

PLAN DE ESTUDIOS: 2004 Ajuste 2011
AÑO ACADEMICO: 2013
CARRERA: Ingeniería Electrónica

1. OBJETIVOS:

Luego de completado el curso, los alumnos tendrán los conocimientos y aptitudes para:

- Señalar las características de un sistema de tiempo real, y qué lo diferencia de otros sistemas operativos de propósitos generales.
- Caracterizar un lenguaje de programación y un sistema operativo de acuerdo a sus características de tiempo real.
- Desarrollar un proyecto con requerimientos de *tiempo real*.
- Aplicar distintas soluciones de tiempo real a casos específicos en el campo de la Electrónica.

2. CONTENIDO:

UNIDAD 1 : Introducción y definiciones

1. Presentación del curso.
2. Repaso de conceptos básicos de arquitectura de computadoras.
3. Sincronismo entre el mundo exterior y el programa. Sincronismo de entrada y de salida. Métodos de sincronismo: *polling*, interrupciones. Ejemplos típicos con distintos tipos de periféricos.
4. Interrupciones: división entre *foreground* y *background*.
5. Interrupciones: Concepto de *thread* y de tarea o proceso.
6. Sistemas de Tiempo Real: definiciones, tipos y clasificaciones. Ejemplos.

UNIDAD 2: Consideraciones de hardware

1. Arquitectura básica
2. Interfases hacia el hardware
3. Unidad central de proceso
4. Memoria
5. Entrada / Salida
6. Otros dispositivos especiales: ASIC's, FPGA's, transductores, conversores A/D y D/A

UNIDAD 3: Diseño, Desarrollo y Depuración.

1. Técnicas de diseño y desarrollo de circuitos y programas. Análisis de herramientas CAD y Entornos de desarrollo y depuración.
2. Fragmentación en módulos de fácil depuración. Elaboración de macros y bibliotecas con criterio de reuso de los mismos.
3. Sistemas secuenciales. Repaso de diagramas de estados. Especificación y limitación de métodos formales: máquinas de estado finito, Empleo de tablas. Parsing. Gráficos de estado (*statecharts*), Redes de Petri, UML.

4. Técnicas de puesta a punto y depuración. Simulación y emulación. Programas monitores, simuladores por software y emuladores de hardware. Kits de evaluación y desarrollo. JTAG y serial wire protocol. Principios de operación y uso. Breakpoints sobre instrucciones y data watchpoints. Trace.
5. Técnicas de cálculo de la duración de un proyecto. Costeo.

UNIDAD 4: Microcontroladores de 32 bits

1. Arquitectura ARM. Estructura interna. Arquitectura Risc. Registros.
2. Instrucciones condicionales. Modos de direccionamiento, instrucciones
3. Modos de trabajo.
4. Manejo de excepciones. Uso de registros adicionales.
5. Familia Cortex. Arquitectura y Thumb2
6. Herramientas de desarrollo y depuración. Programación y ejemplos.

UNIDAD 5: Lenguajes de Programación para STR

1. Características y requisitos de los lenguajes de programación para STR.
2. Lenguaje ensamblador.
3. Lenguajes procedurales: características,
4. Lenguajes orientados a objeto.
5. Revisión de los lenguajes más usados y sus características.

UNIDAD 6: Sistemas Operativos de Tiempo Real

1. Núcleo de Tiempo Real: Pseudo-kernels, sistemas *foreground/background*, sistemas manejados por interrupciones: por prevaciado o cooperativos.
2. Fundamentos teóricos de Sistemas Operativos de Tiempo Real: Planificación de tareas; tipos de planificadores.
3. Comunicación entre tareas y sincronización.
4. Gerenciación de memoria.
5. Casos de estudio:
 - a. Un sencillo núcleo de tiempo real.
 - b. POSIX.
 - c.

UNIDAD 7: Requerimientos en Software

1. Proceso de requerimientos: tipos de requerimientos.
2. Especificaciones de requerimientos para Sistemas de Tiempo Real.
3. Especificación y limitación de métodos formales: máquinas de estado finito, gráficos de estado (*statecharts*), Redes de Petri.
4. Análisis y diseño estructurado.
5. Propiedades del Software.
6. Principios básicos de Ingeniería de Software.

