

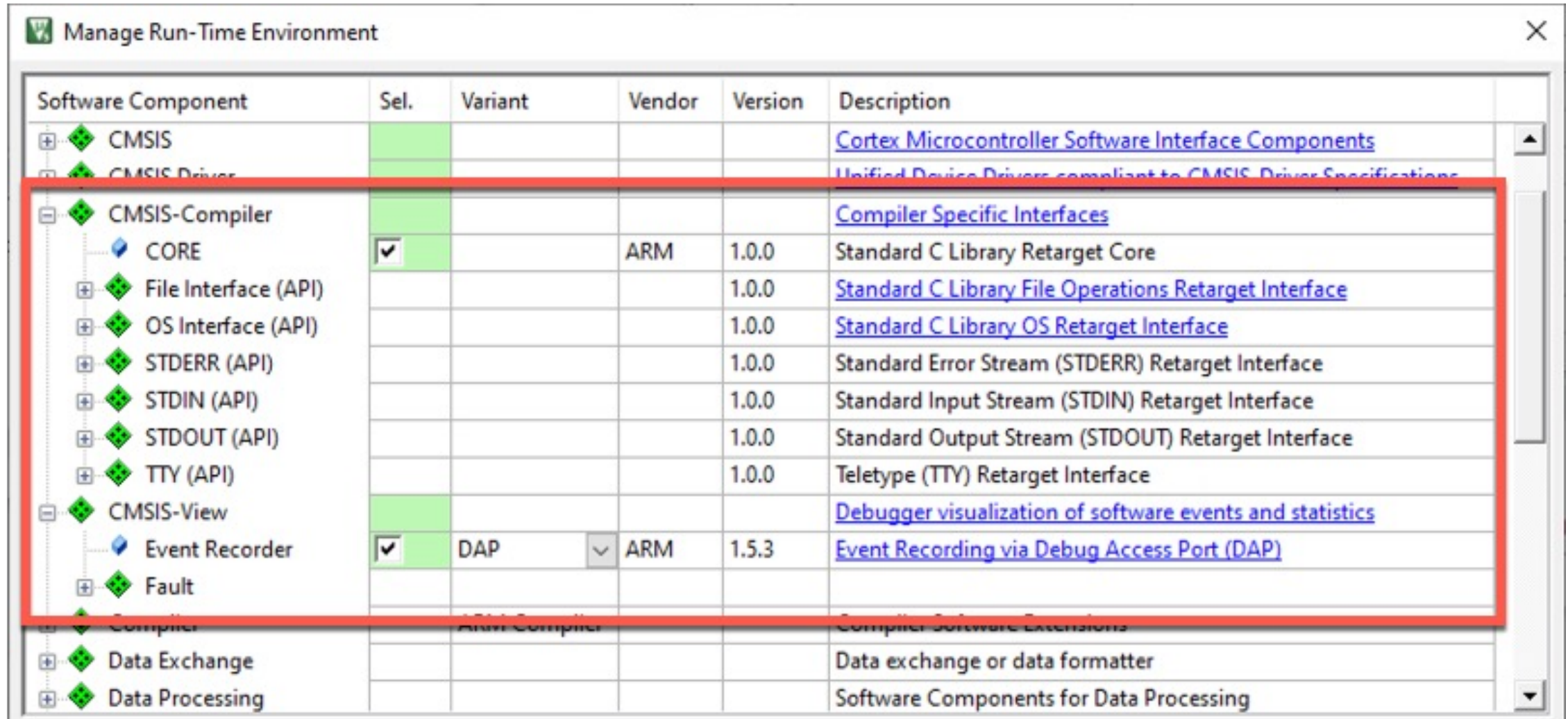
arm



CMSIS-Compiler and CMSIS-View

Christopher Seidl, Matthias Hertel – Arm
30 January 2024

New in Version 6: CMSIS-Compiler and CMSIS-View

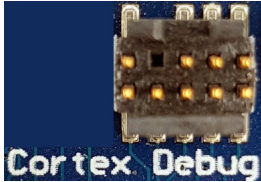
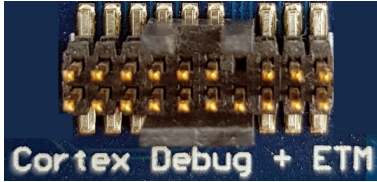


The screenshot shows the 'Manage Run-Time Environment' window with a table of software components. A red rectangle highlights the 'CMSIS-Compiler' and 'CMSIS-View' sections. In the 'CMSIS-Compiler' section, the 'CORE' component is selected with a checkmark. In the 'CMSIS-View' section, the 'Event Recorder' component is selected with a checkmark, and its variant is set to 'DAP' in a dropdown menu.

