

Firmware



Un dispositivo de **firmware** muy común: el control remoto de la televisión.



Memoria de solo lectura que contiene el BIOS de una vieja placa base.

El **firmware** es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo, es el *software* que tiene directa interacción con el *hardware*, siendo así el encargado de controlarlo para ejecutar correctamente las instrucciones externas. De hecho el firmware es uno de los tres principales pilares del diseño electrónico.

En resumen, un *firmware* es un software que maneja físicamente al hardware.

El programa BIOS de una computadora es un *firmware* cuyo propósito es activar una máquina desde su encendido y preparar el entorno para cargar un sistema operativo en la memoria RAM.

Origen del término

El término fue acuñado por Ascher Opler en un artículo de la revista *Datamation*, publicado en 1967.¹ Originalmente, se refería al microshiva - contenido en un almacenamiento de control escribible (una área pequeña especializada de memoria RAM), que definía e implementaba el conjunto de instrucciones del computador. Si fuera necesario, el *firmware* podía ser recargado para especializar o para modificar las instrucciones que podría ejecutar la Unidad Central de Procesamiento (CPU). Según el uso original, el *firmware* contrastaba tanto con el soporte físico (la CPU en sí misma) como con el software (las instrucciones normales que se ejecutan en una CPU). El *firmware* no estaba compuesto de instrucciones de máquina de la CPU, sino del microcódigo de nivel inferior implicado en la implementación de las instrucciones de máquina que iría a ejecutar la CPU. El *firmware* existía en el límite o frontera entre el hardware y el software, por ello el término de *firmware* (que significa 'software firme, fijo o sólido').

Posteriormente, el término fue ensanchado para incluir cualquier tipo de microcódigo, ya fuera en RAM o ROM.


Aún más adelante, el término fue ensanchado otra vez más, en el uso popular, para denotar cualquier cosa residente en ROM, incluyendo las instrucciones de máquina del procesador para el BIOS, los cargadores de arranque, o aplicaciones especializadas.

Firmware hoy en día


El *firmware* ha evolucionado para significar casi cualquier contenido programable de un dispositivo de *hardware*, no solo código de máquina para un procesador, sino también configuraciones y datos para los circuitos integrados para aplicaciones específicas (ASIC), dispositivos de lógica programable, etc.

Hasta mediados de los años 1990 el procedimiento típico para actualizar un *firmware* a una nueva versión era reemplazar el medio de almacenamiento que contenía el *firmware*, usualmente un chip de memoria ROM enchufado en un zócalo. Hoy en día este procedimiento no es habitual ya que los fabricantes han añadido una nueva funcionalidad que permite grabar las nuevas instrucciones en la misma memoria, haciendo de la actualización un proceso mucho más cómodo y dinámico. Aun así el proceso de actualización de un *firmware* hay que realizarlo con mucho cuidado, ya que al ser un componente vital cualquier fallo puede dejar al equipo inservible. Por ejemplo, un fallo de alimentación a mitad del proceso de actualización evitaría la carga completa del código que gobierna el equipo, quizá incluso la carga del código que se encarga de actualizar el *firmware*, así que no podríamos actualizarlo de nuevo y por lo tanto el equipo dejaría de funcionar.


Periféricos de computador

La mayoría de los periféricos del COMPUTADOR  son de hecho computadores de propósito especial. Mientras que los dispositivos externos tienen el *firmware* almacenado internamente, las modernas tarjetas de los periféricos de computadores típicamente tienen grandes partes de *firmware* que es cargado en el arranque por el sistema huésped, pues esto es más flexible. Dicho *hardware* por lo tanto puede no funcionar completamente hasta que el computador huésped le haya cargado el *firmware* indispensable, a menudo por medio del driver de dispositivo específico (o, más exactamente, por medio de un subsistema dentro del paquete del driver de dispositivo). Los modernos drivers de dispositivo también pueden exponer una interfaz directa de usuario para la configuración además de las llamadas/interfaces del sistema operativo o de las Interfaces de programación de aplicaciones (APIs).

Retos del *firmware* en los PC

En algún sentido, los varios componentes del *firmware* son tan IMPORTANTES  como el sistema operativo en un computador. Sin embargo, a diferencia de la mayoría de los sistemas operativos modernos, el *firmware* tiene raramente un mecanismo automático bien desarrollado para actualizarse a sí mismo para corregir los problemas de funcionalidad que son detectados después de que la unidad es despachada.

