Common Microcontroller Software Interface Standard

Christopher Seidl, Matthias Hertel – Arm

16 January 2024

CMSIS - Common Microcontroller Software Interface Standard

Consistent software foundation for the embedded industry

What?

- For a wide range of devices CMSIS provides consistent and efficient interfaces to processor, peripherals, and real-time operating systems.
- Simplifies delivery of device support and software components with a packing system that facilities project build.
- Enables debug access with event trace, fault analysis, and system views to peripherals.

Why?

- Standardization enables software reuse and reduces the learning curve for developers. ML and DSP applications can utilize the different compute performance of Cortex-M and Ethos-U.
- Easy access to new devices, boards, and related software components. Simplifies reproducible project builds with CI and MLOps integration.
- Allows developers to correct issues, optimized performance, and enhance the overall stability of embedded systems.

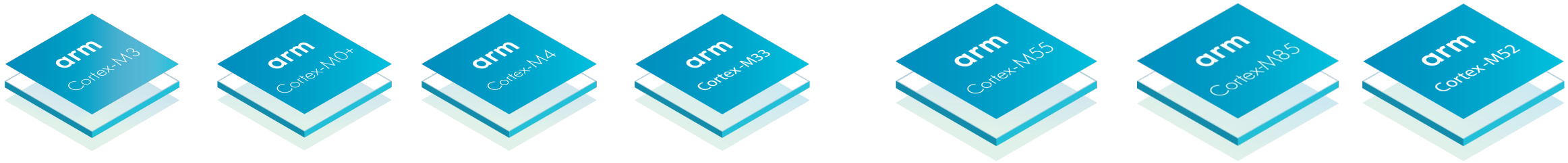
Two Decades of Microcontroller Innovation

32-bit Compute

TrustZone Security

Helium ML/DSP

2004	2009	2010	2016	2020	2022	2023
Birth of the Low Cost, Low Power 32-bit Microcontroller	Focus on ultra-low cost / power / area	Delivering DSP Extensions	Introducing TrustZone advanced security for Cortex-M	Delivering Helium architecture technology for ML & DSP	Addressing higher performance ML & DSP applications	Today: Delivering high-efficiency ML & DSP



arm

CMSIS - Fifteen Years of Software Evolution

10,000 supported devices

900+ packs from 60 vendors

6 million GitHub projects

Multiple toolchains

CMSIS v1

Access to processor core, interrupts, and peripherals

CMSIS v2

Optimized DSP compute functions for all Cortex-M

Peripheral debug description (SVD)