Software Component	Sel.	Variant	Vendor	Version	Description
CMSIS	☐				Cortex Microcontroller Software Interface Components
CMSIS Driver	☐				Unified Device Drivers compliant to CMSIS Driver Specifications
CMSIS-Compiler	☐				Compiler Specific Interfaces
CORE	☒		ARM	1.0.0	Standard C Library Retarget Core
File Interface (API)	☐			1.0.0	Standard C Library File Operations Retarget Interface
OS Interface (API)	☐			1.0.0	Standard C Library OS Retarget Interface
STDERR (API)	☐			1.0.0	Standard Error Stream (STDERR) Retarget Interface
STDIN (API)	☐			1.0.0	Standard Input Stream (STDIN) Retarget Interface
STDOUT (API)	☐			1.0.0	Standard Output Stream (STDOUT) Retarget Interface
TTY (API)	☐			1.0.0	Teletype (TTY) Retarget Interface
CMSIS-View	☐				Debugger visualization of software events and statistics
Event Recorder	☒	DAP	ARM	1.5.3	Event Recording via Debug Access Port (DAP)
Fault	☐				
Compiler	☐		ARM Compiler		Compiler Software Extensions
Data Exchange	☐				Data exchange or data formatter
Data Processing	☐				Software Components for Data Processing

Debug and Trace interfaces on Cortex-M microcontrollers

Arm CoreSight™ for debug/trace – UART for printf-style debugging

JTAG / SWD	SWO	ETM	UART
4-pin/2-pin	1-pin (extends SWD)	4-pin data / 1-pin clock	1-pin TxD / 1-pin RxD
			☐ TxD ☐ RxD ☐ GND
Synchronous ~50..100Mbit/sec	Manchester or USART encoding ~100Mbit/sec or 25Mbit/sec	Synchronous up to 800Mbit/sec	115200 baud
• Memory read/write access – even while executing • Run control: stop / go • Program breakpoints • Memory access breakpoints • ETB or MTB (trace buffer)	Events (optional with timestamp): • Access watch to 4 memory locations • Interrupt analysis • Event monitor (i.e. waitstate) • ITM ports for data output	• All SWO event features • Instruction trace • On some devices data trace	• Typical usage for printf-style debug output
<i>Standard debug interface</i>	<i>Not available on some devices</i>	<i>Only on few devices</i>	<i>Requires peripheral + extra pins</i>

Software analysis that works across all devices

Application developers need high-level views to operation of software components.

Event Recorder

Timing and program execution analysis

- View the dynamic program execution flow
- Adds RTOS awareness
- Store, record, and analyze fault information

Component Viewer

Static view to the program operation

- View the static user relevant information
- Obtained via debug symbols in memory
- Hot-pluggable

