



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Programa de **SISTEMAS EMBEBIDOS PARA TIEMPO REAL**

1. NOMBRE DE LA UNIDAD CURRICULAR

Sistemas embebidos para tiempo real.

2. CRÉDITOS

10 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

Al finalizar el curso el estudiante será capaz de:

- Definir sistemas embebidos y sistemas embebidos de tiempo real, e identificar sus características.
- Describir el proceso de compilación de una aplicación, la inicialización de datos, el arranque de programa y el cargado de programa.
- Programar utilizando control de versiones.
- Describir el hardware asociado a un sistema embebido. Describir los conceptos y bloques básicos de microcontroladores y periféricos más relevantes.
- Utilizar e interpretar documentación técnica compleja (manuales de usuario, hojas de datos, etc.).
- Describir el flujo de ejecución de un microcontrolador: contexto principal y de interrupciones. Describir la secuencia de ejecución de interrupciones. Reconocer el problema de datos compartidos. Identificar los tiempos involucrados en las interrupciones.
- Describir las principales arquitecturas de software e identificar sus características. Comparar las ventajas y desventajas de cada una. Seleccionar la arquitectura adecuada para una aplicación particular.
- Conocer los conceptos básicos de los sistemas operativos para tiempo real. En particular el concepto de tareas, estados de una tarea, el rol del planificador, y la vinculación entre tareas y datos.
- Analizar y aplicar técnicas de reducción del consumo a nivel circuito y a nivel sistema, y de programación para bajo consumo.
- Definir los principales conceptos de test. Identificar las diferencias entre test en PC y test en sistemas embebidos. Describir los métodos de test embebido: en host y con simuladores.

- Diseñar el hardware y desarrollar el software embebido para implementar un sistema basado en microcontrolador. En particular, balanceando adecuadamente el procesamiento que se realiza dentro de las interrupciones y en la tarea, y separando el código dependiente del independiente de hardware.
- Enfrentarse a un problema real e implementar una solución al mismo utilizando un sistema embebido para tiempo real.

4. METODOLOGÍA DE ENSEÑANZA

El curso comprende clases teóricas, laboratorios y la realización de un proyecto final. Se propone una metodología que promueva la participación activa de los estudiantes y el trabajo en grupo, en todas las actividades del curso. En las clases de teórico se propondrán actividades de trabajo grupales e individuales. Asimismo, el trabajo de laboratorio y el proyecto final se realizará en forma grupal. A continuación se desglosa la dedicación estudiantil esperada al curso:

Tipo	Cantidad de Horas
Clase teórica	22
Clase de laboratorio	16
Clase de consulta (laboratorio y seguimiento proyecto)	10
Evaluación (presentaciones)	2
Subtotal horas presenciales	50
Estudio	4
Preparación de laboratorio	16
Proyecto final	80
Total de horas no presenciales	100
Total de horas dedicadas por estudiante	150

Los laboratorios son de asistencia obligatoria, y tienen una evaluación individual y grupal. Se entregarán a los estudiantes un kit de desarrollo que permite, conectándolo a una computadora, la preparación y realización de la práctica.

El proyecto final también se desarrollará sobre la base de un kit de desarrollo provisto por los docentes. A lo largo del curso los estudiantes deberán realizar al menos dos presentaciones, y se deberá entregar una documentación final.

5. TEMARIO

1. Introducción y conceptos básicos: introducción a los sistemas embebidos y sistemas embebidos de tiempo real, hardware en sistemas embebidos, metodologías de desarrollo de software embebido.
2. Microcontroladores e interrupciones: arquitectura de microprocesadores y microcontroladores, periféricos, fundamentos de las interrupciones, latencia, el problema de datos compartidos.

3. Arquitecturas de software embebido: Round-Robin, Round-Robin con interrupciones, planificación por encolado de funciones, y sistema operativo para tiempo real. Análisis y selección de arquitectura.
4. Introducción a los sistemas operativos para tiempo real: tareas y el estado de una tarea, el planificador, vinculación entre tareas y datos, semáforos y datos compartidos. Cola de mensajes, buzones y pipes, funciones de temporizado, eventos, manejo de memoria, manejo de rutinas de atención a interrupciones.
5. Sistemas embebidos de bajo consumo: técnicas de reducción del consumo a nivel circuito y a nivel sistema. Programación para bajo consumo.
6. Criterios básicos de diseño de un sistema embebido: criterios generales y específicos de sistemas operativos para tiempo real.
7. Otros temas de sistemas embebidos: diseño basado en modelos, test y técnicas para confiabilidad en sistemas críticos, etc.

6. BIBLIOGRAFÍA

6.1 Básica

1. Simon, David E. (1999). An Embedded Software Primer. Addison-Wesley Professional. ISBN-10: 0-201-61569-X
2. Fan, Xiacong (2015). Real-Time Embedded Systems. Design principles and engineering practices. Elsevier. ISBN: 978-0-12-801507-0.

