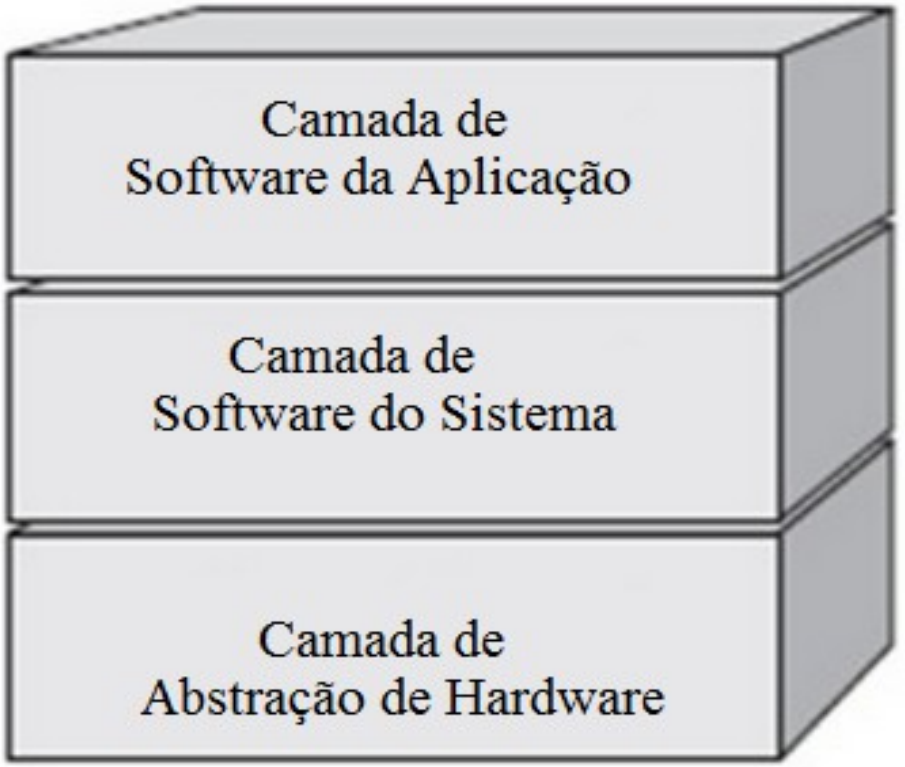
Sistemas embarcados

Introdução ao projeto de software de sistemas embarcados

Projeto de software

O projeto de software (em geral, bem como para sistemas embarcados) é organizado seguindo um **modelo de camadas.**



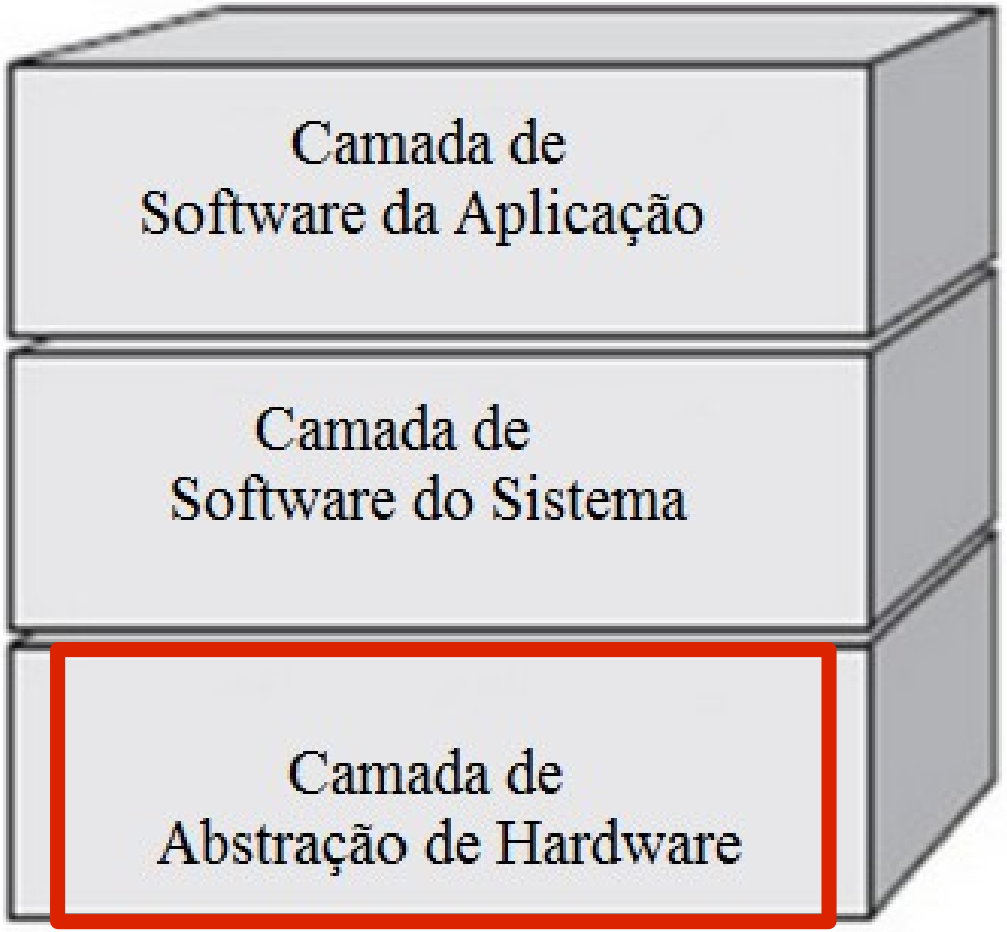
Aplicação (ex.: codificador/decodificador MP3, servidor HTTP, interface gráfica (GUI)

Sistema operacional (SO), bibliotecas do sistema (ex.: TCP/IP, sistema de arquivos, etc)

Acesso a CPU, memórias, E/S e periféricos através de drivers de dispositivos

rcados

Projeto de software



A **camada de abstração de hardware** (HAL) é aquela que tem acesso **direto aos registradores dos periféricos e da CPU.**

