Co-funded by the
Erasmus+ Programme
of the European Union

Sistemas Embebidos

Conceptos generales

¿Qué es un Sistema Embebido?

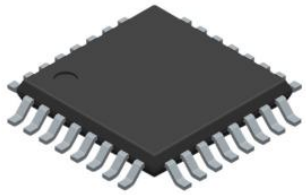
¿Qué es un Sistema Embebido?

Es un sistema de computación basado en un microprocesador o un microcontrolador diseñado para realizar una o algunas pocas funciones dedicadas.

Desarrollo de un Sistema Embebido

- Distintas disciplinas – electrónica, software, diseño industrial...
- Diseño específico
- Confiable y robusto (marcapasos, sistema de frenado, satélite)
- Tiempo real
- Mínima interfaz de usuario (o nula)
- Bajos recursos. Bajo consumo. Bajo costo.

Procesadores usados



Microcontroladores

- B / KB de memoria

Microchip (PIC)

Atmel (AVR, ATMega)

Espressif ESP32

Microprocesadores / SOM

- MB / GB de memoria

ARM

RISC-V

ASICs

Lenguajes de programación

- C es el lenguaje por excelencia para SSEE
  - Compromiso entre facilidad para entender y mantener (comparado con Assembly) y eficiencia (comparado con lenguajes de más alto nivel)
- También se usan muchas otros lenguajes, dependiendo el caso, por ejemplo C++, Python, Java, Lua, Rust, entre otros.
- Los lenguajes de más alto nivel se suelen usar en micros más potentes en combinación con algún sistema operativo.

Eficiencia

- Eficiencia en tiempo de ejecución:

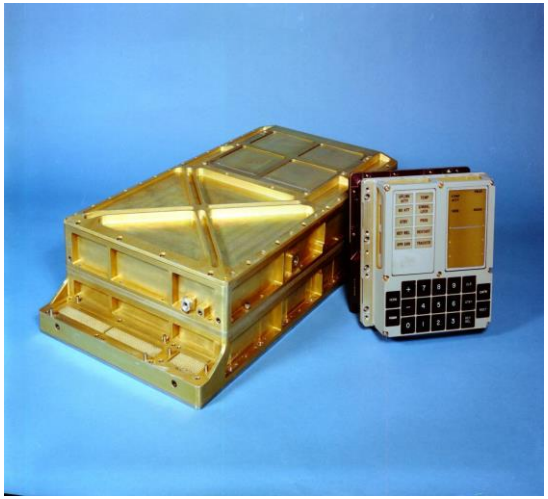
	Time (ms)
(c) C	1.00
(c) Rust	1.04
(c) C++	1.56
(c) Ada	1.85
(v) Java	1.89
(c) Chapel	2.14
(c) Go	2.83
(c) Pascal	3.02
(c) Ocaml	3.09
(v) C#	3.14
(v) Lisp	3.40
(c) Haskell	3.55
(c) Swift	4.20
(c) Fortran	4.20

	Time (ms)
(v) F#	6.30
(i) JavaScript	6.52
(i) Dart	6.67
(v) Racket	11.27
(i) Hack	26.99
(i) PHP	27.64
(v) Erlang	36.71
(i) Jruby	43.44
(i) TypeScript	46.20
(i) Ruby	59.34
(i) Perl	65.79
(i) Python	71.90
(i) Lua	82.91

Fuente: https://haslab.github.io/SAFER/scp21.pdf

Eficiencia

Apollo Guidance Computer



2Mhz Clock
4KB RAM
32KB ROM

¿Cómo se programan?

- Bare Metal
- RTOS – Real Time Operating System
- Sistema Operativo Embebido

La decisión depende de varios factores, como la potencia del micro o la complejidad y necesidad de control del proyecto

Bare Metal

- No hay sistema operativo, ni ninguna capa de abstracción.
- El único código que corre es el que se programa.
- Positivo: no existe overhead y el manejo de recursos se hace de forma más eficiente.
- Negativo: se torna muy complejo para proyectos grandes, con varias tareas, que necesitan escalabilidad.
- Lenguajes de programación: Assembly, C.

RTOS - Real Time Operating System

- Funcionalidades core de un sistema operativo – scheduler, manejo de memoria, sincronización y comunicación entre tareas (mutex, semáforos, queues...)
- Clave: tiempo real
- Lenguajes de programación: generalmente C o C++
- Ejemplos: FreeRTOS, Zephyr, embOS

Sistema Operativo Embebido

- Versión más compacta y eficiente (recortada) de un sistema operativo tradicional
- Mayor cantidad de servicios en comparación con opciones anteriores
- Lenguajes de programación: Cualquiera...
- Ejemplos: OpenWRT, Ubuntu Core.
- Generadores de imágenes: Yocto, Buildroot

Firmware vs Software Embebido

- Frontera difusa
- Firmware es un tipo de software
  - Asociado históricamente a software grabado en ROM durante fabricación
  - El software que hace que el hardware funcione
  - Puede ser una parte (o todo) el software en un sistema embebido
  - No necesariamente presente en un sistema embebido. Por ejemplo: BIOS
- Software embebido es simplemente software que corre sobre un sistema embebido