

Sistemas Empotrados

Código: 102791
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	1
2502441 Ingeniería Informática	OT	4	1

Contacto

Nombre: Lluís Ribas Xirgo
Correo electrónico: lluis.ribas@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)
Algún grupo íntegramente en inglés: No
Algún grupo íntegramente en catalán: No
Algún grupo íntegramente en español: No

Otras observaciones sobre los idiomas

El material está, principalmente, en inglés.

Equipo docente

Joaquín Saiz Alcaine
Ismael Fabricio Chaile Alfaro

Prerequisitos

Para la plena comprensión de los contenidos de la asignatura conviene tener una habilidad básica en la programación y un buen conocimiento de cómo se ejecutan los programas en los computadores. Para ello, se tiene que haber hecho el Laboratorio de programación y la Ingeniería del software (programación) y haber cursado Estructura de computadores, Sistemas operativos y Arquitectura de computadores (modelo de ejecución de los programas).

Objetivos y contextualización

Esta asignatura es la primera de la materia de Diseño de sistemas de cómputo orientado a aplicaciones. En esta materia trata del desarrollo de los sistemas que ejecutan los algoritmos de unas aplicaciones específicas respetando unos requerimientos que suelen ser muy exigentes. Por ejemplo, no es suficiente que un dispositivo móvil sea capaz de mostrar un vídeo, sino que lo tiene que hacer a 25 imágenes por segundo, sincronizándose con la información auditiva y consumiendo la mínima energía posible. Así pues, el objetivo final de la materia es que se sepa diseñar algoritmos y las implicaciones que cada diseño tiene en el coste de la aplicación según el mismo algoritmo y cómo se implementa, es decir, según la plataforma de ejecución que se elija.

En este contexto, en la asignatura de Sistemas empotrados se trata de que el alumnado alcance los objetivos siguientes:

- Conocer los diversos ámbitos de aplicación de los sistemas empotrados.
- Tener nociones de los requerimientos habituales de cada dominio de aplicación, incluidos los de tiempo real.
- Comprender los aspectos de seguridad, fiabilidad y robustez de los sistemas.
- Conocer la metodología del desarrollo de los sistemas empotrados.
- Entender los diversos modelos de cálculo de los sistemas.
- Tener habilidad práctica con el diseño y manipulación de los modelos de cálculo orientados a estados.
- Conocer los elementos básicos de las arquitecturas de los sistemas empotrados.
- Haber adquirido los rudimentos del diseño basado en plataformas.
- Saber estimar costes de implementación a partir de los modelos de cálculo de los sistemas.
- Conocer el problema de la partición de los sistemas y diversas estrategias para solucionarlo.
- Tener los rudimentos de programación y descripción de hardware para la implementación de los sistemas.

Competencias

Ingeniería Informática

- Adquirir hábitos de trabajo personal.
- Capacidad de desarrollar procesadores específicos y sistemas empotrados, así como desarrollar y optimizar el software de dichos sistemas.
- Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- 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.
- Comunicación.
- Trabajar en equipo.

Resultados de aprendizaje

1. Analizar los requerimientos de las aplicaciones informáticas.
2. Asumir y respetar el rol de los diversos miembros del equipo, así como los distintos niveles de dependencia del mismo.
3. Comparar y evaluar las posibles plataformas para cumplir los requerimientos de las aplicaciones.
4. Gestionar el tiempo y los recursos disponibles. Trabajar de forma organizada.
5. Identificar las necesidades de la aplicación específica que se desea resolver.
6. Identificar las necesidades de seguridad que deben cumplir los sistemas empotrados.
7. Trabajar de forma autónoma.
8. Utilizar el inglés como idioma de comunicación y relación profesional de referencia.