Camada de abstração de HW

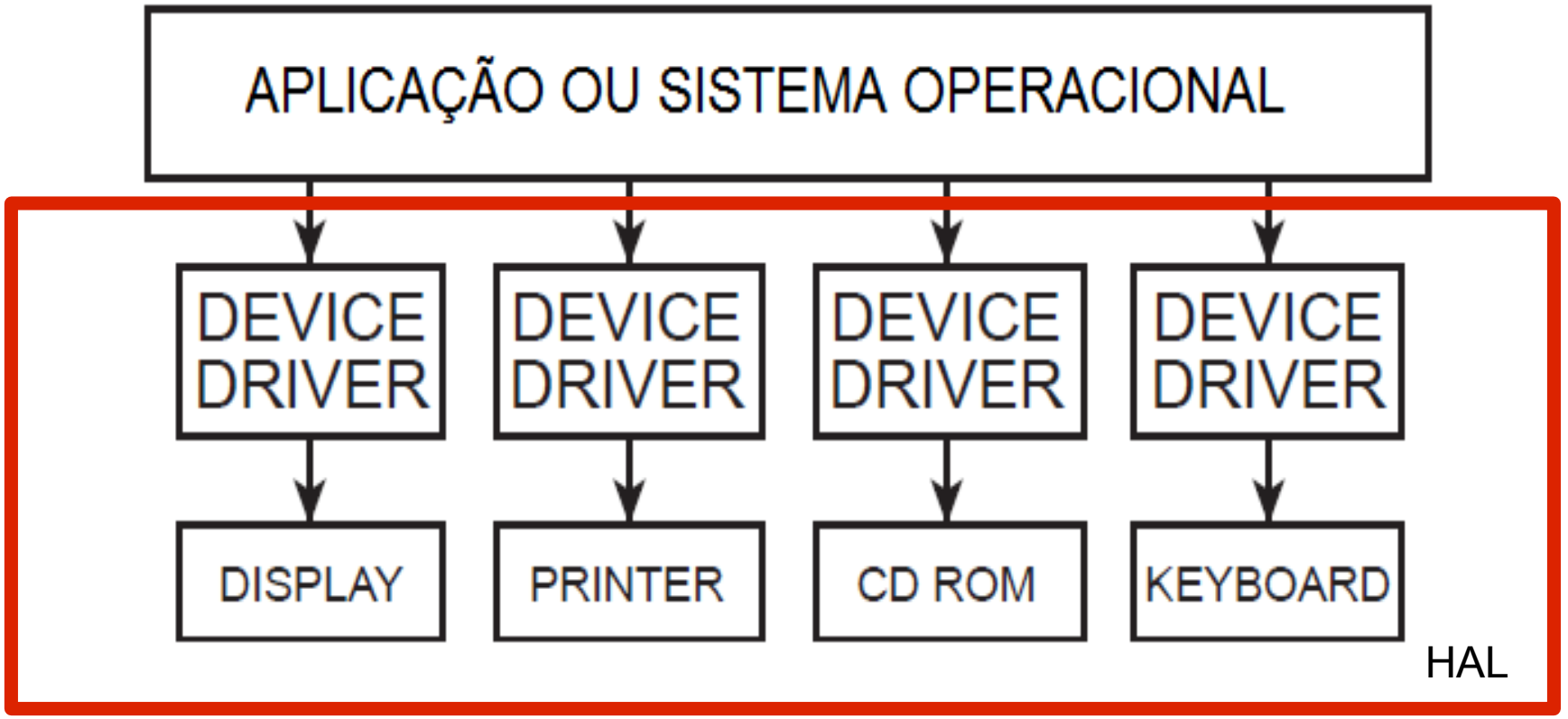
A **camada de abstração de hardware** (Hardware Abstraction Layer - HAL) é composta pelos ***drivers de dispositivos (device drivers)*****.**

Um **driver de dispositivo** funciona como um **tradutor** entre o **dispositivo** e as **aplicações** ou o sistema operacional.

Permite a **interação** com o dispositivo através de **comandos** de software **abstratos**.

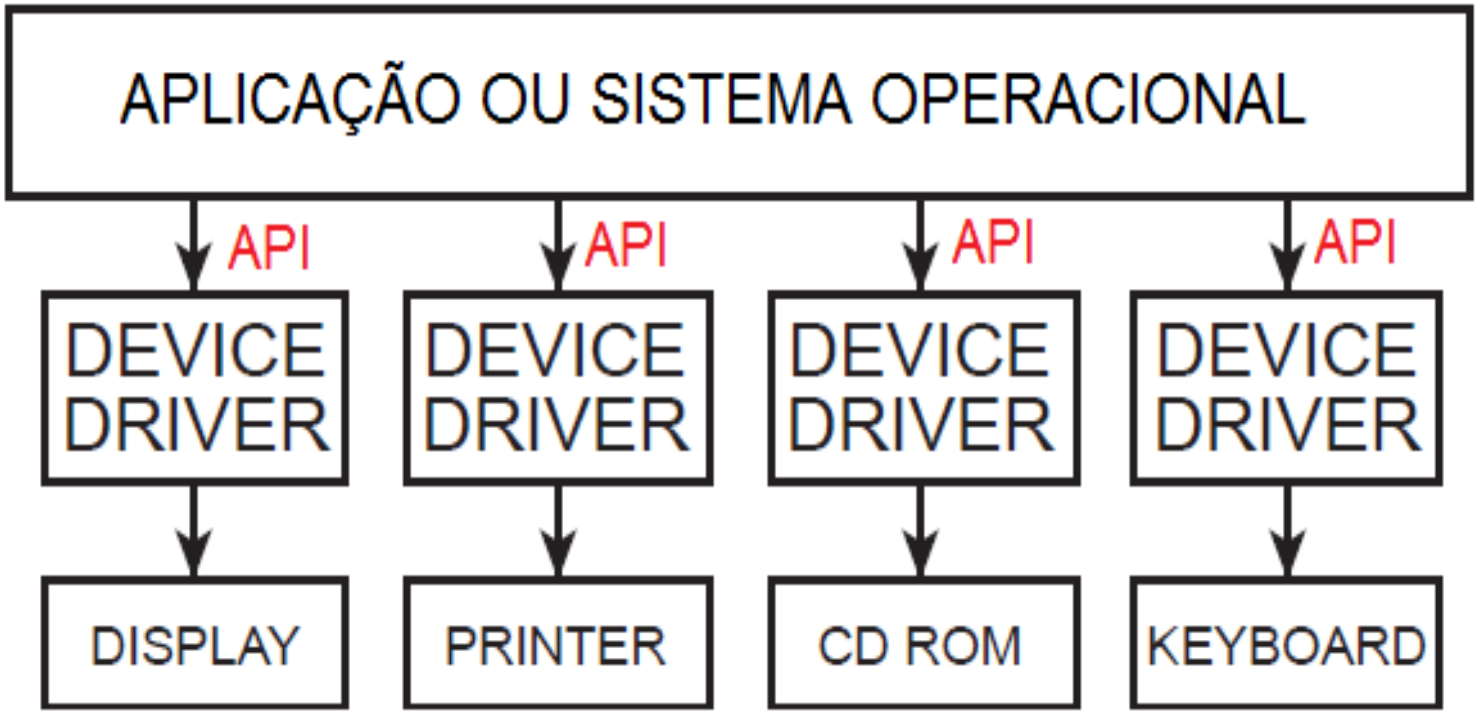
Torna desnecessário para as demais partes de software saber **detalhes** de implementação do **dispositivo** (ex.: endereço de **registradores**, programação dos registradores, etc).

