

ANX-PR/CL/001-01
GUÍA DE APRENDIZAJE

ASIGNATURA

Sistemas empotrados

CURSO ACADÉMICO - SEMESTRE

2016-17 - Primer semestre

Datos Descriptivos

Nombre de la Asignatura	Sistemas empotrados
Titulación	09AN - Master Universitario en Ingeniería de Sistemas Electronicos
Centro responsable de la titulación	Escuela Tecnica Superior de Ingenieros de Telecomunicacion
Semestre/s de impartición	Primer semestre
Módulos	Optativas sistemas electronicos
Materias	Optativas sistemas electronicos
Carácter	Optativa
Código UPM	93000708
Nombre en inglés	Embedded systems

Datos Generales

Créditos	4	Curso	1
Curso Académico	2016-17	Período de impartición	Septiembre-Enero
Idioma de impartición	Castellano	Otros idiomas de impartición	

Requisitos Previos Obligatorios

Asignaturas Previas Requeridas

El plan de estudios Master Universitario en Ingeniería de Sistemas Electronicos no tiene definidas asignaturas previas superadas para esta asignatura.

Otros Requisitos

El plan de estudios Master Universitario en Ingeniería de Sistemas Electronicos no tiene definidos otros requisitos para esta asignatura.

Conocimientos Previos

Asignaturas Previas Recomendadas

El coordinador de la asignatura no ha definido asignaturas previas recomendadas.

Otros Conocimientos Previos Recomendados

El coordinador de la asignatura no ha definido otros conocimientos previos recomendados.

Competencias

CE1 - Capacidad para evaluar las ventajas e inconvenientes de las alternativas tecnológicas en el diseño o fabricación de sistemas electrónicos analógicos, digitales, centrales o distribuidos

CE2 - Capacidad para aplicar herramientas, técnicas y metodologías avanzadas de diseño de sistemas o subsistemas electrónicos

CE4 - Capacidad para diseñar un dispositivo, sistema, aplicación o servicio que cumpla unas especificaciones dadas, empleando un enfoque sistémico y multidisciplinar e integrando los módulos y herramientas avanzadas disponibles en el campo de la Ingeniería Electrónica

CE5.1 - Capacidad para seleccionar, especificar, proyectar, documentar o poner a punto sistemas electrónicos para proporcionar o explotar servicios o infraestructuras en áreas de aplicación de interés

CE5.3 - Capacidad para diseñar un sistema electrónico digital de complejidad media-alta empleando una plataforma basada en dispositivos programables, empleando un lenguaje de programación o aplicando metodologías, estrategias y herramientas CAD para la integración o el codiseño de componentes hardware y software

CG1 - Uso de la lengua inglesa: comprender los contenidos de clases magistrales, conferencias y seminarios en lengua inglesa; redactar en inglés informes y artículos científicos o técnicos usando herramientas informáticas; realizar exposiciones públicas en inglés de trabajos, resultados y conclusiones, por ejemplo, en las asignaturas del máster, todo ello con la ayuda de medios informáticos audiovisuales

CO6 - Diseñar, construir y validar prototipos funcionales de sistemas electrónicos empotrados de control, de comunicaciones... pasando por todas las fases del proceso dentro de un esquema de trabajo en equipo

Resultados de Aprendizaje

RA145 - Conocimiento de los modelos de computación y las herramientas de desarrollo de software empotrado.

RA143 - Capacidad para optimizar el uso de la memoria para un diseño concreto, teniendo en cuenta el comportamiento de la aplicación, las diferentes alternativas de jerarquía de memoria, técnicas de compilación y modificaciones de código.

RA146 - Capacidad para diseñar sistemas concurrentes asegurando el cumplimiento de los requisitos temporales y teniendo en cuenta aspectos como el uso de memoria, la flexibilidad, el rendimiento y el consumo.

RA144 - Capacidad para analizar los factores que incluyen en el consumo y el rendimiento de sistemas basados en microprocesador y para seleccionar las alternativas más apropiadas en cada escenario.

RA140 - Conocimiento de las principales herramientas de desarrollo de software para sistemas empotrados y capacidad para utilizar ese conocimiento para detectar y corregir errores en el diseño o la implementación, y para analizar el rendimiento, la utilización de recursos y el consumo.

RA139 - Aprender técnicas para la depuración de sistemas hardware mediante simulación

RA141 - Conocimiento de las características de un sistema empotrado o un sistema ciber-físico y su influencia en la evaluación de alternativas de diseño

RA142 - Capacidad para seleccionar la arquitectura HW y SW más adecuada para sistemas empotrados con restricciones.

