



Universidad  
de Alcalá

# GUÍA DOCENTE

## SISTEMAS EMPOTRADOS

**Grado en Ingeniería de Computadores**  
**Universidad de Alcalá**

**Curso Académico 2018/19**  
**3º Curso - Segundo Cuatrimestre**

## GUÍA DOCENTE

Nombre de la asignatura:	<b>Sistemas Empotrados</b>
Código:	<b>590010</b>
Titulación en la que se imparte:	<b>Ingeniería de Computadores</b>
Departamento y Área de Conocimiento:	<b>Automática</b>
Carácter:	<b>Obligatoria</b>
Créditos ECTS:	<b>6</b>
Curso y cuatrimestre:	<b>3º, C2</b>
Profesorado:	Óscar Rodríguez Polo, Pablo Parra Espada
Horario de Tutoría:	
Idioma en el que se imparte:	Español/English Friendly

### 1.a PRESENTACIÓN

Esta asignatura, junto con “Sistemas de Tiempo Real” y “Percepción y Control”, conforma la materia de Ingeniería de Sistemas. La asignatura se centra en los sistemas empotrados, los elementos que se utilizan para su construcción y los conocimientos necesarios para comprender su funcionamiento, afrontar su diseño y tomar decisiones sobre su implementación.

La asignatura comenzará con una introducción dónde se presentan las características fundamentales de los sistemas empotrados, explicando los elementos principales que los forman, las métricas más importantes que afectan a su desarrollo, así como las distintas posibilidades de abordar la implementación de su hardware de computo.

Tras la introducción, se expondrán los distintos niveles de diseño de un sistema empotrado, así como las técnicas empleadas, tanto de diseño como de validación, que permiten mejorar la productividad y la calidad de los sistemas desarrollados. Entre las técnicas explicadas, se describirá, mediante un caso de uso, la técnica de co-diseño hardware-software, con el fin de que el alumno pueda conocer los retos y beneficios que supone la utilización de la misma en el desarrollo de sistemas empotrados.

A continuación se definirá el concepto de arquitectura de un sistema empotrado, y se describirán los elementos más característicos que se emplean en las estructuras que forman parte de la arquitectura, así como la funcionalidad que proporcionan, justificando su idoneidad para formar parte de este tipo de sistemas.

Finalmente, se estudiará el proceso de desarrollo de los sistemas empotrados, definiendo el ciclo de vida que permite su especificación y validación, así como la posterior implementación y despliegue del sistema. Se hará, además, una breve introducción a algunas herramientas que facilitan el proceso.

Paralelamente a la exposición teórica, el alumno llevará a cabo prácticas que refuercen los contenidos teóricos. En este sentido tendrá que ser capaz de desarrollar, desplegar y depurar software sobre el hardware propio de los sistemas empotrados, así como familiarizarse con la simulación un sistema empotrado que implique aspectos hardware y software. Para las prácticas se utilizará primeramente el hardware básico de un sistema empotrado, implementándose drivers sencillos que permitan una comunicación serie, la gestión de las interrupciones y los servicios básicos de temporización. Posteriormente se utilizarán servicios más avanzados, que faciliten la programación de funciones en el tiempo sobre un ejecutivo cíclico. Finalmente, el alumno realizará prácticas sobre sistemas multitarea, familiarizándose con la utilización de servicios de temporización, mecanismos de protección de acceso a recursos compartidos, y mecanismos de comunicación, tanto síncrona como asíncrona, entre esas tareas.

## 1.b COURSE SUMMARY

This course shows the introduction to embedded systems. It allows to understand the behaviour of this type of systems and to provide skills for its design and implementation.

The course explains the metrics that driven the embedded systems development. It shows the different design levels and, for each level, the design alternatives and development techniques that make feasible to meet the requirement related to these metrics. The hardware-software co-design techniques will be introduced, focusing in those that facilitate the partitioning, so, after the system level definition, the hardware and software components can be implemented and validated separately. The embedded systems development process is studied. The architecture of an embedded system is defined and the life cycle that allows its specification, implementation and validation is explained.

