

DATOS GENERALES

Curso académico

Tipo de curso	Experto Universitario
Número de créditos	15,00 Créditos ECTS
Matrícula	0 euros (importe precio público)
Requisitos de acceso	<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>

Modalidad	Presencial
Lugar de impartición	ETSE
Horario	Tardes y sábado

Dirección

Organizador	Escola Tècnica Superior d'Enginyeria (ETSE-UV)
Dirección	<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>

Plazos

Preinscripción al curso	Hasta 13/12/2024
Fecha inicio	Enero 2025
Fecha fin	Junio 2025

Más información

Teléfono	961 603 000
E-mail	informacion@adeituv.es

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.

- ¿ Lenguajes de scripting útiles
- ¿ Interfaz entre diferentes lenguajes de programación
- Tema 3: Arquitecturas SW/HW para sistemas embebidos (1h)
 - ¿ CPUs
 - ¿ Memorias
 - ¿ Hosted/Hostless
 - ¿ Flash/Flashless
 - ¿ SDK y API de clientes
 - ¿ EVKs
- Tema 4: Desarrollo de software para sistemas embebidos (1h)
 - ¿ Ciclo de vida del desarrollo de software para sistemas embebidos
 - ¿ Metodologías ágiles de desarrollo de software para sistemas embebidos
 - ¿ Herramientas de desarrollo de software para sistemas embebidos
- Tema 5: Descripción de características deseables del flujo de desarrollo (2h)
 - ¿ Reducción del Time-to-market
 - ¿ Definición de requisitos
 - o Funcionales
 - o Temporales
 - o Coste
 - o Etc.
 - ¿ Arquitectura de Sistema
 - ¿ Co-diseño HW/SW
 - o Plataformas de pruebas HW/SW
 - ¿ Simulaciones
 - ¿ Emuladores
 - ¿ FPGAs
 - o Definición de interfaces HW/SW
 - o HW drivers
 - o Proceso de `bringup`
 - ¿ Test Driven Development (TDD)
 - ¿ Control de versiones (SCM)
 - ¿ Gestión de tareas y errores en proyectos (Agile + Jira)
 - ¿ Tests de Sistema
 - ¿ Sistemas de Integración Continua
 - ¿ Test benches
 - ¿ Documentación
- Tema 6: Diseño de software para sistemas embebidos (3h)
 - ¿ SW product line: HW and SW configurations
 - ¿ Arquitectura SW de capas, componentes e interfaces
 - o Diseño para el re-uso
 - o Capas de abstracción, HAL, OSAL.
 - o Código independiente de aplicación
 - o Código dependiente de aplicación
 - o Componentes de terceros y cuestiones legales
 - ¿ Diseño para compatibilidad hacia atrás
 - ¿ Diseño escalable
 - ¿ Uso extensivo de técnicas de programación defensiva (assert)
 - o Chequeo de hard/soft deadlines
 - o Chequeo de problemas con la memoria (overwrites, stack overflows, etc)
 - ¿ Sistemas operativos de tiempo real (RTOS)
 - o Configuración
 - o Threads y prioridades
 - o Interrupciones
 - o Timers
 - o Stacks
 - o Primitivas de comunicación
 - o Aplicaciones multiprocesador
 - o Utilidades de debug y análisis del rendimiento
 - o Problemas recurrentes:
 - ¿ Thread preemption
 - ¿ Tiempo de respuesta a interrupciones
 - ¿ Inversión de prioridades
 - ¿ Tipos de Componentes
 - o HOST SW: drivers, apps, libs
 - o Firmware: dev, prod, BIST, loader
 - o Herramientas y scripts
 - o Interfaces públicos / privados
- Tema 7: Funcionalidades usuales en sistemas embebidos (2h)
 - ¿ Configuración específica del producto
 - o Producción
 - o Remota
 - ¿ SW upgrades
 - ¿ SDK para extender/cambiar funcionalidad

- ¿ Flash FS
- ¿ Interfaces
 - o JTAG
 - o UART
 - o SPI
 - o Consola de debug/operacion
- ¿ Watchdog
- ¿ Herramientas de debug
- ¿ Memoria dinámica