CMSIS v3

Real-time operating systems API

CMSIS v4

Enable middleware with Driver and RTOS interfaces

Device support and software delivered with CMSIS packs

CMSIS v5

Development on GitHub

CMSIS-NN for Machine Learning

Optimized compute libraries for Helium

CMSIS v6

Optimized structure and enhancements for software development

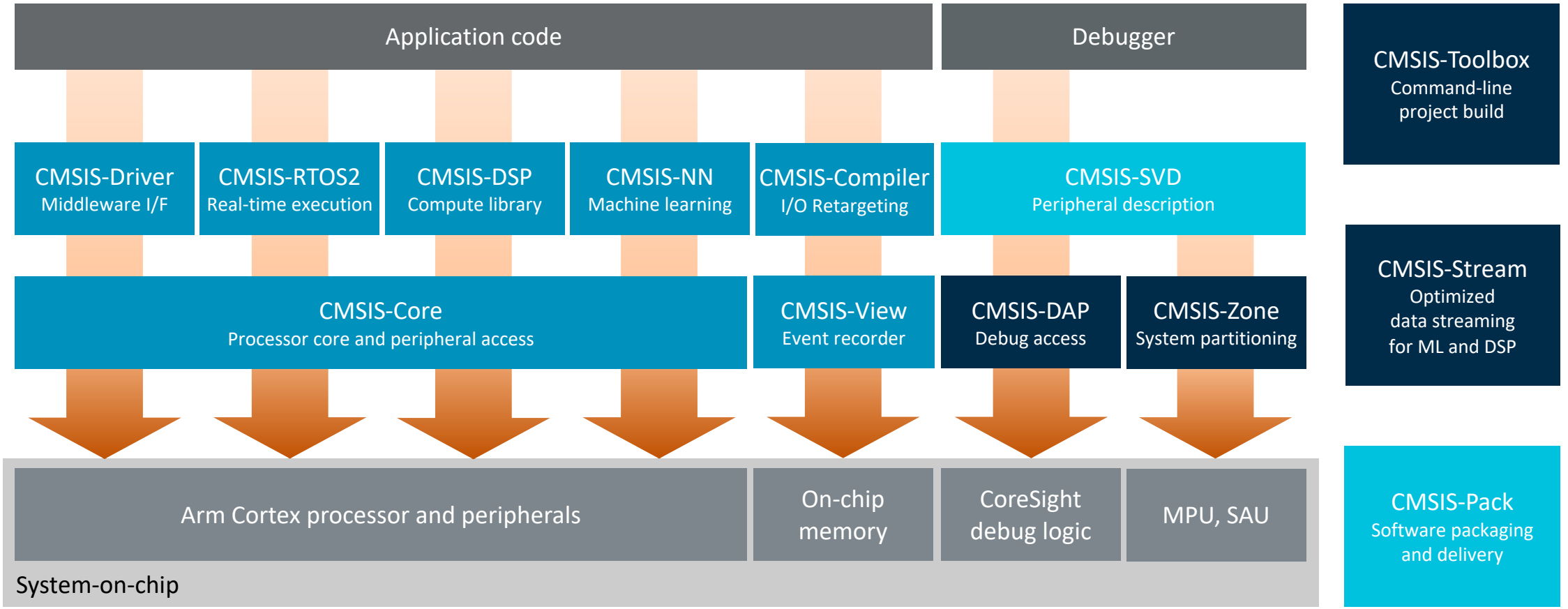
CLI build system for CI, MLOps, and IDE integration

arm

CMSIS Version 6

Consistent software framework for billions of devices

github.com/ARM-software/CMSIS_6



Software components for the Arm Cortex processor target

Tools for optimizing software development flows

Specifications

Demo

Navigating CMSIS v6

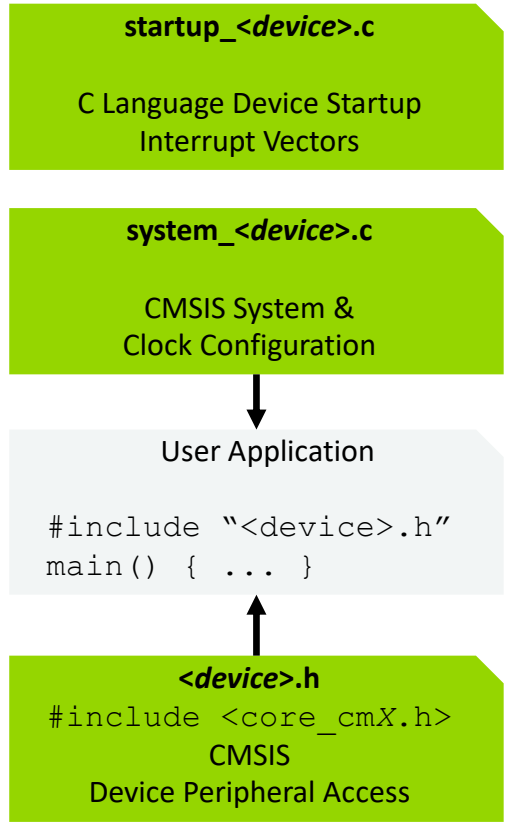
Details

Changes in CMSIS v6

CMSIS-CORE

Processor core and peripheral access

- CMSIS-CORE provides a lightweight peripheral access layer with
  - Consistent layout of all peripheral registers
  - Vector definitions for all exceptions and interrupts
  - Functions to access core registers and core peripherals
  - Debug channel (for printf-style + RTOS Kernel)