In parallel with the theoretical lessons, a set of practices are proposed. These practices are focused on the software development on an embedded platform, and they let the student to work with industrial environments, including a simulator of a system on a chip, and a toolchain that integrates the compiling of the software projects, and also the deployment and debugging on the final target.

## 2. COMPETENCIAS

Esta asignatura contribuye a que el estudiante desarrolle las siguientes competencias y tiene como objetivo un conjunto de resultados del aprendizaje:

Competencias genéricas:

- TR5: Obtener las habilidades de comunicación interpersonal y de trabajo en equipo para el trabajo en proyectos y grupos de trabajo.
- TRU1. Capacidad de análisis y síntesis.
- TRU2 Comunicación oral y escrita.
- TRU3 Capacidad de gestión de la información.
- TRU4 Aprendizaje autónomo.
- TRU5 Trabajo en equipo.

Competencias de carácter profesional:

Esta asignatura contribuye a adquirir las siguientes competencias propias del perfil profesional del Grado de Ingeniería en Computadores:

- CB5: Capacidad para diseñar, sintetizar, construir y validar un computador empleando los elementos hardware y software necesarios.
- CB10: Capacidad para seleccionar técnicas, tecnologías y metodologías adecuadas para desarrollar sistemas controlados por computador.
- CB23: Capacidad de comunicar los conocimientos adquiridos y defensa de la solución propuesta.
- CB25: Capacidad de análisis y síntesis.
- CB26: Capacidad de organización y planificación.
- CB27: Capacidad para la resolución de problemas y la toma de decisiones.
- CB28: Capacidad para el trabajo en equipo y habilidades en las relaciones interpersonales.
- CB29: Capacidad de razonamiento crítico y el aprendizaje autónomo.
- CB31: Capacidad de expresión y comprensión a nivel básico.

Resultados del aprendizaje:

- RA1 Enumerar las características específicas de los sistemas empotrados y describir cómo éstas afectan al su proceso de desarrollo.
- RA2 Enumerar las distintas métricas de diseño de los sistemas empotrados.
- RA3 Describir el papel de la unidad de procesamiento en el contexto de un sistema completo con E/S y memoria.
- RA4: Describir cómo es la comunicación de la unidad de procesamiento con el mundo exterior por medio de los periféricos.
- RA5 Identificar las alternativas de implementación de un sistema empotrado y ser capaz de seleccionar la más adecuada acorde a las métricas de diseño marcadas como objetivo.
- RA6 Identificar los distintos niveles de diseño de un sistema empotrado y describir las técnicas utilizadas para desarrollarlos.
- RA7 Definir el concepto de arquitectura de los sistemas empotrados y describir el proceso seguido para su especificación y validación.
- RA8 Ser capaz de explicar el diseño de un sistema empotrado.
- RA9 Ser capaz de programar un sistema empotrado y describir cómo el lenguaje de alto nivel se convierte en código ejecutable.
- RA10 Ser capaz de emplear compiladores y entornos de desarrollo propios de los sistemas empotrados.

### 3. CONTENIDOS

Bloques de contenido (se pueden especificar los temas si se considera necesario)	Total de clases, créditos u horas
Introducción a los sistemas empotrados	10 horas
Técnicas de diseño y validación de sistemas empotrados	18 horas
Arquitectura de un sistema empotrado y elementos de diseño	16 horas
Proceso y entornos de desarrollo de los sistemas empotrados	12 horas

### 4. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE. ACTIVIDADES FORMATIVAS

#### 4.1. Distribución de créditos (especificar en horas)

Número de horas presenciales:	<ul style="list-style-type: none"><li>Clases en gran grupo: 28 horas (2 horas x 14 semanas)</li><li>Clases en grupo reducido: 28 horas (2 horas x 14 semanas)</li><li>Evaluaciones: 4 horas</li></ul> Total: 60 horas presenciales
Número de horas del trabajo propio del estudiante:	<ul style="list-style-type: none"><li>Preparación de las clases, aprendizaje autónomo, preparación de ejercicios, pruebas y prácticas, preparación de la prueba final.</li></ul> Total: 90 horas
Total horas	150 horas