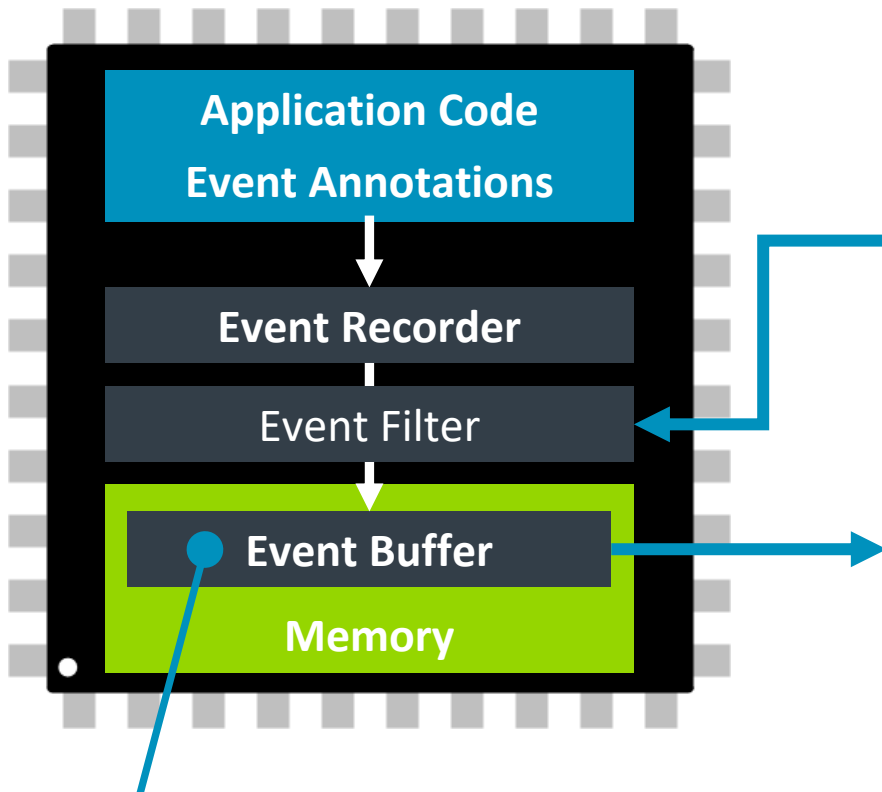
Both work via standard Cortex-M JTAG/SWD debug interface

Utilize memory read to capture data

[Whitepaper: Software Analysis with Event Annotations](#)

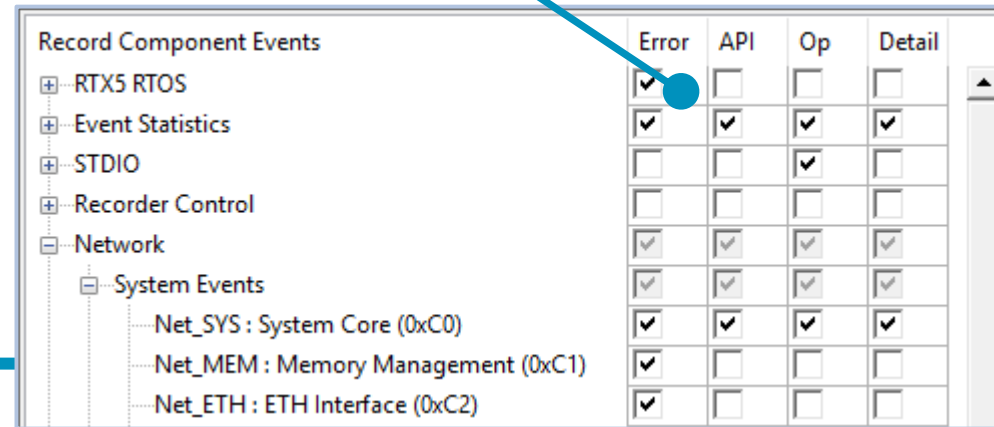
Event Recorder

Dynamic view to operation and timing of software components



Event buffer is configurable in size to cope with event bursts

Filters can be set at debug or system level; select information relevant to current situation



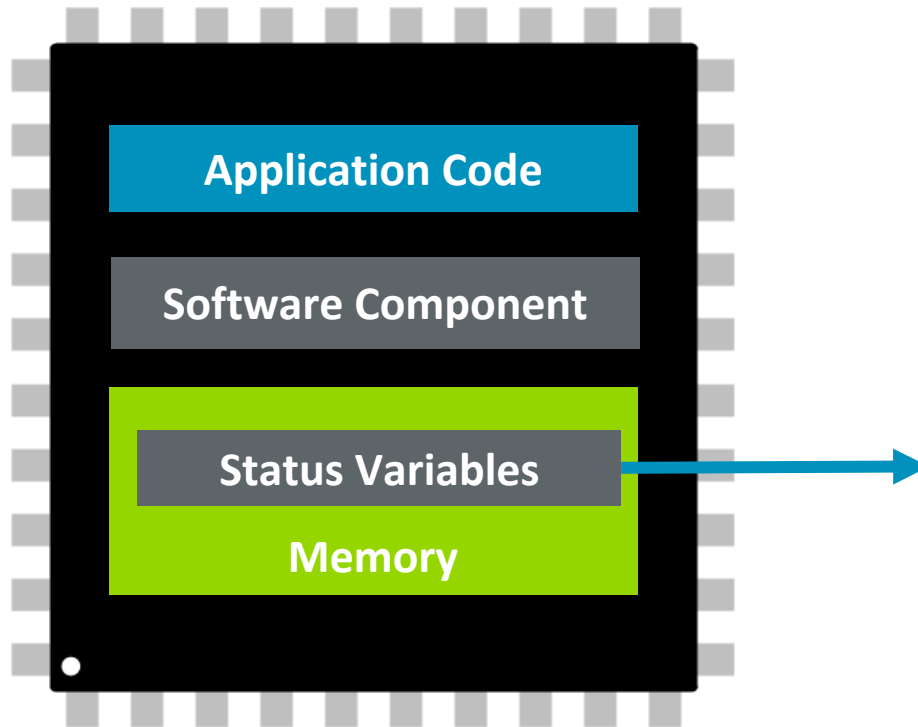
The 'Event Recorder' window shows a table of recorded events. The 'Enable' checkbox is checked. The 'Mark' dropdown is set to 'All Operations'. The 'Stopped' button is visible.