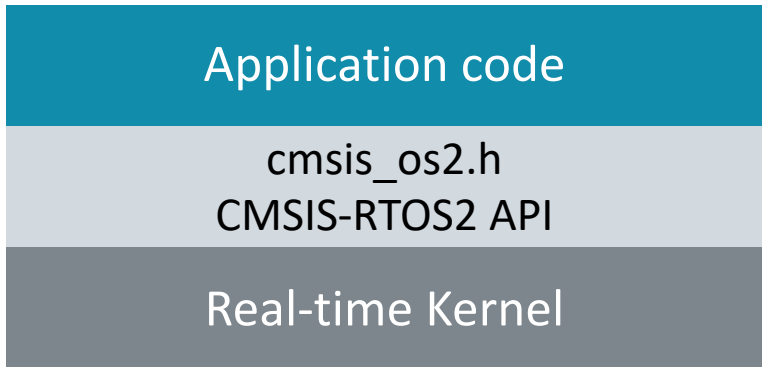


Changes

Use of Arm C Language Extensions (ACLE)
Drop support for Arm Compiler 5
Mandates for C-based startup code
Header files reworked
Deprecated features removed

CMSIS-RTOS2 API

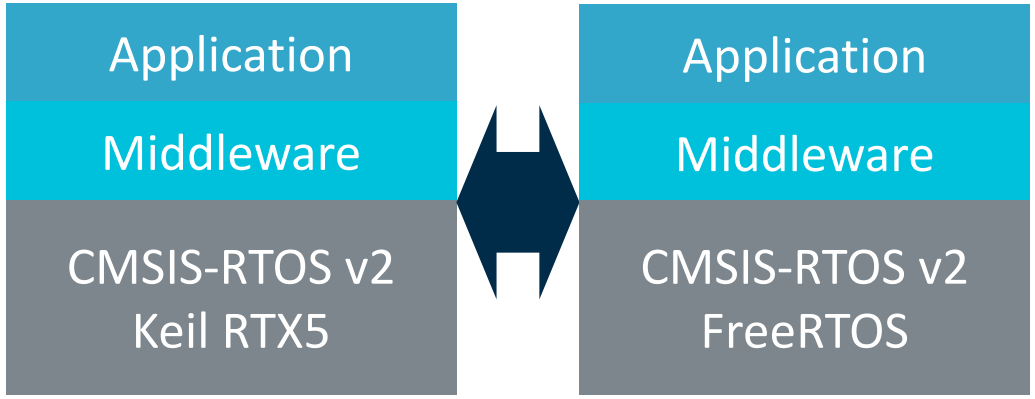
Deterministic real-time software execution using a standardized API



- Provides basic features required in many applications.
- Provides flexibility in choosing the right RTOS:
  - Azure ThreadX, embOS, FreeRTOS, Keil RTX5, Micrium OS, Zephyr RTOS

