

# Principios de Mecatrónica – SDI-11561

## Ingeniería en Mecatrónica

Hugo Rodríguez Cortés

Departamento de Eléctrica y Electrónica

Instituto Tecnológico Autónomo de México

Agosto 2023

Se trata de un sistema de computación diseñado para realizar una o varias funciones **dedicadas** frecuentemente en tiempo real.

**Tiempo real.** El periodo de tiempo durante el cual ocurre algo.

El sistema de computación ejecuta una función en tiempo real, si el resultado se obtiene dentro de un periodo de tiempo que el usuario siente como inmediato o actual.

La latencia debe ser inferior a un valor definido, normalmente medido en segundos.

**Latencia.** Intervalo entre la estimulación y la respuesta.

Al contrario de lo que ocurre con los **ordenadores de propósito general** (como por ejemplo una computadora personal) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para **cubrir necesidades específicas**.

En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base (la tarjeta de vídeo, audio, módem, etc.) y muchas veces los dispositivos resultantes no tienen el aspecto de lo que se suele asociar a una computadora.

Todo sistema que NO es una PC de escritorio, ni un servidor, ni una workstation, ni una supercomputadora, ni un cluster distribuido, en fin, NO es un sistema programable de proposito general.

Ejemplos de sistemas embebidos podrían ser:

- Embebidos en industria automotriz : navegador GPS, de la aviación: piloto automatico, control de aterrizaje
- Embebidos en telecomunicaciones: routers, modems
- Embebidos en comunicaciones: teléfonos celulares
- Embebidos en el hogar: control de heladeras, microondas, robots que aspiran, cortan pasto.

**Los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o en el microprocesador incorporado sobre el mismo**, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos.

La mayoría de los sistemas embebidos funcionan en tiempo real (Real-Time).

**Se encuentran frecuentemente conectados a ambientes físicos a través de sensores y actuadores. Son sistemas híbridos (es decir, poseen partes analógicas + digitales), típicamente son sistemas reactivos, los cuales son “aquellos que están en interacción continua con su entorno y su ejecución es a un ritmo determinado por ese entorno ” (Bergé, 1995).**

Los sistemas embebidos tienen como una de sus partes una computadora con características especiales conocida como microcontrolador que viene a ser el cerebro del sistema, incluye interfaces de entrada/salida. Normalmente estos sistemas poseen un interfaz externo para efectuar un monitoreo del estado y hacer un diagnóstico del sistema.

**Un sistema embebido esta formado por un microprocesador y un software que se ejecute sobre éste.** Sin embargo este debe poder guardarse para luego ser ejecutado. Esto podría realizarse en una **memoria RAM o ROM.**

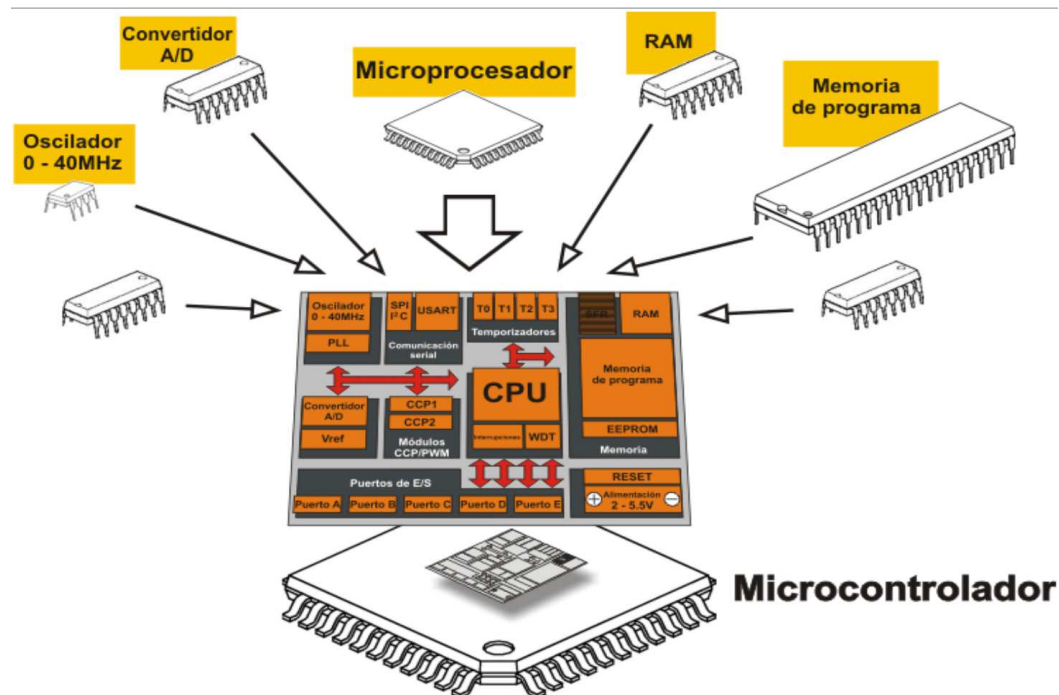
Todo sistema embebido necesitará una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador.