UNIDAD 8: Aplicaciones de Sistemas en Tiempo Real

1. Análisis y estudio de distintas aplicaciones.
2. Sistemas distribuidos en Tiempo Real.

3. BIBLIOGRAFIA

The Definitive Guide to the ARM Cortex-M3, Second Edition – Joseph Yiu – Newnes – 2009 - ISBN-13: 978-1856179638
Cortex-M3 Technical Reference Manual (http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337i/DDI0337I_cortexm3_r2p1_t1rm.pdf)
The Cortex Microcontroller Software Interface Standard (http://www.onarm.com/cmsis/download/10/version-2-0-of-the-cortex-microcontroller-software-interface-standard-cmsis/)
ARM [®] v7-M Architecture Reference Manual (http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.architecture/index.html)
ARM Generic Interrupt Controller (http://infocenter.arm.com/help/topic/com.arm.doc.ih0048a/IH0048A_gic_architecture_spec_v1_0.pdf)
Practical UML Statecharts in C/C++, Second Edition: Event-Driven Programming for Embedded Systems - Miro Samek – Newnes – Octubre 2008 - ISBN-13: 978-0750687065
ARM Architecture Reference Manual – Seal - Addison Wesley – 2000 – ISBN 0 201 737191
ARM system-on-chip architecture – Second edition -Furber – Addison Wesley 2000 – ISBN 0-201-67519-6
ARM System Developer's Guide: Designing and Optimizing System Software – Sloss. Symes. Wright – Morgan Kaufmann – 2004 - ISBN-13: 978-1558608740
Real-Time Concepts for Embedded Systems - Qing Li Caroline Yao – CMP – Julio 2003 - ISBN-13: 978-1578201242
Real Time Embedded Multithreading Second Edition – Lamie – Newnes – 2009 - ISBN-13: 978-1856176019
Real Time Systems Development – Rob Williams - Butterworth-Heinemann (Diciembre 2005) - ISBN-13: 978-0750664714
Simple Real time Operating Systems - Chowdary Venkateswara Penumuchu- Trafford Publishing (Agosto, 2007) - ISBN-13: 978-1425117825
Real Time System Design and Analysis <i>Third Edition</i> . Phillip A. Laplante. John Wiley & Sons.
Introduction to Real-Time Systems: From Design to Networking with C/C++ , Raymond A. Buhr, Donald L. Bailey;

4. METODOLOGIA DE LA ENSEÑANZA

La metodología de la enseñanza / aprendizaje está basada en los siguientes pilares:

- Clases teóricas interactivas.

- Clases prácticas relacionadas con un proyecto a desarrollar durante el cuatrimestre

Teniendo en cuenta la altura de la carrera, se incentivará la participación activa en un proceso típico de Ingeniería de Proyecto como forma de aprendizaje y aplicación de los conceptos.

5. CRITERIOS DE EVALUACION

Para la aprobación del curso de Trabajos Prácticos, los alumnos abordarán un proyecto, el cual tendrá entregas parciales, de acuerdo se describe en la Guía de Trabajos Prácticos y con entregas de la documentación de cada uno de ellos de acuerdo a lo que se establece en el Cronograma.

Las condiciones de aprobación del curso de Trabajos Prácticos serán las siguientes:

- Haber participado en las sesiones de trabajo práctico que se desarrollan cada semana.
- Haber cumplido con los objetivos de cada trabajo práctico
- Haber entregado en tiempo y forma los informes de cada trabajo práctico

Como el trabajo de desarrollo puede ser (y se recomienda que sea) encarado en grupo, la nota del curso de trabajos prácticos se computará de la siguiente forma para cada alumno:

$$0.6 \times \text{Promedio de la nota de los informes de TP} + 0.4 \text{ de la nota conceptual del docente}$$

Los informes de los trabajos prácticos deben ser entregados en el término de 21 días; de entregarse hasta una semana después, se disminuirá la nota del informe en tres puntos.

Si luego de cuatro semanas, el trabajo no fue aprobado, quedará para recuperar en la fecha a fijar por la Universidad.

Para permanecer en el curso de trabajo práctico y debido que, por tratarse de una tarea de desarrollo, los trabajos prácticos están vinculados entre sí, la falta de presentación de dos informes de trabajos prácticos **dejará fuera del curso a los componentes del grupo.**

En forma personal, cada componente del grupo deberá concurrir a las distintas sesiones que componen un trabajo práctico y participar del mismo; la falta persistente a dichas sesiones o la falta persistente de participación, llevarán a la pérdida del curso de trabajos prácticos.