#### 4.2. Estrategias metodológicas, materiales y recursos didácticos

Clases Teóricas (en grupos grandes)	<ul style="list-style-type: none"><li>Presentación y/o revisión de conceptos</li><li>Presentaciones orales y otras actividades</li></ul>
Clases Prácticas (en grupos reducidos)	<ul style="list-style-type: none"><li>Presentación y/o revisión de conceptos de carácter eminentemente práctico.</li><li>Sesiones prácticas de laboratorio: orientadas a consolidar los conceptos presentados previamente, así como a familiarizar al estudiante con herramientas hardware y software de apoyo al estudio de la materia y futuro desempeño profesional.</li><li>Presentaciones orales y otras actividades</li></ul>

Tutorías individuales, grupales y vía web (foro, correo, etc.)

- Resolución de dudas
- Apoyo al aprendizaje autónomo

## 5. EVALUACIÓN: Procedimientos, Criterios de evaluación, Instrumentos y Criterios de calificación

### 5.1. Procedimientos

La evaluación puede realizarse de forma continua o mediante una evaluación final, existiendo para cada caso dos convocatorias por matrícula: ordinaria y extraordinaria.

- **Evaluación continua**

La evaluación continua valora el desarrollo de las competencias durante todo el proceso de aprendizaje de la asignatura (NRPEA, Art. 3) mediante una serie de pruebas de carácter sumativo distribuidas a lo largo del curso, que permiten al estudiante abordar la asignatura de forma progresiva.

Garantiza la retroalimentación temprana en el proceso de aprendizaje del alumno y permite a los profesores, coordinadores y demás elementos del Sistema de Garantía de Calidad hacer un seguimiento global, con la posibilidad de actuar en caso de que lo aconsejen indicadores o situaciones determinadas.

La evaluación de la parte relacionada con las prácticas se realizará al finalizar su bloque correspondiente.

- **Evaluación final**

según la Normativa Reguladora de los Procesos de Evaluación de los Aprendizajes aprobada en Consejo de Gobierno el 24 de marzo de 2011, existe la posibilidad de acogerse a una prueba de evaluación final. Según cita literalmente dicha normativa: para acogerse a la evaluación final, el estudiante tendrá que solicitarlo por escrito al decano o director de centro en las dos primeras semanas de impartición de la asignatura, explicando las razones que le impiden seguir el sistema de evaluación continua.

### Criterios de evaluación

Para determinar si el alumno ha alcanzado los resultados previstos, se tendrán en cuenta los siguientes criterios de evaluación:

CE1: El alumno ha adquirido los conocimientos sobre métricas y técnicas de diseño de sistemas empotrados.

CE2: El alumno ha adquirido los conocimientos sobre el proceso de desarrollo de sistemas empotrados.

CE3: El alumno muestra capacidad de aplicación e integración de los contenidos a problemas, escenarios o casos de estudio afines a la asignatura.

CE4: El alumno muestra capacidad e iniciativa para desarrollar software para sistemas empotrados y utilizar entornos de desarrollo específicos de este tipo de sistemas.

CE5: El alumno demuestra capacidad de argumentación y de emisión de juicios sobre diseños de sistemas empotrados planteados en la asignatura.

CE6: El alumno cumple con las tareas encomendadas.

CE7: El alumno muestra interés por los contenidos y la materia trabajada.

CE8: El alumno demuestra cuidado formal, claridad y rigor en la exposición de ideas y razonamientos.

## 5.2. Instrumentos de calificación

El rendimiento de los alumnos será valorado por su trabajo, conocimientos y destrezas adquiridas y la mejora de su proceso de aprendizaje.

Los instrumentos de evaluación continua a emplear consistirán en la realización de actividades de evaluación continua planteadas por el profesor para cada uno de los temas.