6.2 Complementaria

3. Lewis, Daniel W. (2001). Fundamentals of Embedded Software: Where C and Assembly Meet. Prentice Hall. ISBN: 0130615897.
4. Labrosse, Jean J. (2002). MicroC OS II: The Real Time Kernel. CMP Books. ISBN: 1578201039, 2002.
5. M. Pedram, and J. M. Rabaey, eds (2002). Power aware design methodologies. Springer Science & Business Media. ISBN: 978-0-306-48139-0, 2002.

Tema	Básica	Complementaria
Introducción y conceptos básicos	(1)	(2,3)
Microcontroladores e interrupciones	(1)	(2,3)
Arquitecturas de software embebido	(1)	(2,3)
Introducción a los sistemas operativos para tiempo real	(1)	(2,4)
Sistemas embebidos de bajo consumo	(1,2)	(5)
Criterios básicos de diseño de un sistema embebido	(1)	(2)
Temas avanzados de sistemas embebidos	(1,2)	(4)

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Programación en lenguaje de alto nivel (preferentemente en lenguaje C) con énfasis en tipos de datos avanzados (punteros, arreglos, estructuras y uniones), modularización, y encapsulamiento. Conocimientos básicos de microprocesadores, incluyendo modos de direccionamiento, lenguaje de máquina, lenguaje de ensamblador, set de instrucciones, interrupciones, uso de stack y subrutinas. Electrónica digital, incluyendo el transistor MOS como llave, y circuitos digitales. Niveles lógicos y márgenes de ruido; tiempos de propagación, tiempos de subida, y bajada; consumo estático y dinámico. Estructura básica de compuertas lógicas CMOS, compuertas con etapas de salida de colector/drain abierto y tercer estado. Niveles lógicos TTL y su interfaz con lógica CMOS. Manejo de hojas de datos.

7.2 Conocimientos Previos Recomendados: Conversores analógico-digitales, Interfaces de comunicación (RS232, UART o similares), y redes de datos.

ANEXO A
Para todas las Carreras

A1) INSTITUTO

Instituto de Ingeniería Eléctrica

A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción y conceptos básicos (3 hs de clase teórica).
Semana 2	Introducción y conceptos básicos (1,5 hs de clase teórica), Microcontroladores e interrupciones (1,5 hs de clase teórica).
Semana 3	Microcontroladores e interrupciones (3 hs de clase teórica).
Semana 4	Microcontroladores e interrupciones (1,5 hs de clase teórica). Arquitecturas de software embebido (1,5 hs de clase teórica). Lab1: Introducción y conceptos básicos (4 hs de laboratorio).
Semana 5	Introducción a los sistemas operativos para tiempo real (3 hs teórica) Lab2: Microcontroladores e interrupciones (4 hs de laboratorio).
Semana 6	Introducción a los sistemas operativos para tiempo real (1,5 hs teórica) Sistemas embebidos de bajo consumo (1,5 hs teórica) Lab3: Arquitecturas de software embebido I (4 hs de laboratorio).
Semana 7	Criterios básicos de diseño de un sistema embebido (1,5 hs teórica) Temas avanzados de sistemas embebidos (1,5 hs teórica) Lab4: Arquitecturas de software embebido II (4 hs de laboratorio).
Semana 8	Temas avanzados de sistemas embebidos (1 hs teórica)
Semana 9	Realización proyecto final
Semana 10	Realización proyecto final
Semana 11	Hito intermedio proyecto final (1 hs presentación) Realización proyecto final
Semana 12	Realización proyecto final
Semana 13	Realización proyecto final
Semana 14	Realización proyecto final
Semana 15	Defensa proyecto final (1 hs presentación)

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La asistencia a las clases de laboratorio e instancias presenciales de proyecto es obligatoria.

El curso no tiene examen y para su aprobación se debe aprobar cada una de las partes del curso (laboratorio y proyecto). La calificación final será ponderada de la siguiente manera:

Laboratorio: 40 % considerando cuestionarios individuales, y el desempeño individual y grupal durante la práctica.

Proyecto: 60 % considerando el desempeño individual y grupal durante la ejecución del proyecto, los resultados obtenidos, las presentaciones y la documentación.

Sei

A4) CALIDAD DE LIBRE

No admite la calidad de libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

No hay cupos.

ANEXO B para la carrera Ingeniería Eléctrica
B1) ÁREA DE FORMACIÓN

Sistemas Digitales

B2) UNIDADES CURRICULARES PREVIAS

Curso:

Examen de Programación para Ingeniería Eléctrica.

Examen de Introducción a los Microprocesadores.

CURSO : Electrónica Fundamental.

Examen:

No tiene examen.

APROB. RES. CONSEJO DE FAC. ING.

Fecha 26/11/2019 Exp. 060780-002733-19