Hardware Abstraction Layer (HAL): This is a layer of software that provides an interface between the application software and the underlying hardware. It allows the application software to be written independently of the specific hardware, making it more portable.

Firmware: sits next to the HAL, This is the low-level software that controls the hardware of the embedded system. It is responsible for tasks such as device initialization, peripheral control, and device driver management.

Operating System (OS): This is the software that manages the resources of the embedded system, such as memory and processing power. Examples of embedded OS include real-time operating systems (RTOS) and microcontroller-specific OS.

Middleware: This is software that sits between the OS and the application software, providing additional functionality and connectivity. Examples of middleware include communication protocols, libraries, and software frameworks.

Application software: This is the high-level software that provides the functionality of the embedded system. It can be written in a variety of programming languages, such as C or C++, and is typically tailored to the specific application of the embedded system.