Camada de abstração de HW



Camada de abstração de HW

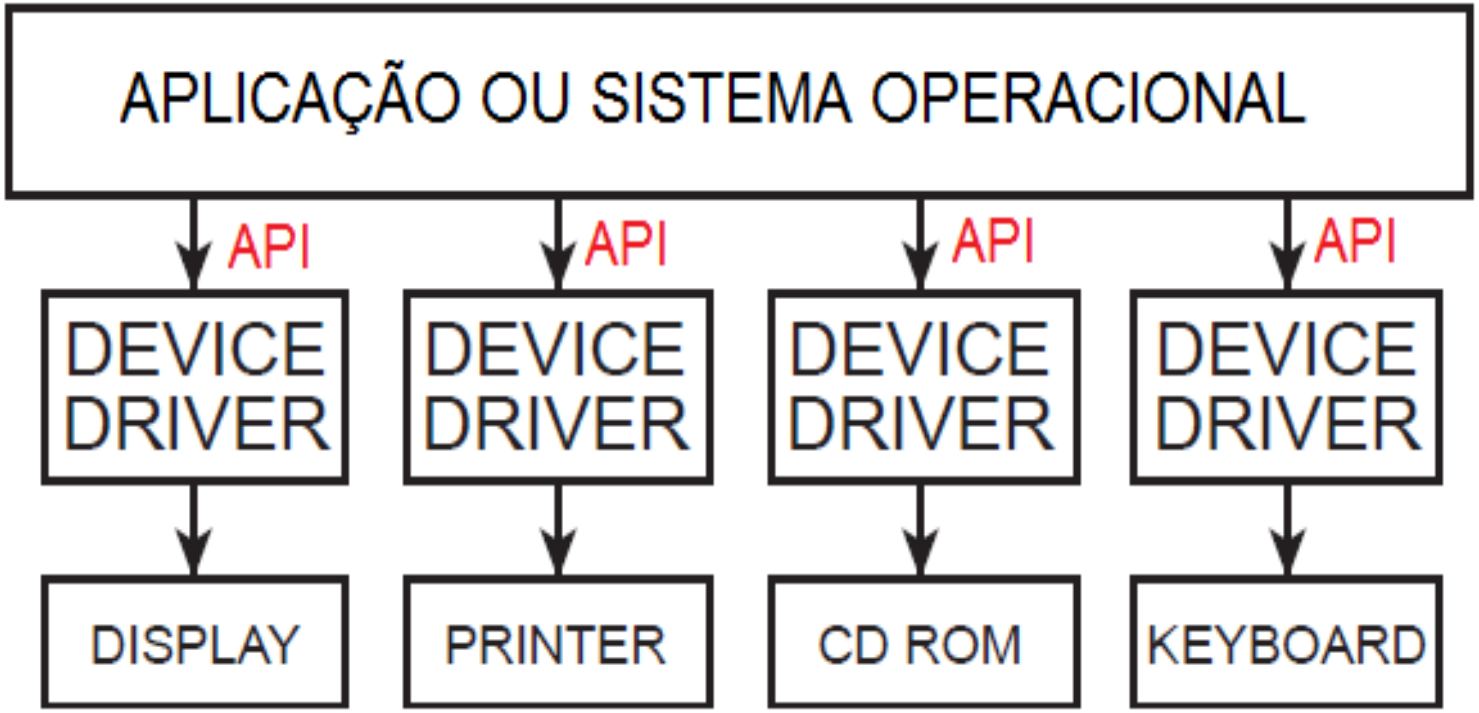
Drivers de dispositivos são acessados através de uma **interface** (**API** – Interface de Programação de Aplicação)