Contará también con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior.

Debido a que las tareas realizadas por sistemas embebidos son de relativa **sencillez**, los procesadores comúnmente empleados cuentan con **registros de 8 o 16 bits.** **En su memoria sólo reside el programa destinado a gobernar una aplicación concreta.**

Sus líneas de **entrada/salida (I/O)** soportan la conexión de los **sensores y actuadores** del dispositivo a controlar y todos los recursos complementarios disponibles tienen como finalidad atender a sus requerimientos.

Estas son las únicas características que tienen en común los sistemas embebidos, todo lo demás será totalmente diferente para cada sistema embebido en particular debido a la variedad de aplicaciones disponibles.



**Sistema mecatrónico:** es el resultado de la integración física de las ramas que comprenden el estudio de sistemas **mecánicos, máquinas eléctricas (en algunos casos sistemas de potencia), sensores, actuadores, sistemas de comunicación (telecomunicación) y cómputo.** Todo, con la finalidad de obtener una integración funcional para procesar información disponible en el sistema a través de herramientas de software resultando en acciones automáticas de control ejecutadas al sistema físico.

**Sistema Mecatrónico:** combina múltiples ramas de la ingeniería para brindar soluciones confiables y eficientes a problemas de manufactura, transporte, medicina, entretenimiento, etc.



Por ejemplo, un microcontrolador de alto desempeño Atmega328P de 8 bits, con arquitectura **RISC** es utilizado en un Arduino Uno.

- 32 KB de memoria flash, 1 KB de memoria EEPROM<sup>a</sup>, 2 KB de SRAM<sup>b</sup>
- 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, una interface serial orientada a byte de 2 cables
- 6 canales 10-bits Conversor A/D, “watchdog timer” programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Su arquitectura le permite ejecutar instrucciones en un solo ciclo de reloj.

---

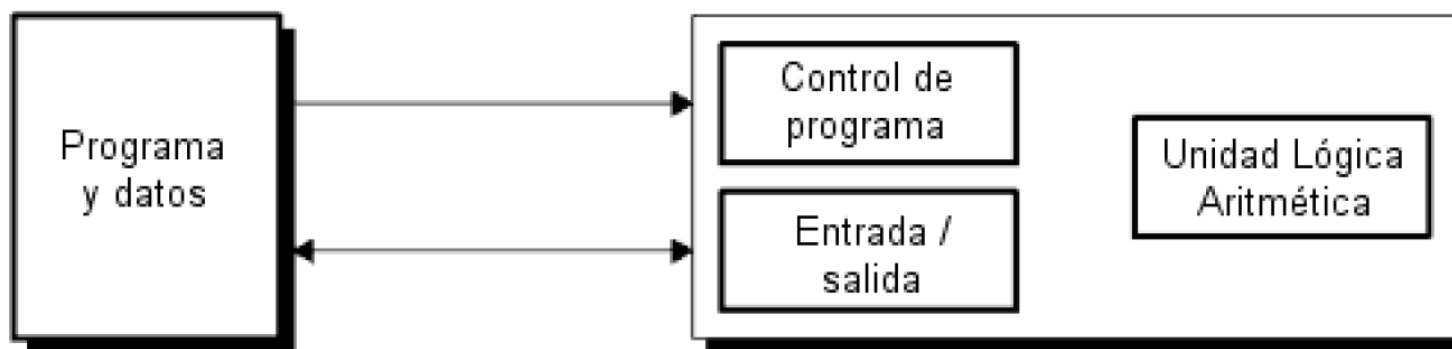
<sup>a</sup>Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioleta. Son memorias no volátiles.

<sup>b</sup>es una memoria RAM estática de acceso aleatorio, se puede denominar como un tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada.