Changes

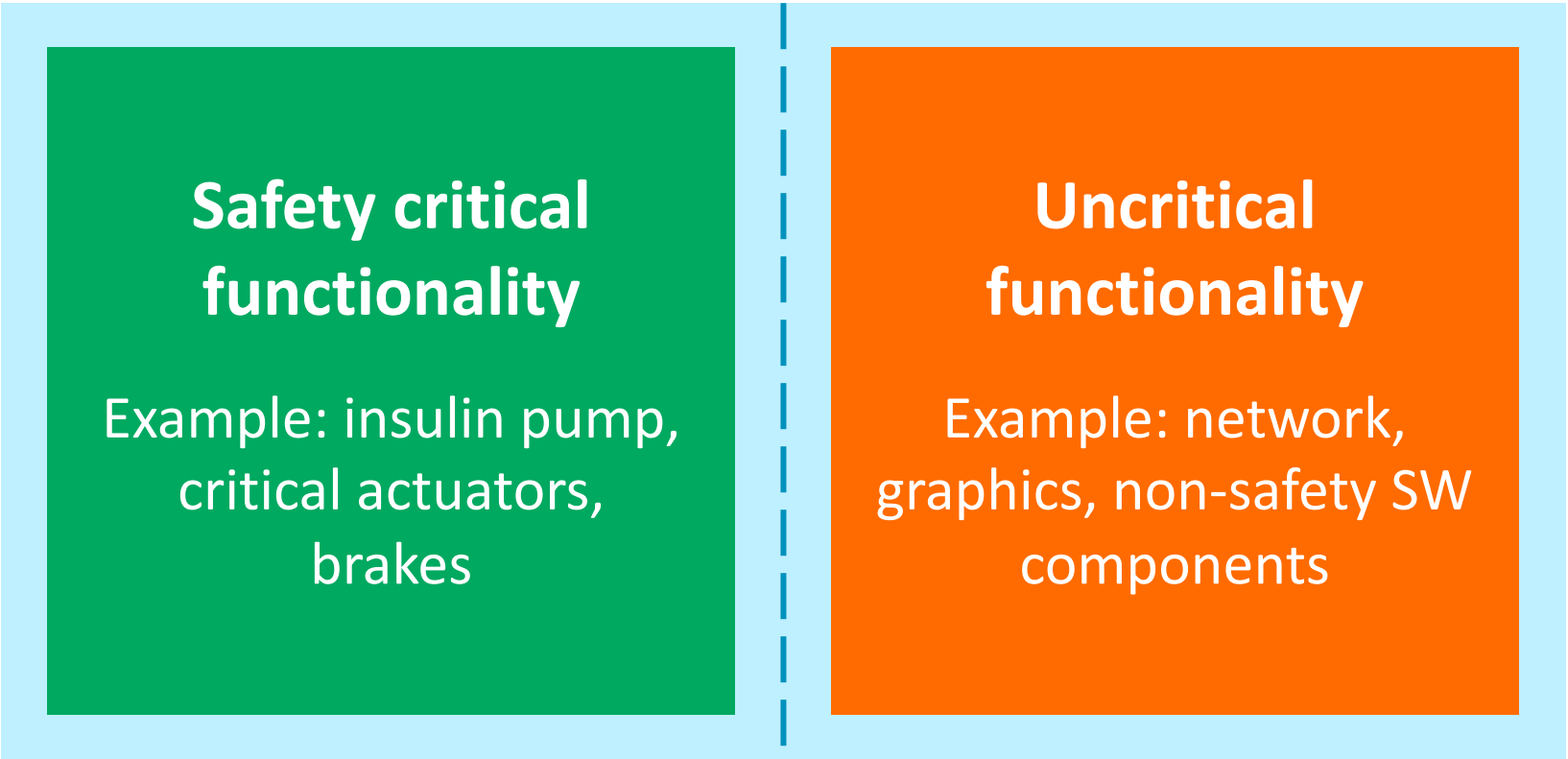
RTOS v1 API removed
OS Tick API moved to CMSIS Cclass
Added support for process isolation
Provisions for SMP (Symmetric Multi-Processing)



CMSIS-RTX CMSIS-FreeRTOS

Process Isolation

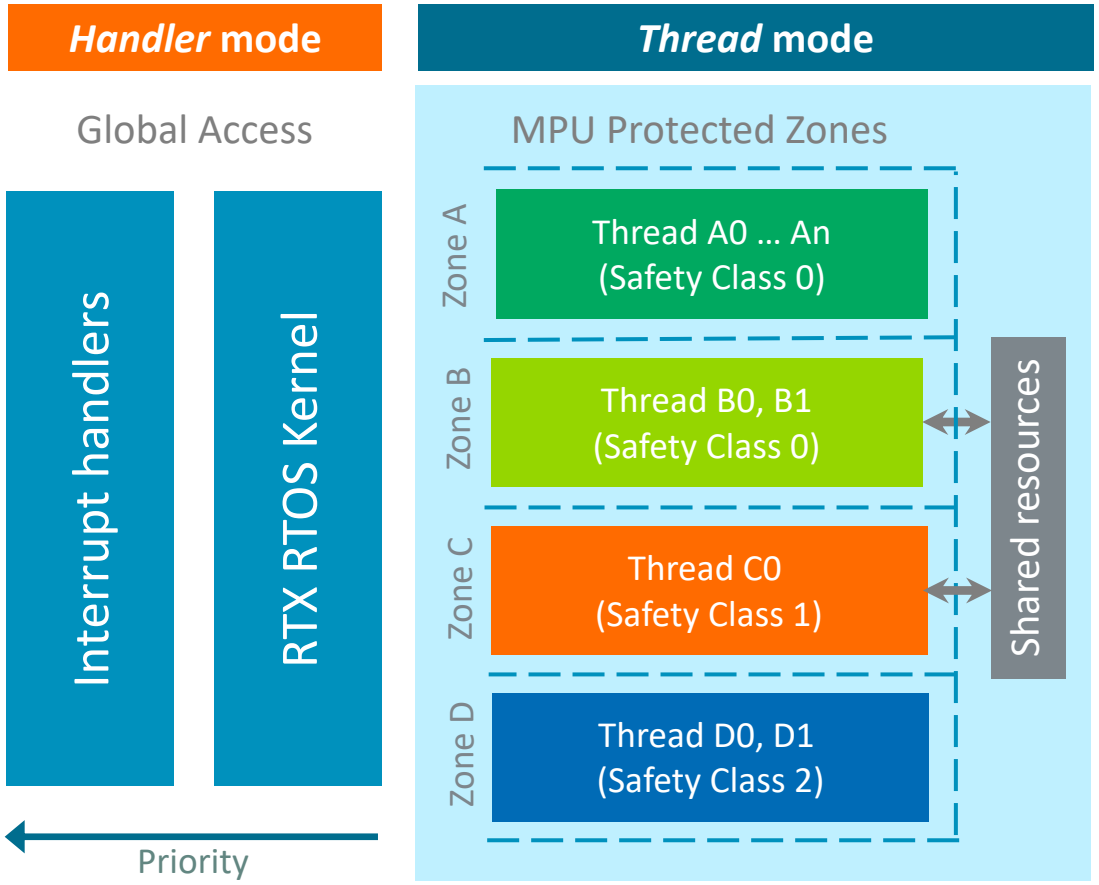
Enables use of software with different safety integrity levels



