

**Master Course Syllabus**  
School of Engineering and Computer Science  
Washington State University Vancouver

**CS 466/566**  
**Embedded Systems**  
3 Semester Hours  
(2 lecture hours, 3 laboratory hours)

**Course Instructor**

David 'Miller' Lowe (Adjunct, Firmware Engineer, Polaris Industries)

Office Hours: Planned, Before/After Class

E-Mail: [miller.lowe@wsu.edu](mailto:miller.lowe@wsu.edu), [milhead@gmail.com](mailto:milhead@gmail.com)

**Class Hours**

Tuesday (lecture) 5:45pm to 7:40pm VUB 1

Thursday (lab) 5:45pm-8:15pm VECS 220

**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.

### **Required Textbooks**

None,

Old Textbook, David Simon, An Embedded Software Primer, Addison-Wesley, 1999, ISBN 0-201-61569-X, I will provide excerpts if required.

### **Reference Material**

Embedded Systems Journal (various articles specified)  
 ARM Info Center ( <http://infocenter.arm.com> )  
 Multiple Web references given during lectures  
 Google Search

### **Grading**

Homework/Labs	30%
Midterm	30%
Final	40%

### **Major Topics Covered in the Course**

1. Review of microprocessor architecture and interfacing
2. Hardware architectures of embedded systems
3. Software architecture of embedded systems
4. Real-time operating systems (RTOS)
5. Communication 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. Typical projects are shown below.

Lab	Description	Weeks
1	Get your development environment and compile the simple blinky.c firmware <ul style="list-style-type: none"> <li>• Get the hardware running..</li> <li>• LED Output / Button Inputs</li> <li>• Bare-metal 'blinky.c' single-threaded implementation.</li> <li>• TM4C123GH6PM Microcontroller DATA SHEET</li> <li>• Empirical clock measurement using lab instruments</li> </ul>	1
2	Blinky Firmware revisit under RTOS control <ul style="list-style-type: none"> <li>• Setup FreeRTOS scheduler and heartbeat task.</li> <li>• Generic task structure and blinky.c refactor</li> <li>• Using USB-Serial for output.</li> <li>• OS clock measurement using lab instruments</li> </ul>	1
3	RTOS Producer/Consumer using threads and Queues. <ul style="list-style-type: none"> <li>• Use tasks, Queues, assert()</li> <li>• Connect GDB and debug over ICDI using OCD</li> <li>• Use GDB debugger to analyze Assert</li> </ul>	1
4	Simple Remote Device Communication <ul style="list-style-type: none"> <li>• Pin Muxing</li> <li>• SPI master communications (bitBang and ASIC peripheral)</li> <li>• External interrupt handling.</li> <li>• Use Timers to measure round trip times.</li> <li>• Watchdog hang protection.</li> </ul>	2
5	Inter-Processor Communication via SPI <ul style="list-style-type: none"> <li>• SPI Slave Device interface implementation.</li> <li>• DC Motor Operation Basics</li> </ul>	2
6	DC Motor Control <ul style="list-style-type: none"> <li>• Manual Quadrature position sensing</li> <li>• Position servo PID implementation</li> <li>• Tuning a Wikipedia PID control loop</li> <li>• Velocity servo PID implementation</li> </ul>	2
7	Final Project, Connected, Multi-Processor, Lab. <ul style="list-style-type: none"> <li>• Application will be determined later in the semester</li> </ul>	5

### **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**

A single assignment is dedicated to an oral presentation. Students must obtain pre-approval from the instructor of a topic of interest in embedded systems. Student presents a PowerPoint based slide presentation of 10 to 20 minutes 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

### **CS566**

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

ENCS-GO1 – Understanding of code generation of mundane and interrupt specific code sections.	5
ENCS-GO3 – Report and Presentation of final lab	2