Embedded Systems & RTOS (404204)

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hrs/ Week	In Semester Assessment: Phase I : 30 End Semester Examination: Phase II: 70

Course Objectives:

- To understand the different design metrics of embedded system
- To learn real time operating system concepts.
- To understand the Embedded Linux environment
- To apply concept RTOS for different embedded system application

Course Outcomes:

After successfully completing the course students will be able to

- Consider the different constraints of embedded system
- Understand Real time systems concepts.
- Do the analysis Linux operating system as real time operating system.
- To use RTOS for different embedded systems

Unit I :Introduction to Embedded Systems 6L

Introduction to Embedded Systems, Architecture, Classification and Characteristics of Embedded System, Design Process, Design Metrics and optimization of various parameters of embedded system. Embedded processor technology, IC technology, Design technology. Software development life cycle. Various models like waterfall, spiral, V , Rapid Prototyping models and Comparison, Embedded system such as vending machine, temperature Controller, automatic cruise control system, antilog braking system and traction control in vehicles.

Unit II : **RTOS Concepts** 6L

Foreground and background systems, Critical Session, Shared resources, Tasks, Multitasking, Context switching, Kernels, Pre-emptive and non-preemptive Schedulers, Static and Dynamic Priorities, Priority inversion, Mutual exclusion, Synchronization, Inter task communication mechanisms, Interrupts: Latency, Response and Recovery, Clock Tick, Memory requirements.

Unit III: **Structure of uCOS – II** 6L

Kernel Structure: Tasks, Task States, TCB, Ready list, Task Scheduling, Task Level Context Switching, Locking and unlocking of scheduler, Idle Task, Statistics Task, Interrupts, Clock Tick, Initialization, Starting the OS. Task Management: Creating/Deleting and Suspending/Resuming Task, Task Stacks and checking, Changing Task’s Priority. Time Management: Delaying/Resuming task, System Time.

Unit IV: **Synchronization in µCOS- II** 6L

Semaphore Management: Creation/Delation, Pending/Posting/Acceptance/Query. Mutual Exclusion Semaphores: Creation/Delation, ending/Posting/Acceptance/Query. Event Flag Management: Internals, Creation/ Delation of Event Flag groups, Waiting/Setting/Clearing/Looking for/Querying an Event Flag Group.

Unit V : **Communication in µCOS- II** 6L

Message Mailbox Management: Creating/Deleting a Mailbox, Waiting/ Sending /Getting without waiting a Message from Mailbox, Status of Mailbox, Alternate uses of Mailbox. Message Queue Management: Creating/Deleting/ Flushing a Message Queue, Waiting/Sending/Getting without waiting a Message from Queue, Status and Alternate use of Message Queue. Memory Management: MCB, Crating a partition, Obtaining /Returning/Waiting for a memory Block, Partition Status. Porting of µCOS- II: Development tools, Directories and Files, Configuration and testing of Port.

Unit VI : **Linux Kernel Construction** 6L

Need of Linux, Embedded Linux Today, Open Source and the GPL, BIOS Versus Boot loader Linux Kernel Background, Linux Kernel Construction, Kernel Build System, Kernel Configuration. Role of a Bootloader, Bootloader Challenges. A Universal Bootloader: Das U-Boot. Porting U-Boot. Device Driver Concepts, Module Utilities, Driver Methods. Linux File System & Concepts.

Text Books

1. Jean J. *Labrosse, "*MicroC OS II, The Real-Time Kernel", 2nd edition, CMP Books.
2. Christopher Hallinan, "Embedded Linux Primer -A Practical, Real-World Approach "2nd edition, Prentice Hall.
3. Raj Kamal, "Embedded Systems – Architecture, Programming and Design" 2nd edition, Mc Graw Hill

Reference Books

1. Dr. K.V.K.K. Prasad **"**Embedded / Real Time Systems Programming Black Book" Dreamtech Press
2. Frank Vahid and Tony Givargis, "Embedded System Design – A Unified hardware/ Software introduction" 3rd edition, Wiley.

List of Experiments

Perform any 5 from 1 to 7 and any 2 from 8 to 10.

1. RTOS porting on available micro controller board.
2. Interfacing of 4X4 Keyboard to a micro controller using µCOS- II task
3. Interfacing of 4X4 Keyboard, 16X2 LCD display and ADC to a micro controller using µCOS- II task
4. Implement a semaphore for any given task switching on a micro controller
5. Implementation of mutual exclusion in tasks as per 3.
6. Implementation of mailbox and message queue management in tasks as per 3.
7. Implementation of memory management in tasks as per 3.
8. Interfacing of LEDS and Keyboard using Linux OS
9. Interfacing of graphic LCD using Linux OS.
10. Interfacing ADC and DAC using Linux OS.