

## DATOS GENERALES

<b>Curso académico</b>	Curso 2024/2025
<b>Tipo de curso</b>	Experto Universitario
<b>Número de créditos</b>	15,00 Créditos ECTS
<b>Matrícula</b>	0 euros (importe precio público)
<b>Requisitos de acceso</b>	<p>Los perfiles de ingreso recomendado se corresponderán a los perfiles formativos de los siguientes planes de estudio a nivel de grado: Ingeniería Electrónica de Telecomunicación, Ingeniería en Tecnologías y Servicios de Telecomunicación, Ingeniería de Tecnologías de Telecomunicación, Ingeniería Electrónica Industrial, Ingeniería Electrónica y Automática Industrial, Ingeniería Electroónica Industrial y Automática, Informática Industrial y Robótica, Ingeniería Informática, o grados, nacionales o extranjeros, con una alta afinidad a los aquí enumerados.</p> <p>En menor medida, podrían considerarse graduados en: Ingeniería de la Energía, Ingeniería Aeroespacial, Ingeniería Telemática, Ingeniería Robótica, Ingeniería Física, o títulos afines.</p> <p>En casos excepcionales, podrían considerarse perfiles científicos tradicionales como Grado en Física, Grado en Matemáticas o Grado en Ciencia de Datos.</p> <p>Se permitirá el acceso al estudiantado que le falte menos de un 10% de los créditos para terminar los estudios de grado, de forma condicionada a que se aprueben durante el mismo curso académico.</p> <p>Respecto al perfil personal del estudiante que mejor se adapta, corresponde a personas que quieran profundizar con rigor en los conocimientos y las habilidades que se requieren para especializarse en las áreas descritas para la orientación profesional. Así deberían tener una afinidad con la microelectrónica como motor en amplios sectores socioeconómicos, y en especial, como contribución esencial en sectores estratégicos industriales. Perfiles profesionales del sector con voluntad de asentar, reorientar o complementar su formación también serán adecuados para este título.</p>
<b>Modalidad</b>	Presencial
<b>Lugar de impartición</b>	ETSE
<b>Horario</b>	Tardes y sábado
<a href="#">Dirección</a>	
<b>Organizador</b>	Escola Tècnica Superior d'Enginyeria (ETSE-UV)
<b>Dirección</b>	<p>Abilio Candido Reig Escriva          Profesor/a Titular de Universidad. Departament d'Enginyeria Electrònica. Universitat de València          Jesús Soret Medel          Profesor/a Titular de Universidad. Departament d'Enginyeria Electrònica. Universitat de València          Francisco Javier Jiménez Marquina          Director de Ingeniería.MaxLinear</p>
<a href="#">Plazos</a>	
<b>Preinscripción al curso</b>	Hasta 13/12/2024
<b>Fecha inicio</b>	Enero 2025
<b>Fecha fin</b>	Junio 2025
<a href="#">Más información</a>	
<b>Teléfono</b>	961 603 000
<b>E-mail</b>	<a href="mailto:informacion@adeituv.es">informacion@adeituv.es</a>

## PROGRAMA

[Ingeniería de software para sistemas embebidos \(S1\)](#)

Tema 1: Introducción a los sistemas embebidos (1h)  
 ¿ Conceptos básicos de los sistemas embebidos. Características diferenciales  
 ¿ Tipos de sistemas embebidos  
 ¿ Aplicaciones de los sistemas embebidos  
 ¿ Seguridad de los sistemas embebidos  
 Tema 2: Lenguajes de programación para sistemas embebidos (1h)  
 ¿ Lenguajes de bajo nivel para sistemas embebidos. Ensamblador  
 ¿ Lenguajes de alto nivel para sistemas embebidos. C y herramientas de compilación.

