



1. Identificación

1.1. De la Asignatura

Curso Académico	2016/2017
Titulación	GRADO EN INGENIERIA INFORMÁTICA y PROG CONJUNTA DE ESTUDIOS OFICIALES GRADO MATEMÁTICAS Y GRADO ING. INFORMÁTICA
Nombre de la Asignatura	SISTEMAS EMPOTRADOS Y DE TIEMPO REAL
Código	3867
Curso	CUARTO y QUINTO(IC)
Carácter	OPTATIVA
N.º Grupos	2
Créditos ECTS	6
Estimación del volumen de trabajo del alumno	150
Organización Temporal/Temporalidad	1º Cuatrimestre y 1º Cuatrimestre(IC)
Idiomas en que se imparte	ESPAÑOL
Tipo de Enseñanza	Presencial

1.2. Del profesorado: Equipo Docente

Coordinación de la asignatura ANTONIO FLORES GIL Grupo: 1 y 2	Área/Departamento	INGENIERÍA Y TECNOLOGÍA DE COMPUTADORES
	Categoría	PROFESORES TITULARES DE UNIVERSIDAD
	Correo	aflores@um.es
	Electrónico / Página web / Tutoría electrónica	http://ditec.um.es/personal/19 Tutoría Electrónica: SÍ



	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar
		Anual	Viernes	10:00- 13:00	868884638, Facultad de Informática B1.3.041

2. Presentación

La asignatura Sistemas Empotrados y de Tiempo Real contempla los conceptos específicos que se deben tener en cuenta a la hora de programar e implementar este tipo de sistemas. La definición temporal de tareas y actividades está focalizada a conseguir la consecución de los objetivos de la asignatura, teniendo en cuenta que se condensan muchos principios y paradigmas relativos al funcionamiento de los sistemas operativos: ejecución de tareas, memoria compartida, planificación de ejecución, etc. Es importante asimilar los conceptos teóricos para poder realizar correctamente la práctica final de la asignatura.

3. Condiciones de acceso a la asignatura

3.1 Incompatibilidades

3.2 Recomendaciones

Conocimientos del lenguaje de programación en Java sobre: programación concurrente, interfaz gráfica, sistemas distribuidos.

4. Competencias

4.1 Competencias Básicas

- CB4. Que los estudiantes puedan transmitir información, ideas problemas y soluciones a un público tanto especializado como no especializado.

4.2 Competencias de la titulación

- CEI13: Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
- CEI14: Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.



- CEII6: Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- CEII8: Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- CEII9: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
- CEII10: Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.
- IC2: Capacidad de desarrollar procesadores específicos y sistemas empotrados, así como desarrollar y optimizar el software de dichos sistemas.
- IC3: Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software de para las mismas.
- IC5: Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empotradas y de tiempo real.
- IC7: Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos.

4.3 Competencias transversales y de materia

- Competencia 1. Capacidad para enumerar el conjunto de tecnologías disponibles para llevar a cabo sistemas empotrados y de tiempo real
- Competencia 2. Capacidad para enunciar las características fundamentales que satisfacen los sistemas empotrados y de tiempo real
- Competencia 3. Ser capaz de diseñar un sistema fiable y distribuido que responda a las necesidades y características
- Competencia 4. Capacidad para identificar y afrontar los problemas de satisfacción de restricciones temporales
- Competencia 5. Capacidad de organización y planificación: desarrollo de un proyecto de complejidad media que aplique los conocimientos adquiridos en el curso

5. Contenidos

Bloque 1: Sistemas de Tiempo Real

TEMA 1. Introducción a los sistemas de tiempo real

TEMA 2. Diseño de sistemas de tiempo real

TEMA 3. Fiabilidad y tolerancia a fallos

TEMA 4. Excepciones y manejo de excepciones

TEMA 5. Temporización de las tareas en un sistema en tiempo real

Bloque 2: Sistemas empotrados

TEMA 1. Introducción a los sistemas empotrados

TEMA 2. Codiseño Hardware/Software

TEMA 3. Depuración de Sistemas empotrados



TEMA 4. Diseño de sistemas empotrados basados en arduino/beaglebone

PRÁCTICAS

Práctica 1. Instalación de entorno de trabajo en tiempo real: *Relacionada con los contenidos Bloque 1, Tema 1 (Bloque 1) y Tema 2 (Bloque 1)*

Para desarrollar cualquier sistema de tiempo real, lo primero que se necesita es un sistema operativo de tiempo real, que de soporte a las características de este tipo de sistemas. En este caso se verá como instalar/configurar un sistema RTOS basado en RT Linux. Por tanto, el primer paso se corresponderá con la instalación, configuración y comprobación de las características de dicho entorno.