Profesorado

Profesorado

Nombre	Despacho	e-mail	Tutorías
Moya Fernandez, Jose Manuel (Coordinador/a)	B-104.1b	jm.moya@upm.es	M - 11:00 - 12:00 M - 14:00 - 16:00 J - 11:00 - 12:00 J - 14:00 - 16:00 Hora preferente: J 11:00 a 12:00. Para cualquier otra hora, concertar cita por e-mail.
Araujo Pinto, Alvaro	B-104.1b	alvaro.araujo@upm.es	L - 16:00 - 18:00 M - 16:00 - 18:00 X - 16:00 - 18:00 Hora preferente: L 16:00 a 17:00. Para cualquier otra hora, concertar cita por e-mail.

Nota.- Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

Descripción de la Asignatura

Este curso contempla dos aspectos de manera simultánea: computación y restricciones. Está claro que los sistemas de computación tienen un impacto muy importante en nuestras vidas, y está claro que todo ingeniero o científico debe tener unos conocimientos básicos sobre su funcionamiento interno. Pero, ¿por qué deberíamos preocuparnos de las restricciones?

Los sistemas empujados, como cualquier sistema de computación, tienen que realizar una funcionalidad. Pero también tienen que cumplir unas restricciones a menudo muy estrictas:

- Restricciones de tiempo: El ABS de un coche tiene que activar los frenos en un tiempo muy corto para evitar accidentes.
- Una reducción en los requisitos de memoria y tamaño implica dispositivos más ligeros, más portables y más baratos.
- Los teléfonos móviles, los dispositivos multimedia portátiles y las redes de sensores inalámbricas suelen tener restricciones muy fuertes de consumo de energía.
- Por último, con tan escasos recursos, la seguridad se convierte en un reto muy difícil.

Además, un sistema empujado tiene que funcionar en el peor escenario posible, debe ser diseñado para cumplir las restricciones incluso en el caso peor.

En este curso los alumnos aprenderán a programar sistemas empujados basados en microprocesador y diseñar extensiones hardware para funcionar en el caso peor, considerando todas las restricciones durante el diseño y la implementación. Empezaremos por los conceptos más básicos para pronto avanzar a las técnicas más avanzadas.

Esta asignatura aporta el contenido teórico necesario para la asignatura "Laboratorio de Sistemas Electrónicos", que se imparte en el segundo semestre. El entorno de desarrollo y las herramientas que se presentan en esta asignatura se utilizarán también en el laboratorio. Y las prácticas de este laboratorio están diseñadas para complementar el enfoque dado en esta asignatura.

El curso está organizado en torno a varios proyectos utilizando la Raspberry-Pi, un sistema de computación del tamaño de una tarjeta de crédito y muy barato que se enchufa al televisor y un teclado.

Al final del curso, el alumno:

- Utilizará de forma eficiente las herramientas de desarrollo más ampliamente utilizadas (las herramientas de desarrollo del proyecto GNU): compilador GCC, GNU make, binutils, profilers y depuradores.
- Utilizará de forma eficiente el sistema operativo Linux, incluyendo extensiones de tiempo real basadas en Xenomai, y será capaz de describir su funcionamiento interno.
- Será capaz de escribir programas en C bien estructurados, formalmente correctos y eficientes, teniendo en cuenta restricciones de tiempo real estricto, restricciones de memoria, restricciones de consumo y restricciones de seguridad física.
- Será capaz de diseñar e implementar sistemas empujados completos basados en la Raspberry-Pi, conectando otros componentes hardware.

Temario

1. Conceptos básicos. Incluye los conceptos teóricos necesarios para la realización de sistemas empujados. Son los pilares sobre los que se sustenta el resto de aspectos y, en muchos casos, se trata de un enfoque diferente a conocimientos previos.

- 1.1. Introducción y conceptos básicos. Sistemas empujados. Sistemas ciber-físicos. Restricciones. Desarrollo cruzado.
- 1.2. Procesadores para sistemas empujados. Tipos de procesadores. Pipelining.
- 1.3. Arquitecturas de memoria. Tecnologías de memorias. Jerarquías de memoria. Modelos de memoria.
- 1.4. Programación de sistemas empujados. Introducción al desarrollo cruzado. Toolchain. Threads. Procesos. Planificación basada en interrupciones. Sistemas Operativos, microkernels, planificador FIFO.

2. Entorno de desarrollo. Contempla los aspectos prácticos. Se realizarán ejercicios sencillos utilizando herramientas de amplia difusión. De forma iterativa, se estudia el entorno de desarrollo cruzado de GNU y la plataforma de prototipado Raspberry Pi.

