

selección de sistemas embebido.

PROGRAMACIÓN DE SISTEMAS EMBEBIDOS.



31 de enero de 2020

MURGUÍA CHÁVEZ NADIA SARAHI

ING. MECATRÓNICA 8vo A

Contenido

[¿Qué son los sistemas embebidos? 2](#_Toc31336463)

[¿Cuáles son las características básicas de los sistemas embebidos (ES)? 3](#_Toc31336464)

[¿Cuáles son sus requerimientos? 4](#_Toc31336465)

[¿Qué conocimiento se necesita? 5](#_Toc31336466)

# ¿Qué son los sistemas embebidos?

Los sistemas embebidos, también llamados sistemas empotrados, son sistemas operativos creados con el fin de ser controlados por microprocesadores o microcontroladores, de igual manera los sistemas normales, pero llevados a un fin completamente sistematizado y sin llevar tantas tareas, son, mejor dicho, sistemas que cumplen con una tarea en específico. Tal es el caso de un creador de sistemas embebidos de open source[[1]](#footnote-1) llamado arduino[[2]](#footnote-2), esté por lo regular, las tareas que quieren llevar a cabo se crean en un archivo .ino[[3]](#footnote-3) el cual esta creado con base en C++[[4]](#footnote-4), esto con el fin de mencionar uno de los sistemas de Hardware más populares en estos tiempos.

Los sistemas embebidos se pueden catalogar como:

* Sistemas embebidos pequeños (SES)
* Sistemas embebidos grandes (LES)

Dentro de los cuales se pueden encontrar actuadores, sensores y módulos E/S[[5]](#footnote-5), los cuales se incluyen dentro de las principales funciones de los mismos, por lo regular y como se ha mencionado anteriormente tiene como tarea satisfacer necesidades específicas tales como los celulares, routers, reproductores multimedia, sistemas satelitales, etc.

Estos sistemas están programados en lenguajes nativos, esto con el fin de satisfacer las necesidades de eficacia, excelencia y pronta respuesta, se dice que la mayoría actúan en tiempo real, pero también se sabe que este término es completamente equivoco y no se puede realizar aun, aunque se diga que lo es, por las micras de segundo no se realiza en ese mismo punto de tiempo, de alguna manera el hombre no lo alcanza a percibir, por eso es que se tiende a decir que realmente lo es.

Dentro de los sistemas en “tiempo real” se tienen a aquellos que interactúan en su entorno físico y que responden a los estímulos, esto en un plazo de tiempo determinado, por mencionar algunos ejemplos tenemos aquellos que tienen que tener un muy corto plazo de tiempo a la hora de respuesta, tal y como un sistema de frenos, también se encuentran otros que tienden a ser un poco más flexibles tales como la adquisición de datos en una zona meteorológica, o también se tienen aquellos que su tiempo de respuesta equivale a los datos de realización tales como los sistemas multimedia.

Los sistemas embebidos en cuanto a tamaño se categorizan por diversos factores tales como el tamaño físico que ocupan, y sus capacidades que lo componen, capacidades tales como el procesador, tamaño de memoria, etc.

Los sistemas embebidos son fabricados para una tarea en específico y por medo de estos se pueden generar infinidad de dispositivos, en breve descripción es el futuro de la nueva manera de vivir, hablo de IOE[[6]](#footnote-6), prácticamente todos los sistemas generados con esta estructura y con estas bases cada vez son más factibles.

# ¿Cuáles son las características básicas de los sistemas embebidos (ES)?

Las características básicas de los sistemas embebidos son las siguientes:

* Deben ser confiables
* La confiabilidad en ingles reliability R(t), es la probabilidad de que el sistema trabaje correctamente dado que está funcionando en t=0.
* La mantenibilidad, en ingles 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 sistema está funcionando en el tiempo t.
* La seguridad informática: consiste en disponer de una comunicación confidencial y autentica.
* 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.