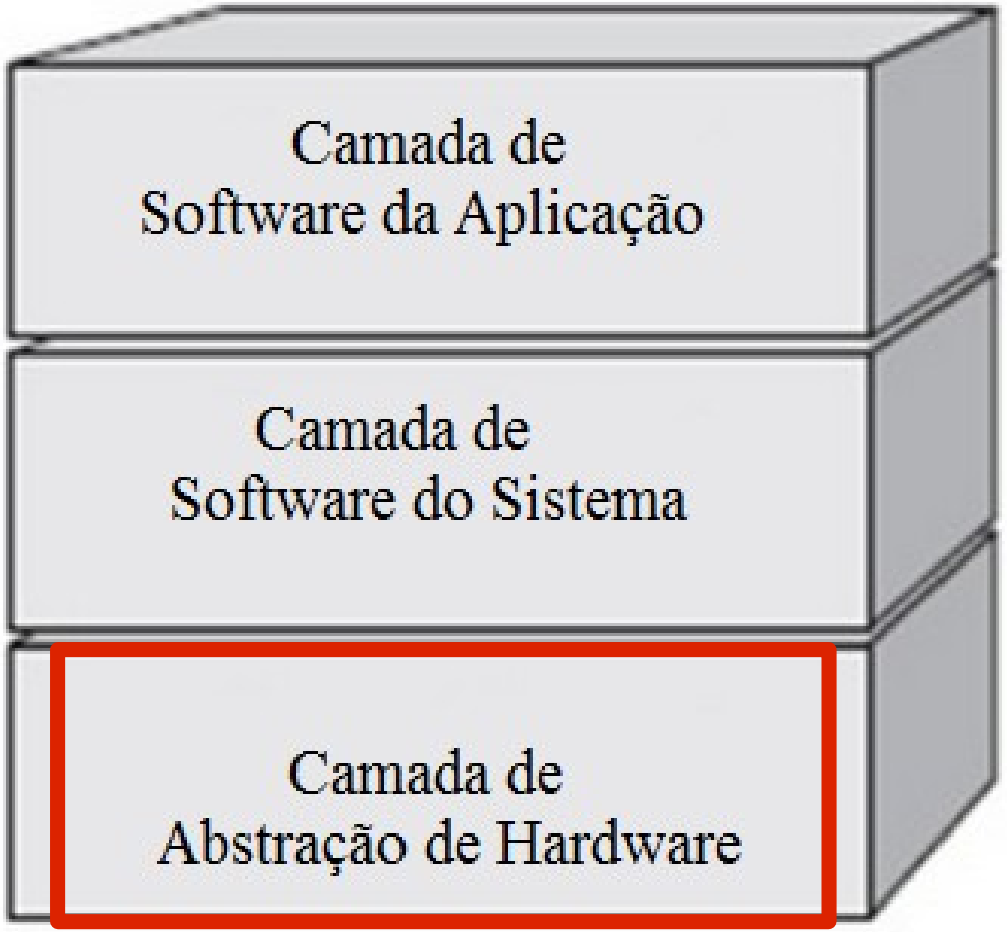
Camada de abstração de HW

Uma API define uma biblioteca de funções. Ex.:

- para imprimir na impressora usa-se a função “Printer_print (char c)”
- para imprimir no display usa-se “Display_print (char c)”

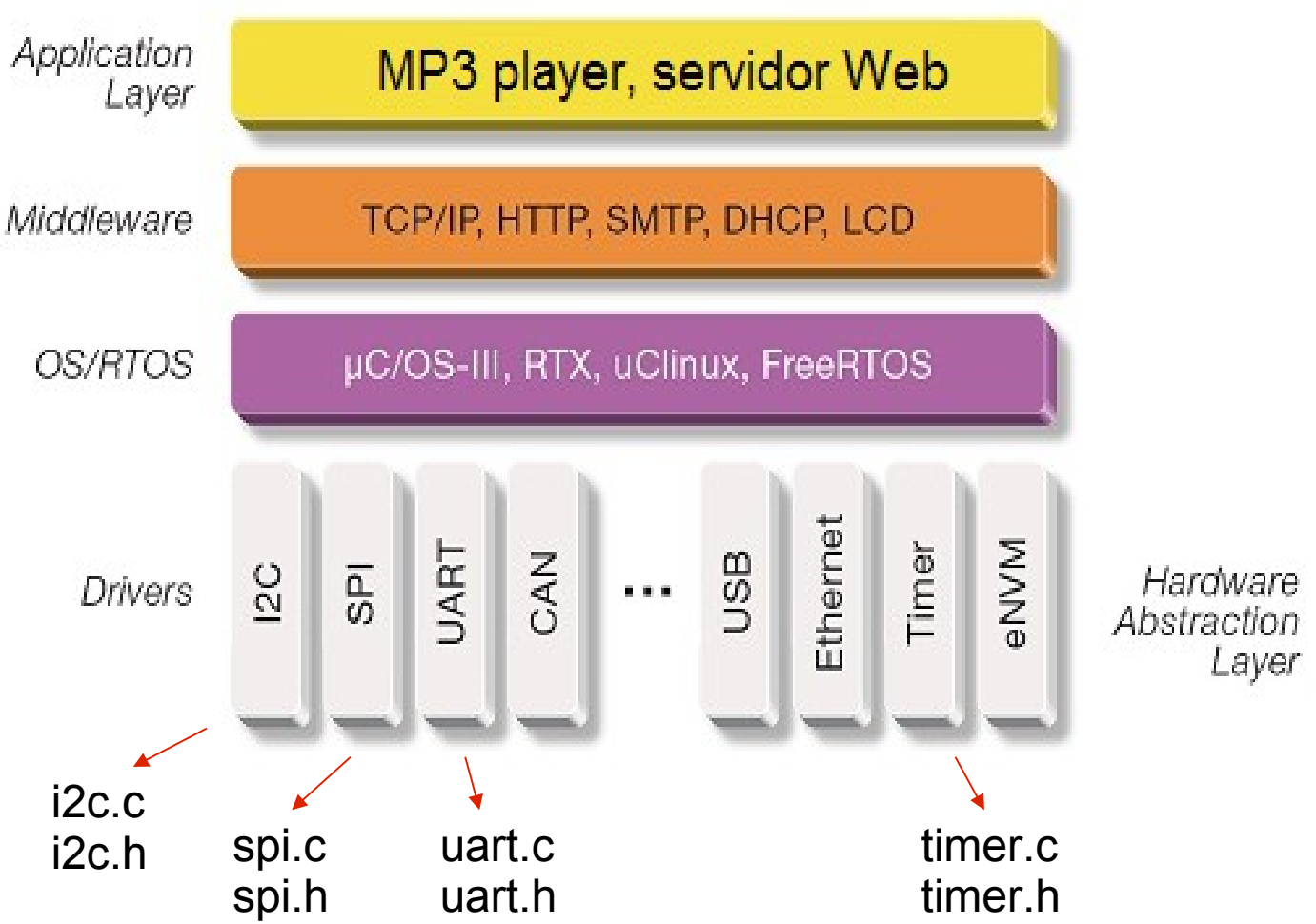


Projeto de software



A **camada de abstração de hardware** (HAL) é aquela que tem acesso **direto aos registradores dos periféricos e da CPU.**

Camada de abstração de HW



Camada de abstração de HW

Exemplo de API:

