



ELECTRÓNICA DIGITAL III

CLASE 1-B

Arquitectura en N-Capas para Software de Sistemas Embebidos en Lenguaje C

Ing. Emiliano Migliore

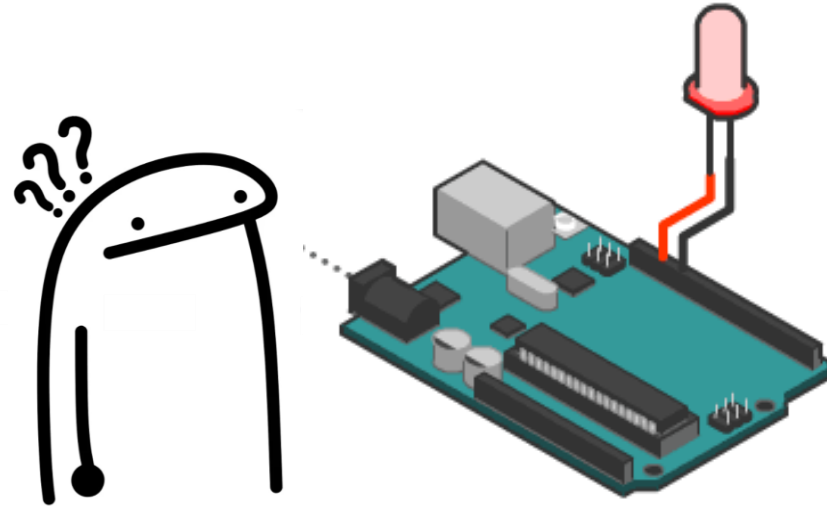
Programación en C para Sistemas Embebidos

Alguna vez te has preguntado que hay detrás de las funciones para programar microcontroladores como Arduino?

```
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
}
```

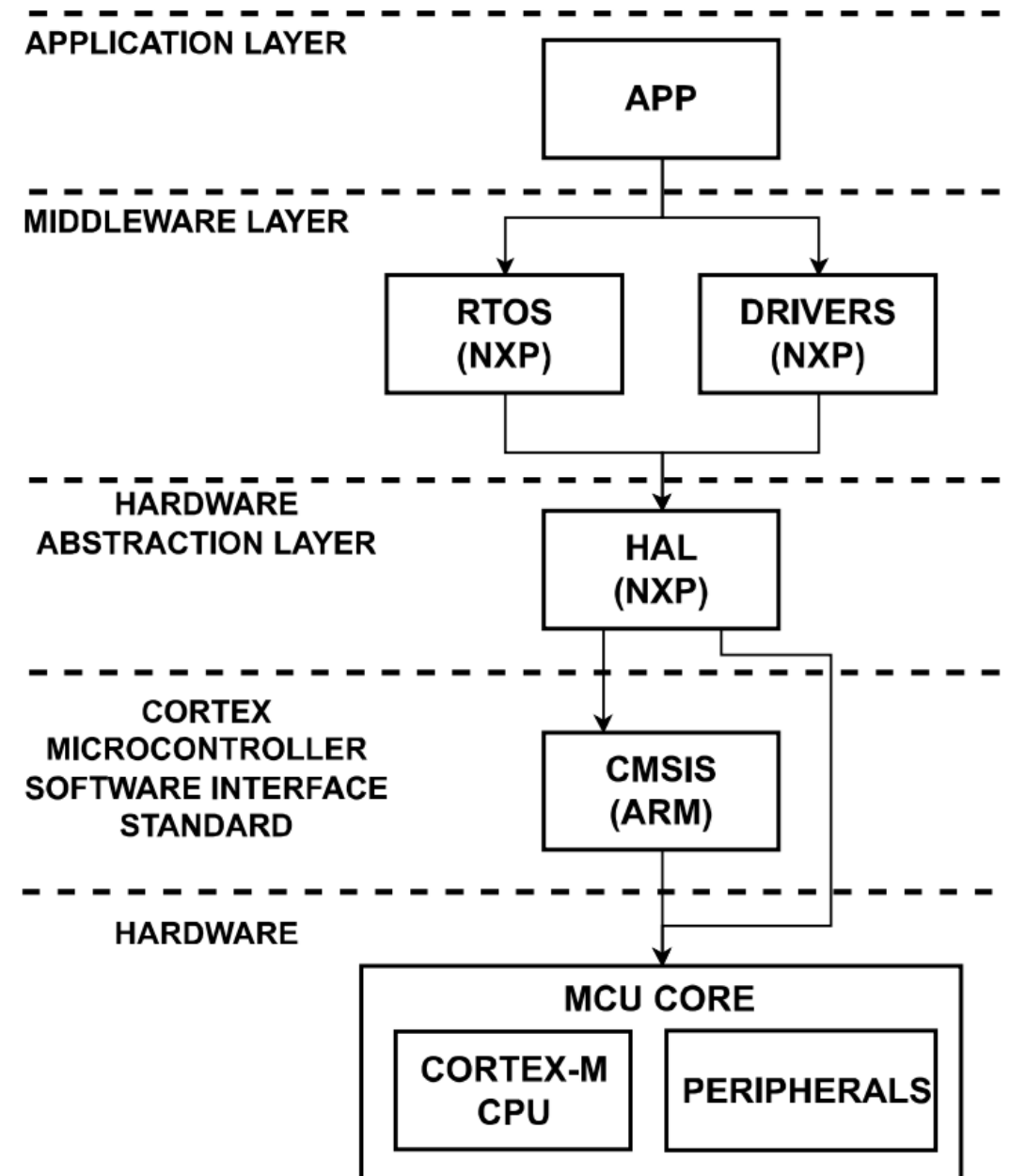


Arquitectura de Software en N-Capas en Sistemas Embebidos

En el contexto de software embebido, una arquitectura en n-capas (o multicapa) es un modelo de diseño estructurado que divide el software en capas lógicas independientes, cada una encargada de una función específica dentro del sistema. Esta separación permite una mejor modularidad, escalabilidad, mantenimiento y reutilización del código.

PRINCIPALES CARACTERÍSTICAS:

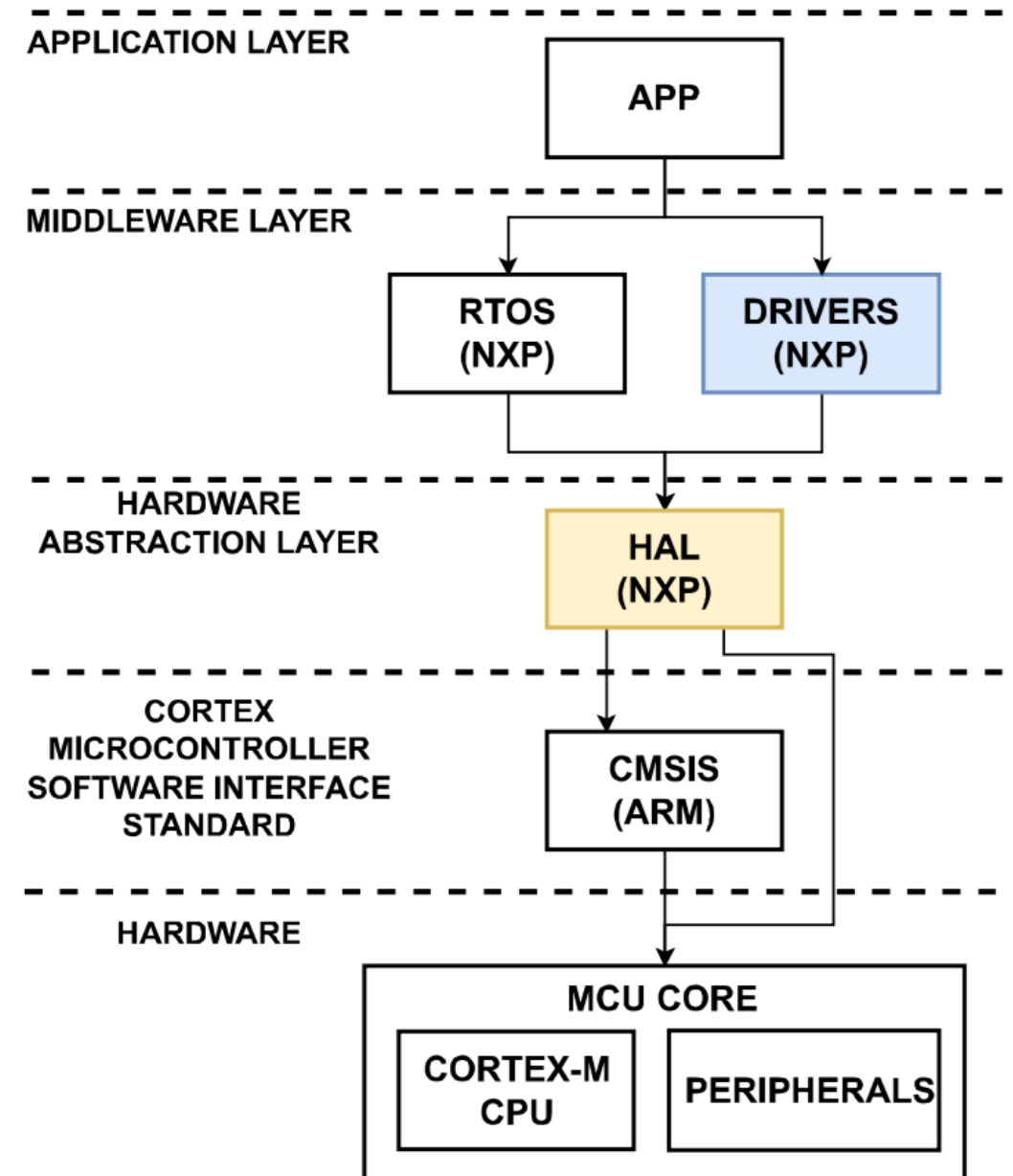
- **Separación de responsabilidades:** cada capa tiene un propósito claro.
- **Aislamiento de cambios:** cambios en una capa tienden a no afectar otras.
- **Comunicación unidireccional:** las capas superiores pueden usar servicios de capas inferiores, pero no al revés.
- **Fácil depuración y prueba:** se puede probar una capa de manera independiente (unit testing).



Capa de Abstracción de Hardware de un MCU

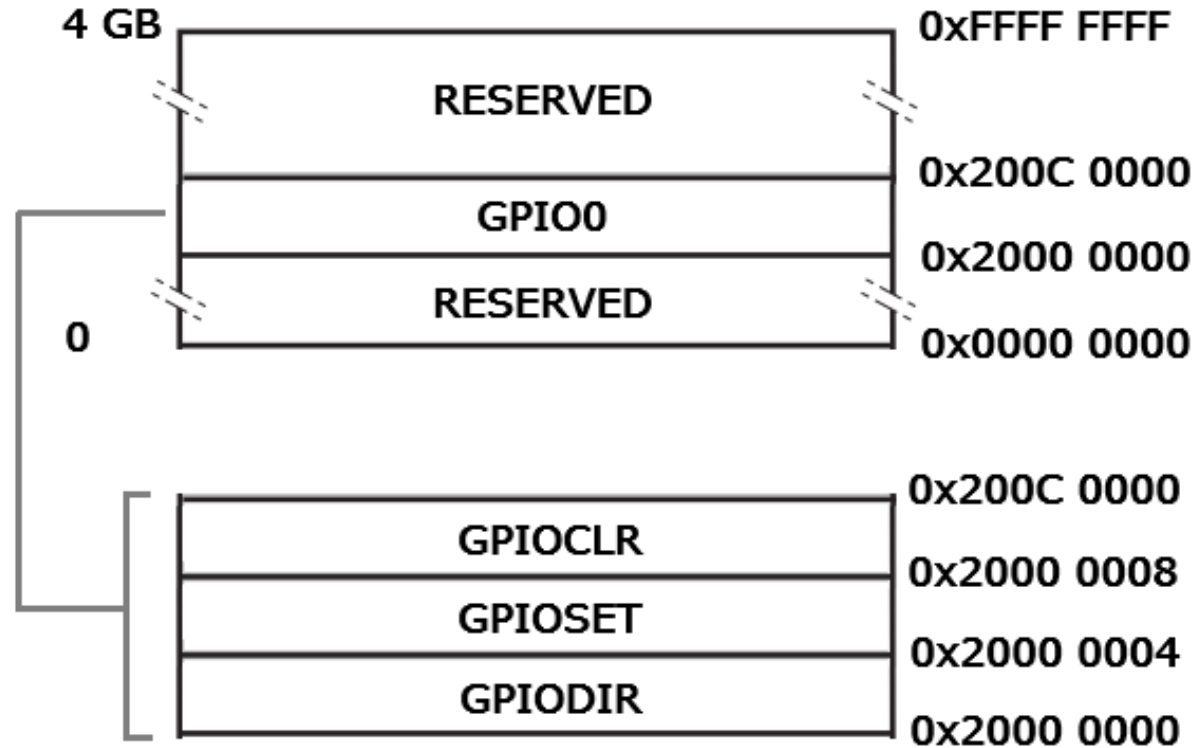
La capa HAL (Hardware Abstraction Layer) es una capa de software que funciona como intermediaria entre el hardware físico y el firmware de aplicación.

- Abstrae o "esconde" los detalles específicos del hardware.
- Permite que el software use el hardware sin tener que conocer cómo está construido o cómo funciona internamente.
- Facilita que un mismo programa funcione en diferentes dispositivos o microcontroladores con pocos o ningún cambio, proveyendo funciones o drivers estandarizados para manejar periféricos (como puertos GPIO, UART, ADC, TIMERS).



Diseño de la Capa de Abstracción de Hardware de un MCU

Dado el siguiente fragmento del mapa de memoria de un MCU de 32-bits, implementar en lenguaje C la capa de abstracción de hardware (HAL) para el periférico GPIO0 del MCU.



$$C_{sist} = \frac{N_{sist}^A}{1GB} = \frac{2^{n_{sist}^A}}{2^{30}} = \frac{2^{32}}{2^{30}} = 2^2 = 4GB$$

$$4GB \equiv 0x100000000_H \Rightarrow [0x00000000, 0x\underbrace{\text{FFFFFFFF}}_{8B}]_H$$