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

- ¿ Memoria vs Rendimiento
- ¿ Coste vs Facilidad en el desarrollo
- ¿ Requisitos HW
- ¿ Optimización de Memoria
 - o Compactación de estructuras de datos
 - o Datos en memorias compartidas
 - o Asignación de memoria (linker script)
 - o Reutilización de memoria
 - o Implementación de subsets de librerías (matemáticas, libc)
- ¿ Optimización de rendimiento
 - o Aritmética de punto fijo
 - o Uso de ensamblador.
 - o Re-escritura de código para uso de HW específico (DSP)
 - o DMA

Tema 9: Análisis de fallos en Sistemas Embebidos (2h)

- ¿ Requisitos HW
- ¿ JTAG
- ¿ Unit Tests
- ¿ Logs (serial, ethernet, files).
- ¿ Debug Buffers
- ¿ CPU Trace buffer
- ¿ Memory dumps
- ¿ Análisis de problemas en tiempo. Profiling
- ¿ Parseo y visualización de datos.
- ¿ Debug en sistemas multiprocesador

LABORATORIO:

La duración de cada laboratorio será de 1,5h:

Laboratorio 1: Diseño de la arquitectura de un sistema embebido

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

Laboratorio 2: Máquina virtualizada para desarrollo

- ¿ Virtualización de la plataforma de diseño (Docker)
- ¿ Herramientas de desarrollo (GNU)
- ¿ Sistema de control de versiones (GIT)
- ¿ Unit tests (Google Test)
- ¿ Análisis Estático de Código (CppCheck)
- ¿ Cobertura de testeo (Gcov)
- ¿ Análisis dinámico de Código (Valgrind/Electric Fence)

Laboratorio 3: Implementación de plataforma SW independiente de la aplicación sobre un simulador de RTOS

Laboratorio 4: Implementación de utilidades para debug (I)

Laboratorio 5: Implementación de utilidades para debug (I)

Laboratorio 6: Implementación de Mock para simular un HW específico

Laboratorio 7: Implementación de una aplicación real time ejemplo sobre la plataforma anterior (I)

Laboratorio 8: Implementación de una aplicación real time ejemplo sobre la plataforma anterior (II)

Laboratorio 9: Debug funcional de la aplicación

Laboratorio 10: Análisis de rendimiento y uso de memoria

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

TEORIA (15h)

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

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

Tema 1: Definición de requisitos

- ¿ Requisitos hardware
- ¿ Requisitos software

- ¿ Requisitos de sistema
- ¿ Herramientas de gestión de requisitos
- Tema 2: Definición de prueba a partir de requisitos
- ¿ Herramientas existentes vs propias
- ¿ Tipos de pruebas (caja negra, caja blanca, funcionales, no funcionales...)
- ¿ Definición de indicadores clave de rendimiento (KPI) a partir de requisitos
- ¿ Definición de criterios de aceptación ¿pass/fail¿
- Tema 3: Introducción a la validación de sistemas embebidos
- ¿ Validación de sistemas embebidos vs sistemas software:
- ¿ Disponibilidad de recursos
- ¿ Tiempos de ejecución
- ¿ Posibilidad de automatización
- ¿ Definición de sistema bajo prueba
- ¿ Interacción con el sistema bajo prueba
- Tema 4: Ejecución y automatización de pruebas
- ¿ Definición de bancos de prueba
- ¿ Pruebas automáticas vs manuales: Por qué automatizar
- ¿ Automatización:
- ¿ Codificación: Control de cambios
- ¿ Instrumentación
- ¿ Ejecución
- ¿ Informes
- ¿ Herramientas:
- ¿ Lenguajes de programación para pruebas automáticos
- ¿ Herramientas para registrar progreso de pruebas
- ¿ Frameworks de prueba
- ¿ Recolección organizada de datos para informe de defectos
- Tema 5: Defectos
- ¿ Qué es un defecto y cómo identificarlos
- ¿ Cómo reportar un defecto correctamente
- ¿ Herramientas de informe y trazabilidad de defectos
- Tema 6: CI para productos embebidos
- ¿ Qué ofrece una herramienta de CI en la validación de productos embebidos:
- ¿ Definición de trabajos fácilmente repetibles
- ¿ Distribución de los trabajos entre hardware disponible
- ¿ Ventajas de tener un CI automatizado:
- ¿ Optimización de tiempo máquina
- ¿ Distribución de recursos
- ¿ Informes de resultados automáticos
- ¿ Gestión de bancos de prueba:
- ¿ Coexistencia entre automatización y uso manual de recursos
- ¿ Interacción de los componentes de un entorno de CI
- ¿ Herramientas habituales de CI:
- ¿ Jenkins, Teamcity, Jira Workflow
- Tema 7: Obtención y análisis de indicadores clave de rendimiento (KPI)
- ¿ KPI relacionados con el producto:
- ¿ Rendimiento
- ¿ Estabilidad
- ¿ Repetibilidad
- ¿ KPI relacionados con el entorno de CI:
- ¿ Cobertura de requisitos
- ¿ Tiempo entre detección de defecto y el arreglo
- ¿ Uso de recursos
- ¿

