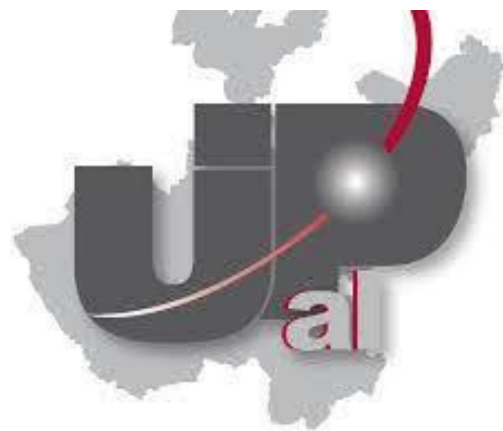




---

# SELECCIÓN DE SISTEMAS EMBEBIDOS

---



**UNIVERSIDAD POLITÉCNICA**  
DE LA ZONA METROPOLITANA DE GUADALAJARA

Amaury Efraín Gutiérrez Chávez  
Ing. Mecatrónica 8°A  
Sistemas embebidos

# ¿Qué es un sistema embebido?

Se trata de un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas. Los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes como JAVA.

## Características

Las características básicas de los sistemas embebidos son las siguientes:

Deben ser confiables,

- La confiabilidad, en inglés reliability  $R(t)$ , es la probabilidad de que el sistema trabaje correctamente dado que está funcionando en  $t=0$ .
- La mantenibilidad, en inglés Maintainability  $M(d)$ , es la probabilidad de que el sistema vuelva a trabajar correctamente  $d$  unidades de tiempo después de un fallo.
- La disponibilidad, en inglés Availability  $A(t)$ , es la probabilidad de que el sistema esté funcionando en el tiempo  $t$ .
- La seguridad informática: consiste en disponer de una comunicación confidencial y autenticada.
- La creación de un sistema confiable debe ser considerada desde un comienzo, no como una consideración posterior.
- Deben ser eficientes en cuanto a la energía, al tamaño de código, al peso y al costo.
- Están dedicados a ciertas aplicaciones.
- Interfaces de usuario dedicadas (sin ratón, keyboard y pantalla)

Muchos sistemas embebidos deben cumplir restricciones de tiempo real. Un sistema de tiempo real debe reaccionar a estímulos del objeto controlado (u operador) dentro de un intervalo definido por el ambiente

Existen varias interfaces:

- Las interfaces de operador (Hombre-Máquina-HMI) – monitores, interruptores, botones, indicadores, emisores individuales o grupales de los diferentes tipos de señales, motores eléctricos, solenoides y otros. Se puede aplicar en los trenes. Las características del software

son las siguientes: robustez, facilidad de uso, presentación clara de la información, diseño atractivo, flexibilidad de proyecto.

- Las interfaces eléctricas (interfaces con otros componentes y dispositivos): Interno - I2C, SPI, ISA y otros.

- Las interfaces Exteriores - RS232, TTY, Ethernet, Centronics, FlexRay, CAN, LIN, RF y otros

## **Estructura**

Las principales características de un sistema embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser producidos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño. Normalmente, los sistemas embebidos emplean procesadores muy básicos, relativamente lentos y memorias pequeñas para minimizar los costos. La velocidad no solo está dada por la velocidad del reloj del procesador, sino que la totalidad de la arquitectura se simplifica para reducir costos. Usualmente un ES (sistema embebido) utiliza periféricos controlados por interfaces seriales sincrónicas, las cuales son muchas veces más lentas que los periféricos de un PC. Los programas en estos sistemas se ejecutan minimizando los tiempos muertos y afrontando fuertes limitaciones de hardware, ya que usualmente no tienen discos duros, ni teclados o monitores, una memoria flash reemplaza los discos, y algunos botones y una pantalla LCD normalmente reemplazan los dispositivos de interfaz. El software que controla un dispositivo de hardware, se conoce como Firmware. La programación en estos dispositivos se realiza en lenguaje ensamblador o en lenguaje C, actualmente se han desarrollado algunas máquinas virtuales y otros compiladores que permiten el diseño de programas más complejos.

