systems
6. Real-time system specifications
7. Debugging real-time systems
8. Security & protection for embedded systems
9. Object-oriented design in RTOS
10. Advanced topics in embedded systems

Laboratory Projects

The students are required to implement embedded computer systems that sample a variety of sensors and actuators, constructing hardware or software interfaces to the sensors and actuators. Students also learn to use logic analyzers and digital oscilloscopes.

Programming Project Area	Weeks
Develop finite state machine firmware to de-bounce a switch	1
Introductory programming of microprocessor development board	1
Concurrent data sharing/process synchronization	1
Stepper motor control and interval timer usage	1
Use of logic analyzers to instrument and debug a system	1
Sensor interfacing	1
Develop embedded software and hardware interfaces to sense and control multiple devices, meeting real-time system constraints	4

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	1
Algorithm & Software Design	0	2	Concepts of Programming Languages	0	0

Oral and Written Communications

One assignment is dedicated to an oral presentation. Students must obtain pre-approval from the instructor of a topic of interest in embedded systems. Students make presentations of 10 to 20 minutes supported by visual aids and must answer questions from the instructor and other students. Students are graded on the presentation and their ability to answer questions.

Social and Ethical Issues

This course contains no significant coverage of social and ethical issues.

Theoretical Content

This course contains no significant theoretical content.

Problem Analysis

For the programming projects, students determine requirements for the programs in consultation with the instructor and must perform sufficient analyses of the requirements to arrive at an effective program design.

Solution Design

Programming projects require the student to perform substantial design to arrive an implementation that fulfils the functional requirements and is both robust and well organized. Typically, the final programming project consists of >500 lines of code, some of which may be assembly language.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

OS4. Scheduling and dispatch [core]	3
OS9. Real-time and embedded systems [elective]	16
SE12. Specialized systems development [elective]	4

Course Coordinator:	Dick Lang
Last Updated:	October 10, 2005
Syllabus Version Number:	1.3