Hay que tener en cuenta que respuestas correctas pero tardías son erróneas. Una restricción de tiempo real se dice DURA ó ESTRICTA (hard) si su incumplimeinto puede resultar una catástrofe.

Toda otra restricción de tiempo es blanda (soft).

La mayoría de los sistemas embebidos son de tiempo real (Real-Time) y la mayoría de los sistemas de tiempo real son embebidos.

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 retorno” (Berge, 1995)

Podemos añadir también que los sistemas embebidos poseen un número limitado de funciones predefinidas para actuar, tienen una fuente de alimentación limitada y una administración de energía efectiva.

Poseen disponibilidad de recursos de reservas para situaciones inesperadas.

Los sistemas embebidos suelen tener en unas de sus partes una computadora con características especiales conocida como microcontrolador que viene a ser el cerebro del sistema, el cual incluye interfaces de entrada/salida en el mismo chip. Normalmente estos sistemas poseen una interfaz externa para efectuar un monitoreo del estado y hacer diagnósticos del sistema.

Además cabe reseñar que el uso de sistemas embebidos en productos complejos implica un desafío de la seguridad en TI[[7]](#footnote-7) para proteger la información contenida en el sistemas embebido y también la que es transmitida desde y hacia el dispositivo por redes privadas o Internet. Por tanto, cabe verificación, así como servicios de pruebas de seguridad, así como evaluación específica para sistemas embebidos.

El diseño de un producto que incorpora sistemas embebidos está orientado a minimizar los costos y máxima la confiabilidad, pero también es imprescindible incorporar en el diseño consideraciones de seguridad, incluyendo funciones y protocolos criptográficos que protejan la información durante todas las fases. Los sistemas embebidos a menudo operan en un ambiente dedicado con condiciones operacionales y escenarios muy específicos. Es importantes que dichas condiciones y amenazas se tengan en cuenta cuando se diseñan las funciones de seguridad.

Tradicionalmente esta ha sido realizada a través de una terminal serie, pero con el tiempo la industria ha observado las ventajas del monitoreo a distancia, así como también la posibilidad de efectuar pequeños ajustes sin necesidad de estar físicamente en el mismo lugar donde surten efecto dichos cambios.

Existen varias interfaces:

* Las interfaces de operador (Hombre-Máquina-HMI) – monitoreo, interruptores, botones, indicadores, emisores individuales o grupales y otros.
* Se puede aplicar en los trenes. Las características del software son la información, diseño atractivo, flexibilidad de proyecto.
* Las interfaces eléctricas (interfaces con otros componentes y dispositivos): interno – I2C, SPI, ISA y otros.
* Las interfaces exteriors – RS232, TTY, Ethernet, Centronics, FlexRay, CAN, LIN, RF y otros.

# ¿Cuáles son sus requerimientos?

Históricamente sea cual fuese la función específica del sistema embebido se ha requerido:

Contar con las conectividades en uso (USB, Ethernet, Wifi, Bluetooth, Zigbee, etc.)

Contar con las interfaces de usuario en uso (display LED, touch screen, multimedia, etc.)

Para cumplir éstos requerimientos (en permanente evolución) se hace necesario contar con plataformas (micros/DSP/FPGA/ASIC/etc.) de rendimiento y recursos en crecimiento que permitan atender el incremento del procesamiento necesario para soportar periféricos avanzados con capacidad de atender las nuevas conectividades e interfaces de usuario requeridas por el mercado.

# ¿Qué conocimiento se necesita?

Conocer y evaluar las tecnologías actuales de implementación de sistemas embebidos, en especial microcontroladores de 32 bits.

Conocer y aplicar las mejores prácticas de ingeniería de software, útiles en el desarrollo de ES, que sirva para organizar el ciclo de vida de un proyecto y mejorar la eficiencia del trabajo en equipo.

Explicar el rol de modelado en el desarrollo de ES. Modelar soluciones utilizando los diagramas y las notaciones de uso más frecuente en Es (Diagramas de Estado, de Actividad, etc.)

