

SISTEMAS EMPOTRADOS

Tema 1- Introducción a la programación de Sistemas Empotrados

Profesor: Carlos Diego Moreno Moreno

Dpto. de Ingeniería Electrónica y de Computadores.
Área de Arquitectura y Tecnología de Computadores.
Escuela Politécnica Superior. Universidad de Córdoba.



Objetivos.

- Que el alumnado adquiriera el concepto de sistema empotrado a partir de las definiciones que nos podemos encontrar en la bibliografía.
- Que queden claras las diferencias con un sistema empotrado y un sistema en tiempo real.
- Conocer algo del inicio de los sistemas empotrados a lo largo del tiempo hasta la actualidad.
- Que conozca la clasificación de los sistemas empotrados y sus áreas de aplicación.
- Estudiar los requisitos que debe tener cualquier sistema empotrado para ser útil.
- Aprender el concepto de co–diseño *hardware/software* para el desarrollo de aplicaciones en sistemas empotrados.
- Que conozca las características de los sistemas operativos necesarios que son específicos para los sistemas empotrados.

TEMA 1: Introducción a la programación de Sistemas Empotrados

1.1.– Definición de sistema empotrado.

1.2.– Comparación de sistema empotrado con sistema en tiempo real.

1.3.– Historia de los sistemas empotrados.

1.4.– Clasificación de los sistemas empotrados.

1.5.– Áreas de aplicación de los sistemas empotrados.

1.6.– Requisitos de los sistemas empotrados.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.








1.8.– Sistemas operativos empotrados.

1.1.– Definición de Sistema Empotrado.

- El interés por los **sistemas** que hoy día se denominan **empotrados** (*embedded system*), ha experimentado un aumento considerable en las últimas décadas con su aplicación a los sistemas denominados de tiempo real.
- No obstante, podemos encontrar referencias a este tipo de sistemas que datan de la década de los 70.
- Pero antes de entrar en materia, debemos dar una definición de los sistemas que vamos a tratar. En primer lugar, debemos indicar que no existe ninguna definición estándar de lo que se entiende por sistema empotrado; es más, dependiendo del área en el que se le pregunte nos darán respuestas diferentes.

1.1.– Definición de Sistema Empotrado.

 Nos podemos encontrar con las siguientes definiciones:

-  Un sistema empotrado es un sistema operativo ejecutándose en un micro de pocos recursos.
-  Un sistema empotrado es un dispositivo (*hardware* + *software*) no susceptible de modificación del algoritmo que define su comportamiento.
-  Un sistema empotrado es un procesador, con sus elementos externos que desarrolla una función específica de manera autónoma.
-  Un sistema empotrado es un sistema computador de propósito especial construido en un dispositivo mayor.
-  Un sistema empotrado es una mezcla de *hardware* y *software* que constituye un componente dentro de un sistema más complejo y se espera que funcione sin intervención humana.
-  Un sistema empotrado es un sistema que usa un computador para realizar una función específica, pero ni es usado ni es percibido como un computador.
-  Un sistema empotrado es un sistema que contiene al menos un procesador, periféricos y un *software*, utilizados para una aplicación específica.

1.1.– Definición de Sistema Empotrado.