- 2.1. Entorno de desarrollo en C de GNU. Invocación básica de gcc, nm, objdump, make.
- 2.2. Raspberry Pi. Introducción a Raspbian y las herramientas de compilación nativa. Introducción a la depuración y herramientas básicas. Profiling. Análisis de FreeRTOS. Introducción a Xenomai
- 2.3. Desarrollo cruzado. Introducción al desarrollo cruzado. Depuración cruzada.

3. Modelado. Se describen técnicas para crear descripciones abstractas, no ambiguas y fácilmente analizables, de un sistema.

3.1. Diseño basado en modelos. Sistemas concurrentes. Tareas como funciones. Patrón Reactor. Modelos continuos, discretos e híbridos. Máquinas de estados y composición de modelos. Modelos de computación concurrentes.

3.2. Modelos concurrentes síncronos. Máquinas de estados finitos. Equivalencia y refinamiento.

4. Diseño con restricciones. Técnicas de diseño y análisis de sistemas, concentrando el esfuerzo en criterios de optimización que no son los habituales en sistemas de propósito general: requisitos de tiempo real, fiabilidad, uso de memoria y consumo.

4.1. Sistemas de tiempo real. Introducción a la planificación. Ejecutivos cíclicos. Planificación con prioridades fijas y desalojo. Planificación monótona en plazo. Recursos compartidos. Protocolos de herencia de prioridad y techo de prioridad inmediato.

4.2. Fiabilidad y técnicas de análisis formal. Reachability Analysis & Model Checking

4.3. Diseño con restricciones de memoria. Patrones de diseño para minimizar el uso de memoria. Análisis cuantitativo del uso de memoria.

4.4. Diseño de bajo consumo. Introducción al diseño de bajo consumo. Análisis cuantitativo del consumo medio.

Cronograma

Horas totales: 104 horas y 30 minutos

Horas presenciales: 46 horas (44.2%)

Peso total de actividades de evaluación continua:
100%

Peso total de actividades de evaluación sólo prueba final:
100%

Semana	Actividad Presencial en Aula	Actividad Presencial en Laboratorio	Otra Actividad Presencial	Actividades Evaluación
Semana 1	<p>1.1. Introducción y conceptos básicos. Sistemas empotrados. Sistemas ciber-físicos. Restricciones. Desarrollo cruzado.</p> <p>Duración: 02:00 LM: Actividad del tipo Lección Magistral</p>		<p>2.2. Raspberry Pi. Introducción a Raspbian y las herramientas de compilación nativa.</p> <p>Duración: 01:00 PL: Actividad del tipo Prácticas de Laboratorio</p>	
Semana 2	<p>1.4. Programación de sistemas empotrados. Introducción al desarrollo cruzado. Toolchain. Threads. Procesos. Planificación basada en interrupciones. Sistemas Operativos, microkernels, planificador FIFO.</p> <p>Duración: 03:00 LM: Actividad del tipo Lección Magistral</p>			<p>Ejercicios individuales con entrega en moodle. Evaluación por taller de moodle.</p> <p>Duración: 00:30 ET: Técnica del tipo Prueba Telemática Evaluación continua Actividad no presencial</p>
Semana 3	<p>2.1. Entorno de desarrollo en C de GNU. Invocación básica de gcc, nm, objdump, make.</p> <p>Duración: 02:00 LM: Actividad del tipo Lección Magistral</p>		<p>2.2. Raspberry Pi (II). Introducción a la depuración y herramientas básicas. Profiling. Análisis de FreeRTOS. Introducción a Xenomai</p> <p>Duración: 01:00 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Práctica 1: Familiarización con el entorno de programación de la Raspberry Pi. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 08:00 TG: Técnica del tipo Trabajo en Grupo Evaluación continua Actividad no presencial</p>
Semana 4	<p>4.1. Sistemas de tiempo real. Introducción a la planificación. Ejecutivos cíclicos. Planificación con prioridades fijas y desalojo. Planificación monótona en frecuencia. Planificación monótona en plazo.</p> <p>Duración: 03:00 LM: Actividad del tipo Lección Magistral</p>			<p>Práctica 2: Ejecutivo cíclico con Raspbian y wiringPi. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 06:00 TG: Técnica del tipo Trabajo en Grupo Evaluación continua Actividad no presencial</p>
Semana 5	<p>4.1. Sistemas de tiempo real (II). Recursos compartidos. Protocolos de herencia de prioridad. Protocolo de techo de prioridad inmediato. Sistemas reactivos con requisitos estrictos de tiempo.</p> <p>Duración: 03:00 LM: Actividad del tipo Lección Magistral</p>			<p>Práctica 3: Sistema con prioridades fijas y desalojo. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 10:00 TG: Técnica del tipo Trabajo en Grupo Evaluación continua Actividad no presencial</p>