Event	Time (sec)	Component	Event Property	Value
0	0.05831509	EvCtrl	EventRecorderInitialize	Restart Count = 2
1	0.05831701	EvCtrl	EventRecorderStart	
2	0.05840147	Net_SYS	InitSystem	ver=7.12.0
3	0.05894136	Net_HTTPs	InitServer	sessions=6, port=0
4	0.05896325	Net_HTTPs	StartService	port=80
5	0.05899101	Net_DHCP	InitClient	vcid=0, opt=0
6	0.05899451	Net_DHCP	StartClient	start
7	0.05899959	Net_DHCP6	InitClient	opt=0
8	0.05900543	Net_DHCP6	StartClient	mode= Stateful
9	0.05901859	Net_SYS	InitComplete	success
10	0.06477536	Net_SYS	GetOption	netif=ETH0, opt=IP6_LLAddr

Component Viewer

Detailed status information of software components helps to understand operation and setup

Easy to add using XML file; pre-configured for Keil RTX5, FreeRTOS, and MDK-Middleware



The screenshot shows the 'Network' component viewer window. It has a tree view on the left and a table of values on the right. The tree view shows the following structure:

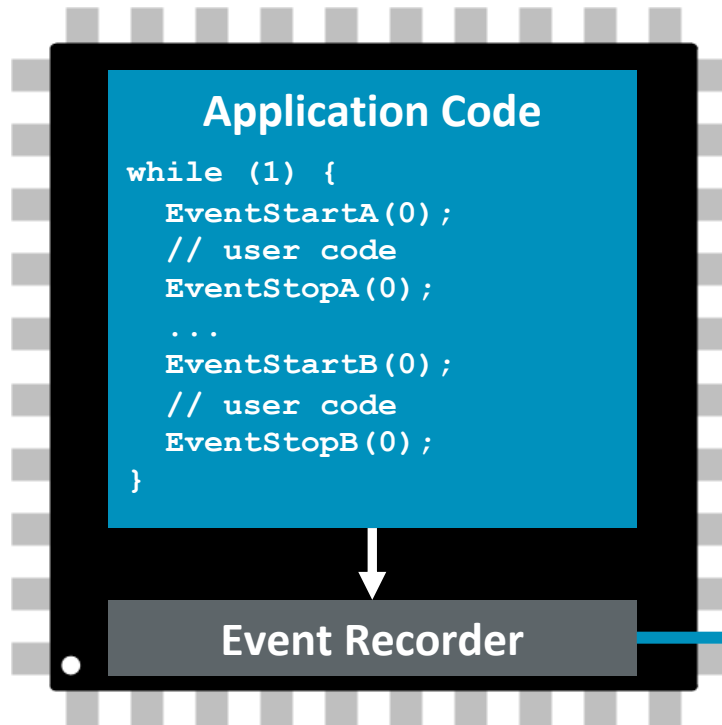
- Property
 - IPv4/IPv6
 - ETH interface
 - MAC address
 - VLAN
 - IPv4 settings
 - IP address
 - Network mask
 - Default gateway
 - Primary DNS server
 - Secondary DNS server
 - IP Fragmentation
 - IPv6 settings
 - UDP sockets
 - Socket 1
 - Socket 2
 - Socket 3
 - TCP sockets
 - Socket 1
 - Local Port
 - Callback Function
 - Options
 - Socket 2
 - Socket 3
 - Socket 4