Contenido

1. Introducción

1.1. Dominios de aplicación de los sistemas embebidos

1.2. Casos de estudio

1.3. Ingeniería de sistemas

2. Metodología de diseño

2.1. Flujo de diseño

2.2. Análisis de requerimientos funcionales y no funcionales

2.3. Modelos de cálculo basados en estados

2.4. Redes de máquinas de estado

2.5. Arquitecturas del hardware

2.6. Sistemas operativos en tiempo real

2.7. Arquitecturas del software

2.8. Plataformas arquitecturales

3. Metodología de implementación

3.1. Refinamiento de especificaciones

3.2. Reusabilidad

3.3. Particionado y planificación

3.4. Síntesis de software y de hardware

Metodología

La docencia se estructura a partir de las actividades presenciales siguientes:

Clases de teoría: Son sesiones de exposición de contenidos, con una primera parte que se dedica a la divulgación de los conocimientos necesarios para el análisis y el diseño de los sistemas embebidos y a explicar casos que sitúen en contexto el conocimiento y las habilidades que se adquieren en la asignatura. La segunda parte se dedicará a plantear los problemas que se tratarán en los seminarios correspondientes.

Seminarios de problemas: Discusión de pequeños casos de estudio que sirvan para consolidar los conocimientos teóricos en cuanto al análisis y el diseño de los sistemas embebidos.

Prácticas en laboratorio: Sesiones de trabajo en grupo, siguiendo un guion y supervisadas por un profesor o una profesora. En cada sesión se tratará un aspecto concreto en cuanto a la implementación de los sistemas embebidos.

Hay una parte muy importante de trabajo en equipo fuera del aula, tanto en cuanto a los problemas propuestos en clase como para la realización de las prácticas. En este sentido, cada miembro de cada equipo deberá asumir diferentes roles para cada trabajo que se le encargue al equipo. Esto también supone que trabajar de forma organizada y saber trabajar de forma autónoma cuando convenga.

COMPETENCIAS TRANSVERSALES

En esta asignatura se pretende que el alumnado adquiera autonomía y capacidad de organización en el trabajo propio, así como una competencia básica en el trabajo de equipo y en inglés.

En este sentido, habrá una parte de la evaluación específica de cada uno de los resultados de aprendizaje correspondientes:

T03.02. Asumir y respetar el rol de los diversos miembros del equipo, así como los diferentes niveles de dependencia del equipo: El proyecto de las prácticas se tendrá que hacer en equipos y la presentación final habrá de incluir, necesariamente, la descripción de qué ha hecho cada persona.

T02.01. Trabajar de manera autónoma: Los problemas y la parte que cada persona asuma del proyecto de la práctica en su equipo se tiene que resolver y hacer de forma individual.

T02.03. Gestionar el tiempo y los recursos disponibles. Trabajar de manera organizada: En el proyecto de prácticas, la organización del trabajo propio queda sujeto al rol en el equipo. En cualquier caso, todas las

entregas hechas con retraso tienen penalizaciones, lo que obliga a las y los estudiantes a gestionar el tiempo de manera que su equipo pueda cumplir con los requerimientos temporales que se le piden.

T04.03. Utilizar el inglés como el idioma de comunicación y de relación profesional de referencia: La mayoría del material es en inglés para fomentar el aprendizaje de este idioma y se valora positivamente que los informes de las prácticas o los problemas se entreguen en inglés.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Problemas: Propuestas de soluciones, discusión de problemas y resolución de dudas	12	0,48	1, 2, 3, 4, 7
Prácticas: Desarrollo del proyecto de asignatura	12	0,48	1, 2, 3, 4, 7
Teoría: Asistencia y participación en clases magistrales	22	0,88	5, 6
Tipo: Supervisadas			
Resolución de problemas adicionales	6	0,24	1, 3, 4, 7
Seguimiento del trabajo del proyecto del curso	6	0,24	1, 3, 4, 7
Tipo: Autónomas			
Estudio	26	1,04	5, 6, 7
Prácticas: Desarrollo del proyecto y elaboración de informes	12	0,48	1, 2, 3, 4, 7, 8
Resolución de problemas y elaboración de informes	24	0,96	1, 2, 3, 4, 7, 8

Evaluación

a) Proceso y actividades de evaluación programadas

La evaluación es continua con actividades específicas (exámenes y trabajos) a lo largo del curso. Estas actividades de evaluación generan una serie de notas que determinan la nota final.