El total de las actividades planteadas, sus contenidos y temporización, se comunicarán al alumno durante la clase de presentación o a la finalización del bloque de contenido correspondiente. Dichas actividades comprenden:

PEI: Pruebas de evaluación intermedia, de tipo test. A lo largo del curso se realizarán exámenes objetivos cuyo contenido versará sobre los temas tratados o sobre presentaciones de trabajos en clase. Los contenidos y la temporalización de estos exámenes se fijarán durante los primeros días de clase

PL: Pruebas de laboratorio. Sobre las prácticas realizadas se plantearán ampliaciones o variaciones de las mismas que deberán ser resueltas por los alumnos.

En: Entregables. Un trabajo individual y otro grupal serán entregados. El trabajo grupal irá acompañado, además, de una memoria y será presentado en clase.

PEF: Prueba de evaluación final, consistente en un examen escrito de resolución de problemas a la finalización del periodo docente.

### Convocatoria ordinaria – Evaluación continua

Resultados de aprendizaje	Criterios de evaluación	Instrumentos de evaluación	Peso en la calificación
RA1 – RA6	CE1	PEI1	10%
RA7	CE2	PEI2	10%
RA8	CE5, CE6, CE7, CE8	E1	7%
RA9-RA10	CE4,CE6,CE7	PL1	10%
RA9-RA10	CE4,CE6,CE7	PL2	10%
RA9-RA10	CE4,CE6,CE7	PL3	10%
RA9-RA10	CE4,CE6,CE7	PL4	10%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1-RA8	CE3, CE5, CE7	PEI3	10%

### Convocatoria ordinaria – Evaluación final

Resultados de aprendizaje	Criterios de evaluación	Instrumentos de evaluación	Peso en la calificación
RA8	CE5, CE6, CE7, CE8	E1	7%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1 – RA10	CE1 – CE8	PEF	70%

### Convocatoria extraordinaria

Resultados de aprendizaje	Criterios de evaluación	Instrumentos de evaluación	Peso en la calificación
RA8	CE5, CE6, CE7, CE8	E1	7%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1 – RA10	CE1 – CE8	PEF	70%

## 6. BIBLIOGRAFÍA

### Bibliografía Básica

- Frank Vahid and Tony Givargis. Embedded System Design: A Unified Hardware/Software Introduction. John Wiley & Sons; ISBN: 0471386782. 2002. El libro describe conceptos básicos de sistemas empotrados, las métricas empleadas en su diseño y las alternativas de implementación. Se analizan las técnicas arquitectura de un sistema empotrado en la que se integra el hardware de procesamiento, el software, los periféricos, la memoria, etc. Se tratan algunos ejemplos de estudio y se presentan soluciones de diseño de alto nivel basadas en tareas concurrentes y en la especificación del comportamiento mediante máquinas de estados.
- David E. Simon. An Embedded Software Primer. Addison-Wesley, 1999. El libro presenta los fundamentos hardware y software necesarios para desarrollar un sistema empotrado. En él se describen las distintas técnicas empleadas en su resolución, profundizando en aquellas



basadas en la utilización de un sistema operativo de tiempo real. Los contenidos son adecuados para tener una visión de conjunto de este tipo de sistemas e introduce nociones básicas sobre herramientas de desarrollo y depuración.

- Arnold S. Berger. *Embedded Systems Design: An Introduction to Processes, Tools and Techniques*. CMP Books; 1st edition (December 15, 2010). El libro aborda desde el punto de vista práctico el diseño de sistemas empotrados. Centra sus contenidos en las características técnicas de las distintas alternativas y herramientas, tanto hardware como software, que se emplean en el desarrollo y depuración de este tipo de sistemas. Tammy Noergaard. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*. Editorial Newnes. 2013. Este libro también cubre ampliamente la temática de los sistemas empotrados. Los temas tratados son: elementos hardware, sistema operativo, lenguajes ensambladores y de alto nivel, redes y casos de estudio.