Desarrollar aplicaciones embebidas en lenguaje C (C++ / etc.), con y sin un sistema operativo de tiempo real (RTOS), empleando técnicas de programación específicas para lograr eficiencia, confiabilidad y re-usabilidad.

Analizar y sintetizar circuitos de apoyo e interfaz.

Contar con la habilidad para escribir documentos relacionados a un desarrollo de ingeniería.

Contar con una base teórico-práctica general sobre SE tal que permita continuar el aprendizaje relativo al área (consolidar los conocimientos teóricos con la práctica debida).

* Existen algunos lineamientos a seguir:
* Simplicidad (KISS “Keep It Simple, Stupid”)
* Excelencia (DFE “Design for Excellence”)
* Documentarse debidamente antes de comenzar el diseño.
* Método de trabajo (contar con uno)

Los sistemas embebidos son herramientas que permiten ofrecer soluciones a aplicaciones antes resueltas mediante el uso de otras tecnologías/técnicas. Sepamos que puede ser necesario recurrir al auxilio de expertos en disciplinas específicas de la aplicación.

# Simplicidad (KISS)

La simplicidad (KISS) es la clave del éxito de un diseño en ingeniería. En el desarrollo de sistemas complejos en ingeniería debemos:

* Desarrollar empleando partes sencillas y comprensibles que redundará en errores de fácil detección y corrección.
* Debemos rechazar lo rebuscado e innecesario.

En otras palabras, advierte al diseñador para que en su labor no compre problemas, sino que venda soluciones.

# Excelencia (DEF)

Excellence (DFE): Manufacture and Assembly, Reliability, Testing and Service, Disassembly and Reassembly, Use and Operability, Green, Environment and Recycling, Quality and Cost, Logistic, Inspection and International, etc.

No implica el cumplimiento de todos estos ítems, pues la actividad de diseño estará fuertemente condicionando por dos factores:

* La idiosincrasia tanto del diseñador como el medio en que este se desempeña
* El contexto en que se lleve a cabo de diseño

Se debe establecer que ítems de excelencia se procura alcanzar y concentrarse en lograrlos.

# Documentar debidamente.

Es recomendable recopilar documentación referida al diseño, a las técnicas y/o herramientas, etc. (Hojas de Datos, Fe de Erratas, Notas de Aplicación, Ejercicios o Ejemplos de Diseño, Manuales de Usuario, Manuales de Referencia Técnica, etc.)

Aprovechar las facilidades de ofrecer las vías de comunicación actuales para la búsqueda de información.

Encarar la búsqueda con sentido común y criterio

La búsqueda en sí misma es un medio y no un fin

Leer, analizar y clasificar la documentación recopilada

Durante la etapa de diseño saque provecho de la información recopilada aprendiendo del trabajo de los demás.

# Método de trabajo

Optamos por el más usado, simple y seguro

Recomendaciones:

* Procure aprender del método
* Procure adaptarlo a su gusto
* Si no está conforme con el método:
* Genere su propio método, pero use no, pues:
* Sin método cada diseño nos obliga a comenzar de cero

Recuerde que cada diseño tiene una solución adecuada

# Sistemas de tiempo real

Un sistema embebido de tiempo real es un que:

Interacciona repetidamente con su entorno físico

Responde a los estímulos que recibe del mismo dentro de un plazo de tiempo determinado

Para que el funcionamiento del sistema sea correcto no basta con que las acciones sean correctas (validez lógica), sino que tienen que ejecutarse dentro del intervalo de tiempo especificado (instante en que se produce la corrección)

El tiempo en que se ejecutan las acciones del sistema es significativo (asegurar: determinismo & tiempo de respuesta; p/tiempo de ejecución: contemplar el caso más desfavorable; atención: tiempo real no es igual a rápido (ver caso por caso)).

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)
5. [↑](#footnote-ref-5)
6. [↑](#footnote-ref-6)
7. [↑](#footnote-ref-7)