- Already implemented in CMSIS-RTX (Keil RTX5)

Spatial Isolation

Each thread can access only memory and peripherals assigned for its operation



Zones are configured using CMSIS-Zone

MPU Protected Zones

- One or more threads are assigned to an MPU Protected Zone with specified access rights to R/W memory and peripherals.
- Shared resources can be accessed from multiple MPU Protected Zones.
- Code and R/O memory has global access to allow common library code and Link-Time Optimizations (LTO).
- IRQs have global access with no extra overhead to retain full time-deterministic operation.

Safety Classes

- RTOS objects and MPU Protected Zones can be assigned to a safety class.
- Thread code execution in a lower safety class cannot create or modify RTOS objects with a higher safety class.

arm

CMSIS-Driver

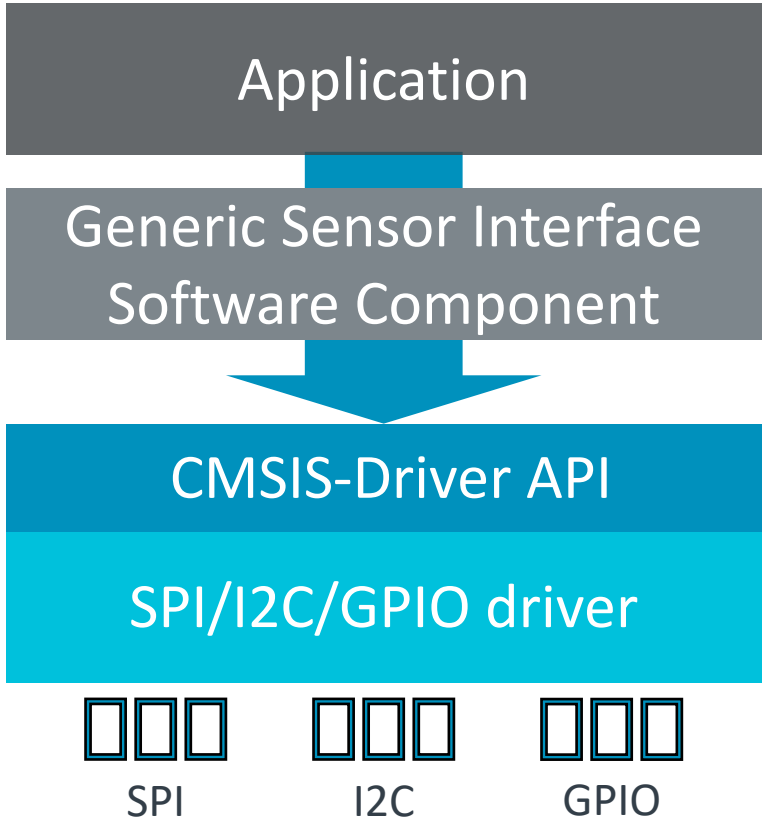
Middleware interface

- CMSIS-Driver describes peripheral driver interfaces for middleware components and user applications
- It offers an effective interface to middleware and central configuration

New Driver APIs

GPIO support of control pins

VIO test interface for physical or virtual access to LEDs and buttons



Computing components

CMSIS-DSP

- CMSIS-DSP provides optimized compute kernels for all Arm Cortex-M and for Cortex-A.