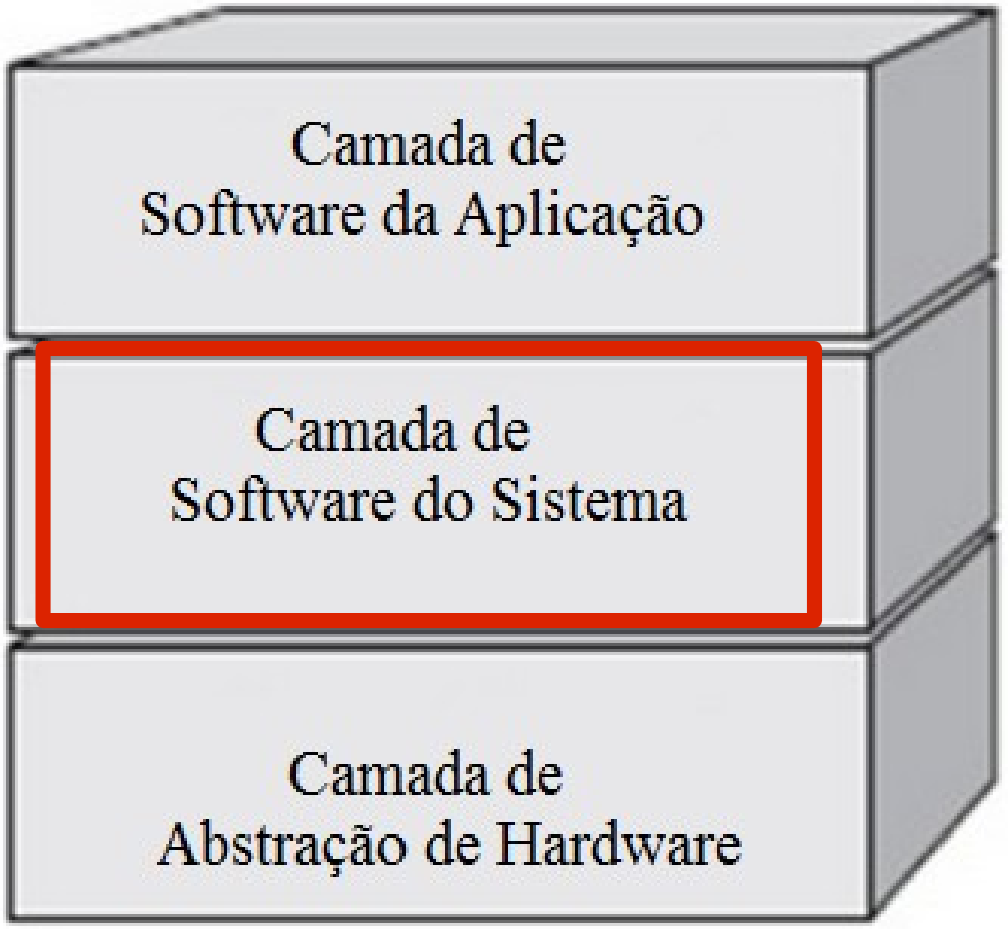
Driver UART.c/UART.h

UART_ini (baud_t BAUDRATE, par_t PARIDADE, …)

UART_tx(char)

UART_rx(char *)

Projeto de software



A **camada de software do sistema** geralmente é implementada como um **sistema multitarefas**:

como uma máquina de estados finitos (**FSM**) ou

um sistema operacional de tempo real (**RTOS**)

Camada de software do sistema

A **camada de software do sistema** define a estrutura principal de todo o software.

O projeto da camada de software do sistema se dá entre duas formas básicas (modelos de programação):

baseado em eventos ou **baseado em *threads***

Threads x eventos

Threads

Execução de uma aplicação através da **divisão** da mesma em **duas ou mais tarefas** executadas **concorrentemente** (*multithreading*).

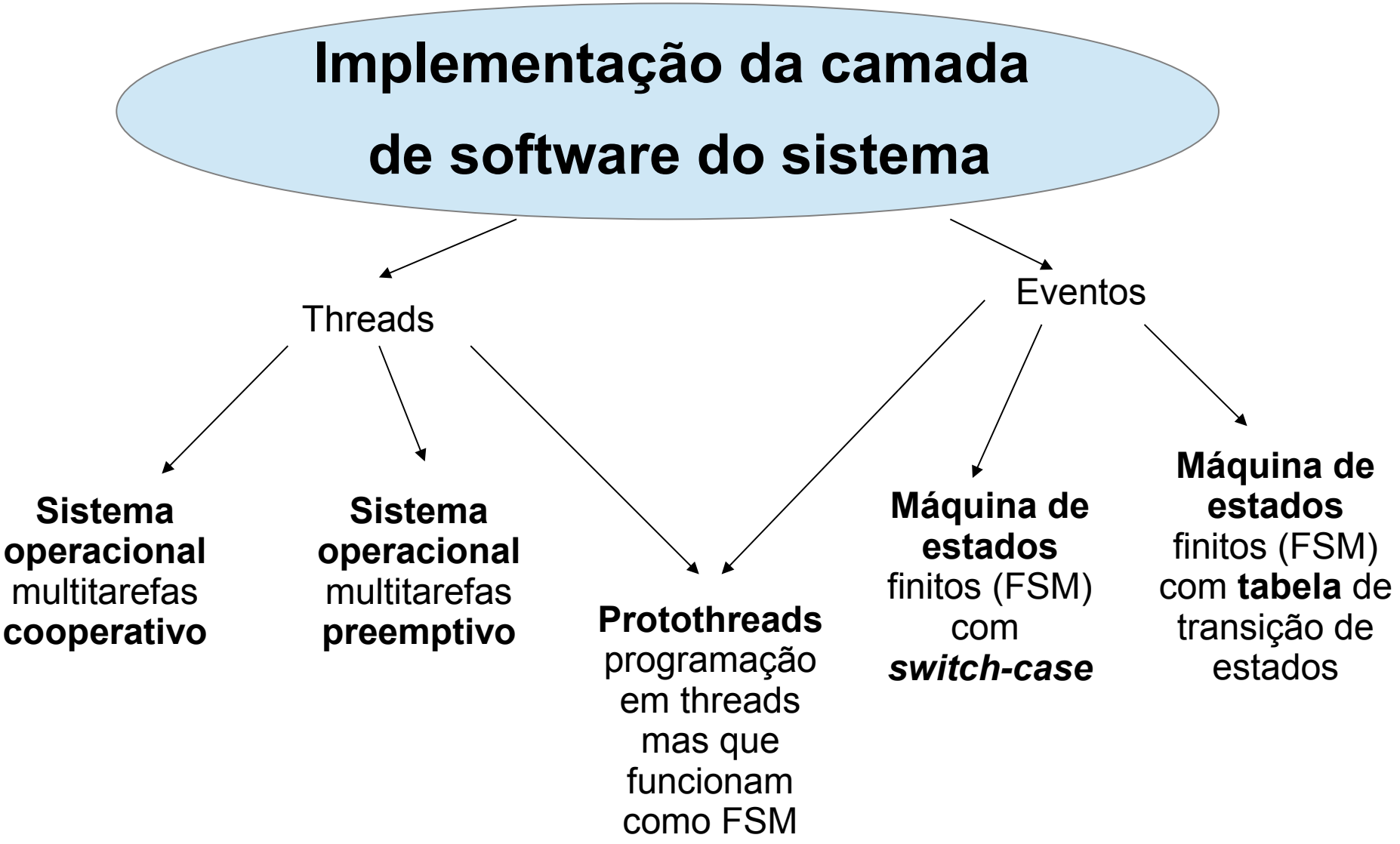
O gerenciamento e a **execução de threads** é uma função básica de um **sistema operacional** (SO)

Eventos

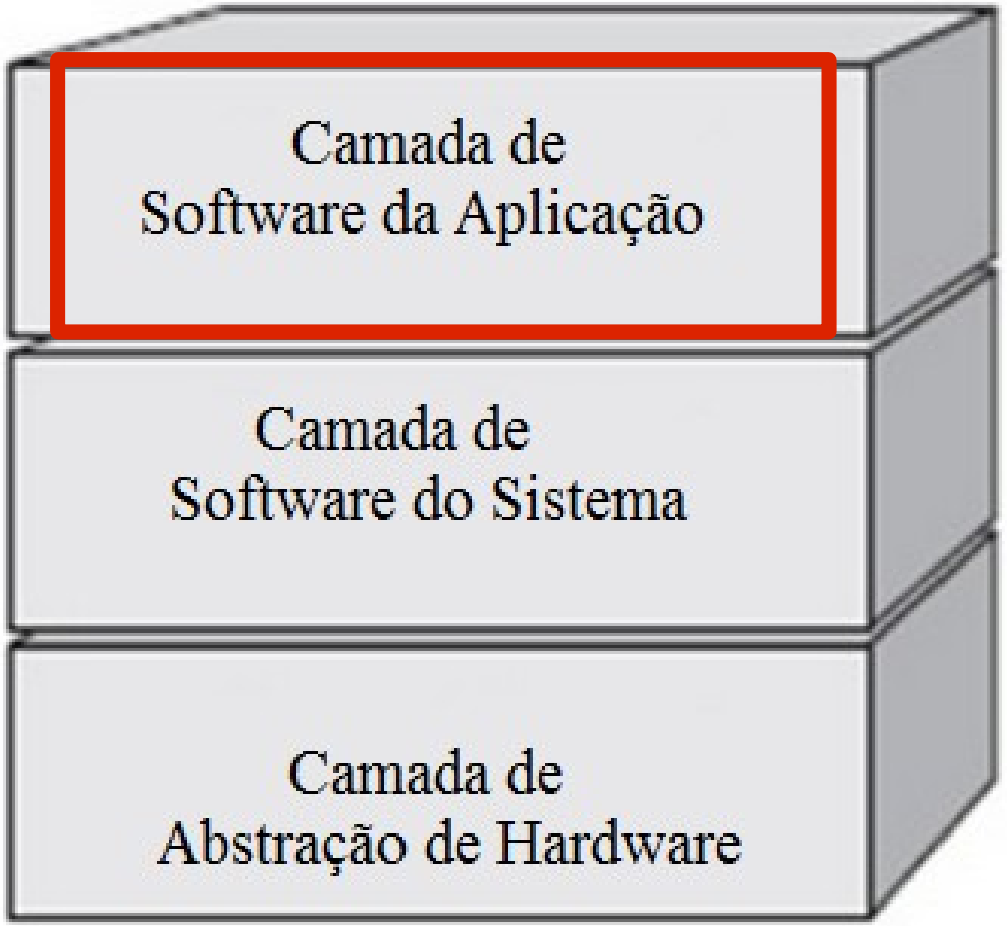
Em um **sistema guiado por eventos** as tarefas são compostas por **estados** do sistema, onde a **transição entre estados** é disparada por eventos.

Sistemas **guiados por eventos** são modelados como **máquinas de estados finitos** (FSM)

Threads x eventos



Projeto de software



A **camada de software do aplicação** é aquela que implementa os **aplicativos** e determina as **funcionalidades** do **sistema embarcado**.

Camada de software de aplicação

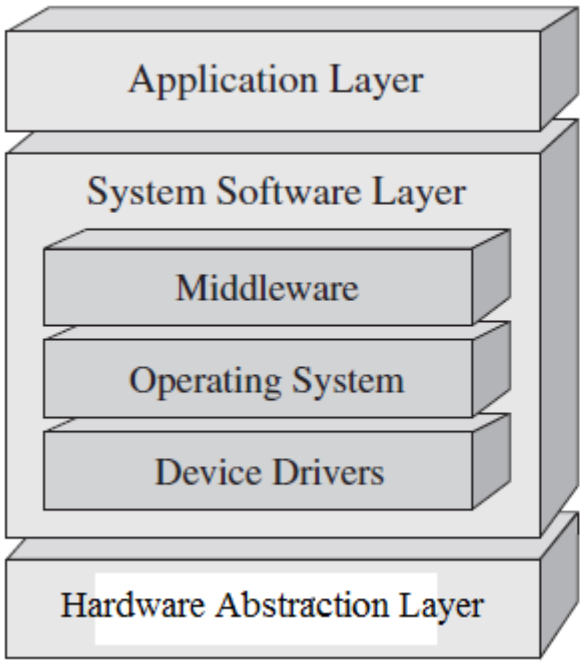
A **camada de software de aplicação** determina as funcionalides do sistema embarcado.

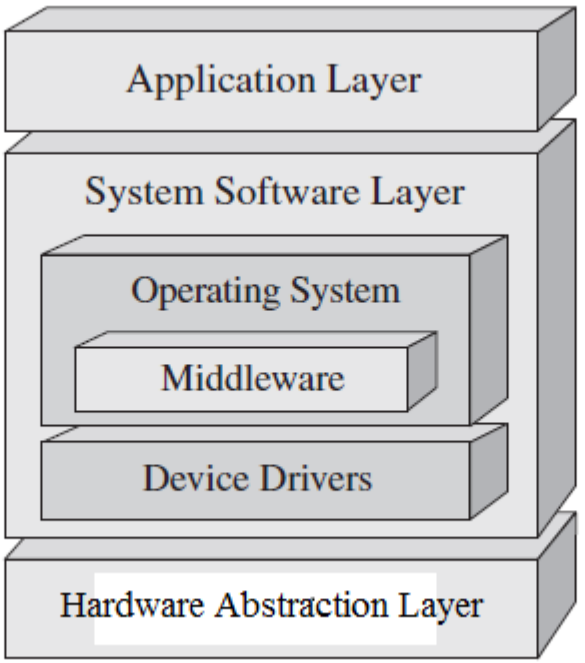
O projeto da camada de software de aplicação segue o modelo adotado pela camada de software do sistema (**guiado por eventos** ou **baseado em *threads).***

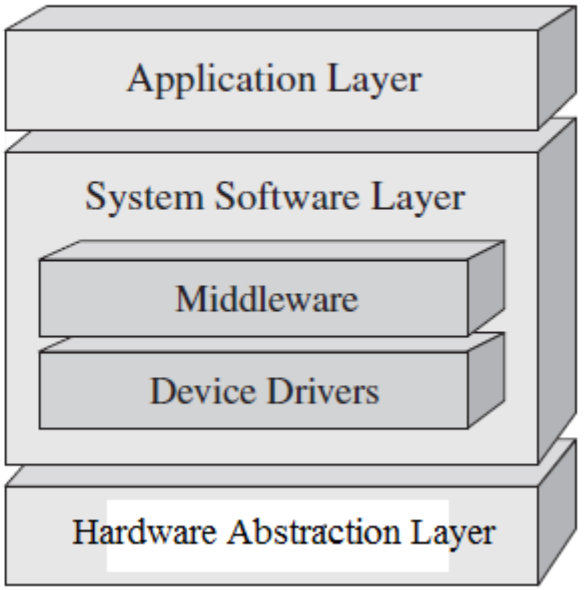
O software de aplicação é baseado na utilização da **API** fornecida pelo software do sistema e/ou de ***middleware.***

Middleware

Middleware é uma subcamada de software que fica na camada de software do sistema, sendo geralmente utilizada pelo software de aplicação como uma biblioteca de software separada ou integrada ao um OS.

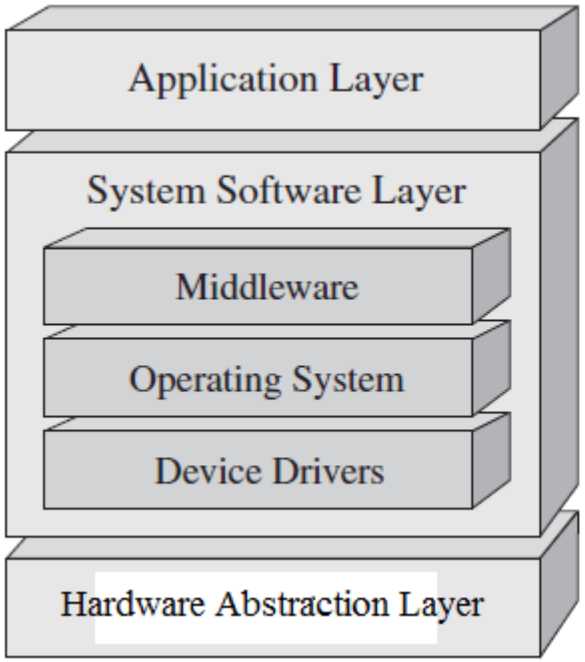


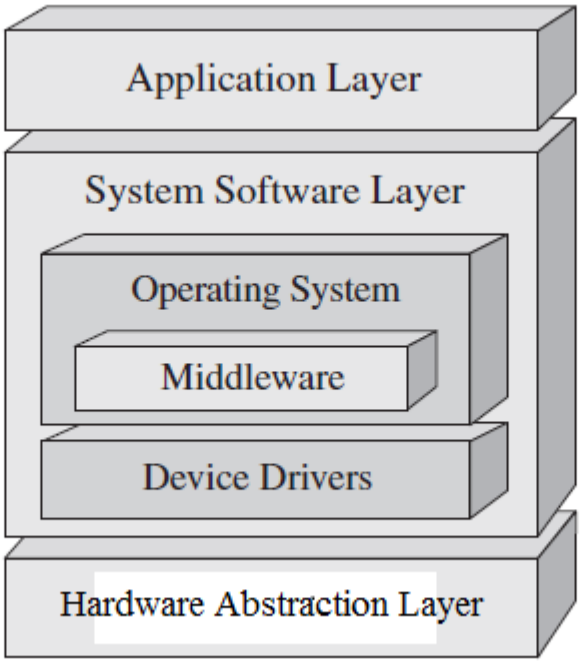


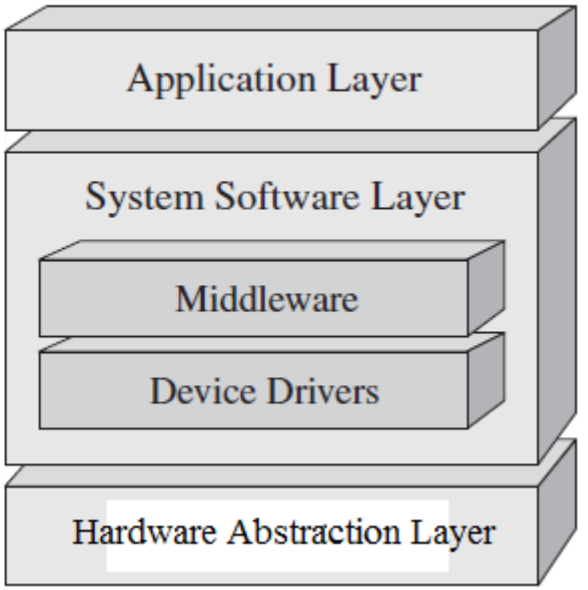


Middleware

Middleware permite **reusar software** entre várias aplicações, **evitando replicação** do código, aumentando a flexibilidade, a **portabilidade**, a segurança e a intercomunicação entre aplicações, e diminuindo a **complexidade** das aplicações.