The table on the right shows the following values:

Property	Value
Debug	Debug
Link-Up	Link-Up
MAC address	1E-30-6C-A2-45-5E
VLAN	Disabled
Address: 10.41.0.69	Address: 10.41.0.69
IP address	10.41.0.69
Network mask	255.255.255.0
Default gateway	10.41.0.11
Primary DNS server	10.58.2.242
Secondary DNS server	10.58.2.68
IP Fragmentation	Enabled
Address: ::	Address: ::
Used: 3, Available: 5	Used: 3, Available: 5
Opened, Port: 137	Opened, Port: 137
Opened, Port: 68	Opened, Port: 68
Opened, Port: 546	Opened, Port: 546
Used: 6, Available: 6	Used: 6, Available: 6
Listen, Port: 80	Listen, Port: 80
80	80
http_listener	http_listener
Keep-alive: Off, Flow-ctrl: Off, Delay-ACK: On	Keep-alive: Off, Flow-ctrl: Off, Delay-ACK: On
Listen, Port: 80	Listen, Port: 80
Listen, Port: 80	Listen, Port: 80
Listen, Port: 80	Listen, Port: 80

Event Statistics – record execution time of application code

- START: The basic function call is EventStartG(slot) and EventStartGv(slot, v1, v2)
- STOP: The basic function call is EventStopG(slot) and EventStopGv(slot, v1, v2)
 - Calls are arranged in four groups (G = A, B, C, D). v1, v2 is for data value
 - Each group has 15 slots (0 to 15); slot = 15 is global stop for all slots of a group



Event Statistics		
Source	Count	Execution Timing
Event Start/Stop Group A		
Slot=0	22	Total=710.71ms Avg=32.30ms Min=19.21ms Max=40.23ms First=10.23ms Last=1.25s
Min t: Start: val0=501 val1=0		Stop: "main.c" (49) t=19.21ms
Max t: Start: val0=917 val1=0		Stop: "main.c" (45) t=40.23ms
Slot=15	22 (+1)	Total=947.86ms Avg=41.21ms Min=27.14ms Max=62.31ms First=2.30ms Last=1.29s
Min t: Start: val0=501 val1=0		Stop: "main.c" (49) t=27.14ms
Max t: Start: val0=995 val1=0		Stop: "main.c" (49) t=62.31ms
Event Start/Stop Group B		
Slot=0	22	Total=336.00ms Avg=15.27ms Min=3.11ms Max=20.24ms First=45.49ms Last=1.27s
Min t: Start: "main.c" (83)		Stop: val0=1934 val1=203 t=3.11ms
Max t: Start: "main.c" (83)		Stop: val0=32440 val1=1333 t=20.24ms
Event Start/Stop Group C		
Slot=0	0 (+1)	
Event Start/Stop Group D		

Annotated Software Components

+ RTOS:

- Keil RTX5
- CMSIS-FreeRTOS

+ Debug:

- CMSIS-View:Fault

+ Middleware

- File System (MDK-Middleware)
- Network (MDK-Middleware)
- USB (MDK-Middleware)

Learn more

[Application Note KAN320: Using Event Recorder for debugging a network performance issue](https://arm-software.github.io/CMSIS-View/latest/er_use.html)

Event Recorder				
Enable Recorder: ☒  				
Event	Time (sec)	Component	Event Property	Value
0	0.02112980	RTX Threa	EventRecorderInitialize	Restart Count = 2
1	0.02113258	RTX Threa	EventRecorderStart	
2	0.02113722	RTX Threa	InitSystem	ver=7.12.0
3	0.02114145	RTX Threa	InitServer	sessions=6, port=0
4	0.02114462	RTX Threa	StartService	port=80
5	0.02114803	RTX Threa	InitClient	vcid=0, opt=0
6	0.04112960	RTX Threa	StartClient	start
7	0.04113213	RTX Threa	InitClient	opt=0
8	0.04113648	RTX Threa	StartClient	mode=Stateful
9	0.04114045	RTX Threa	InitComplete	success
10	0.04114362	RTX Threa	GetOption	netif=ETH0, opt=IP6_LLAddr
11	0.04114700	RTX Threa	ClientState	state=INIT
12	0.06112956	RTX Threa	SendDhcpMessage	type=DHCP_DISCOVER, bcast=1
13	0.06113209	RTX Threa	NextState	next=SELECTING
14	0.06113633	RTX Threa	ClientState	state=INIT
15	0.06114008	RTX Threa	NextStateDelay	next=START, delay=1000ms
16	0.06114325	RTX Threa	ReceiveFrame	server=10.41.0.11, len=300
17	0.06114657	RTX Threa	ClientState	state=SELECTING
18	0.08112954	RTX Thread	ThreadSwitch	thread_id=0x1FFF92B4
19	0.08113208	RTX Thread	ThreadDelayCompleted	
20	0.08113632	RTX Thread	ThreadUnblocked	thread_id=0x1FFF8250, ret_val=osOK
21	0.08114007	RTX Thread	ThreadSwitch	thread_id=0x1FFF8250
22	0.08114323	RTX Thread	ThreadDelay	ticks=20
23	0.08114655	RTX Thread	ThreadBlocked	thread_id=0x1FFF8250, timeout=20
24	0.10112954	RTX Thread	ThreadSwitch	thread_id=0x1FFF92B4
25	0.10113208	RTX Thread	ThreadDelayCompleted	
26	0.10113632	RTX Thread	ThreadUnblocked	thread_id=0x1FFF8250, ret_val=osOK
27	0.10114008	RTX Thread	ThreadSwitch	thread_id=0x1FFF8250
28	0.10114325	RTX Thread	ThreadDelay	ticks=20