### **Componentes de un sistema embebido**

Un ES estaría formado por un microprocesador y un software que se ejecute sobre éste. Sin embargo, este software necesitará sin duda un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM. Todo sistema embebido necesitará una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador. También contará 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 el conexionado 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.

### Plataforma de sistemas integrados:

- El Microprocesador (MP o  $\mu$ P) y los microcontroladores (MCU), que tienen menos poder de cómputo, pero varios periféricos;
- Arquitecturas de base - Von Neumann and Harvard;
- Utilizado  $\mu$ P y MCU - CISC (Complex Instruction Set Computer) y más a menudo RISC (Reduced Instruction Set Computer);
- Popular RISC familias de procesos: ARC (ARC International), ARM (ARM Holdings), AVR (Atmel), PIC (Microchip), MSP430 (TI) y otros;
- CISC CPUs: Intel y Motorola;
- Por lo general en el interior hay una memoria caché y procesamiento canalización de instrucciones;
- Memoria para datos e instrucciones: RAM, PROM - OTP (OneTime Programmable), EEPROM o memoria Flash;
- Periféricos: General Purpose Input / Output - GPIO, temporizadores, ADC, DAC y mucho más.

### Comunicación:

- RS-232, RS-422, RS-485, UART / USART (Receptor Universal Sincrono y Asíncrono / Transmisor);
- I2C (Circuito Inter-Integrado), SPI (Bus de Interface periférico en serie), SSC y ESSI (Interfaz mejorada serie síncrona), USB (Universal Serial Bus);
- Protocolos de comunicación de red: Ethernet, CAN (Controlador del área de red), LonWorks etc.
- Software: Popular OS – QNX4 RIOS, Linux embebido y Linux-base (Android, etc.), iOS, Windows CE, etc.

### Herramientas para probar y corregir (Depuración)

- JTAG (Joint Test Action Group) – una interfaz especializada para la prueba saturada PCB;
- ISP (In-System Programming) – Programación de Circuito;

- ICSP (circuito de programación en serie) - un método para la programación directa del microcontrolador, por ejemplo, de la serie PIC and AVR;
- BDM (Modo de depuración de fondo) – utilizado principalmente en productos de Freescale;
- IDE (Entorno de desarrollo integrado) – para el desarrollo de programas

## Memoria

En ella se encuentra almacenado el código de los programas que el sistema puede ejecutar, así como los datos. Su característica principal es que debe tener un acceso de lectura y escritura lo más rápido posible para que el microprocesador no pierda tiempo en tareas que no son meramente de cálculo. Al ser volátil el sistema requiere de un soporte donde se almacenen los datos aun sin disponer de alimentación o energía.

## Puertos de Entrada / Salida (I/O)

Son puntos (nodos) en los que los dispositivos periféricos se pueden conectar y pueden intercambiar información con la memoria y el procesador central. Los puertos contienen en sí mismos un número definido de registros, los cuales se utilizan para el almacenamiento temporal de varios tipos de datos. Las direcciones de los registros y sus funciones están definidas con precisión (standard).

Tipos:

- puertos serie, en los que los datos se transfieren bit a bit de forma secuencial (COM1, COM2);
- puertos paralelos, en los que los datos se transfieren en paralelo (en un byte, una palabra, etc.) (LPT1).
- puertos universales (USB).

Conclusión:

Aquí vemos una pequeña descripción de lo que es un microcontrolador, sus componentes y hasta que llega a manejar como lenguaje. Esta es una herramienta muy útil en nuestra carrera lo cual es bueno que sepamos su estructura

Bibliografía

[http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion\\_de\\_referencia\\_ISE5\\_3\\_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE5_3_1.pdf)