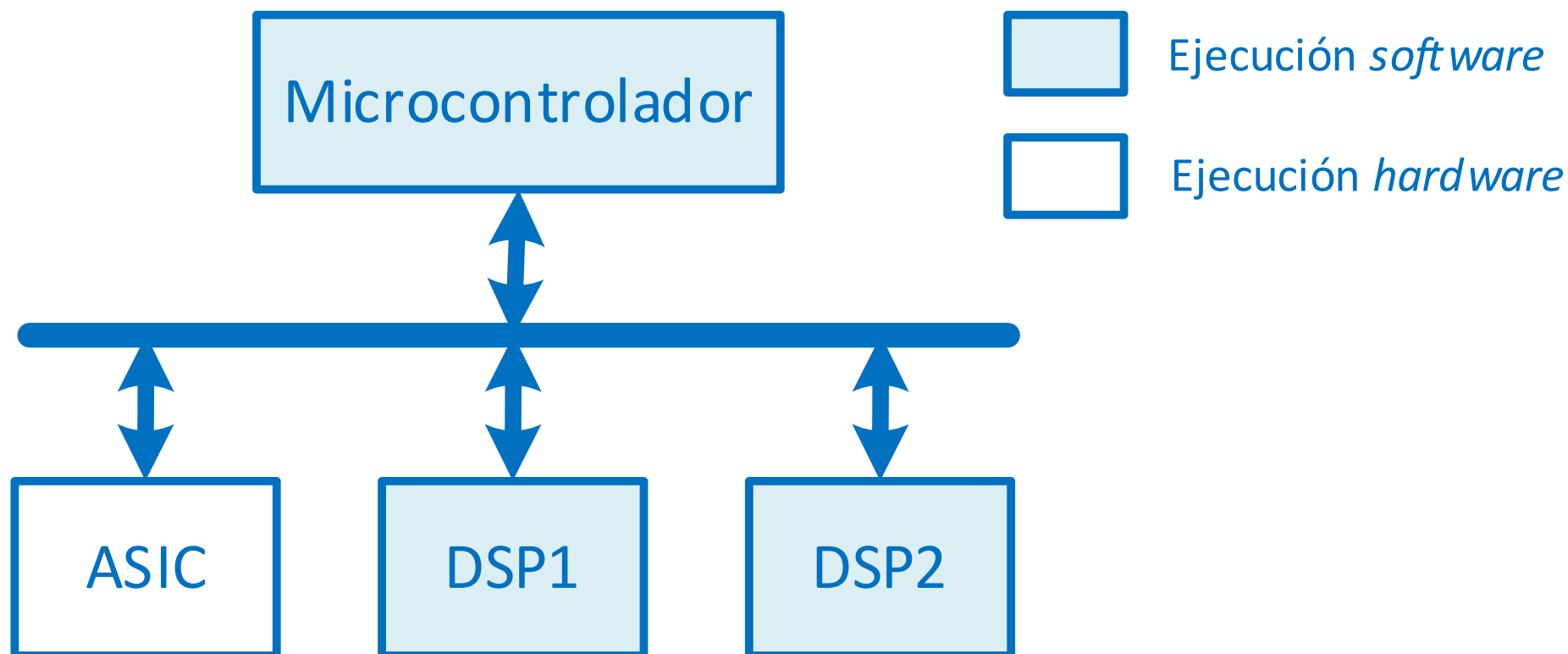
- Un sistema empotrado (o sistema embebido) lo vamos a definir como un sistema electrónico diseñado específicamente para realizar unas determinadas funciones, habitualmente formando parte de un sistema de mayor entidad.
- La característica principal es que emplea para ello uno o varios procesadores digitales (CPUs) en formato microprocesador, microcontrolador o DSP lo que le permite aportar “inteligencia” al sistema anfitrión al que ayuda a gobernar y del que forma parte.
- Un sistema empotrado es un sistema de computación que, a diferencia de los ordenadores de propósito general diseñados para cubrir un amplio rango de necesidades, tiene como objetivo efectuar unas funciones específicas dentro de un sistema de computación en tiempo real.

1.1.– Definición de Sistema Empotrado.

- Por lo general los sistemas empotrados se pueden programar directamente en el lenguaje ensamblador del microcontrolador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++.
- Como todo sistema que se precie, está formado por un microprocesador o microcontrolador y un *firmware* que se ejecute sobre éste haciendo que los distintos componentes realicen las funciones en los tiempos definidos y de acuerdo a sus requerimientos.
- Hay que resaltar que **no** se puede considerar un sistema empotrado a una aplicación genérica que se ejecuta en la CPU principal de un ordenador personal o de un servidor.
- Un sistema empotrado tiene como integrante fundamental al computador, ya sea en sus versiones de microprocesador o de microcontrolador.

1.1.– Definición de Sistema Empotrado.

Un sistema empotrado genérico seguirá el esquema mostrado en la figura siguiente:

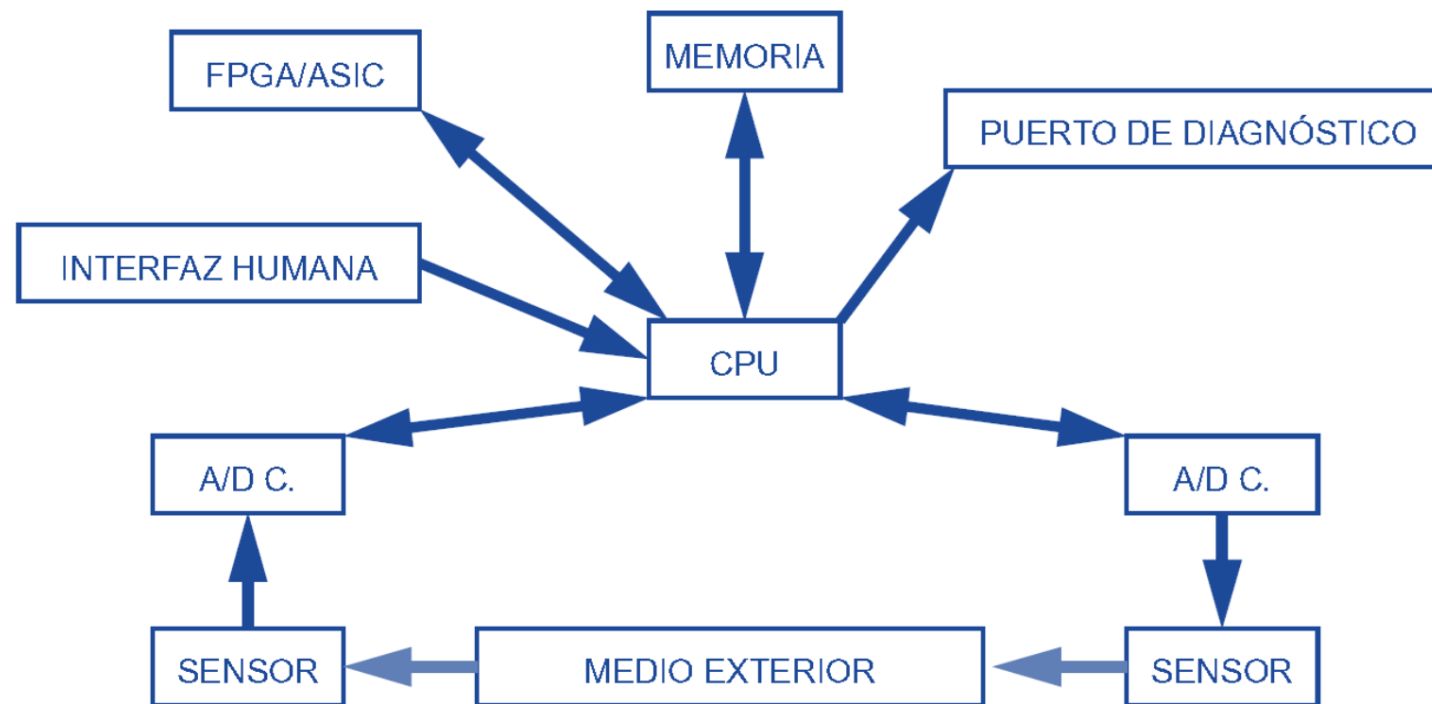


1.1.– Definición de Sistema Empotrado.

- 🔗 Luego un sistema empotrado es un sistema con un relativo grado de heterogeneidad por la combinación de *hardware* a medida (como es el dispositivo ASIC) y *software* empotrado (como son los programas que controlan tanto al microcontrolador como a los DSP o elementos programables).
- 🔗 Incluso, dentro de cada uno de estos grandes grupos, existe heterogeneidad ya que los dispositivos ASIC utilizarán diferentes estilos de diseño (por lo general); y en cuanto al componente *software*, habrá diferencia de estilos en la programación del microcontrolador y de los DSP.
- 🔗 Esta heterogeneidad impone una de las principales partes del diseño de cualquier sistema empotrado: decidir qué funcionalidad se implementará vía *software* y cuál vía *hardware*.

1.1.– Definición de Sistema Empotrado.

El sistema de la figura es un ejemplo más detallado donde podemos apreciar la CPU y la memoria, junto con una amplia variedad de interfaces que permite al sistema medir, manipular e interactuar con el entorno exterior. Existen dos tipos de comunicación: una comunicación en el interior del sistema (que se realizará a través de un bus de sistema), y una comunicación externa (del sistema con el exterior).








→ Comunicación exterior

→ Bus de sistema

1.1.– Definición de Sistema Empotrado.

 Algunas diferencias con un sistema computador pueden ser las siguientes:

-  La interfaz humana (que en un sistema computador es generalmente la pantalla y el teclado) puede ser tan simple como un diodo led o tan compleja como el sistema de visión de un robot.
-  El puerto de diagnóstico ya no sirve únicamente para chequear el sistema computador (o, extrapolando, el sistema empotrado), sino que también puede chequear el entorno que hay que controlar.
-  Se suelen utilizar dispositivos de aplicación específica para aumentar las prestaciones del sistema completo, ya sean digitales o analógicos.
-  Por lo general, el *software* tiene una misión fija y específica a la aplicación.

 La mayoría de estas diferencias vienen determinadas por las diferencias en los campos de aplicación. Mientras que en el sistema computador, las principales tareas serán la ejecución de hojas de cálculo, procesadores de textos o análisis mediante simulación; en los sistemas empotrados se realizarán tareas de ejecución de leyes de control, máquinas de estados finitos y algoritmos de procesamiento de señal para poder realizar la manipulación del entorno.

TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.**
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.

1.2.– Comparación de sistema empotrado con sistema en tiempo real.

- Un sistema de tiempo real es un sistema informático en el que es significativo el tiempo en el que se producen sus acciones. No basta que las acciones de un sistema sean correctas, sino que, además, deben ocurrir dentro de un intervalo de tiempo determinado.
- Sistema en tiempo real es un sistema que tiene que producir unas salidas en respuesta a unas entradas dentro de unos límites de tiempo específicos.
- En un sistema de tiempo real es tan importante el resultado lógico de las funciones que realiza como el tiempo en que este resultado se producen.
- Los sistemas de tiempo real suelen estar integrados en un sistema de ingeniería más general, en el que realizan funciones de control, procesamiento y/o monitorización: sistemas empotrados (*embedded systems*).

1.2.– Comparación de sistema empotrado con sistema en tiempo real.

- La mayoría de los sistemas empotrados tienen requerimientos de tiempo real.
- La mayoría de los sistemas de tiempo real van empotrados.



TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.**
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.

1.3.– Historia de los sistemas empotrados.

- ✎ Cuando los sistemas empotrados entraron en el rango de disciplina académica, se vio que uno de los grandes problemas radica en el particionado *hardware/software*; y para dar solución a este problema era necesaria la creación de modelos y algoritmos.
- ✎ Al principio de la década de los 90, se presentaron dos sistemas para dar solución a este problema, denominados VULCAN y COSYMA, los cuales tomaron aproximaciones complementarias para realizar este particionado, es decir, para decidir qué funciones serían realizadas vía *software*, y cuáles vía *hardware*.
 - 🌿 VULCAN colocaba todas las funciones en los dispositivos *hardware*, e iba pasando algunas funciones a *software* para minimizar el coste.
 - 🌿 COSYMA colocaba todas las funciones vía *software* e iba pasando funciones a los dispositivos *hardware* para obtener las prestaciones requeridas.

1.3.– Historia de los sistemas empotrados.

- Los diseñadores de los sistemas *hardware/software* tenían que analizar las prestaciones en tres dimensiones:
- Las prestaciones **hardware** tenían como meta determinar la máxima frecuencia de reloj de la unidad *hardware*.
 - Las prestaciones **software** tenían como meta determinar el tiempo de ejecución del caso peor (en esencia, el mismo problema que en el caso *hardware*); no obstante su solución es más difícil de hallar debido a la menor investigación desarrollada en este campo.
 - Las prestaciones **de sistema** también son complicadas, ya que varios programas pueden ser ejecutados de forma concurrente (o varias partes de varios programas), que a su vez, pueden ser ejecutados concurrentemente con el procesado de aplicaciones específicas. Ambos sistemas (VULCAN y COSYMA) tratan de reducir la complejidad, evitando estas concurrencias, y por lo tanto, todas las operaciones (*software* y *hardware*) son consideradas como secuenciales.

1.3.– Historia de los sistemas empotrados.

- Para realizar estos análisis de prestaciones se encontró en la co-simulación un aliado muy importante, incorporándolo como elemento esencial a la metodología de co-diseño. El reto consiste en la realización de una co-simulación en niveles de abstracción mixtos para ejecutar los suficientes vectores de *test* para validar el diseño. Uno de los co-simuladores que primero surgieron fue PTOLEMY.
- Una vez que las prestaciones anteriores, tiempos de ejecución y de procesado, iban siendo resueltas; otras prestaciones iban adquiriendo una importancia creciente. Entre estas prestaciones podemos encontrar la estimación del coste *hardware*, el consumo de potencia, arquitecturas más generales, protocolos.
- Actualmente, y gracias al auge de las FPGA, el particionado *hardware/software* es una tarea de diseño práctica. Varios fabricantes han desarrollado placas que combinan FPGA (en la que se implementaría los dispositivos *hardware*) y CPUs (en la que se implementaría el desarrollo *software*), conteniendo la arquitectura típica de un sistema empotrado.




TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.**
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.

1.4.– Clasificación de los sistemas empotrados.

- Podemos encontrar diferentes clasificaciones de los sistemas empotrados. Una de las principales se puede encontrar si consideramos su interacción con el resto del entorno. Atendiendo a esta característica, podemos encontrar los siguientes sistemas:
- 🌿 Sistemas **reactivos**: siempre interactúan con el exterior de tal forma que la velocidad de operación del sistema deberá ser la velocidad del entorno exterior.
 - 🌿 Sistemas **interactivos**: siempre interactúan con el exterior de tal forma que la velocidad de operación del sistema deberá ser la velocidad del propio sistema empotrado.
 - 🌿 Sistemas **transformacionales**: son aquellos sistemas que no interactúan con el exterior, únicamente toma un bloque de datos de entrada y lo transforma en un bloque de datos de salida, que no es necesario en el entorno.

1.4.– Clasificación de los sistemas empotrados.

-  **Ejemplo** de sistema **reactivo**: el sistema de control aéreo de un aeropuerto, ya que la velocidad del sistema dependerá de la velocidad con la que lleguen los datos de los diferentes aviones que se acerquen o salgan del mismo.
-  **Ejemplo** de sistema **interactivo**: una máquina de videojuegos, ya que la velocidad del sistema depende de él mismo, y el exterior (es decir el usuario del videojuego) se debe adecuar a su velocidad.
-  **Ejemplo** de sistema **transformacional**: los postes de publicidad electrónicos, en los que no existe ningún tipo de interactividad excepto la entrada de datos iniciales y la salida de datos finales.

TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.**
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.

1.5.– Áreas de aplicación de los sistemas empotrados.

- Los sistemas empotrados están ampliamente presentes en nuestra vida cotidiana, y de hecho, raro es el momento en el que no nos encontramos cerca de uno de ellos.
- Un usuario no técnico de un sistema empotrado puede no ser consciente de que está usando un sistema computador.
- En algunos hogares las personas, que no tienen por qué ser usuarias de un ordenador personal estándar (PC), utilizan del orden de diez o más sistemas empotrados cada día: TV, móvil, cámara de fotos, frigorífico, lavadora, coche, etc.
- Las aplicaciones más numerosas y habituales de los sistemas empotrados suelen ser del tipo industrial y gran consumo.

1.5.– Áreas de aplicación de los sistemas empotrados.

- En la práctica totalidad de las áreas de nuestra vida nos encontramos con sistemas empotrados que prácticamente nos pasan desapercibidos. Sirva como ejemplo el sector del automóvil, que en pocos años ha introducido notables avances en lo referente a la seguridad, confort, info-movilidad, etc.
- Existen en el mercado de semiconductores una amplia variedad de familias de microprocesadores, microcontroladores y DSPs dirigidos a este sector.
- En la actualidad, todos los fabricantes de semiconductores ofrecen su gama de productos relacionándolos con el amplio rango de aplicaciones a los que van dirigidos.
- A continuación, se describen las principales aplicaciones de los sistemas empotrados:

1.5.– Áreas de aplicación de los sistemas empotrados.

- 📖 Aplicaciones basadas en pequeños microcontroladores.
 - 🌿 Realizan un control simple de un pequeño dispositivo.
 - 🌿 Origen de la programación empotrada. Se utilizaban microcontroladores de 4 y 8 *bits*.
 - 🌿 Área con unas severas restricciones en coste y por tanto muy limitada en recursos (potencia CPU, memoria).
 - 🌿 Ejemplo: Lector de una tarjeta con *chip*, teléfono, pequeños electrodomésticos, etc.

1.5.– Áreas de aplicación de los sistemas empotrados.

Control de sistemas.

- 🌿 La programación se centra en la realización óptima de los lazos de control.
- 🌿 Ej: Sistema de control de velocidad de crucero de un automóvil o sistema de frenado ABS.

Control de sistemas empotrados distribuidos:

- 🌿 Múltiples CPUs conectadas mediante red de comunicaciones de tiempo real con un ancho de banda bajo, del orden de 1Mbit/sec.
- 🌿 Ejemplo: Red “*Controller Area Network*” en automóviles.

Sistemas en un chip (SoC).

1.5.– Áreas de aplicación de los sistemas empotrados.



Redes:








- La mayoría de los dispositivos comerciales para redes son sistemas empotrados.
- Ejemplo: *routers, bridges*, puntos de accesos inalámbricos.

PCs empotrados.

Sistemas críticos:

- Sistemas que realizan tareas críticas. Requieren en caso de fallo, al menos un funcionamiento degradado.
- Ejemplo: dispositivos médicos, centrales nucleares, aviónica.

1.5.– Áreas de aplicación de los sistemas empotrados.

-  Robótica.
-  Periféricos de computadores: por ejemplo impresoras, discos duros, etc.
-  Procesamiento digital de señales:
 -  Procesamiento de audio y vídeo, y en general, procesamiento de *streams* de datos.
 -  Ejemplo: MP4, equipos de audio, cámaras digitales, multimedia en general.
-  Equipos industriales de instrumentación, automatización, producción, etc.
-  Sistemas de transmisión de datos inalámbricos: Sistemas compuestos por una red de nodos que realizan adquisición de datos. Ejemplo: red de sensores inalámbricos (WSN).

1.5.– Áreas de aplicación de los sistemas empotrados.

- Como ejemplo, mostramos en la figura siguiente los tipos de sistemas empotrados que podemos encontrarnos en un hogar medio.
- En ella podemos distinguir cuatro grandes grupos de elementos:
 - Los basados en PC (ordenador, PDA o cualquier elemento de entrada/salida como una impresora).
 - Los relativos a las comunicaciones (teléfonos o intercomunicadores).
 - Relativos al entretenimiento (televisores, vídeos, consolas, etc.).
 - Relativos al confort y comodidad (electrodomésticos y climatizadores).

1.5.– Áreas de aplicación de los sistemas empotrados.









TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.**
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.




1.6.— Requisitos de los sistemas empotrados.

- 🏠 Las principales características de un sistema empotrado son las siguientes:
 - 🌿 Emplean una combinación de recursos *hardware* y *software* para realizar una función específica.
 - 🌿 Realizan una única función o un conjunto muy limitado de funciones (no suelen ser de propósito general).
 - 🌿 El diseño de procesadores de aplicación específica suelen ser un componente significativo de estos sistemas.
- 🏠 Aunque todas las características anteriores son inherentes a cualquier sistema empotrado, para que el sistema sea útil, también deberá tener las siguientes:

1.6.— Requisitos de los sistemas empotrados.

-  **Concurrencia:** Los componentes del sistema funcionan simultáneamente, por lo que el sistema deberá operar a la vez.
-  **Fiabilidad y seguridad:**
 -  El sistema debe ser fiable y seguro frente a errores, ya que puede requerir un comportamiento autónomo.
 -  El sistema debe funcionar sin producirse *resets* o paradas inesperadas que requieran la intervención humana.
 -  El sistema no puede tener fallos impredecibles.
 -  El manejo de estos errores puede ser vía *hardware* o *software*; aunque la utilización *software* nos dará un sistema menos robusto.

1.6.— Requisitos de los sistemas empotrados.

-  **Interacción con dispositivos físicos:** Los sistemas empotrados interaccionan con el entorno a través de dispositivos E/S no usuales, por lo que suele ser necesario un acondicionamiento de las diferentes señales.
-  **Robustez:** Al sistema empotrado se le impondrá la necesidad de la máxima robustez ya que las condiciones de uso no tienen por qué ser óptimas, sino que pueden estar en el interior de un vehículo con diferentes condiciones de operación.
-  **Pequeñas dimensiones:** Las dimensiones de un sistema empotrado no dependen sólo de sí mismo sino también del espacio disponible en el cual dicho sistema va a ser ubicado.

1.6.— Requisitos de los sistemas empotrados.

Bajo consumo de potencia:


- 🌿 El hecho de poder utilizar el sistema en ambientes hostiles puede implicar la necesidad de operaciones sin cables.
- 🌿 Por lo tanto, un menor consumo implica una mayor autonomía de operación.
- 🌿 Muchos sistemas empotrados están alimentados con baterías.

Relación coste/efectividad:


- 🌿 En caso de diseños de propósito especial el coste no tendría por qué ser un requisito.
- 🌿 En cambio, en producción masiva, el coste es el principal requisito.

1.6.— Requisitos de los sistemas empotrados.

Eficiente utilización de la potencia de procesamiento:

-  El diseñador debe tener en mente cuales deben ser sus requisitos de potencia para la utilización del procesador adecuado, para no aumentar la complejidad de su sistema, el coste o el consumo de potencia.

Eficiente utilización de la memoria:

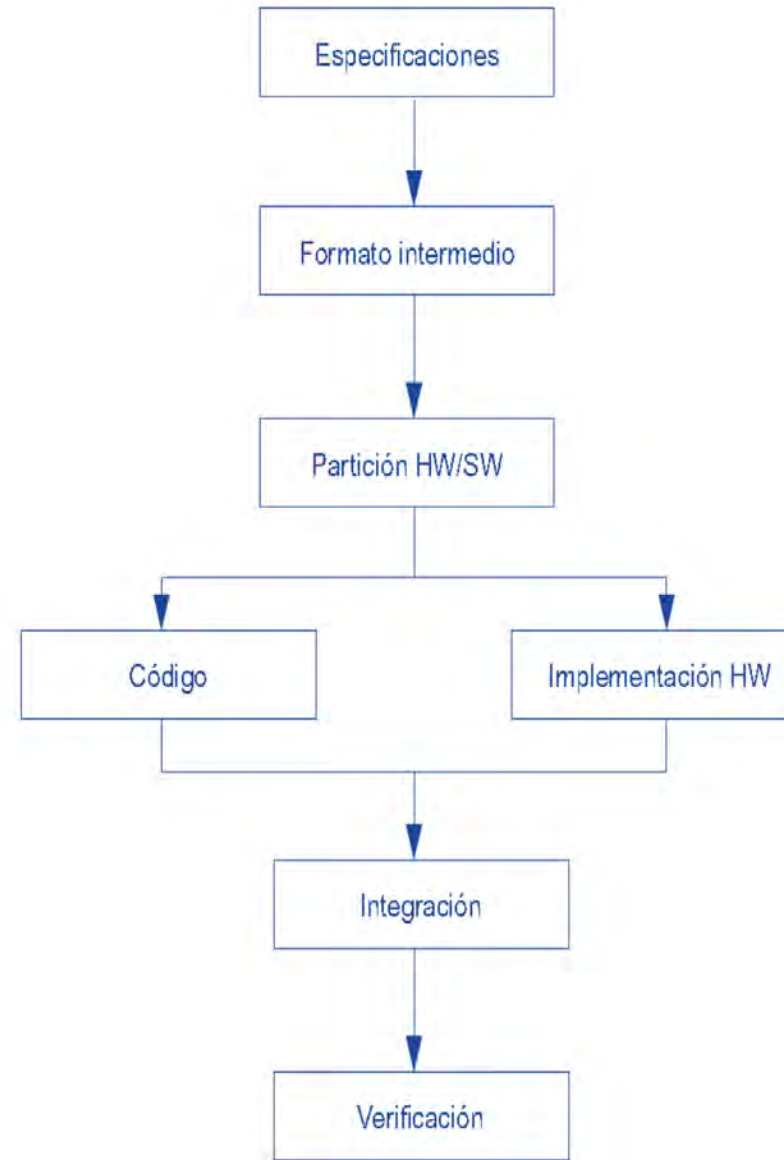
-  Gran parte de los sistemas empotrados no tienen memoria secundaria masiva por lo que el diseñador debe determinar el tamaño de memoria dedicada a programa y/o a datos.

TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.**
- 1.8.– Sistemas operativos empotrados.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.




Para los sistemas empotrados, un flujo de diseño de una tarea o aplicación dividida en subtareas podría ser el siguiente:






1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.

- 📖 **Especificaciones:** en primer lugar, se debe realizar un modelado del sistema completo.
- 📖 El modelo obtenido de esta subtarea servirá para fijar los diferentes algoritmos que estén involucrados en la operación del sistema (ya sean algoritmos de control o de operación con datos).
- 📖 Un modelo formal debería contar con los siguientes componentes:
 - 🌿 Una descripción funcional.
 - 🌿 Un conjunto de propiedades.
 - 🌿 Un conjunto de índices de desarrollo.
 - 🌿 Un conjunto de restricciones.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.

-  **Particionado del sistema:** Una vez realizado tendremos el comportamiento del sistema dividido en varias funciones de menor complejidad, con el propósito de abordar el problema completo como varios problemas desconectados entre sí.
-  **Particionado *hardware/software*:** Determinaremos qué funciones se realizarán mediante dispositivos de aplicación específica (*hardware*) o mediante programación (*software*).
-  **Síntesis *hardware y software*:** Estos procesos de síntesis no deben realizarse de forma separada ni independiente, sino que hay que considerar una tarea de *scheduling* (planificación), en la cual se realizará una temporización de todos los dispositivos para que el flujo de datos esté de acuerdo con la operación a realizar.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.

-  **Mapeado:** Finalmente habrá que realizar una subtarea de mapeado que traducirá la descripción funcional en programas que pueden ser ejecutados en unos determinados procesadores o en una conexión de dispositivos *hardware* digitales y/o analógicos (dependiendo de la zona de particionado de la descripción funcional).
-  Aunque se ha tratado como un flujo secuencial de tareas todas las subtareas están relacionadas entre sí, y no únicamente las adyacentes.
-  Por ejemplo, el hecho de tener un determinado procesador con sus propias funciones (estaríamos hablando de la subtarea de mapeado), puede causar que el comportamiento global sea particionado en una serie de funciones (subtarea de particionado), que podrían cambiar si se dispusiese de otro tipo de procesador.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.

- 📖 El término de co–diseño *hardware/software* es relativamente reciente, de principios de la década de los 90. Este término describe la confluencia de dos problemas en el diseño de circuitos integrados:
 - 🌱 Las metodologías de diseño de sistemas tendrían que considerar grandes CPUs prediseñadas, como componentes de los circuitos.
 - 🌱 El *software* debería ser tratado como un componente en el diseño de circuitos, el cual estaría incluido en las CPUs mencionadas anteriormente.
- 📖 Esta situación fue predicha según la ley de *Moore* que predecía que, en esta fecha, los circuitos integrados serían lo suficientemente grandes para poder albergar una CPU y otros sistemas.