- Â¿ Lenguajes de scripting Ãtiles
- Â¿ Interfaz entre diferentes lenguajes de programaciÃ³n
- Tema 3: Arquitecturas SW/HW para sistemas embebidos (1h)
  - Â¿ CPUs
  - Â¿ Memorias
  - Â¿ Hosted/Hostless
  - Â¿ Flash/Flashless
  - Â¿ SDK y API de clientes
  - Â¿ EVKs
- Tema 4: Desarrollo de software para sistemas embebidos (1h)
  - Â¿ Ciclo de vida del desarrollo de software para sistemas embebidos
  - Â¿ MetodologÃas ágiles de desarrollo de software para sistemas embebidos
  - Â¿ Herramientas de desarrollo de software para sistemas embebidos
- Tema 5: DescripciÃ³n de caracterÃsticas deseables del flujo de desarrollo (2h)
  - Â¿ Reducci3n del Time-to-market
  - Â¿ Defini3n de requisitos
    - o Funcionales
    - o Temporales
    - o Coste
    - o Etc.
  - Â¿ Arquitectura de Sistema
  - Â¿ Co-diseÃ±o HW/SW
    - o Plataformas de pruebas HW/SW
  - Â¿ Simulaciones
  - Â¿ Emuladores
  - Â¿ FPGAs
    - o Defini3n de interfaces HW/SW
    - o HW drivers
    - o Proceso de `bringup`
  - Â¿ Test Driven Development (TDD)
  - Â¿ Control de versiones (SCM)
  - Â¿ GestiÃ³n de tareas y errores en proyectos (Agile + Jira)
  - Â¿ Tests de Sistema
  - Â¿ Sistemas de IntegraciÃ³n Continua
  - Â¿ Test benches
  - Â¿ DocumentaciÃ³n
- Tema 6: DiseÃ±o de software para sistemas embebidos (3h)
  - Â¿ SW product line: HW and SW configurations
  - Â¿ Arquitectura SW de capas, componentes e interfaces
    - o DiseÃ±o para el re-uso
    - o Capas de abstracciÃ³n, HAL, OSAL.
    - o CÃ³digo independiente de aplicaciÃ³n
    - o CÃ³digo dependiente de aplicaciÃ³n
    - o Componentes de terceros y cuestiones legales
  - Â¿ DiseÃ±o para compatibilidad hacia atrÃas
  - Â¿ DiseÃ±o escalable
  - Â¿ Uso extensivo de tecnicas de programacion defensiva (assert)
    - o Chequeo de hard/soft deadlines
    - o Chequeo de problemas con la memoria (overwrites, stack overflows, etc)
  - Â¿ Sistemas operativos de tiempo real (RTOS)
    - o Configuraci3n
    - o Threads y prioridades
    - o Interrupciones
    - o Timers
    - o Stacks
    - o Primitivas de comunicacion
    - o Aplicaciones multiprocesador
    - o Utilidades de debug y analisis del rendimiento
  - o Problemas recurrentes:
    - Â¿ Thread preemption
    - Â¿ Tiempo de respuesta a interrupciones
    - Â¿ Inversion de prioridades
  - Â¿ Tipos de Componentes
    - o HOST SW: drivers, apps, libs
    - o Firmware: dev, prod, BIST, loader
    - o Herramientas y scripts
    - o Interface pÃblicos / privados
- Tema 7: Funcionalidades usuales en sistemas embebidos (2h)
  - Â¿ ConfiguraciÃ³n especifica del producto
    - o ProducciÃ³n
    - o Remota
  - Â¿ SW upgrades
  - Â¿ SDK para extender/cambiar funcionalidad

- Â¿ Flash FS
- Â¿ Interfaces
  - o JTAG
  - o UART
  - o SPI
  - o Consola de debug/operacion
- Â¿ Watchdog
- Â¿ Herramientas de debug
- Â¿ Memoria dinÃ¡mica

#### Tema 8: OptimizaciÃ³n de SW en Sistemas Embebidos (2h)

- Â¿ Memoria vs Rendimiento
- Â¿ Coste vs Facilidad en el desarrollo
- Â¿ Requisitos HW
- Â¿ OptimizaciÃ³n de Memoria
  - o CompactaciÃ³n de estructuras de datos
  - o Datos en memorias compartidas
  - o AsignaciÃ³n de memoria (linker script)
  - o ReutilizaciÃ³n de memoria
  - o ImplementaciÃ³n de subsets de librerÃ­as (matemÃ¡ticas, libcÃ¡)
- Â¿ OptimizaciÃ³n de rendimiento
  - o AritmÃ©tica de punto fijo
  - o Uso de ensamblador.
  - o Re-escritura de cÃ³digo para uso de HW especÃ­fico (DSP)
  - o DMA

#### Tema 9: AnÃ¡lisis de fallos en Sistemas Embebidos (2h)

- Â¿ Requisitos HW
- Â¿ JTAG
- Â¿ Unit Tests
- Â¿ Logs (serial, ethernet, filesÃ¡).
- Â¿ Debug Buffers
- Â¿ CPU Trace buffer
- Â¿ Memory dumps
- Â¿ AnÃ¡lisis de problemas en tiempo. Profiling
- Â¿ Parseo y visualizaciÃ³n de datos.
- Â¿ Debug en sistemas multiprocesador

#### LABORATORIO:

La duraciÃ³n de cada laboratorio serÃ¡ de 1,5h:

##### Laboratorio 1: DiseÃ±o de la arquitectura de un sistema embebido

- Â¿ Requisitos
- Â¿ Arquitectura y codiseÃ±o HW/SW
- Â¿ Ventajas y desventajas de las distintas opciones en cuanto a coste, facilidad de desarrollo, rendimiento, etc.

##### Laboratorio 2: MÃ¡quina virtualizada para desarrollo

- Â¿ VirtualizaciÃ³n de la plataforma de diseÃ±o (Docker)
- Â¿ Herramientas de desarrollo (GNU)
- Â¿ Sistema de control de versiones (GIT)
- Â¿ Unit tests (Google Test)
- Â¿ AnÃ¡lisis EstÃ¡tico de CÃ³digo (CppCheck)
- Â¿ Cobertura de testeo (Gcov)
- Â¿ AnÃ¡lisis dinÃ¡mico de CÃ³digo (Valgrind/Electric Fence)

##### Laboratorio 3: ImplementaciÃ³n de plataforma SW independiente de la aplicaciÃ³n sobre un simulador de RTOS

##### Laboratorio 4: ImplementaciÃ³n de utilidades para debug (I)

##### Laboratorio 5: ImplementaciÃ³n de utilidades para debug (I)

##### Laboratorio 6: ImplementaciÃ³n de Mock para simular un HW especÃ­fico

##### Laboratorio 7: ImplementaciÃ³n de una aplicaciÃ³n real time ejemplo sobre la plataforma anterior (I)

##### Laboratorio 8: ImplementaciÃ³n de una aplicaciÃ³n real time ejemplo sobre la plataforma anterior (II)

##### Laboratorio 9: Debug funcional de la aplicaciÃ³n

##### Laboratorio 10: AnÃ¡lisis de rendimiento y uso de memoria

#### [Control de calidad \(QA\) en sistemas embebidos \(S2\)](#)

#### TEORIA (15h)

##### Tema 0: Conceptos de validaciÃ³n y aseguramiento de la calidad (QA)

- Â¿ Modelo en V
- Â¿ Requisitos
- Â¿ Pruebas
- Â¿ DepuraciÃ³n y defectos
- Â¿ Conceptos de CI y de CD (Continuous Integration & Continuous Delivery)

##### Tema 1: DefiniciÃ³n de requisitos

- Â¿ Requisitos hardware
- Â¿ Requisitos software

- Â¿ Requisitos de sistema
- Â¿ Herramientas de gesti3n de requisitos
- Tema 2: Defini3n de prueba a partir de requisitos
- Â¿ Herramientas existentes vs propias
- Â¿ Tipos de pruebas (caja negra, caja blanca, funcionales, no funcionales...)
- Â¿ Defini3n de indicadores clave de rendimiento (KPI) a partir de requisitos
- Â¿ Defini3n de criterios de aceptaci3n Â¿pass/failÂ¿
- Tema 3: Introducci3n a la validaci3n de sistemas embebidos
- Â¿ Validaci3n de sistemas embebidos vs sistemas software:
- Â¿ Disponibilidad de recursos
- Â¿ Tiempos de ejecuci3n
- Â¿ Posibilidad de automatizaci3n
- Â¿ Defini3n de sistema bajo prueba
- Â¿ Interacci3n con el sistema bajo prueba
- Tema 4: Ejecuci3n y automatizaci3n de pruebas
- Â¿ Defini3n de bancos de prueba
- Â¿ Pruebas autom1ticas vs manuales: Por qu3 automatizar
- Â¿ Automatizaci3n:
- Â¿ Codificaci3n: Control de cambios
- Â¿ Instrumentaci3n
- Â¿ Ejecuci3n
- Â¿ Informes
- Â¿ Herramientas:
- Â¿ Lenguajes de programaci3n para pruebas autom1ticos
- Â¿ Herramientas para registrar progreso de pruebas
- Â¿ Frameworks de prueba
- Â¿ Recolecci3n organizada de datos para informe de defectos
- Tema 5: Defectos
- Â¿ Qu3 es un defecto y c3mo identificarlos
- Â¿ C3mo reportar un defecto correctamente
- Â¿ Herramientas de informe y trazabilidad de defectos
- Tema 6: CI para productos embebidos
- Â¿ Qu3 ofrece una herramienta de CI en la validaci3n de productos embebidos:
- Â¿ Defini3n de trabajos f1cilmente repetibles
- Â¿ Distribuci3n de los trabajos entre hardware disponible
- Â¿ Ventajas de tener un CI automatizado:
- Â¿ Optimizaci3n de tiempo maquina
- Â¿ Distribuci3n de recursos
- Â¿ Informes de resultados autom1ticos
- Â¿ Gesti3n de bancos de prueba:
- Â¿ Coexistencia entre automatizaci3n y uso manual de recursos
- Â¿ Interacci3n de los componentes de un entorno de CI
- Â¿ Herramientas habituales de CI:
- Â¿ Jenkins, Teamcity, Jira Workflow
- Tema 7: Obtenci3n y an1lisis de indicadores clave de rendimiento (KPI)
- Â¿ KPI relacionados con el producto:
- Â¿ Rendimiento
- Â¿ Estabilidad
- Â¿ Repetibilidad
- Â¿ KPI relacionados con el entorno de CI:
- Â¿ Cobertura de requisitos
- Â¿ Tiempo entre detecci3n de defecto y el arreglo
- Â¿ Uso de recursos
- Â¿

#### LABORATORIO (15h)

Laboratorio 1: Definir requisitos a partir de una descripci3n breve de un producto.  
 Laboratorio 2: Defini3n de pruebas a partir de requisitos. Plan de prueba.  
 Laboratorio 3: Ejecuci3n manual de un plan de prueba. Pruebas exploratorias.  
 Laboratorio 4: Automatizaci3n de las pruebas definidas.  
 Laboratorio 5: Identificaci3n y reporte de defectos a partir de los resultados de las pruebas.  
 Laboratorio 6: Creaci3n de un entorno CI completo.  
 Laboratorio 7: Defini3n de KPI a partir de requisitos y resultados de las pruebas.

#### Sistemas operativos en tiempo real (S3)

##### TEORIA (10h)

- Â¿ Conceptos generales de arquitectura de computadores
  - o La arquitectura (ISA: Instruction Set Architecture)
- Â¿ Diferencias entre CISC y RISC
  - o La CPU (Unidad Central de Proceso)
- Â¿ Unidad de control
- Â¿ ALU (Unidad Aritm3tica-L3gica)
- Â¿ Buses
- Â¿ Cache de instrucciones

- Â¿ Pipelining
- Â¿ Registros
- o Cores
- o Memoria (ROM, RAM)
- Â¿ Memoria cachÃ©
- o PerifÃ©ricos de entrada/salida
- Â¿ Interrupciones
- o Microcontroladores

- Â¿ Conceptos generales de software
- o Lenguajes de programaci3n
- Â¿ Relaci3n entre juego de instrucciones, c3digo objeto y lenguaje ensamblador
- o Tipos de archivo ejecutable
- o Compiladores
- o El linker o enlazador
- Â¿ Linker scripts y scatter files
- o Memoria estÃtica y dinÃmica
- Â¿ El stack y el heap
- o El cargador de arranque o bootloader

- Â¿ Tiempo real y conceptos de RTOS
- o RTOS vs GPOS
- Â¿ Latency
- o RTOS vs Â¿bare-metalÂ¿ (Â¿super loopÂ¿)
- o Kernel / Scheduler
- Â¿ preemptive time-slicing
- Â¿ cooperative time-slicing
- Â¿ Tick, Idle task, Ticless idle
- o Interrupciones hardware y software
- o Tareas e hilos
- Â¿ thread stack
- Â¿ prioridades
- o Paralelismo y concurrencia
- Â¿ Secci3n CrÃtica
- Â¿ SemÃforos, mutex y operaciones at3micas
- Â¿ Colas de mensajes o otros mecanismos
- Â¿ Condition variables
- Â¿ Problemas clÃsicos
- Â¿ productor/consumidor
- Â¿ condici3n de carrera
- Â¿ inversi3n de prioridades
- o Procesadores multi-core
- Â¿ SMP (Symetric Multi-Processing)
- Â¿ AMP (Asymmetric Multi-Processing)
- o Comunicaci3n entre tareas

- Â¿ AnÃlisis de los RTOS mÃs com3nmente usados
- o FreeRTOS
- o MicroC/OS-II (uCOS)
- o ThreadX
- o RTEMS
- o Zephyr
- o VxWorks

- Â¿ RTOS en FPGAs
- o Procesadores softcore. Ejemplos
- Â¿ Xilinx Microblaze
- Â¿ Tensilica Xtensa
- Â¿ Implementaciones RISC-V (Mi-V RV32, NEORV32, FEMTORV32)

#### LABORATORIO (20h)

Proyecto de un pequeÃ±o sistema controlado por RTOS:

Control de un ventilador accionado por motor DC (PWM) de forma automÃtica mediante sensor de temperatura y controlable mediante comandos por Ethernet.

#### [Proyecto industrial en microelectr3nica](#)

Los contenidos del "Proyecto Industrial en Microelectr3nica" serÃn diferentes dependiendo de los objetivos concretos del proyecto a realizar. Pueden ser objeto de tema de aquellos que sean propios de los estudios del tÃtulo. En particular, se podrÃn proyectar toda clase de sistemas y dispositivos microelectr3nicos por cuantos procedimientos permita realizar la ingenierÃa actual. Tambi3n podrÃ ser objeto del Proyecto Industrial en Microelectr3nica los trabajos de investigaci3n y desarrollo, y el modelado te3rico o num3rico de los dispositivos, circuitos o sistemas microelectr3nicos. Se podrÃn considerar asimismo los estudios relacionados con los contenidos del tÃtulo relativos a equipos, fÃbricas, instalaciones, servicios o su planificaci3n, gesti3n o explotaci3n.

## PROFESORADO

### María Teresa Bacete Castellá<sup>3</sup>

Site Director. Maxlinear

### Javier Calpe Maravilla

Profesor/a Titular de Universidad. Departament d'Enginyeria Electrònica. Universitat de València

### Miguel Chanca Martán

IC Lead. Robert Bosch

### Francisco Escuder Roberto

Responsable de software. MaxLinear Hispania, S.L.

### Luis Alfonso Espinosa Ortega

Ingeniero de Firmware. Analog Devices, S.L.U

### José Luis García Navas

Ingeniero de calidad de software. MaxLinear Hispania, S.L.

### Francisco Javier Jiménez Marquina

Director de Ingeniería. MaxLinear

### Fernando Pardo Carpio

Catedrático/a de Universidad. Departament d'Informàtica. Universitat de València

### Abilio Candido Reig Escrivá

Profesor/a Titular de Universidad. Departament d'Enginyeria Electrònica. Universitat de València

### Antonio Jesús Rubio Salcedo

Ingeniero de Verificación de Diseño. Analog Devices, S.L.U

### Rafael Serrano-Gotarredona

Director General. ams-OSRAM

### José Soret Medel

Profesor/a Titular de Universidad. Departament d'Enginyeria Electrònica. Universitat de València

## OBJETIVOS

Las salidas profesionales que tiene el curso son:

Las salidas profesionales previstas están estrechamente vinculadas a los perfiles más demandados en este ámbito, entre los que se podrán destacar:

Diseñadores de sistema (arquitectura del chip, partición Hw/Sw, algorítmica, DSP...); desarrolladores de flujo de diseño digital; diseñadores de procesadores y memorias embebidas; ingenieros de diseño físico (P&R); diseñadores de dispositivos (layout); ingenieros de estándares; ingenieros de firmware; ingenieros de software; ingenieros de test; ingenieros de calidad y fiabilidad; ingenieros de packaging (chiplet...); tecnólogos; diseñadores de dispositivos optoelectrónicos; diseñadores de PCBs; ...

En los últimos tiempos, las empresas del sector de la microelectrónica y los semiconductores han visto como sus ofertas de trabajo quedaban frecuentemente vacantes por falta de personas con la formación demandada para puestos tan específicos. Por tanto, el objetivo fundamental de este título es ofrecer al ecosistema VaSiC perfiles profesionales con las competencias necesarias para ser directamente incorporados en sus plantillas.

## METODOLOGÍA

El centro responsable del Título de Experto en Sistemas-en-Chip (SoC) es la Escola Tècnica Superior d'Enginyeria (ETSE), y se impartirá de forma presencial, en castellano, en horario de viernes y sábado. La matrícula será gratuita. Este título propio se desarrolla en 15 ECTS, entre los que se incluyen 6 de proyecto industrial en microelectrónica con empresa, fundamentalmente de VaSiC. Los contenidos de este título serán impartidos en su práctica totalidad por profesorado especialista de las empresas de VaSiC. Todas las asignaturas, tanto las fundamentales como las optativas, incluirán contenidos teóricos y contenidos prácticos. El alumnado dispondrá de ordenadores adecuados con las herramientas utilizadas en el estándar industrial para el

desarrollo y análisis de los sistemas que se propongan: Cadence, Synopsys, desarrollo ARM, Matlab... También dispondrán del instrumental necesario para el test y caracterización de los dispositivos que se estudien. Se prevén prácticas en sala blanca.