## Bibliografía Complementaria (optativo)

- [Jack Ganssle](#). *The Art of Designing Embedded Systems*. Newnes. 2008. Libro que refleja la experiencia de un desarrollador de sistemas empotrados. Resulta de especial utilidad una vez que se han adquirido los fundamentos teóricos y comienzan los primeros trabajos prácticos del alumno.
- Chris Rowen. *Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors*. Ed. Prentice Hall PTR. 2004. Es un libro que cubre de un modo unificado el diseño de SOC para procesadores configurables y extensibles. Aborda temas como cogeneración hardware/software, particionado, comunicaciones y sistemas con varios procesadores.
- Richard Zurawski. *Embedded Systems Handbook (Industrial Information Technology)*. 2009. Es un libro que cubre la temática de los sistemas empotrados de un modo amplio enfatizando en nuevas tendencias y tecnologías sin olvidarse de los fundamentos. Cubre todos los aspectos ligados a los sistemas empotrados tanto desde la perspectiva del software como de la del hardware. Entre ellos citaremos sistemas de tiempo real, seguridad, system-on-chip y network-on-chip, redes, etc. También incluye ejemplos y aplicaciones en el campo industrial, automoción y sensores inteligentes.
- Axel Jantsch. *Modeling Embedded Systems and SoC's: Concurrency and Time in Models of Computation (The Morgan Kaufmann Series in Systems on Silicon)*. Ed. Morgan Kaufmann. 2003. El libro se centra en la presentación de técnicas de modelado para sistemas empotrados. Cubre típicos como máquinas de estado, redes de Petri o modelos de computación síncronos. También aporta ejemplos reales de uso de estas técnicas.
- David C. Black, Jack Donovan. *SystemC: From the Ground Up*. Ed. Springer. 2010. SystemC proporciona un conjunto de extensiones a C++ que permite el desarrollo rápido de sistemas hardware/software. Este libro se centra en aspectos prácticos del lenguaje para modelar sistemas reales. El amplio surtido de ejemplos y de programas disponibles de forma gratuita permite al lector una rápida introducción a SystemC.



Universidad  
de Alcalá

# TEACHING GUIDE

## EMBEDDED SYSTEMS (Sistemas Empotrados)

**Bachelor Degree in Computer  
Engineering  
University of Alcalá**

**Academic Year 2018/19  
3<sup>rd</sup> Year - 2<sup>nd</sup> Term**

## TEACHING GUIDE

Course name:	<b>Embedded Systems</b>
Code:	<b>590010</b>
Degrees:	<b>Bachelor's Degree in: Graduate on Engineering on computer Engineering</b>
Department and Knowledge Area:	<b>Computer Engineering Department</b>
Type:	<b>Obligatory</b>
ECTS Credits:	<b>6</b>
Year and term:	<b>3<sup>rd</sup>, 2<sup>nd</sup></b>
Teaching staff:	<b>Óscar Rodríguez Polo, Pablo Parra Espada</b>
Office Hours:	<b>Check the course webpage</b>
Language:	<b>Spanish/English Friendly</b>

### 1.a INTRODUCTION

This subject, together with "Real Time Systems" and "Perception and Control", covers the subject of Systems Engineering. It is focused on embedded systems, the elements used for their construction, and the knowledge necessary to understand their operation, face their design and take decisions about their implementation. The subject begins with an introduction in which the main characteristics of the embedded systems are presented, and the main elements that form them are explained, together with the most important metrics that affect their development, as well as the different possibilities of implementation of their processing hardware.

After the introduction, the different design levels of the embedded system will be exposed, as well as the design and validation techniques used in order to improve the productivity and quality of the developed systems. Among the techniques explained, the co-design hardware-software will be described, through a use case, in order that the student can know the challenges and benefits of using it in the development of embedded systems.

Next, the concept of an embedded system architecture will be defined, and the most significant elements used in the structures that are part of the architecture will be described, as well as the functionality they provide, justifying their suitability to be part of this type of systems.

Finally, the development process of the embedded systems will be studied, defining the life cycle that includes its specification and validation, as well as the subsequent implementation and deployment of the system. A set of tools that facilitate the process will be also introduced.