LABORATORIO (15h)

Laboratorio 1: Definir requisitos a partir de una descripción breve de un producto.
 Laboratorio 2: Definición de pruebas a partir de requisitos. Plan de prueba.
 Laboratorio 3: Ejecución manual de un plan de prueba. Pruebas exploratorias.
 Laboratorio 4: Automatización de las pruebas definidas.
 Laboratorio 5: Identificación y reporte de defectos a partir de los resultados de las pruebas.
 Laboratorio 6: Creación de un entorno CI completo.
 Laboratorio 7: Definición de KPI a partir de requisitos y resultados de las pruebas.

[Sistemas operativos en tiempo real \(S3\)](#)

TEORIA (10h)

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

- ¿ Pipelining
- ¿ Registros
- o Cores
- o Memoria (ROM, RAM)
- ¿ Memoria caché
- o Periféricos de entrada/salida
- ¿ Interrupciones
- o Microcontroladores

- ¿ Conceptos generales de software
- o Lenguajes de programación
- ¿ Relación entre juego de instrucciones, código objeto y lenguaje ensamblador
- o Tipos de archivo ejecutable
- o Compiladores
- o El linker o enlazador
- ¿ Linker scripts y scatter files
- o Memoria estática y dinámica
- ¿ El stack y el heap
- o El cargador de arranque o bootloader

- ¿ Tiempo real y conceptos de RTOS
- o RTOS vs GPOS
- ¿ Latency
- o RTOS vs ¿bare-metal¿ (¿super loop¿)
- o Kernel / Scheduler
- ¿ preemptive time-slicing
- ¿ cooperative time-slicing
- ¿ Tick, Idle task, Ticless idle
- o Interrupciones hardware y software
- o Tareas e hilos
- ¿ thread stack
- ¿ prioridades
- o Paralelismo y concurrencia
- ¿ Sección Crítica
- ¿ Semáforos, mutex y operaciones atómicas
- ¿ Colas de mensajes o otros mecanismos
- ¿ Condition variables
- ¿ Problemas clásicos
- ¿ productor/consumidor
- ¿ condición de carrera
- ¿ inversión de prioridades
- o Procesadores multi-core
- ¿ SMP (Symetric Multi-Processing)
- ¿ AMP (Asymmetric Multi-Processing)
- o Comunicación entre tareas

- ¿ Análisis de los RTOS más comúnmente usados
- o FreeRTOS
- o MicroC/OS-II (uCOS)
- o ThreadX
- o RTEMS
- o Zephyr
- o VxWorks

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

LABORATORIO (20h)

Proyecto de un pequeño sistema controlado por RTOS:

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

[Proyecto industrial en microelectrónica](#)

Los contenidos del "Proyecto Industrial en Microelectrónica" serán diferentes dependiendo de los objetivos concretos del proyecto a realizar. Pueden ser objeto de tema de aquellos que sean propios de los estudios del título. En particular, se podrán proyectar toda clase de sistemas y dispositivos microelectrónicos por cuantos procedimientos permita realizar la ingeniería actual. También podrá ser objeto del Proyecto Industrial en Microelectrónica los trabajos de investigación y desarrollo, y el modelado teórico o numérico de los dispositivos, circuitos o sistemas microelectrónicos. Se podrán considerar asimismo los estudios relacionados con los contenidos del título relativos a equipos, fábricas, instalaciones, servicios o su planificación, gestión o explotación.

PROFESORADO

María Teresa Bacete Castelló

Site Director. Maxlinear

Javier Calpe Maravilla

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

Miguel Chanca Martín

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 Escriva

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

Jesús 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 podrían 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 (chiplat...); 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.