Semana 6	<p>2.3. Desarrollo cruzado. Introducción al desarrollo cruzado. Depuración cruzada.</p> <p>Duración: 02:00</p> <p>LM: Actividad del tipo Lección Magistral</p>		<p>Puesta en común de los ejercicios de planificación</p> <p>Duración: 01:00</p> <p>AC: Actividad del tipo Acciones Cooperativas</p>	<p>Práctica 4: Sistema reactivo de tiempo real basado en Reactor. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 10:00</p> <p>TG: Técnica del tipo Trabajo en Grupo</p> <p>Evaluación continua</p> <p>Actividad no presencial</p>
Semana 7	<p>1.2. Procesadores para sistemas empotrados. Tipos de procesadores. Pipelining.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			
Semana 8	<p>1.3. Arquitecturas de memoria. Tecnologías de memorias. Jerarquías de memoria. Modelos de memoria.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			<p>Panel sobre el diseño de jerarquías de memoria para diferentes escenarios.</p> <p>Duración: 02:00</p> <p>PG: Técnica del tipo Presentación en Grupo</p> <p>Evaluación continua</p> <p>Actividad presencial</p>
Semana 9	<p>3.1. Diseño basado en modelos. Sistemas concurrentes. Tareas como funciones. Patrón Reactor. Modelos continuos, discretos e híbridos. Máquinas de estados y composición de modelos. Modelos de computación concurrentes.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			
Semana 10	<p>3.2. Modelos concurrentes síncronos. Máquinas de estados finitos. Equivalencia y refinamiento.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			<p>Práctica 5: Modelado, diseño y análisis de un sistema empotrado con requisitos estrictos de tiempo a partir de los requisitos (FSM). Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 16:00</p> <p>TG: Técnica del tipo Trabajo en Grupo</p> <p>Evaluación continua y sólo prueba final</p> <p>Actividad no presencial</p>
Semana 11	<p>4.2. Fiabilidad y técnicas de análisis formal. Reachability Analysis & Model Checking</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			
Semana 12	<p>4.3. Diseño con restricciones de memoria. Patrones de diseño para minimizar el uso de memoria. Análisis cuantitativo del uso de memoria.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			

Semana 13	<p>4.4. Diseño de bajo consumo. Introducción al diseño de bajo consumo. Análisis cuantitativo del consumo medio.</p> <p>Duración: 03:00</p> <p>LM: Actividad del tipo Lección Magistral</p>			<p>Práctica 6: Optimización de memoria y consumo. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.</p> <p>Duración: 08:00</p> <p>TG: Técnica del tipo Trabajo en Grupo</p> <p>Evaluación continua y sólo prueba final</p> <p>Actividad no presencial</p>
Semana 14	<p>Ejercicios y ejemplos.</p> <p>Duración: 03:00</p> <p>PR: Actividad del tipo Clase de Problemas</p>			
Semana 15				<p>Examen para evaluación continua</p> <p>Duración: 02:00</p> <p>EX: Técnica del tipo Examen Escrito</p> <p>Evaluación continua</p> <p>Actividad presencial</p> <p>Examen final</p> <p>Duración: 02:00</p> <p>EX: Técnica del tipo Examen Escrito</p> <p>Evaluación sólo prueba final</p> <p>Actividad presencial</p>
Semana 16				
Semana 17				

Nota.- El cronograma sigue una planificación teórica de la asignatura que puede sufrir modificaciones durante el curso.

Nota 2.- Para poder calcular correctamente la dedicación de un alumno, la duración de las actividades que se repiten en el tiempo (por ejemplo, subgrupos de prácticas") únicamente se indican la primera vez que se definen.