Parallel to the theoretical exposition, the student will carry out practices to consolidate the acquired theoretical contents. In this sense, the student will develop, deploy and debug software on an embedded system hardware, and he also will use a simulation environment of an embedded system that involves hardware and software aspects. To support the laboratory, the basic hardware of an embedded system will be used first, implementing simple drivers that allow serial communication, interrupts management and the basic timing services. Subsequently, more advanced services will be used, that facilitate the programming of functions over time on a cyclical executive. Finally, the student will perform practices on multitasking systems, becoming familiar with the use of timing services, mutual exclusion mechanisms for access to shared resources, and synchronous and asynchronous communication mechanisms.

## 1.b COURSE SUMMARY

This course shows the introduction to embedded systems. It allows to understand the behaviour of this type of systems and to provide skills for its design and implementation.

The course explains the metrics that driven the embedded systems development. It shows the different design levels and, for each level, the design alternatives and development techniques that make feasible to meet the requirement related to these metrics. The hardware-software co-design techniques will be introduced, focusing in those that facilitate the partitioning, so, after the system level definition, the hardware and software components can be implemented and validated separately. The embedded systems development process is studied. The architecture of an embedded system is defined and the life cycle that allows its specification, implementation and validation is explained.

In parallel with the theoretical lessons, a set of practices are proposed. These practices are focused on the software development on an embedded platform, and they let the student to work with industrial environments, including a simulator of a system on a chip, and a toolchain that integrates the compiling of the software projects, and also the deployment and debugging on the final target.

## 2. COMPETENCES

This course intends the student to develop the following competences and it has as objective a set of learning outcomes:

Generic Competences:

- TR5: Obtain interpersonal communication and teamwork skills for work in projects and work groups.
- TRU1 Capacity for analysis and synthesis.
- TRU2 Oral and written communication.
- TRU3 Ability to manage information.
- TRU4 autonomous learning.
- TRU5 Teamwork.

Professional Competences:

This course contributes to acquire the following competences of the professional profile of the Degree in Computer Engineering:

- CB5: Ability to design, synthesize, build and validate a computer using the necessary hardware and software elements.
- CB10: Ability to select suitable techniques, technologies and methodologies to develop computer-controlled systems.
- CB23: Ability to communicate the knowledge acquired and defend the proposed solution.
- CB25: Capacity for analysis and synthesis.
- CB26: Capacity for organization and planning.
- CB27: Ability to solve problems and take decisions.
- CB28: Ability to work in teams and skills in interpersonal relationships.
- CB29: Capacity for critical reasoning and autonomous learning.
- CB31: Capacity for basic expression and compression.

Learning Outcomes:

- RA1 Enumerate the specific characteristics of embedded systems and describe how they condition their development process.
- RA2 Enumerate the specific design metrics of embedded systems.
- RA3 Describe the role of the processing unit in the context of a complete embedded system with I/O and memory.
- RA4 Describe how the peripherals enable the interaction of the processing unit with the working environment of an embedded system.
- RA5 Identify the implementation alternatives of an embedded system and be able to select the most suitable one according to the design metrics marked as objective.
- RA6 Identify the design levels of an embedded system and describe the most significant techniques used to develop them.
- RA7 Define the concept of architecture of embedded systems and describe the process followed for its specification and validation.
- RA8 Be able to explain the design of an embedded system.
- RA9 Be able to program an embedded system and describe how high-level language becomes executable code.
- RA10 Be able to use compilers and developing environments suitable for embedded systems.

### 3. CONTENTS

Content Blocks	Number of hours
Introduction to embedded systems	10 hours
Design and validation techniques for embedded systems	18 hours
Embedded systems architecture and design elements	16 hours
Development process and environments for embedded systems	12 hours

### 4. METHODOLOGICAL STRATEGIES, MATERIALS AND RESOURCE

#### 4.1. Credit distribution

Theory sessions, Lab sessions and exams:	<ul style="list-style-type: none"> <li>Sessions in large groups: 28 hours (2 hours x 14 weeks)</li> <li>Sessions en small groups: 28 hours (2 hours x 14 weeks)</li> <li>Exams: 4 hours</li> </ul> Total: 60 hours
Autonomous student work:	<ul style="list-style-type: none"> <li>Information look up, work memory redaction, exercises, autonomous learning, preparation of exams and labs.</li> </ul> Total: 90 hours
Total	150 hours

#### 4.2. Methodological strategy, materials y didactic resources

Theory sessions (large groups)	<ul style="list-style-type: none"> <li>Exposition and/or concepts reviews</li> <li>Oral presentations and other activities</li> </ul>
Practical sessions (small groups)	<ul style="list-style-type: none"> <li>Exposition and/or review of practical concepts.</li> <li>Lab sessions: they are focused on consolidate the theoretical concepts previously explained, and on practising with software and hardware tools that assist in the understanding of the subject and in the acquisition of relevant skills for the professional future.</li> <li>Oral expositions and other activities</li> </ul>
Individual, group and online office hour	<ul style="list-style-type: none"> <li>Solving student questions</li> <li>Support to autonomous learning</li> </ul>

## 5. ASSESSMENT: Procedures and assessment and grading criteria

### 5.1. Procedures

The evaluation can be continuous or through a final evaluation. There are two opportunities in each case: ordinary and extraordinary.

- **Continuous Evaluation**

The competences acquired by the student are assessed during the whole learning process (NRPEA, Art. 3) by means a series of summative tests distributed throughout the course, which allow the student to acquire the competencies progressively.

This evaluation guarantees early feedback in the learning process of the student and allows teachers, coordinators and other elements of the Quality Assurance System to get a global traceability, with the possibility of intervention in case of being advised by certain indicators or situations.

The evaluation of the part related to the practices will be done at the end of its corresponding block.

- **Final Assessment**

According to the regulation approved by the Government Council on March 24, 2011, there is the possibility of taking a final evaluation test. As this regulation literally says: to qualify for the final evaluation, the student will have to request it in writing to the dean or director of the center in the first two weeks of the course term, explaining the reasons that prevent him from following the continuous assessment system..

### **Assessment Criteria**

In order to determine if the student has achieved the expected results, the following evaluation criteria will be taken into account:

CE1: The student has acquired knowledge about metrics and embedded systems design techniques.

CE2: The student has acquired knowledge about the development process of embedded systems.

CE3: The student shows the ability to apply and integrate the contents to problems, scenarios or case studies related to the subject.

CE4: The student shows capacity and initiative to develop software for embedded systems and use specific development environments of this type of systems.

CE5: The student demonstrates the ability to argue and reasoning about designs of embedded systems.

CE6: The student overcomes the tasks assigned.

CE7: The student shows interest in the contents and the work assigned.

CE8: The student demonstrates interest, clarity and rigor in the presentation of ideas and reasoning.

## 5.2. Assessment Instruments

The performance of the students will be assessed for their work, knowledge and skills acquired and the improvement of their learning process.

The continuous assessment instruments will consist of the fulfilment of continuous assessment activities proposed by the teacher for each of the topics.

The proposed activities, their contents and timing, will be communicated to the student during the presentation class or at the end of the corresponding content block. These activities comprise:

- PEI: Intermediate Assessment Tests (Pruebas de Evaluación Intermedia, PEI). During the course, several exams will be carried out, the content of which will focus on the topics covered or on expositions of group works. The contents and timing of these exams will be set during the first days of class
- PL: Lab Tests (Pruebas de Laboratorio, PL). These tests consist on expansions or variations of the practices carried out that must be resolved by the students.
- En: Deliverables (Entregable, En). An individual work and a group work will be delivered. The group work will be accompanied, in addition, by a work memory and it also will be presented in class.
- PEF: Final Assessment Test (Prueba de Evaluación Final, PEF). It consists of a written test, at the end of the teaching period, that requires the solving of a set of problems.



