



Cortex Microcontroller Software Interface Standard (CMSIS)

CMSIS : Pronunciation

See-M-Sys

CMSIS : Logo

CMSIS compliant software development tools and middleware stacks are allowed to carry the CMSIS logo.



CMSIS : Introduction

- **CMSIS is a vendor-independent hardware abstraction layer for microcontrollers that are based on Arm Cortex processors.**
- **CMSIS defines generic tool interfaces and enables consistent device support**
- **CMSIS provides simple software interfaces to the processor and the peripherals.**

CMSIS : Aim

- **The aim of CMSIS is to improve software portability and reusability across different microcontrollers and toolchains.**
- **It allows software from different sources to integrate seamlessly together.**
- **CMSIS simplifies software reuse, reduces the learning curve for microcontroller developers.**

CMSIS : Components

- **CMSIS(M)**
- **CMSIS(A)**
- **Driver**
- **DSP**
- **Neural Network (NN)**
- **RTOS v1**
- **RTOS v2**
- **Build**
- **SVD**
- **DAP**
- **Zone**

CMSIS : CMSIS (M)

- **For all Cortex-M and SecurCore processors**
- Provides standardized API for configuring the Cortex-M processor core and peripherals.
- **Also standardizes the naming of the device peripheral registers.**
- Includes intrinsic functions for Cortex-M4/M7/M33/M35P SIMD instructions.

SIMD: Single Instruction Multiple Data

SecurCore: ARM Securcore Processor Family provides 32-bit solutions for smart card and secure IC development

CMSIS : Driver

- **Provides generic peripheral driver interfaces for middleware.**
- **Connects microcontroller peripherals with middleware that implements for example communication stacks, file systems, or graphic user interfaces.**

CMSIS : DSP

- **For all Cortex-M processors**
- **DSP library collection with over 60 functions for various data types.**
- **Implementations optimized for the SIMD instruction set are available for Cortex-M4/M7/M33/M35P.**

- **For all Cortex-M processors**
- **Collection of efficient neural network kernels developed to maximize the performance and minimize the memory footprint on Cortex-M processor cores.**

CMSIS : RTOS v1

- **For Cortex-M0/M0+/M3/M4/M7**
- **Common API for real-time operating systems along with a reference implementation based on RTX.**

CMSIS : RTOS v2

- **For all Cortex-M and Cortex-A5/A7/A9 processors.**
- **Extends CMSIS-RTOS v1 with Armv8-M support, dynamic object creation, provisions for multi-core systems, binary compatible interface**

CMSIS : Pack

- **For all Cortex-M, Cortex-A5/A7/A9 processors and SecurCore processors.**
- **Describes a delivery mechanism for software components, device parameters, and evaluation board support**
- **It simplifies software re-use and product life-cycle management (PLM).**

CMSIS : Build

- **For all Cortex-M, Cortex-A5/A7/A9 processors and SecurCore processors.**
- **A set of tools, software frameworks, and work flows that improve productivity.**

CMSIS : SVD

- **For all Cortex-M and SecurCore processors.**
- Defines device description file .
- **This file is provided and maintained by the silicon vendor and contains a complete description of the microcontroller peripheral registers in an XML format.**
- This file is then imported by the development tool, which uses it to automatically construct the peripheral debug windows for the microcontroller.
- **This approach allows full debugger support to be available as new microcontrollers are released.**

CMSIS : DAP

- **For all Cortex processors.**
- Defines the interface protocol for a hardware debug unit that sits between the host PC and the debug access port (DAP) of the microcontroller.
- **This allows any software toolchain that supports CMSIS DAP to connect to any hardware debug unit that also supports CMSIS DAP.**

- **For all Cortex-M processors.**
- Defines methods to describe system resources and to partition these resources into multiple projects and execution areas.

CMSIS : Basics

- The CMSIS core specification provides a standard set of low-level functions, macros, and peripheral register definitions that allow your application code to easily access the Cortex-M processor and microcontroller peripheral registers.

CMSIS : Coding Rules

- Coding standard is based on MISRA C

CMSIS : Variable Types

Standard ANSI C Type

MISRA C Type

Signed char

int8_t

Signed short

int16_t

Signed int

int32_t

Signed __int64

int64_t

Unsigned char

uint8_t

Unsigned short

uint16_t

Unsigned int

uint32_t

Unsigned __int16

uint64_t

CMSIS : Input/Output Qualifiers

<u>MISRA C Qualifier</u>	<u>ANSI C Type</u>	<u>Meaning</u>
#define __I	volatile const	read only
#define __O	volatile	write only
#define __IO	volatile	read and write

CMSIS : Adding CMSIS to a project

Three files need to be added:

- **The Startup file**

This file contains the default CMSIS vector table.

- **The system_<device> file**

This file contains the necessary code to initialize the microcontroller system peripherals.

- **The device include file**

This file imports the CMSIS header file that contains the CMSIS core functions and macros.



The end.