El cálculo de la nota final, n , sigue la siguiente expresión:

$$n = \max(x \cdot 50\% + c \cdot 25\% + p \cdot 25\%, x \cdot 75\% + p \cdot 25\%)$$

donde x es la nota del examen, c , la de la evaluación continuada, y p , la del proyecto.

La nota final será, como máximo, un 4,5 si x o $p < 5$. En otras palabras, debe aprobarse el examen y el proyecto por separado.

Hay que tener en cuenta que, si la nota de la evaluación continuada no mejora la nota final, no se tiene en cuenta para su cálculo. Por eso, la nota final es la máxima entre las notas con y sin evaluación continuada.

La nota del examen (x) es la nota del examen final, que se podrá recuperar en un segundo examen.

La nota de la evaluación continuada (c) se obtiene de una media ponderada de las pruebas de evaluación continuada que se hagan a lo largo del curso. Se prevé que se hagan tres.

La nota del proyecto (p) será el resultado de una media ponderada de todas las entregas de seguimiento y de la entrega y defensa final.

b) Programación de las actividades de evaluación

Las fechas de las pruebas de evaluación continua, de la defensa del proyecto y de los plazos de las entregas se publicarán en el campus virtual (CV) y pueden estar sujetos a posibles cambios de programación por motivos de adaptación a posibles incidencias: siempre se informará previamente a través del CV ya que se entiende que es el mecanismo habitual de intercambio de información entre profesorado y estudiantes fuera del aula.

c) Proceso de recuperación

Las entregas fuera de plazo, siempre que haya previo aviso, serán aceptados y penalizados con una nota más baja. En ningún caso se admitirán entregas fuera de plazo sin previo aviso o justificación de fuerza mayor. Se podrá abrir un segundo plazo de entrega por los informes que reciban una evaluación negativa. Los trabajos no entregados recibirán una nota de 0 y no tendrán opción a una segunda evaluación.

De acuerdo con la coordinación del Grado y la dirección de la Escuela de Ingeniería las siguientes actividades no se podrán recuperar en el examen final:

- Proyecto, 25% de la calificación final

La evaluación continuada se puede recuperar con el examen final.

El examen final se puede recuperar con un segundo examen.

d) Procedimiento de revisión de las calificaciones

Las revisiones se podrán hacer en cualquier momento después de la publicación de las notas y antes del final del plazo de revisión del examen final.

Si, como resultado de una revisión, se acuerda el cambio de una nota, la nueva nota no se podrá modificar en una revisión posterior.

Una vez pasado el plazo de revisión del examen final, solo se harán revisiones del examen de recuperación.

e) Calificaciones

La calificación de "no evaluable" solo se otorgará a las personas que no realicen ninguna actividad evaluable. La participación en cualquier actividad evaluable implica que el resto de actividades que no se realicen se computen como 0 en el cálculo de la nota final.

Las matrículas de honor se concederán a quienes obtengan una nota superior o igual a 9,0 en cada parte, hasta el 5% de los matriculados según orden descendente de nota final. A criterio del profesorado, también se podrán conceder en otros casos, siempre que no se exceda del 5% y la nota final sea igual o superior a 9,0.

f) Irregularidades, copia y plagio

Las copias se refieren a las evidencias de que el trabajo o el examen se ha hecho en parte o totalmente sin contribución intelectual del autor. En esta definición se incluyen también las tentativas probadas de copia en exámenes y entregas de trabajos y las violaciones de las normas que aseguran la autoría intelectual. Los plagios hacen referencia a los trabajos y textos de otros autores que se hacen pasar como propios. Son un delito contra la propiedad intelectual. Para evitar incurrir en plagio, hay que citar las fuentes que utiliza a la hora de escribir el informe de un trabajo.

De acuerdo con la normativa de la UAB, tanto copias como plagios o cualquier intento de alterar el resultado de la evaluación, propia o ajena -dejando copiar, por ejemplo, implican una nota final de la parte correspondiente (examen, evaluación continuada o proyecto) de 0, a efectos de calcular un valor cuantitativo de la nota, y suspender la asignatura, sin que ello limite el derecho a emprender acciones en contra de quienes hayan participado en estos actos, tanto en el ámbito académico como en el penal.

g) Evaluación de estudiantes que repiten

No hay ningún tratamiento diferenciado para alumnos que repitan la asignatura, pero pueden aprovechar material propio del curso anterior siempre que lo indiquen así en los informes correspondientes.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Defensa del proyecto	12,5	10	0,4	1, 2, 3, 4, 7, 8
Examen de recuperación	50	2	0,08	5, 6
Examen final	50	2	0,08	5, 6
Informes de los trabajos de prácticas (5)	12,5	10	0,4	1, 2, 3, 4, 7, 8
Pruebas de evaluación continuada (3)	25	6	0,24	1, 3, 5, 7

Bibliografía

LI. Ribas Xirgo. (2014). *How to code finite state machines (FSMs) in C. A systematic approach*. TR01.102791 Embedded Systems. Universitat Autònoma de Barcelona.
[https://www.researchgate.net/publication/273636602_How_to_code_finite_state_machines_FSMs_in_C_A_syste]

Explica un método de programación de máquinas de estado en C que es muy similar al que se ve en la asignatura.

LI. Ribas Xirgo. (2011). "Estructura bàsica d'un computador", Capítulo 5 de Montse Peiron Guàrdia, Lluís Ribas i Xirgo, Fermín Sánchez Carracedo y A. Josep Velasco González: *Fonaments de computadores*. Material docent de la UOC. OpenCourseWare de la UOC.
[<http://openaccess.uoc.edu/webapps/o2/handle/10609/12901>]

Trata del modelo de máquinas de estado, de las máquinas algorítmicas y de las arquitecturas básicas de los sistemas digitales, que coinciden con los temas correspondientes en la asignatura.

Edward A. Lee and Sanjit A. Seshia. (2017) *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, Second Edition, MIT Press.

Un curso con contenidos similares, desde una perspectiva más formal. Ver también:
<https://ptolemy.berkeley.edu/>

M. J. Pont. (2005). *Embedded C*. Pearson Education Ltd.: Essex, England.

Trata de cómo programar sistemas embebidos, tema que coincide con lo que se trata en la parte de problemas y prácticas de la asignatura. Por lo tanto, es un material complementario muy interesante.

Brian Bailey, Grant Martin and Andrew Piziali. (2007). *ESL Design and Verification. A Prescription for Electronic System-Level Methodology*. Elsevier.

Hace un repaso de todo el proceso de síntesis de sistemas embebidos y pone en contexto el material de la asignatura. Por lo tanto, es un buen material complementario.

Tim Wilmshurst. (2010). *Designing Embedded Systems with PIC Microcontrollers. Principles and Applications (Second Edition)*. Elsevier.

Información complementaria a la de la asignatura, que presenta un posible sistema embebido para el control de un robot.

Oliver H. Bailey. (2005). *Embedded Systems Desktop Integration*. Wordware Publishing.

Información complementaria a la de la asignatura que incide, sobre todo, en el aspecto de la comunicación entre el hardware y el software.

Software

CoppeliaSim, EDU Version, Coppelia Robotics [<https://www.coppeliarobotics.com/>]

ZeroBrane Studio, ZeroBrane [<https://studio.zerobrane.com/>]

Draw.io, diagrams.net [<https://app.diagrams.net/>]