1. Catalog Description

Programming of embedded computer systems, that is, computing devices that are parts of a larger installation, from watches and cell phones, to medical devices, cars, and space vehicles. Software design principles, specific implementation constructs, and operating system kernels will be taught, supported by the use of software development tools.

2. Course Objectives and Learning Outcomes

Students will learn fundamental concepts of designing and programming embedded computer systems, including requirements specifications, architectural and detailed design, and implementation, focusing on real-time aspects of programming languages, operating system kernels, and hardware architectures. Object-oriented software engineering approach, supported by software tools in the design and implementation process will be emphasized. Learning the concepts will be enforced by a Project to design and develop a software application of practical importance. Specifically, after completing this course, the student will acquire:

- understanding of the specifics of embedded computer systems
- the ability to design for and program selected hardware architectures
- the ability to understand design trade-offs and take design decisions
- the ability to present engineering designs in writing and verbally.

3. Prerequisites

COP 2006 Introduction to Programming or ISM 3230 Introduction to Business Programming or Instructor’s consent.

4. Reading Materials

Currently, there is no required textbook for this class. Recommended books are listed at the end of this syllabus. All class readings are based on materials available from electronic sources or provided by Instructor on a weekly basis. By special arrangement with some professional magazines:

- Circuit Cellar, http://www.circuitcellar.com/

students will have access to selected monthly issues throughout the year.
5. Class Meetings

This is a web based course, so face-to-face lectures are replaced by web modules and Internet readings, however, project meetings are essential for reporting progress and can be scheduled individually. The Orientation Meeting is on Friday, August 28, 11:00-12:15, Holmes 303.

6. Course Outline

Week 2: High-Level Design Principles & Tools
Week 3: Designing the Software Architecture – Static and Dynamic Descriptions
Week 4: Integrated Development Environments (with Eclipse)
Week 5: Programming Language Features for Embedded Applications: C/C++, Java, J2ME, exception handling, mixed language programming
Week 6: Advanced Programming: Real-Time Benchmark and Time-Triggered Systems
Week 7: Real-Time Kernels and Operating Systems: tasks and threads, scheduling, intertask communication
Week 8: Midterm Project Verification
Week 9: Advanced RTOS Concepts: pre-emption, re-entrancy, priority inversion, rate monotonic scheduling, device drivers
Week 10: External Hardware Devices, Computer Buses, Interrupts, Clocks/Timing
Week 11: Microcontrollers and FPGA’s
Week 12: Safety, Reliability and Fault Tolerance
Week 13: Neglected Topics & Comprehensive Test (offered via ANGEL)
Week 14: Extra week for project completion and discussions
Week 15: Project Finalization and Verification (with demos)

7. Administrative Issues

Project: Software Development Project is an essential part of this class. Assessment will be based on 3 phases: Software Specification, Software Design, Software Implementation with Project Demonstration. Detailed topics and schedule will be announced in due course.

Readings and Assignments: Readings related to current topic will be assigned every week. Programming Assignments and other Homework will be given throughout the semester, focusing on the concepts learned from these readings.

Quizzes and Exams: Three to four Quizzes and a Comprehensive Test will be offered (dates TBD); Final Exam is optional for students whose combined class/project score is 80% or higher.

Grading Policy: Project 40%, Programming/Homework Assignments 30%, Quizzes and Comprehensive 30%. A: 90-100%; B: 80-89.9%; C: 70-79.9%; D: 60-69.9%; F: < 60%; (plus/minus grades at the discretion of Instructor). If the Final is taken, its score is averaged with the combined class score to calculate the course grade: CourseGrade = (ClassScore + Final)/2.

Attendance: Since this is a web-based course, no attendance of lectures is required. However, the student is responsible for all assignments on a weekly basis, including meeting the Instructor, if requested, to report progress on the projects. No make-up will be given for missed quizzes, tests or assignments, unless a case is made in advance with Instructor’s approval.

Ethic, Disabilities Act, and Observance of Religious Holidays.
Instructor follows general university policies as spelled out, respectively, in:
• Academic Behavior Standards & Academic Dishonesty Policy in the Student Guidebook (sections on “Student Code of Conduct” and “Policies and Procedures”).
  See: http://studentservices.fgcu.edu/JudicialAffairs/
• Americans with Disabilities Act of 1990 – services provided by Office of Adaptive Services
  See: http://studentservices.fgcu.edu/adaptive/
• Policy 4.005 Student Observance of Religious Holidays
  See: http://www.fgcu.edu/generalcounsel/policies-view.asp

Disclaimer: This syllabus has been prepared to the best of Instructor’s knowledge, but the right is reserved to modify or adjust it slightly depending on circumstances beyond Instructor’s control.

Recommended Books on Embedded Systems

S. Ball, Analog Interfacing to Embedded Microprocessor Systems, Newness, 2003
M. Barr, Programming Embedded Systems in C and C++, O’Reilly, 2002
R. Bishop, Learning with Labview 7 Express, Prentive-Hall, 2004
D. Brickner, Test Driving Linux, O’Reilly, 2005
R. Burdick, Essential Windows CE Application Programming, John Wiley & Sons, 1999
D. Clingman et al., Practical Java Game Programming, Charles River Media, 2004
B.P. Douglass, Real-Time Design Patterns, Addison-Wesley, 2003
B.P. Douglass, Real-Time UML, Addison-Wesley, 2004
D.V. Gadre, Programming the Parallel Port, R&D Books, 1998
R. Grehan et al., Real-Time Programming, Addison-Wesley, 1998
C. Hamer, J2ME Games with MIDP2, Apress, 2004
C. Hollabaugh, Embedded Linux: H/W, S/W, and Interfacing, Addison-Wesley, 2005
S. Holzner, Eclipse, O’Reilly, 2004
E.L. Lamie, Real-Time Embedded Multithreading: ThreadX & ARM, CMP Books, 2005
T. Noergaard, Embedded Systems Architecture, Newess Press, 2005
J. Orwant, Designing Embedded Hardware, O’Reilly, 2002
B. Selic et al., Real-Time Object-Oriented Modeling, JohnWiley and Sons, 1994
D. Simon, Embedded Software Primer, Addison-Wesley, 1999
K. Topley, J2ME in a Nutshell, O’Reilly, 2002
K. Yaghmour, Building Embedded Linux Systems, O’Reilly, 2003
J.Y. Wilson, Building Powerful Platforms with Windows CE, Addison-Wesley, 2001