1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.

- 📖 El término de co–diseño *hardware/software* puede ser definido de la forma siguiente:
 - 🌿 El codiseño *hardware/software* es la tarea de diseñar el sistema *hardware* y la generación de código del sistema *software* de un sistema mixto, de tal forma que el comportamiento del sistema global está formado por la ejecución del sistema *hardware y software*.
- 📖 Es decir, es una de las subtarefas del flujo de diseño de sistemas empotrados. No obstante, se le suele igualar al problema de diseño completo relegando las tareas de modelado y particionado a papeles secundarios.


TEMA 1: Introducción a la programación de Sistemas Empotrados

- 1.1.– Definición de sistema empotrado.
- 1.2.– Comparación de sistema empotrado con sistema en tiempo real.
- 1.3.– Historia de los sistemas empotrados.
- 1.4.– Clasificación de los sistemas empotrados.
- 1.5.– Áreas de aplicación de los sistemas empotrados.
- 1.6.– Requisitos de los sistemas empotrados.
- 1.7.– Desarrollo de *software* para sistemas empotrados. Co–diseño HW–SW.
- 1.8.– Sistemas operativos empotrados.**

1.8.– Sistemas operativos empotrados.

- Un microcontrolador es un dispositivo totalmente vacío de *software*, siendo el desarrollador el responsable de “llenarlo” con código.
- Dada la creciente complejidad en los desarrollos con microcontrolador, se puede recurrir a un sistema operativo para microcontrolador o *microkernel*, que simplifica enormemente de aplicaciones y que no es más que código añadido que proporciona los servicios de más bajo nivel de un SO; por ejemplo, multitarea, sincronización, regiones críticas, semáforos, *buffers*, etc.
- En cuanto a los sistemas operativos necesarios para que un sistema basado en microprocesador pueda funcionar y ejecutar programas suelen ser específicos para los sistemas empotrados. Así nos encontramos con sistemas operativos de bajos requisitos de memoria, posibilidad de ejecución de aplicaciones de tiempo real, modulares.
- Ejemplos: Windows CE, QNX y VxWorks de WindRiver.

1.8.– Sistemas operativos empotrados.

- 
- Es posible desarrollar *software* para sistemas empotrados sin sistema operativo, de hecho, existen muchas aplicaciones sencillas que no lo llevan.
 - Es posible implementar fácilmente administradores de memoria, manejadores de entrada/salida.
 - Ventajas: Código compacto y seguro.
 - Desventajas: portabilidad.
 - Actualmente la potencia de los procesadores, incluso de 8 *bits*, permite tener un mínimo sistema operativo empotrado.

1.8.– Sistemas operativos empotrados.

En lo que se refiere al *software*, se tendrán requisitos específicos según la aplicación. En general, para el diseño de un sistema empotrado no se dispone de recursos ilimitados, sino que la cantidad de memoria será escasa, la capacidad de cálculo y dispositivos externos será limitada, etc. Podemos hablar de las siguientes necesidades:

- Trabajo en tiempo real.
- Optimizar al máximo los recursos disponibles.
- Disponer de un sistema de desarrollo específico para cada familia de microprocesadores empleados.
- Programación en ensamblador, aunque en los últimos años, los fabricantes o empresas externas han mejorado la oferta de compiladores que nos permiten trabajar en lenguajes de alto nivel, tales como C.

1.8.– Sistemas operativos empotrados.

- 📖 El empleo de un sistema operativo determinado o no depende del sistema a desarrollar y es una de las principales decisiones que habrá que tomar en la fase de diseño del sistema empotrado.
- 📖 Así, en el caso de decidirse por el empleo de microcontroladores y DSP, por lo general no se usará sistema operativo mientras que, si se emplea algún micro del tipo ARM, PowerPC, Intel X86, etc. sí que lo llevará.
- 📖 La decisión dependerá de los requisitos del sistema, tanto técnicos como económicos.



Bibliografía consultada.

- Dreamtech Software Team, "Programming for Embedded Systems: Cracking the Code", Wiley, 2002, ISBN: 0764549545.
- Tammy Noergaard, "Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers", 2005, ISBN: 0-7506-7792-9.

¡Muchas gracias por su atención!

Carlos Diego Moreno Moreno



Área de Arquitectura y Tecnología de Computadores
Departamento de Ingeniería Electrónica y de Computadores.
Escuela Politécnica Superior. Universidad de Córdoba