### Ordinary Call – Continuous assessment

Learning outcomes	Assessment Criteria	Assessment Instrument	Weight in final grade
RA1 – RA6	CE1	PEI1	10%
RA7	CE2	PEI2	10%
RA8	CE5, CE6, CE7, CE8	E1	7%
RA9-RA10	CE4,CE6,CE7	PL1	10%
RA9-RA10	CE4,CE6,CE7	PL2	10%
RA9-RA10	CE4,CE6,CE7	PL3	10%
RA9-RA10	CE4,CE6,CE7	PL4	10%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1-RA8	CE3, CE5, CE7	PEI3	10%

### Ordinary Call – Final Assessment

Learning outcomes	Assessment Criteria	Assessment Instrument	Weight in final grade
RA8	CE5, CE6, CE7, CE8	E1	7%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1 – RA10	CE1 – CE8	PEF	70%

### Extraordinary Call

Learning outcomes	Assessment Criteria	Assessment Instrument	Weight in final grade
RA8	CE5, CE6, CE7, CE8	E1	7%
RA1-RA8	CE5, CE6, CE7, CE8	E2	23%
RA1 – RA10	CE1 – CE8	PEF	70%

## 6. BIBLIOGRAPHY

### Basic Bibliography

- Frank Vahid and Tony Givargis. Embedded System Design: A Unified Hardware/Software Introduction. John Wiley & Sons; ISBN: 0471386782. 2002. The book describes basic concepts of embedded systems, the metrics used in their design and their implementation alternatives. The design and validation techniques of an embedded system are analysed. Some study examples are discussed and some high-level design solutions are presented based on concurrent tasks and the specification of their behaviour by state machines.

- David E. Simon. *An Embedded Software Primer*. Addison-Wesley, 1999. The book presents the hardware and software fundamentals needed to develop an embedded system. It describes the different techniques used in its resolution, deepening on those based on the use of a real-time operating system. The contents are suitable to have an overview of this type of systems and introduce basic notions about development tools and debugging.
- Arnold S. Berger. *Embedded Systems Design: An Introduction to Processes, Tools and Techniques*. CMP Books; 1st edition (December 15, 2010). The book addresses from the practical point of view the design of embedded systems. Focuses its contents on the technical characteristics of the different alternatives and tools, both hardware and software, that are used in the development and debugging of this type of systems.
- Tammy Noergaard. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*. Editorial Newnes. 2013. This book covers extensively the theme of embedded systems architecture. The topics covered are: hardware elements, operating system, assembly and high-level languages, networks and case studies that help to understand the embedded systems development process.

## Complementary Bibliography

- Jack Ganssle. *The Art of Designing Embedded Systems*. Newnes. 2008. Book that reflects the experience of a developer of embedded systems. It is especially useful once you have acquired the theoretical foundations and begin the first practical work of the student.
- Chris Rowen. *Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors*. Ed. Prentice Hall PTR. 2004. It is a book that covers in a unified way the design of SOC for configurable and extensible processors. Address issues such as cogeneration hardware / software, partitioning, communications and systems with several processors.
- Richard Zurawski. *Embedded Systems Handbook (Industrial Information Technology)*. 2009. It is a book that covers the theme of embedded systems in a broad way emphasizing new trends and technologies without forgetting the fundamentals. It covers all the aspects linked to embedded systems from both the software and the hardware perspective, and different scopes as real-time systems, security, system-on-chip and network-on-chip, networks, etc. It also includes examples and applications in the industrial field, automotive and smart sensors.
- Axel Jantsch. *Modeling Embedded Systems and SoC's: Concurrency and Time in Models of Computation (The Morgan Kaufmann Series in Systems on Silicon)*. Ed. Morgan Kaufmann. 2003. The book focuses on the presentation of modeling techniques for embedded systems. It covers typical such as state machines, Petri nets or synchronous computer models. It also provides real examples of the use of these techniques.
- David C. Black, Jack Donovan. *SystemC: From the Ground Up*. Ed. Springer. 2010. SystemC provides a set of extensions to C++ that allows the rapid development of hardware / software systems. This book focuses on practical aspects of language to model real systems. The wide assortment of examples and programs available for free allows the reader a quick introduction to SystemC.