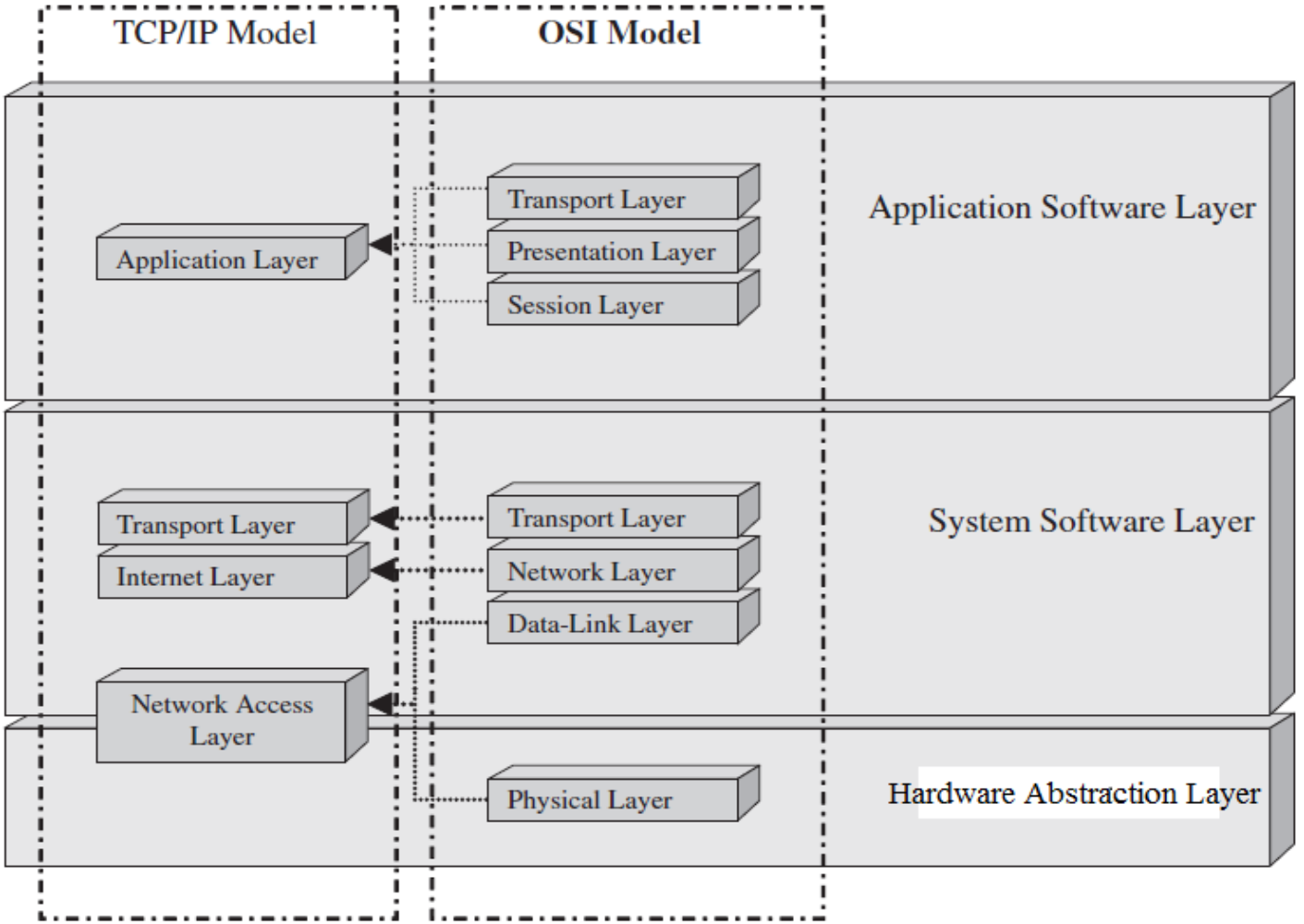






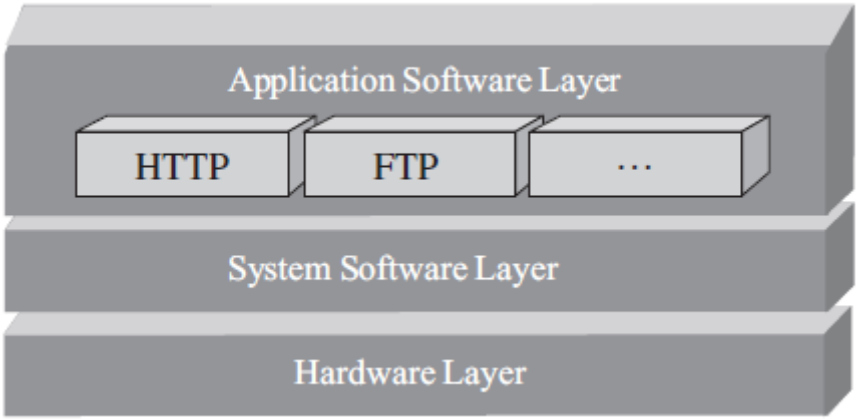
Middleware

Exemplo: ***Middleware*** para comunicação em redes de dados, como protocolos TCP/IP.

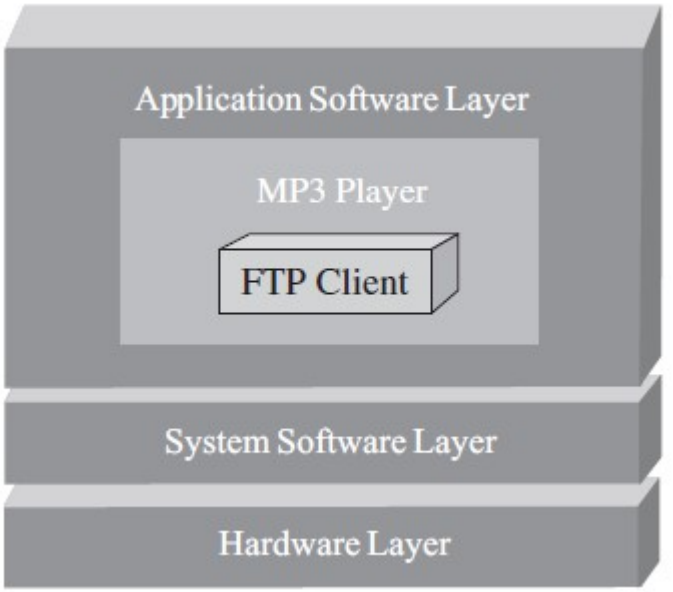


Camada de software de aplicação

A **camada de software de aplicação** determina as funcionalides do sistema embarcado.



Exemplo: servidor web HTTP e FTP



Exemplo: MP3 player com cliente FTP

Referências

Tammy Noergaard. 2005. **Embedded Systems Architecture:** ***A Comprehensive Guide for Engineers and Programmers***. Newnes.