Una vez que se dispone de un sistema RTOS, es necesario seleccionar un lenguaje de programación para implementar el código del sistema en tiempo real. En este caso se va a emplear Java, con las extensiones de tiempo real (RTJ). Para ello, es necesario usar una máquina virtual Java que de soporte a estas características y en este caso, se va a emplear la JVM (Java Virtual Machine) de IBM, denominada IBM Real Time WebSphere (<http://www-01.ibm.com/software/webservers/realtime/>). En consecuencia, el siguiente paso es disponer de la máquina virtual Java, operativa sobre el RTOS.

Práctica 2. Proyecto de programación en tiempo real: *Relacionada con los contenidos Tema 5 (Bloque 1) y Tema 4 (Bloque 1)*

Una vez que se dispone del entorno de tiempo real (RTOS y lenguaje de programación), ya es posible desarrollar una aplicación que funcione con las especificaciones temporales correspondientes. Para ello se propone la realización de un miniproyecto de programación en java que aplique todos los contenidos vistos en la teoría de la asignatura.

Práctica 3. Diseño de una aplicación empotrada en arduino/beaglebone: *Relacionada con los contenidos Bloque 2*

Finalmente, se pretende que el alumno se familiarice con un entorno de desarrollo para sistemas empotrados como es el basado en la plataforma de prototipado Arduino (o de forma alternativa basado en una beaglebone black), para ello se propone el diseño de una aplicación de control utilizando dicha plataforma.



6. Metodología Docente

Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
Lección magistral y Aprendizaje basado en proyectos	Desde el principio del curso se presentará el proyecto a realizar durante la asignatura. De manera que cuando en las clases presenciales se vean los conocimientos, y pautas necesarias para que el alumno pueda profundizar en la materia, sea capaz de ir relacionándolos con el proyecto a realizar. De forma que el aprendizaje de la asignatura se realice de una manera práctica aplicando los nuevos conocimientos que el alumno va adquiriendo a un problema real.	60	90	150
	Total	60	90	150

7. Horario de la asignatura

<http://www.um.es/web/informatica/contenido/estudios/grados/informatica/2016-17#horarios>

<http://www.um.es/web/matematicas/contenido/estudios/grados/pes/2016-17#horarios>



8. Sistema de Evaluación

Métodos / Instrumentos	La evaluación de los conocimientos teóricos de la asignatura se realizará mediante un examen final tipo test.
Criterios de Valoración	La nota obtenida en esta parte de la asignatura se sumará a la obtenida en prácticas para la obtención de la calificación final. En el caso de que el alumno no se presente a este examen o no haya presentado las prácticas, se considerará No Presentado. En todos los demás casos obtendrá la calificación resultado de realizar la media ponderada de ambas calificaciones. Para las convocatorias extraordinarias se guardará la calificación obtenida en la convocatoria ordinaria.
Ponderación	50%
Métodos / Instrumentos	La parte práctica de la asignatura se evaluará mediante la evaluación de las prácticas realizadas por el alumno de forma individual. Se debe entregar un sistema ejecutable completo, una memoria en formato PDF donde se expongan la toma de decisiones y de diseño más relevantes que se han realizado en la práctica, así como posibles mejoras.
Criterios de Valoración	La aplicación debe satisfacer los requisitos mínimos que indique la práctica para superarla.
Ponderación	50%

Fechas de exámenes

<http://www.um.es/web/informatica/contenido/estudios/grados/informatica/2016-17#examenes>
<http://www.um.es/web/matematicas/contenido/estudios/grados/pes/2016-17#examenes>

9. Bibliografía

Bibliografía Complementaria



Introduction to Embedded Systems: A Cyber-Physical Systems Approach. Edward Ashford Lee & Sanjit Arunkumar Seshia.



Java Real Time



Sistemas de Tiempo Real y Lenguajes de Programación 3ª Edición. Alan Burns, Andy Wellings. Ed. Addison-Wesley, 3ª edición, 2003



Real-Time Systems and Programming Languages (Fourth Edition) Ada 2005, Real-Time Java and C/Real-Time POSIX Alan Burns and Andy Wellings Addison Wesley Longmain ISBN: 978-0-321-41745-9



Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux. Derek Molloy. Ed. Wiley. 2014.

10. Observaciones y recomendaciones