EECS 123 INTRODUCTION TO REAL-TIME DISTRIBUTED PROGRAMMING
(Elective for CpE)

Catalog Data: **EECS 123 Introduction to Real-Time Distributed Programming (Credit Units: 4)** Introduction to the techniques for programming applications involving timing-sensitive actions. Hands-on experiences with object-oriented programming styles. Timing requirements, timing specification, response times, deadlines, application programming interfaces to real-time operating systems and middleware, remote procedure call, and distributed objects. Prerequisites: EECS111, EECS112. Formerly ECE147. (Design units: 2)

Textbook: Tutorial notes on High-Level Real-Time Distributed Programming prepared by Kane Kim.
A selection of research articles and book chapters.

References: *The TMO Programming Toolkit* including a user manual and other research articles available from Web.

Coordinator: K. H. (Kane) Kim

Course Outcomes: Students will:
Measure the amount of time taken for execution of program-segments.
Specify timing requirements to be imposed on various parts of real-time computing programs.
Describe application programming interfaces to real time related parts of operating systems.
Design and program real-time distributed computing application programs using C++.
Students will be able to describe the principles of real-time distributed object based design and programming.

Prerequisites By Topic: Understanding of:
Basic computer architecture including input and output subsystems and interrupts; Data structures and object-oriented programming techniques; Introduction to operating systems.

Lecture Topics: Course overview, Background Quiz, Introduction including characteristics of RT applications. (week 1)
Resource scheduling & time measurement; TCP, IP, UDP, RPC, and relevant APIs (week 2)
Global Time and Clock Synchronization Time referencing and calculation facilities in the TMO Approach. (week 3)
Time-triggered actions in single node systems – TMOs containing SpMs only. Config.ini file and a GUI tool for creating Config.ini. TMOs containing SpMs and ODSSs. (week 4)
I/O Handling 1 Midterm Exam. (week 5) –
Real-time Multicast and Memory replication Channel (RMMC) –Part 1
TMOs containing SvMs only including Remote TMO calls. (week 6)
Further on Config.ini file TMOs containing both SpMs and SvMs I/O handling 2 – Use of callback functions issuing one-way service-requests

for SvMs. (week 7)
RMMC – Part 2 (including RMMC-t-SvM and emulation of ORT) Visual Studio for TMO. (week 8)
Virtual Machine for Auxiliary Threads I/O Handling 3 – Use of slave service functions and slave time-triggered functions. (week 9)
DTS (Distributed Time-triggered Simulation) an introduction. (week 10)

Class Schedule: Meets for 3 hours of lecture and 2 hour of lab each week for 10 weeks.

Computer Usage: PC or workstation running widely used OS such as Windows for exercise in real-time distributed programming.

Laboratory Projects: Five or six project assignments are given in this course.

Professional Component: Contributes toward the Computer Engineering Topics Courses and Major Design experience.

Relationship to Program Outcomes: This course relates to Program Outcomes a, b, c, d, e, k, and m as stated at:
http://undergraduate.eng.uci.edu/degreeprograms/computer/mission

Design Content Description

Approach: Real-time programming activities are essentially design activities. Lectures emphasize relevant design principles and applicable measures and constraints. Lab projects provide hands-on design experiences for the students.
Lectures: 50%
Laboratory Portion: 50%

Grading Criteria:

Homework:	10%
Lab assignments:	20%
Midterm exam:	20%
Final exam:	50%
	100%

Estimated ABET Category Content:

Mathematics and Basic Science: 0 credit units or 0%
Engineering Science: 2 credit units or 50%
Engineering Design: 2 credit units or 50%

Prepared by: K. H. (Kane) Kim **Date:** July 2007
CEP Approved: Fall 2004