CS 5372 - Specifications and Design of Real-Time Systems
Course Information Sheet

Instructor: Dr. Salamah Salamah

office: CCS 3.0608 **phone:** 747-6671
email: isalamah@utep.edu

Office Hours: Tuesday & Wednesday 3:00-5:00 pm, or by appointment

Course Description:
The concepts of real-time software systems from the user, designer, and programmer viewpoint are introduced. The connection of an external process to a digital computer by means of hardware and software interface is discussed. The structure, programming, and basic properties of real-time systems are described with an overview of system software. Related topics as interrupts, concurrent task synchronization and communication, sharing resources, schedulability and reliability are discussed. Real-time software development in a host-target environment is presented. Required is knowledge of computer operations including input/output, interrupt interface, and familiarity with operating systems concepts, and software engineering lifecycle. Required is proficiency in a high level language programming **(a working knowledge of C**) including passing data between program modules, operating with input/output, and using memory resources. The hands- on laboratory experiments working outside the class time with the host-target Wind River System environment in your own machine are integral part of the course.

NOTE: *A working knowledge of C language is the course pre-requisite. The handouts will be posted on SmartCloud. You may also try one of popular on-line tutorials.*

Goals:
The purpose of the course is to have students: understand the concepts of real-time process and control, and of a computer as a real-time machine, represent and implement a real-time software system using established software engineering methodologies and development platforms, understand the concepts of inter-process interface and multitasking, understand issues of time-critical computing, be familiar with a real-time system and application software, appreciate the role of real-time systems in target-based aviation/aerospace applications.

Learning Outcomes:
Upon completion of this course, students should be able to:

- Understand concepts of time-critical computing and identify real-time systems.
- Operate host-target development environment for time-critical systems.
- Write multitasking computer programs with inter-task communication and synchronization.
- Apply concepts of inter-task communication and synchronization via shared memory, message queues, signals, semaphores.
- Understand real-time kernels and task scheduling.
- Understand concepts and operations of interfacing hardware.
- Understand concepts of reliability and safety in relation to real-time software..

Required Text:
Real-Time Concepts for Embedded Systems, by Qing Li with Caroline Yao, CMP books, 2003,
ISBN 1-57820-124-1

Supplemental Materials:

- Real-Time Systems Design and Analysis, P.A. Laplante, IEEE Press, 2004
- Embedded Systems Building Blocks, J.J. Labrosse; R&D Publications/Prentice Hall, 1995
- An Embedded Software Primer, D.E. Simon, Addison Wesley, 1999
- Embedded C, M.J. Pont, Addison Wesley, 2002

- Software engineering for real-time systems, J.E. Cooling; Addison-Wesley, ISBN 2003;
- The Art of Designing Embedded Systems, J. Ganssle; Butterworth, 2002
- An access to the Internet to download the course materials. An access to a library with current articles Transactions of ACM, IEEE Computer, Real Times, etc.
- An access to a computer laboratory (or your own machine) with a fast, user friendly, host-target development environment. For a while forget Java ☺ - the language for the test programs and the project implementation is C.

Assignments:
Assignments consist of lab reports, assigned reading, exercises, special reports, and working as part of a team developing a software product.

Grading Procedures:
1. Each student will work on a team project to gain hands-on experience on developing a small size real-time embedded software component. Teams will be pre-assigned. The project theme will be a Digital Home. However, each team will have the flexibility to choose the aspects of the projects that team wants to work on. Proposals for the project will be due on September 14, 2015 along with budget and justification.

2. Individual lab exercises will be assigned regularly throughout the semester. Although collaborative work on these assignments is allowed and even encouraged, the delivered lab solutions and reports must be the student's own work***. You should expect that I will ask you to stop by my office and explain your work to me***.

3. Two full period exams will be given during the term.

4. Individual and team homework (other than labs) will be assigned as necessary.

5. Participation in class discussion and attending classes is a must in this course

6. Points will be distributed as follows:

Team Project	25%
Lab Reports, Exercises, Homework, etc.	25%
Exam 1	20%
Exam 2	20%
Participation	10%

7. Anyone found cheating on an exam or any assignment will receive an automatic F in the course.

8. There are no make-ups. Assignments and exams will be announced with plenty of time to make arrangements for most everything. Absence from an exam is excused only in a medical emergency.

CS 5372 Specification and Design of Real-Time Systems

Weekly Class Schedule

May be modified

TOPIC	**READING**	**ASSIGNMENT**
Introduction, and platforms		Lab #0 Familiarization (no report)
Holiday (but you are not getting off that easy)	(Ch 1/2/3) Intro, platforms	
RTOS and Workbench/VxWorks	(Ch 4/9) Workbench + setup)	
Concurrency and Multitasking	(Ch 5) concurrency, tasking	Lab #1 Hardware and Execution Control
Clocks/Timers	(Ch 11.1-11.4) timing	Lab # 2 Task Control
Synchronization and Semaphores	(Ch 6, 15.2) semaphores	Lab #3 Watchdog and Timing
Communication and Message Queues	(Ch 7, 15.3) Message queues	
Kernel Objects	(Ch 8) objects sema_a1/a2.mov	Lab #4 Synchronization and Semaphores
I/O and Memory Interface	(Ch 12, 13) i/o memory	**Test #1**
Exceptions/Interrupts/Signals	(Ch 10) exceptions	Lab #5 Communication and Message Queues
RT Design	design (Ch 14,15)	Lab # 6 I/O Interface
RT Programming	programming (Ch 14,15)	
Scheduling	(Ch 14) scheduling	Lab#7 Signals and Exceptions
Scheduling (cont)	(Ch 14, 16) scheduling	
		Test #2
Finals Week: Project Presentations and Demonstrations		