Changes

New repository
Stand-alone CMSIS-Pack
Integration with CMSIS-Stream
Optimized for Cortex-M52/M55/M85

CMSIS-NN

- The CMSIS-NN software library is a collection of efficient neural network kernels developed for all Arm Cortex-M processors.

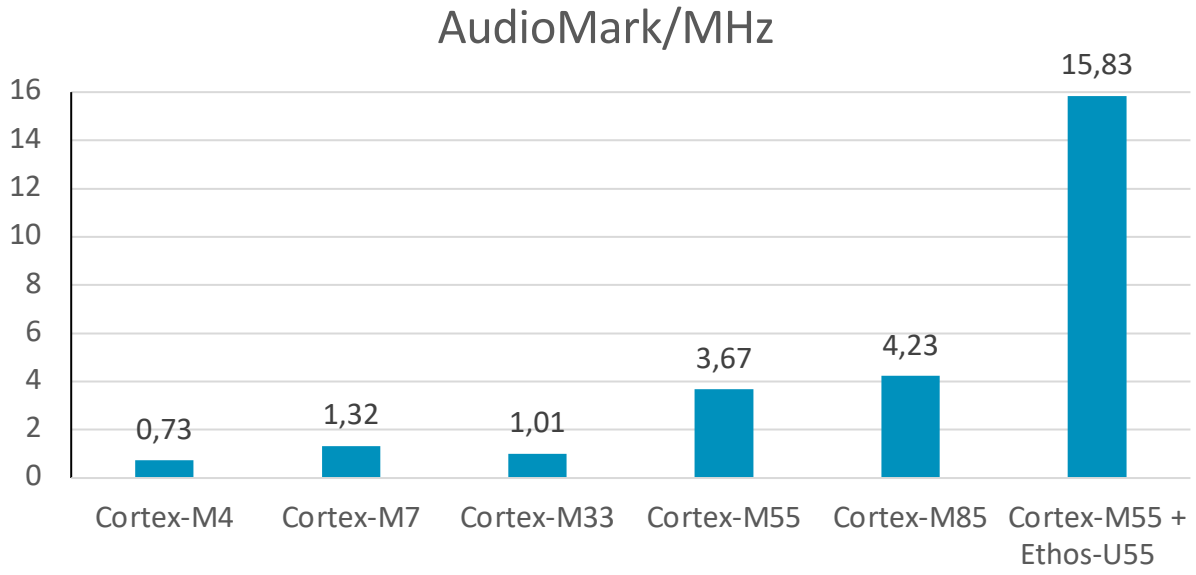
Changes

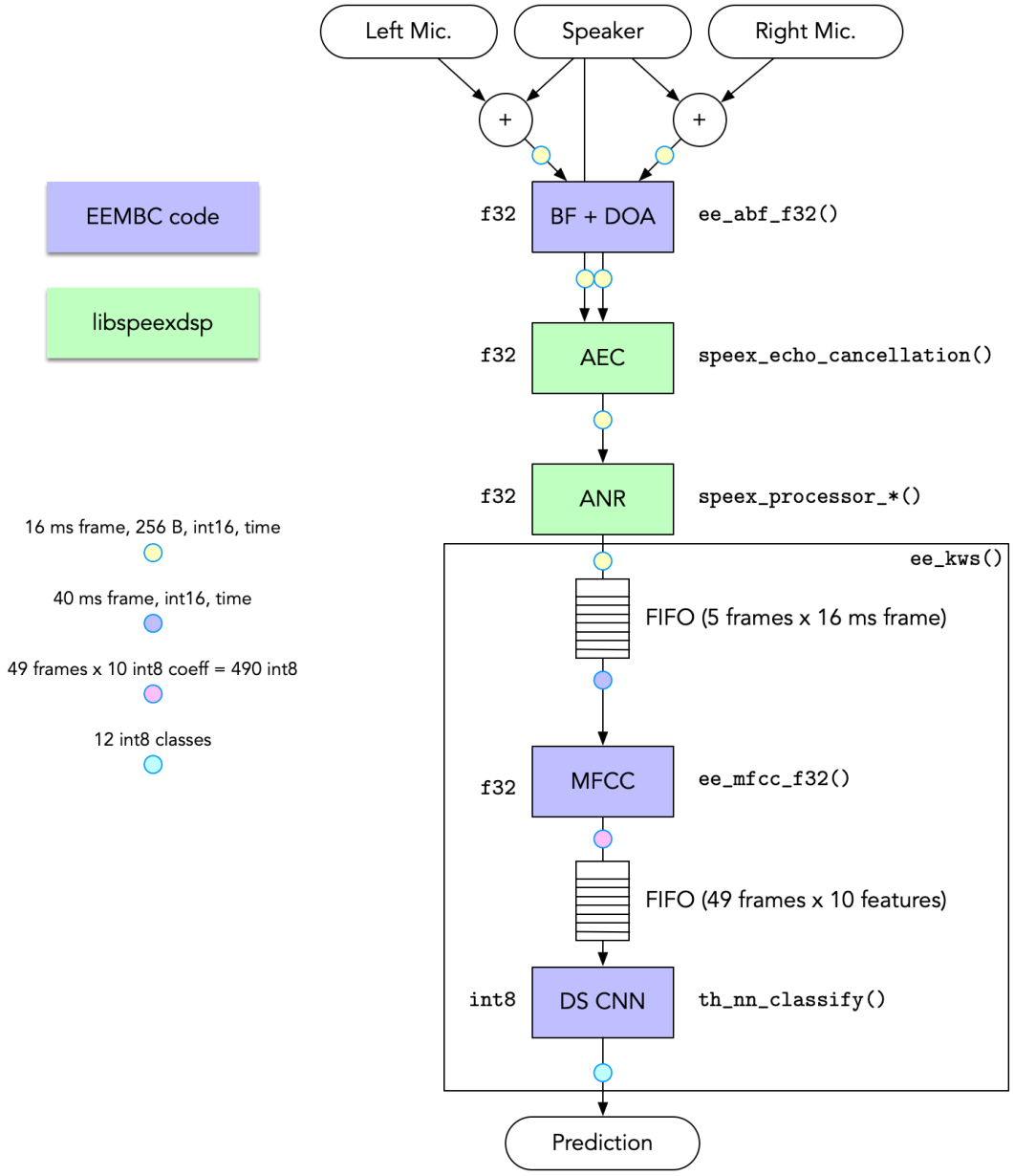
New repository
Stand-alone CMSIS-Pack
Optimized for Cortex-M52/M55/M85

AudioMark

https://github.com/eembc/audiomark

- Emulates the workload in a smart speaker
  - Audio frontend
    - Beamforming, direction of arrival
    - Acoustic echo cancellation
    - Active noise reduction
  - Machine learning
    - Feature extraction (MFCC)
    - Neural network





Demo

Upgrading projects to CMSIS v6

Resources

Blogs

- CMSIS v6 is here: https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/cmsis-v6-is-here
- Which CMSIS components should I care about? https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/which-cmsis-components-should-i-care-about
- What are CMSIS software components? https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/what-are-cmsis-software-components

Learning paths

- Migrating projects to CMSIS v6: https://learn.arm.com/learning-paths/microcontrollers/project-migration-cmsis-v6/
- Migrating CMSIS-Packs to CMSIS v6: https://learn.arm.com/learning-paths/microcontrollers/pack-migration-cmsis-v6/

Upcoming webinars

CMSIS-View/Compiler

- CMSIS-View provides an API for event annotations used to record event timing and data information at run-time.
- CMSIS-Compiler allows individual retargeting configuration for common I/O interfaces: file, stderr, stdin, stdout.

January 30th

CMSIS-Toolbox

- A set of tools, software frameworks, and workflows that improve productivity
- Provides a generic CMSIS-aware project file format that allows IDEs and command-line build tools to share the same projects.

February 13th

CMSIS-Stream and SDS

- CMSIS-Stream provides methods, interfaces, and tools for data block streaming between processing steps of a DSP/ML application.
- SDS implements a data stream management, provides methods and helper tools for DSP and ML projects.

February 27th

arm

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה