

# ARM Cortex-M シリーズの ソフトウェア・ インターフェース規格 CMSIS

最近国内でも ARM マイコンを利用する機会が増えてきている。ARM Cortex-M シリーズにはプログラムの移植性を高めるソフトウェア・インターフェースの規格 CMSIS が用意されており、今までにないマイコン・ソフトウェア開発の標準スタイルとしての期待が高い。本稿ではこの CMSIS を使った簡単な実装を通してその具体像に迫る。

(筆者)



平井 幸広

## 1. CMSIS とは

CMSIS (シーエムシス) は Cortex マイクロコントローラ・ソフトウェア・インターフェース・スタンダード (Cortex Microcontroller Software Interface Standard) の略で、その名の通り、ARM Cortex-M シリーズ用のソフトウェア・インターフェースの規格です(図1)。これはマイコン・ソフトウェア用の HAL (Hardware Abstraction Layer ; ハードウェア抽象化レイヤ) です。

CMSIS は、アプリケーション・ソフトウェアから見た個々のデバイスの違いを吸収することにより、Cortex-M シリーズを使うソフトウェア開発者のソフトウェアの再利用を促進し、プログラム開発を容易にすることが目的です。よりわかりやすくするための CMSIS のコーディング規則や規約に基づいた Cortex-M コア共通のヘッダ・ファイルと、ペリフェラルを定義するデバイス固有のヘッダ・ファイル、また、これらのアクセス用関数やスタートアップ・ルーチンなどのコンポーネント一式が誰でも無償で利用できます。

従来は、これらを各開発担当者、または開発部門などの利用者側が個別に作成していました。しかし、近年のマイコンのペリフェラルはマニュアルだけでも 1,000 ページを超えることが珍しくなく、定義用のヘッダ・ファイルを作成するだけでも容易ではありません。CMSIS を英国 ARM

社や半導体各社が提供することにより、標準化された方法として簡単に利用できます。また、ソフトウェア開発ツールからは独立しており、CMSIS はどの開発ツールでも利用できるようになっています。

ARM Cortex-M コア搭載のマイコンはコアが共通という点でもソフトウェアの再利用に貢献します。しかしさらに CMSIS が必要となったのは、マイコンの性能や機能の進化により、ソフトウェアも増大し、開発/保守のコストが重要になってきているという背景があるからです。

従来のシリアル通信や小容量の不揮発性 ROM に代わって、マイコンでも USB や TCP/IP、大容量フラッシュ ROM が使えるようになりました。それに伴って通信用のソフトウェア・スタックやファイル・システムが必要になり、このようなソフトウェアを自分ですべてゼロから作るのは現実的でなくなってきました。マイコンでもオンチップで 500K バイトを超えるプログラム容量が利用できる今日、ソフトウェア開発の手間を減らすための開発手法の革新が必要になっています。CMSIS はこうした課題をクリアするための方法です。

最新の CMSIS は <http://www.onARM.com/> から無償でダウンロードできます。現在のものは Cortex-M0 と Cortex-M3 の両方に対応しています。

## 2. CMSIS の構成

システムを利用する上での構成を図2に示します。

CMSIS はユーザのアプリケーション・プログラムと MCU (マイクロコントローラ) の間に位置しています。MCU が替わってもアプリケーション・プログラムは極力影響を受けないようにインターフェースします。CMSIS

図1  
CMSIS のロゴ・  
マーク

