

# CMSIS 详解

## CMSIS 的简介

Cortex-M3 微控制器软件接口标准（Cortex Microcontroller Software Interface Standard）是 ARM 和一些编译器厂家以及半导体厂家共同遵循的一套标准，是由 ARM 提出，专门针对 CORTEX-M 系列的标准。在该标准的约定下，ARM 和芯片厂商会提供一些通用的 API 接口来访问 CORTEX 内核以及一些专用外设，以减少更换芯片以及开发工具等移植工作所带来的金钱以及时间上的消耗。只要都是基于 M3 的芯片，代码均是可以复用的。

根据近期的研究调查，发现在嵌入式开发领域，软件的花费在不断提高，相反硬件的花费却逐年降低，因此嵌入式领域的公司，越来越把精力放到了软件上，但软件在更换芯片或是开发工具的更新换代中，代码的重用性不高，随着 CORTEX-M3 处理器大量投放市场，ARM 意识到建立一套软件开发标准的重要性，因此 CMSIS 应运而生。

## CMSIS 的结构：

CMSIS 包含以下组件：

（1）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。

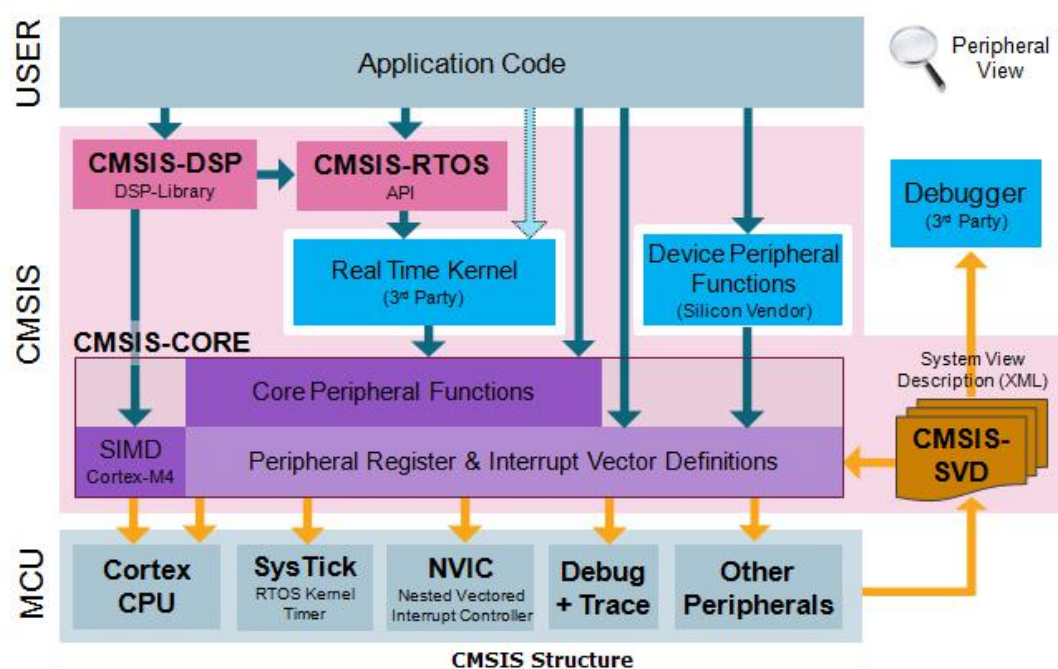
（2）CMSIS-DSP：包含以定点（分数 q7、q15、q31）和单精度浮点（32 位）实现的 60 多种函数的 DSP 库。

（3）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。

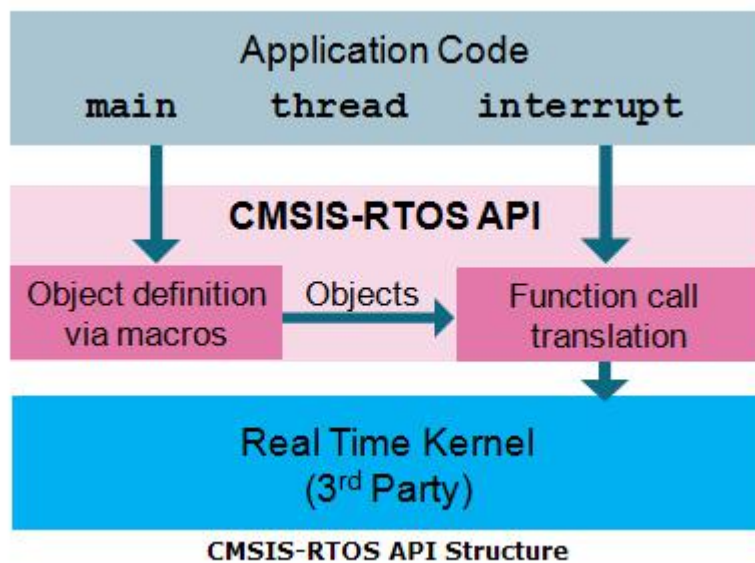
（4）CMSIS-SVD：包含完整微控制器系统（包括外设）的程序员视图的系统视图描述 XML 文件，在调试器可以用来创建外围意识或与外设寄存器和中断定义头文件。

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）。

## 框架图



由上图可知 CMSIS-CORE 和 CMSIS-DSP 比较好理解，CMSIS-DSP 是一个包含了一套常见的信号处理函数用于 Cortex-M 处理器的基础设备的库，而 CMSIS-CORE 主要用来定义一些处理器内部的一些寄存器地址以及功能函数，如对内核寄存器，NVIC，调试子系统的访问。但是 CMSIS-RTOS 好像不好理解，再来看下面一张图吧：



看了这张图更能清楚一些，CMSIS-CORE 在用户的应用代码和第三方的 RTOS Kerner 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。

如何使用 CMSIS，需要那些文件

独立于编译器的文件：

- (1) Cortex-M3 内核及其设备文件 (core\_cm3.h+core\_cm3.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
- (2) 微控制器专用头文件(device.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)
- (3) 微控制器专用系统文件(system\_device.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)

(4) 编译器启动代码(汇编或者 C)(startup\_device.s)

— 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)