



## INGENIERÍA DEL SOFTWARE APLICADO A LOS SISTEMAS EMBEBIDOS\_OL\_18\_19

Al terminar la actividad el asistente podrá (descripción de objetivos de la actividad):

Aprender a diseñar metodologías para la creación de software en el área de los sistemas embebidos de una forma eficiente y robusta. Analizar y diseñar la lógica de control de aplicaciones industriales, para ello se utilizan las Maquinas de Estados Finitas. Para la implementación del software se realiza en el lenguaje "C" aplicado a los procesadores digitales de la Familia C2000 de Texas.

Conocimientos previos necesarios:

Conocimientos básicos microcontroladores y de programación C ansi.

Acción formativa dirigida a:

Ingenieros Telecomunicaciones y de la rama industrial (Grados, Superiores y Técnicos), alumnos de Ciclos Formativos y Formación Profesional, personal de Oficinas Técnicas, Instaladores eléctricos, profesionales que deseen una formación complementaria y aplicada en el ámbito de la optimización de la programación en el lenguaje "C" ANSI aplicado a los sistemas embebidos. Máster de Electrónica, eléctrica y de mecánica

## Temas a desarrollar:

### 1. Introducción a la Ingeniería del Software en sistemas Embebidos.

- Introducción a los Sistemas Embebidos.
- Crisis del Software.
- Ingeniería del Software. Introducción.
- Necesidades en los Sistemas Embebidos.
- Diseño de programas para Sistemas Embebidos.
- Proyectos de Software en los Sistemas Embebidos.
- Modelos de Computación.
- o Técnicas de programación en Hardware in the Loop (HIL).
- o Lenguaje de modelado Unificado (UML). Introducción.
- o Desarrollo basado en Modelos. StateFlow (Simulink).

### 2. Sistemas embebidos.

- Sistemas en Tiempo Real.
- o Clasificación.
- o Características.
- Ejemplo de un sistema en Tiempo Real.
- Sistemas Embebidos de propósito general.
- ¿Qué es un sistema empotrado?. Diferencia entre microcontroladores y DSPs.
- Compañías más relevantes en los sistemas embebidos. Familia C2000 de Texas.
- Diseño de Sistemas Embebidos.
- o Fases de Diseño en sistemas digitales. Especificaciones.
- o Diseño del Hardware.
- o Diseño del Software.

### 3. Lenguaje C ANSI para Sistemas embebidos. Base.

- ¿Qué es un lenguaje de programación?. Clasificación.
- Introducción a lenguaje C Ansi.
- o Tipos de Datos: Enteros y Reales.
- o Modificadores. Auto, extern, register, static y typedef.
- o Casting.
- o Operadores Aritméticos, Lógicos y manejo de bits.
- o Declaración de funciones.
- o Estructuras de control. If, while, for switch,....

### 4. Lenguaje C ANSI para Sistemas embebidos. Estilo de Programación.

- Objetivo del estilo en la programación.
- Introducción al estilo en la programación con lenguaje C Ansi.
- ¿Dónde aplicar el estilo de programación del código C?.
- Comentarios en la programación del código C Ansi.
- Estructura de un programa en código C Ansi.
- Estilo en las sentencias del código C Ansi.
- Archivos en el desarrollo del código C Ansi.
- Ejemplo del Estilo de Programa en C-Ansi.

### 5. Lenguaje C ANSI para Sistemas embebidos. Avanzado.

- Ámbito o alcance (scope) de las variables.
- Duración de las variables en la programación con lenguaje C Ansi. Estáticas y dinámicas.
- Modificadores de las variables en la programación del código C Ansi. Variables externas, register, volatile y const.
- Ejemplos.
- Variables Tipo Enumerados en código C Ansi. Ejemplos.
- Macros aplicadas al código C Ansi. Ejemplos.
- Preprocesado en el código C Ansi. Ejemplos.
- Optimizar el compilador. Ejemplos.
- Utilización optima de estructuras y funciones. Ejemplos.
- Punteros a variables y a funciones en el código C Ansi. Ejemplos.
- Definición de nuevos datos mediante "typedef" en el código C Ansi. Ejemplos.
- Ahorro de memoria mediante la instrucción "unión" en el código C Ansi. Ejemplos.
- Estructura de datos más utilizados. Colas, Pilas y Listas. Ejemplos.

### 6. Interrupciones y Timers en los Sistemas embebidos.

- Introducción a las subrutinas anidadas.
- Tratamiento de interrupciones y Timers en los sistemas embebidos.
- ¿Cómo se gestiona una interrupción en los sistemas embebidos?.
- Latencia de una interrupción en los sistemas embebidos.
- Ventajas de las interrupciones en los sistemas embebidos. Ejemplos.
- Manejo de funciones exclusivas para los Timers.
- Ejemplo del manejo de los Timers en un sistema embebido de la familia C2000 de Texas (TMS320F28027).

### 7. Lenguaje C ANSI para Sistemas embebidos. Objetos.

- Objetivos de la Programación para sistemas embebidos.
- Objetivos de la Programación Orientada a Objetos (POO) para sistemas embebidos.
- Introducción de la Programación Orientada a Objetos para sistemas embebidos. Ejemplo de un la clase de un coche.

## Temas a desarrollar:

- Introducción de la Programación Orientada a Objetos para sistemas embebidos. Ejemplo de un la clase de un coche.
- Terminología de la Programación Orientada a Objetos: Abstracción, encapsulamiento, herencia, poliformismo, constructor.
- Resumen comparativo entre la POO y la Procedural.
- Abordar la POO mediante el lenguaje C Ansi. Ejemplos.
- Ejemplos de cómo implementar una Clase de un filtro paso-bajo en el lenguaje C Ansi.
- Diseño de una aplicación de control de unos leds con unos tiempos manejados por un Timer de la CPU mediante la Programación Orientada a Objetos (POO) para sistemas embebidos, en concreto para el TMS320F28027 de la familia C2000 de Texas Instruments.
- Diseño de una aplicación de un sistemas de medidas local mediante la Programación Orientada a Objetos (POO) para sistemas embebidos, en concreto para el TMS320F28027 de la familia C2000 de Texas Instruments.

### 8. Lenguaje C ANSI para Sistemas embebidos. Optimización.

- Objetivos de la Optimización en la Programación para sistemas embebidos.
- ¿Cómo escribir mejor el código C Ansi para sistemas embebidos?.
- Técnicas de Optimización manual. Ejemplos.
- Mejoras de la Optimización del código mediante:
  - o Declaración de variables. Globales y locales
  - o Manejo de Flags para las condiciones.
  - o Reutilización del código.
- Recomendaciones en programación en C-Ansi.
- Optimización desde el compilador de C Ansi.
- Evaluación de prestaciones del software sobre la CPU.
  - o Definiciones: Tiempo de respuesta, tiempo de CPU y productividad.
  - o Rendimiento de la CPU.
  - o Comparativa entre dos CPU's.
  - o Ley de Amdahl.
- Componentes para medir el tiempo de ejecución de un programa sobre una CPU.
- Medidas de prestaciones sobre la CPU: MIPS y MFLOPs.
- Programas para la evaluación de las prestaciones sobre la CPU. Benchmark.
- Medidas Benchmark: Ocupación de espacio y tiempo de ejecución. Ejemplos.
- Como medir nuestra velocidad de nuestro código. Ejemplos.

### 9. Flujogramas. Representación del código en los Sistemas Embebidos.

- Introducción a las fases de un proyecto Software: visión de los diagramas de flujo.
- Definición de un flujograma o diagrama de flujo. Características principales
- Utilización de los flujogramas o diagrama de flujos.
- Simbología más utilizada (ANSI) en los flujogramas.
- Diseño y Ejemplos de Flujogramas.
- Ejercicio de un diseño de un Flujograma para la lectura de canales analógicos.
- Ejemplo de aplicación de Pseudocodigos.
- Flujo de Datos. Modelo de datos y procesos. Definición y ejemplos.

### 10. Documentación en el diseño del Software en los Sistemas Embebidos.

- Introducción a la documentación del Software.
- Ventajas de aplicar la documentación del Software.
- Tipos de documentación en el Software en los sistemas embebidos.
- Definición de las Especificaciones funcionales.
- Herramientas de Documentación del Software:
  - o Proceso de producción de documentos.
  - o Doxygen (herramienta gratuita). Comandos de documentación.
  - o Ejemplos de Documentación con el Code Composer Studio y Doxygen.
- Normativas del Software.

### 11. Máquinas de Estados Finitas aplicadas a los Sistemas Embebidos.

- Introducción a los Sistemas Reactivos.
- Metodología de programación de la Lógica de Control:
  - o Código Espagueti.
  - o SuperLazo.
  - o Plano Secundario-Principal.
  - o Máquinas de Estados Finitas (MEF).
  - o Planificador de tareas.
- Análisis con ejemplos, aplicando las metodologías de la lógica de control.
- Ejemplo de la metodología de programación del control de una Lavadora.
- Teoría de Automatas. Máquinas de Estados (MEF). Elementos de una MEF.
- Tipos de MEF:
  - o Máquina de Moore.
  - o Máquina de Mealy.
- Diseño de una MEF:
  - o Diagramas de Estados.
  - o Tabla de transiciones.
- Eventos. Generación y su tratamiento.
- Comparativa entre MEF y Flujogramas.

## Temas a desarrollar:

- Comparativa entre MEF y Flujogramas.
- Diseño e implementación de una MEF simple mediante el lenguaje C Ansi.
- Ejemplo de un diseño de una MEF de la secuencia de Encendido/Apagado de un sistema electrónico.
- Máquinas de Estado del tipo Detectoras: Definición y Ejemplos.
- Diseño completo de una MEF para el control de un montacargas (de dos plantas).
- Diseño para el control simple de un ascensor (con dos pisos y puertas) mediante una MEF.
- ¿Como hacer una plantilla de código general para la implementación de una MEF ?. Generación de código.
- Ejemplo completo del diseño de una Máquina de Estado del tipo Detectora manejando el puerto serie de un DSC TMS320F28027.

### 12. StateCharts. Representación del código en los Sistemas Embebidos.

- Introducción a los StateCharts.
- StateCharts de Harel. Características principales.
- Diseño basado en Modelos (MDD).
- Metodología de los StateCharts.
- Representación de un Estado..
- Sintaxis básica de un StateChart.
- Pseudoestados. Definición.
- Máquinas de Estados Extendidas:
  - o Pseudoestados.
  - o Concurrencia.
  - o Jerarquías.
  - o Ejemplos de Máquinas de Estados Extendidas.
- ¿Cómo codificar un Statechart?. Ejemplo.
- Máquinas de Estado Jerárquicas. Ejemplo: Implementación en lenguaje C Ansi.
- Metodología del diseño de una MEF. Diseño de capas de la MEF:
  - o Capa General.
  - o Capa requisitos.
  - o Capa Diseño.
  - o Capa Implementación.
  - o Capa Test o Depuración.
- Ejemplo de un modelado de una aplicación de una MEF. Control de Velocidad Crucero en un coche. Análisis y diseño.
- Conclusiones de los StateCharts.
- Herramientas de apoyo para el diseño de las MEF:
  - o StateFlow (Simulink).
  - o VisualState.

### 13. StateFlow MATLAB.

- Introducción a los StateFlow.
- Características principales del StateFlow.
- Elementos que componen el StateFlow.
- Ejemplo de diagramas de estados con StateFlow.
- Ejercicio de un control de temperatura mediante StateFlow.
- Aplicar tablas de verdad en la lógica de control mediante el StateFlow.
- Generación de informe de un modelo mediante el StateFlow.
- Ejercicio de Modelar la lógica de control de un reconocedor para una secuencia determinada mediante StateFlow.
- Ejemplo completo del diseño de la lógica de control y simulación de un inversor monofasico de conexión a red a partir de energía solar fotovoltaica,
  - o Procedimiento para el diseño.
  - o Definición de Estados de control
  - o Control Antiislanding.
  - o Control del bus de continua.
  - o Control de errores de Hardware y Software.
  - o Diseño maquina de estados del inversor.
- ¿Cómo codificar un Statechart?. Ejemplo.
- Máquinas de Estado Jerárquicas. Ejemplo: Implementación en lenguaje C Ansi.

### 14. Sistemas Operativos en los Sistemas Embebidos (I).

- Introducción a los procesos (threads) dentro del entorno de los sistemas embebidos.
- Procesos y concurrencia.
- Análisis de la problemática de añadir funciones a un programa. Bajo sistemas embebidos.
- Comunicación y sincronización de procesos o tareas.
- Recursos compartidos. Exclusión Mutua.
- Introducción a los sistemas operativos para sistemas embebidos.
- Diseño básico con Sistemas Operativos en Tiempo Real (RTOS).
- Tareas. Definición y criterios de selección.
- Servicio Básicos de los RTOS.
- Planificador de tareas (scheduling). Tipos existentes:
  - o Cooperativos o No expropiativos.
  - o Expropiativos.
- Introducción a la programación dirigida por eventos. Ejemplos.
- Planificador Round-Robin. Características.

## Temas a desarrollar:

- Planificador Round-Robin. Características.
- Ejemplo e implementación en lenguaje C ANSI.
- Planificador Round-Robin con Interrupciones. Ejemplos.
- Ejemplo de un Puente (bridge) de comunicaciones mediante el planificador Round-Robin.
- Comparativa de Round Robin Secuencial & Interrupciones.

### 15. Sistemas Operativos en los Sistemas Embebidos (II).

- Kernel. Algoritmo planificador de Tareas. Conceptos
- ¿Como implementar Algoritmo: Planificador de Tareas en un sistema embebido?
- o Análisis y diseño de Tareas: Sincronas / Asincronas.
- o Planificador de tareas secuenciales.
- o Flujogramas del planificador.
- o Implementación de un planificador de tareas bajo un sistema embebido TMS320F28027 mediante lenguaje C ANSI.
- Implementación de una aplicación de control digital de temperatura en un aula, realizando el diseño para la codificación de un planificador de tareas.
- Diseñar un micro-Kernel, para un sistema embebido de la familia C2000 de Texas Instruments.
- ProtoThreads. ¿Qué son?
- ProtoThreads. LIMITACIONES.
- ProtoThreads. ¿En que se basan?
- Ejemplo-ProtoThreads para un sistema embebido de la familia C2000 de Texas Instruments.
- Sistemas Operativos en Tiempo Real (RTOS). Conceptos avanzados.
- o Tareas Concurrentes (Multitarea).
- o Cambio de contexto de las tareas.
- o Implementación de un sencillo RTOS.
- Características de los Sistemas Operativos en Tiempo Real (RTOS).
- Comparativa de Round Robin y RTOS.
- Selección de la arquitectura de los RTOS.
- Comparativa de las arquitectura de los RTOS.
- Sistemas Operativos en Tiempo Real (RTOS) Comerciales:
- o FreeRTOS.
- o Contiki.
- o TinyOS.
- o QNX.
- o RTEMS.
- o SYS\_BIOS.

### 16. Depuración del Software en los Sistemas Embebidos.

- ¿Cómo abordar un proyecto de un producto Software?
- Conceptos básicos de la Depuración.
- ¿Qué Depurar o Testear?:
- o Test funcional.
- o Test de cobertura
- Problemas al realizar la Depuración en los Sistemas Embebidos.
- Métodos básicos de Depuración en los Sistemas Embebidos.
- ¿Cuándo terminar la Depuración o Testeo?
- Técnicas de depuración en los sistemas embebidos:
- o "Software-in-the-loop" (SiL).
- o "Hardware-in-the-loop" (HiL).
- o "System test" (ST).
- Diseño basado en Modelos (MBD) con Matlab-Simulink.
- Técnica del "Real Time Algorithm in the Loop" (RTAIL).
- Ejemplo de diseño e implementación de un filtro digital mediante la técnica RTAIL.

### 17. Gestión de los proyectos en los Sistemas Embebidos.

- Introducción al procedimiento para la creación de un software en producción.
- El documento de especificaciones.
- El documento de diseño.
- El documento del código.
- Utilización de plantillas documentales.

#### Laboratorios prácticos:

1. Unidad-3: Lenguaje C-Ansi. Base:
2. Unidad-4: Lenguaje C-Ansi. Estilo:
3. Unidad-5: Lenguaje C-Ansi. Avanzado:
4. Unidad-6: Interrupciones y Timer:
5. Unidad-7: Lenguaje C-Ansi. Programación Orientada a Objetos:
6. Unidad-8: Lenguaje C-Ansi. Optimización:
7. Unidad-10: Documentación:
8. Unidad-11: Maquinas de estados Finitos:
9. Unidad-13: StateFlow-MATLAB:
10. Unidad-14: Sistemas Operativos para Sistemas Embebidos (RTOS):
11. Unidad-16: Depuración del Software:

**Metodología didáctica:**

ONLINE

**Otra Información de interés:**

El curso al ser ONLINE, el material de clase serán videos, ejercicios, videos de conferencias y foro de preguntas y otras. Todo a través del acceso de la herramienta software de Poliformat.

**Condiciones generales**

La acción formativa cumple las siguientes condiciones generales: [http://www.cfp.upv.es/cond\\_gen?1](http://www.cfp.upv.es/cond_gen?1)

**Condiciones específicas****Tutorías:**

Las consultas de los alumnos a través de foros, correo electrónico, correo interno serán atendidas de lunes a viernes dentro de un plazo no superior a las 24h. Las consultas realizadas durante sábados, domingos y festivos nacionales en España, serán atendidas en un periodo de 24h a partir del siguiente día laborable.

Las consultas realizadas por los alumnos durante el periodo de vacaciones estivales en España (del 1 al 31 de agosto), se atenderán a partir del día 1 de septiembre.

**Organizadores:**

Responsable de actividad

FRANCISCO JOSE GIMENO SALES

**Datos básicos:**

Tipo de curso

FORMACIÓN ESPECIFICA

Estado

TERMINADO

Duración en horas

45 horas a distancia

Créditos ECTS

4,5

**Dónde y Cuándo:**

Dónde

VALÈNCIA

Horario

INTERNET

Observaciones al horario

Lunes\_Viernes: 9:00h - 18:00h

Lugar de impartición

ONLINE

Fecha Inicio

8/10/18

Fecha Fin

20/08/19

**Datos de matriculación:**

Matrícula desde	22/08/18
Matrícula hasta	6/07/19
Inicio de preinscripción	30/07/18
Mínimo de alumnos	1
Máximo de alumnos	50
Precio	450,00 euros
Observaciones al precio	225,00 € - Alumni UPV PLUS 225,00 € - Alumno UPV 225,00 € - Personal UPV 450,00 € - Público en general

**Profesorado:**

ALIAGA VAREA, RAMÓN JOSÉ  
GIMENO SALES, FRANCISCO JOSE  
LUNDBÄCK MOMPO, DAVID