

**CMSIS 到底是什么?**

**CMSIS 到底是什么？**

先来看看ARM公司对CMSIS的定义：

ARM® Cortex™ 微控制器软件接口标准 (CMSIS) 是 Cortex-M 处理器系列的与供应商无关的硬件抽象层。CMSIS 可实现与处理器和外设之间的一致且简单的软件接口，从而简化软件的重用，缩短微控制器开发人员新手的学习过程，并缩短新设备的上市时间。

软件的创建是嵌入式产品行业的一个主要成本因素。通过跨所有 Cortex-M 芯片供应商产品将软件接口标准化（尤其是在创建新项目或将现有软件迁移到新设备时），可以大大降低成本。

我们知道，不同厂家，比如FSL，ST，Energy Micro等不同厂家的内核都是使用Cortex M，但是这些MCU的外设却大相径庭，外设的设计、接口、寄存器等都不一样，因此，一个能够非常熟练使用STM32软件编程的工程师很难快速地上手开发一款他不熟悉的，尽管是Cortex M内核的芯片。而CMSIS的目的是让不同厂家的Cortex M的MCU至少在内核层次上能够做到一定的一致性，提高软件移植的效率。

**1. CMSIS的结构：**

CMSIS 包含以下组件：

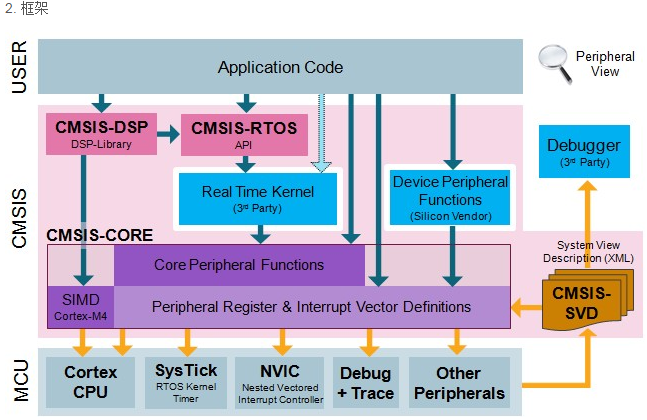
CMSIS-CORE：提供与 Cortex-M0、Cortex-M3、Cortex-M4、SC000 和 SC300 处理器与外围寄存器之间的接口

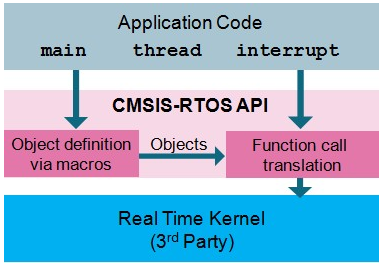
CMSIS-DSP：包含以定点（分数 q7、q15、q31）和单精度浮点（32 位）实现的 60 多种函数的 DSP 库

CMSIS-RTOS API：用于线程控制、资源和时间管理的实时操作系统的标准化编程接口

CMSIS-SVD：包含完整微控制器系统（包括外设）的程序员视图的系统视图描述 XML 文件

此标准可进行全面扩展，以确保适用于所有 Cortex-M 处理器系列微控制器。其中包括所有设备：从最小的 8 KB 设备，直至带有精密通信外设（例如以太网或 USB）的设备。（内核外设功能的内存要求小于 1 KB 代码，低于 10 字节 RAM）。





看了这张图的含义更清楚些，CMSIS-RTOS在用户的应用代码和第三方的RTOS Kernel直接架起一道桥梁，一个设计在不同的RTOS之间移植，或者在不同Cortex MCU直接移植的时候，如果两个RTOS都实现了CMSIS-RTOS，那么用户的应用程序代码完全可以不做修改。

完整的CMSIS文档可以从ARM公司网站下载，大小有100多M字节。

**4. 如何使用CMSIS，需要哪些文件，以Freescale Kinetis L系列举例。**

**独立于编译器的文件：**

● Cortex-M3内核及其设备文件(core\_cm0.h + core\_cm0.c)

─ 访问Cortex-M0内核及其设备：NVIC等

─ 访问Cortex-M0的CPU寄存器和内核外设的函数

● 微控制器专用头文件(device.h) - MKL25Z4.h

─ 指定中断号码(与启动文件一致)

─ 外设寄存器定义(寄存器的基地址和布局)

─ 控制微控制器其他特有的功能的函数(可选)

● 微控制器专用系统文件(system\_device.c) -- system\_MKL25Z4.h + system\_MKL25Z4 .c

─ 函数SystemInit，用来初始化微控制器

--函数 void SystemCoreClockUpdate (void); 用于获取内核时钟频率

─SystemCoreClock，该值代表系统时钟频率

─ 微控制器的其他功能(可选)

● 编译器启动代码(汇编或者C)(startup\_device.s) - startup\_MKL25Z4.s for Keil

─ 微控制器专用的中断处理程序列表(与头文件一致)

─ 弱定义(Weak)的中断处理程序默认函数(可以被用户代码覆盖)