BIOS y *firmware*

El BIOS es bastante fácil de actualizar en un PC moderno; los dispositivos como las tarjetas de vídeo o los módems confían en el *firmware* cargado dinámicamente por un *driver* de dispositivo y a menudo pueden así ser actualizados transparentemente a través de los mecanismos de actualización del sistema operativo. En contraste, el *firmware* en dispositivos de almacenamiento es raramente actualizado; no están estandarizados los mecanismos para detectar las versiones del *firmware* y actualizarlas. Estos dispositivos, por lo tanto, tienden a tener un índice más alto de problemas de funcionalidad, comparados con otras partes de un moderno sistema de COMPUTACIÓN [cita requerida]

Reproductores de música portátiles[editar]

Algunas compañías usan actualizaciones del *firmware* para agregar nuevos formatos de archivos de reproducción de sonido (*codecs*); el iriver agregó el formato ogg de esta manera, por ejemplo. Otras características que pueden cambiar con las actualizaciones del *firmware* son el GUI e incluso la vida de la batería. La mayoría de los reproductores de música modernos soportan las actualizaciones del *firmware*.

Teléfonos móviles

La mayoría de los teléfonos móviles tienen una capacidad de *firmware* actualizable por muchas de las mismas razones que se especificaron anteriormente, pero algunos incluso pueden ser actualizados para mejorar la recepción o la calidad del sonido.

Automóviles

Desde 1996 la mayoría de los automóviles han empleado una computadora a bordo y varios sensores para detectar problemas mecánicos. Los vehículos modernos también emplean sistemas controlados por computador, ABS y sistemas de control de transmisión operados por computadora. El conductor puede también recibir información "in-dash" de esta manera mientras conduce, como, por ejemplo, lecturas en tiempo real de la economía del combustible y de la presión del neumático. La mayoría del *firmware* del vehículo puede ser actualizado en un distribuidor local autorizado.

Definición del IEEE

El glosario estándar de terminología del software del Institute of Electrical and Electronics Engineers (IEEE), Std 610.12-1990, define el firmware como sigue:

"La combinación de instrucciones de un dispositivo de hardware e instrucciones y datos de COMPUTADORA que residen como software de solo lectura en ese dispositivo".

Notas:

(1) este término es a veces usado para referirse solamente al dispositivo de hardware o solamente a las instrucciones o datos de computadora, pero estos significados están desaprobados.

(2) la confusión rodeando este término ha llevado a alguno a sugerir que éste debe ser totalmente evitado.

Ejemplos

Ejemplos de *firmware* incluyen:

- Sistemas de temporización y control para las lavadoras.
- El BIOS encontrado en computadores personales compatibles con el IBM PC.
- El código de plataforma encontrado en los sistemas Itanium, las máquinas Mac OS X basadas en Intel y muchas tarjetas de escritorio de Intel tienen *firmware* EFI.
- Open Firmware, usado en computadores de Sun Microsystems, Apple Computer, y de Genesi.
- ARCS, usado en computadores de Silicon Graphics.
- Kickstart, usado en la línea de computadoras Amiga. Este es un ejemplo único de un firmware que mezclaba las capacidades del BIOS (POST, inicio del *hardware* + autoconfiguración, *plug and play* de periféricos, etc.), del *kernel* en sí mismo del AmigaOS y de partes del código usadas en el sistema operativo (como primitivas y bibliotecas de la interfaz gráfica).
- RTAS (*Run Time Abstraction Services*), usado en computadoras de IBM.
- Chips de EPROM usados en la serie Eventide H-3000 de procesadores digitales de música.
- El Common Firmware Environment (CFE).
- Controlar los atributos de sonido y vídeo y la lista de canales en los televisores modernos.

Hackeo del *firmware*

A veces una nueva o modificada versión no oficial de *firmware* es creada por terceros para proporcionar nuevas características o para abrir una funcionalidad

oculta. Los ejemplos incluyen Whited00r para iPhone e iPod Touch, Rockbox para los reproductores de audio digital, CHDK² y Magic Lantern^{2 3} para las CÁMARAS digitales de Canon, OpenWrt para los enrutadores inalámbricos,² el firmware de región libre para unidades de DVD, que no son de región libre con el *firmware* oficial, así como muchos proyectos de *homebrew* para las consolas de juego. Estos pueden a menudo abrir funcionalidad computacional en dispositivos previamente limitados (ej., ejecutar *Doom* en iPods). La mayoría de los *hacks* de *firmware* son libres y también de código abierto.

Estos *hacks* usualmente utilizan la facilidad de actualización del *firmware* en muchos dispositivos para instalarse o ejecutarse en ellos. Algunos, sin embargo, deben recurrir a *exploits* para funcionar, porque el fabricante ha intentado bloquear el *hardware* para evitar que utilicen código no autorizado.