Actividades de Evaluación

Semana	Descripción	Duración	Tipo evaluación	Técnica evaluativa	Presencial	Peso	Nota mínima	Competencias evaluadas
2	Ejercicios individuales con entrega en moodle. Evaluación por taller de moodle.	00:30	Evaluación continua	ET: Técnica del tipo Prueba Telemática	No	5%	4 / 10	CE2, CE5.1, CG1
3	Práctica 1: Familiarización con el entorno de programación de la Raspberry Pi. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	08:00	Evaluación continua	TG: Técnica del tipo Trabajo en Grupo	No	5%	4 / 10	CG1, CE4, CE2
4	Práctica 2: Ejecutivo cíclico con Raspbian y wiringPi. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	06:00	Evaluación continua	TG: Técnica del tipo Trabajo en Grupo	No	5%	4 / 10	CO6, CG1, CE2
5	Práctica 3: Sistema con prioridades fijas y desalojo. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	10:00	Evaluación continua	TG: Técnica del tipo Trabajo en Grupo	No	5%	4 / 10	CO6, CG1, CE4, CE2
6	Práctica 4: Sistema reactivo de tiempo real basado en Reactor. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	10:00	Evaluación continua	TG: Técnica del tipo Trabajo en Grupo	No	5%	4 / 10	CO6, CG1, CE4, CE2
8	Panel sobre el diseño de jerarquías de memoria para diferentes escenarios.	02:00	Evaluación continua	PG: Técnica del tipo Presentación en Grupo	Sí	5%	4 / 10	CG1, CE1, CE5.1
10	Práctica 5: Modelado, diseño y análisis de un sistema empotrado con requisitos estrictos de tiempo a partir de los requisitos (FSM). Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	16:00	Evaluación continua y sólo prueba final	TG: Técnica del tipo Trabajo en Grupo	No	10%	5 / 10	CO6, CG1, CE5.3, CE1, CE5.1
13	Práctica 6: Optimización de memoria y consumo. Desarrollo y entrega fuera del horario de clase, preguntas de control en clase y comentarios por moodle.	08:00	Evaluación continua y sólo prueba final	TG: Técnica del tipo Trabajo en Grupo	No	10%	5 / 10	CG1, CE1, CE2
15	Examen para evaluación continua	02:00	Evaluación continua	EX: Técnica del tipo Examen Escrito	Sí	50%	4 / 10	CE4, CE1, CE2
15	Examen final	02:00	Evaluación sólo prueba final	EX: Técnica del tipo Examen Escrito	Sí	80%	5 / 10	CG1, CE4, CE1, CE2

Criterios de Evaluación

Los alumnos serán evaluados, por defecto, mediante evaluación continua. No obstante, los alumnos que lo deseen podrán ser evaluados mediante una única prueba final siempre y cuando así lo expresen por escrito al coordinador durante los dos primeros meses del curso. La presentación de este escrito supondrá la renuncia automática a la evaluación continua.

CONVOCATORIA ORDINARIA: MODALIDAD EVALUACIÓN CONTINUA: La asignatura se aprobará cuando se obtenga una calificación mayor o igual a 5 puntos sobre un total de 10. Dicha calificación es la suma de las calificaciones correspondientes a las diferentes actividades de evaluación (6 entregas prácticas, ejercicios en moodle, panel y examen final teórico-práctico por escrito). Además, para poder aprobar en

la modalidad de evaluación continua, es necesario obtener una nota mínima de 4 puntos (sobre 10) en todas las pruebas de evaluación, incluido el examen escrito que se realizará en la convocatoria oficial.

CONVOCATORIA ORDINARIA: EVALUACIÓN MEDIANTE UNA ÚNICA PRUEBA FINAL: el 80% de la calificación de los alumnos que presenten el escrito arriba referido se otorgará en función de una única prueba final a celebrar en la convocatoria oficial. Además, es necesario presentar el día del examen las prácticas 5 y 6, sobre las que se realizarán algunas preguntas de control en el examen. Estas prácticas suponen un 10%, cada una, de la nota final. La contestación incorrecta de las preguntas de control en el examen supone la anulación de la práctica.

CONVOCATORIA EXTRAORDINARIA: la evaluación de la asignatura en su convocatoria extraordinaria se realizará de la misma forma que la modalidad de evaluación mediante una única prueba final, anteriormente

Recursos Didácticos

Descripción	Tipo	Observaciones
Moodle de la asignatura	Recursos web	
Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, http://LeeSeshia.org , ISBN 978-0-557-70857-4, 2011.	Bibliografía	
Christopher Hallinan, Embedded Linux Primer: A Practical, Real-World Approach, Second Edition, Prentice Hall, ISBN-13: 978-0-13-701783-6, 2010	Bibliografía	
Marilyn Wolf, Computers as Components: Principles of Embedded Computing System Design, 3rd edition, Morgan Kaufmann, ISBN 978-0-12-388436-7, 2012.	Bibliografía	Complementario, nivel básico.
Jane W. S. Liu, Real-Time Systems, Prentice Hall, ISBN 0-13-099651-3, 2000	Bibliografía	Complementario, planificación de sistemas de tiempo real.
Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, and Philippe Gerum, Building Embedded Linux Systems, 2nd edition, O'Reilly, ISBN 978-0-596-52968-0, 2008.	Bibliografía	Complementario