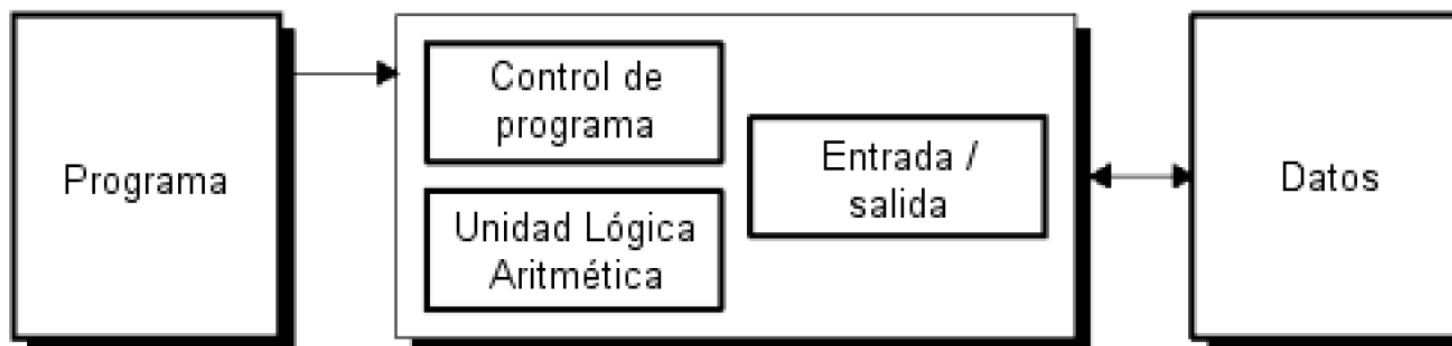
La arquitectura de los sistemas embebidos es de dos tipos:

- **Von Neumann:** Utiliza **el mismo dispositivo de almacenamiento tanto para las instrucciones de programa como para los datos.** La mayoría de las computadoras modernas están basadas en esta arquitectura, pueden incluir dispositivos adicionales (por ejemplo para gestionar las interrupciones de dispositivos externos). **Las operaciones de lectura y escritura no pueden realizarse simultáneamente.**
- **Harvard:** Utiliza **dispositivos de almacenamiento físicamente separados para las instrucciones de programa y para los datos.** El inconveniente radica en que debido a que esta divida la información para que funcione lo mejor posible, es necesario que la frecuencia de lectura de instrucciones y de datos sea aproximadamente la misma.

## Arquitectura Von Neumann



## Arquitectura Harvard



Generalmente, la arquitectura de los sistemas embebidos es de tipo **Harvard**, ya que existe una memoria RAM que almacena los datos y otra de tipo EPROM o EEPROM que almacena el programa. Al usar esta arquitectura **se tiene la ventaja de poder usar simultáneamente los dos tipos de memoria.**

Por otro lado, el **conjunto de instrucciones** define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman lo que conocemos como software.

El conjunto de instrucciones vienen siendo como las letras del alfabeto, el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir palabras, oraciones y cuanto programa se le ocurra.

Existen dos tipos de set de instrucciones: **CISC y RISC**. Las instrucciones **CISC** (Complex Instruction Set Computer, Computadora de Conjunto de Instrucciones Complejo) se caracterizan por tener un repertorio amplio de instrucciones, que pueden hacer cálculos muy complejos, pero se complica su uso.

Además las instrucciones muy complejas no se utilizan con mucha frecuencia y en muchos casos son responsables del retardo del desempeño de todo el sistema.

Esto en particular debido a que la Unidad de Control debe ser más compleja y **si es microprogramada provoca que todas las instrucciones demoren mayor tiempo en su ejecución.**

El tipo **RISC** (Reduced Instruction Set Computer, Computadora con Conjunto de Instrucciones Reducido) se distingue por sus pocas instrucciones, y un tamaño fijo por instrucción, además permite el paralelismo en la ejecución.

Las principales características de estas arquitecturas son las siguientes:

CISC	RISC
Instrucción multiciclo	Instrucción de un único ciclo
Carga/almacenamiento en una instrucción	Carga/almacenamiento son instrucciones separadas
Arquitectura memoria–memoria	Arquitectura registro–registro
Utiliza memoria de microprograma localizado en el circuito integrado	Implementa las instrucciones directamente en hardware
Reduce la dificultad de implementar compiladores	Compiladores complejos
Versatilidad del repertorio de instrucciones	Se añaden nuevas instrucciones sólo si son de uso frecuente
Facilita la depuración de errores	
	Instrucciones de tamaño fijo y en un número reducido de formatos

Una de las mayores ventajas de RISC respecto a CISC es que **utiliza un sistema de direcciones no destructivas en RAM**.

Esto significa que a diferencia de CISC, RISC conserva después de realizar sus operaciones en memoria los dos operandos y su resultado, reduciendo la ejecución de nuevas operaciones.