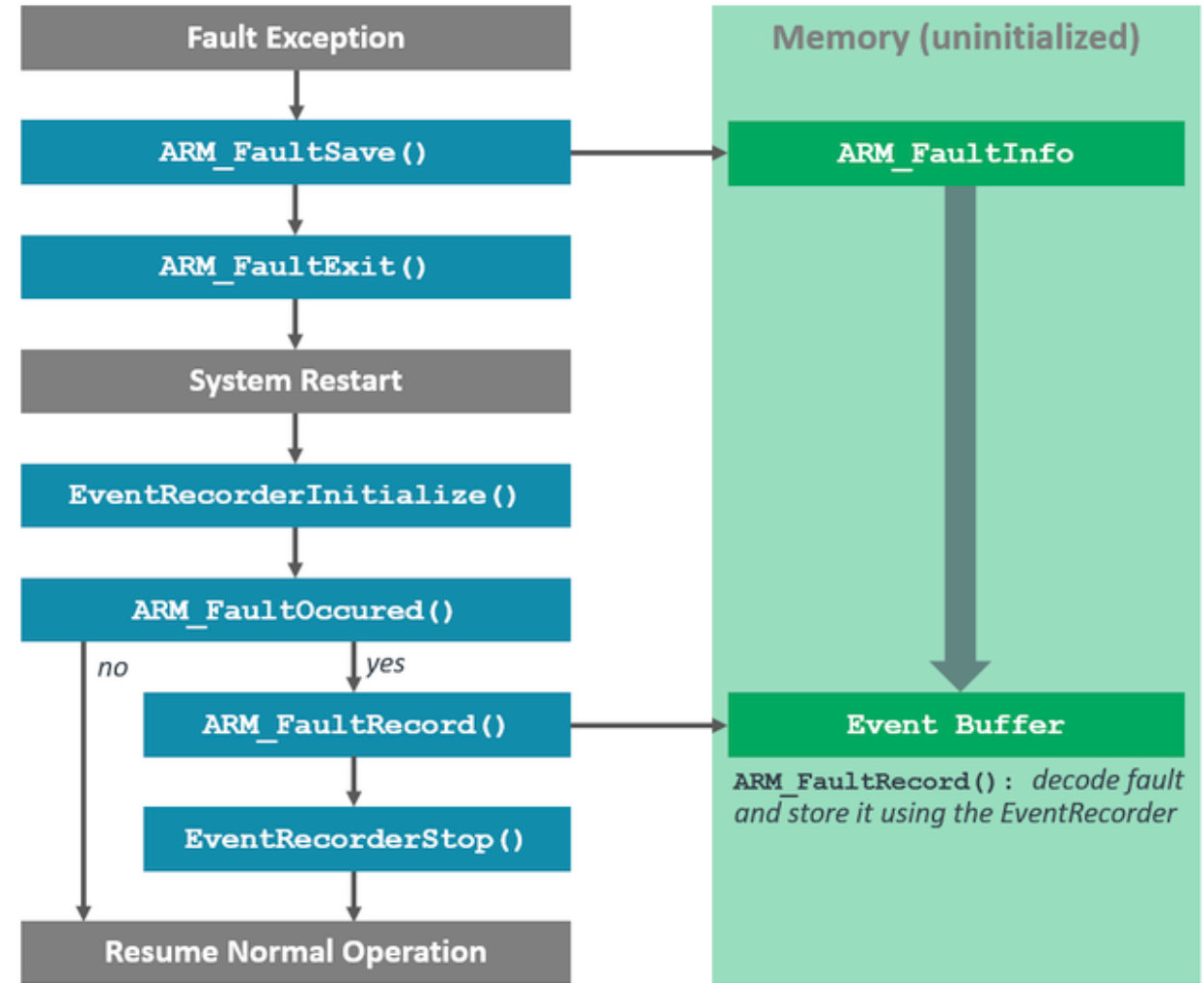
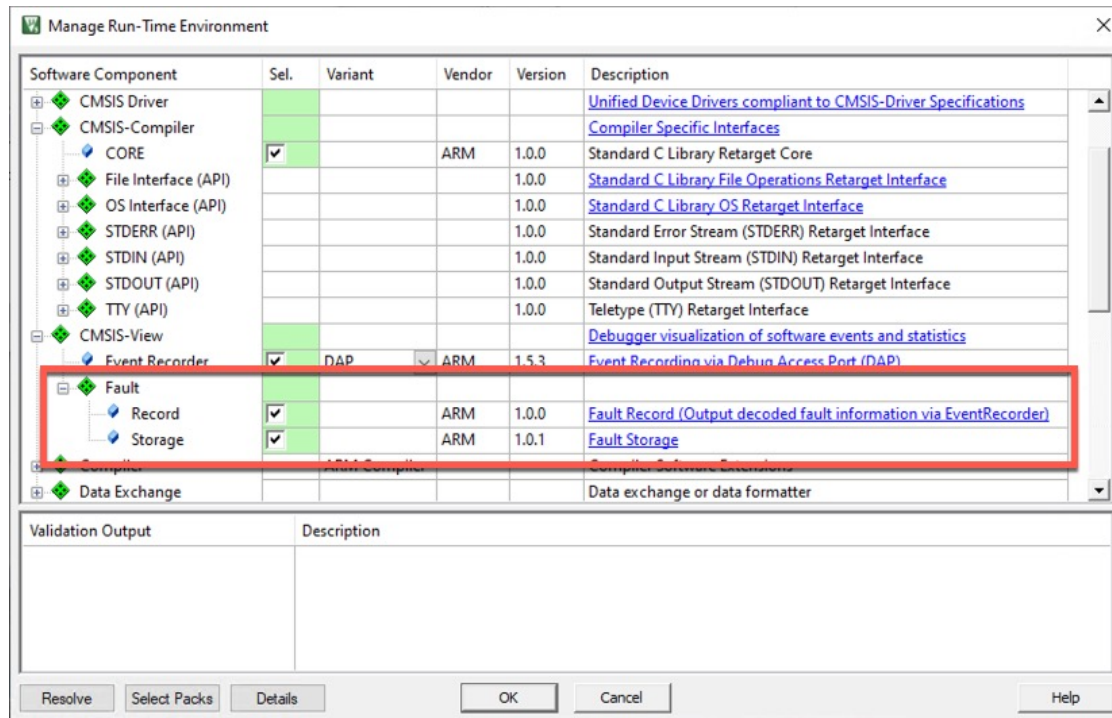
arm

Demo

Event Recorder
Component Viewer

Fault Analysis

- ✦ Provides infrastructure and API function calls to store, record, and analyze the Cortex-M Exception Fault information.

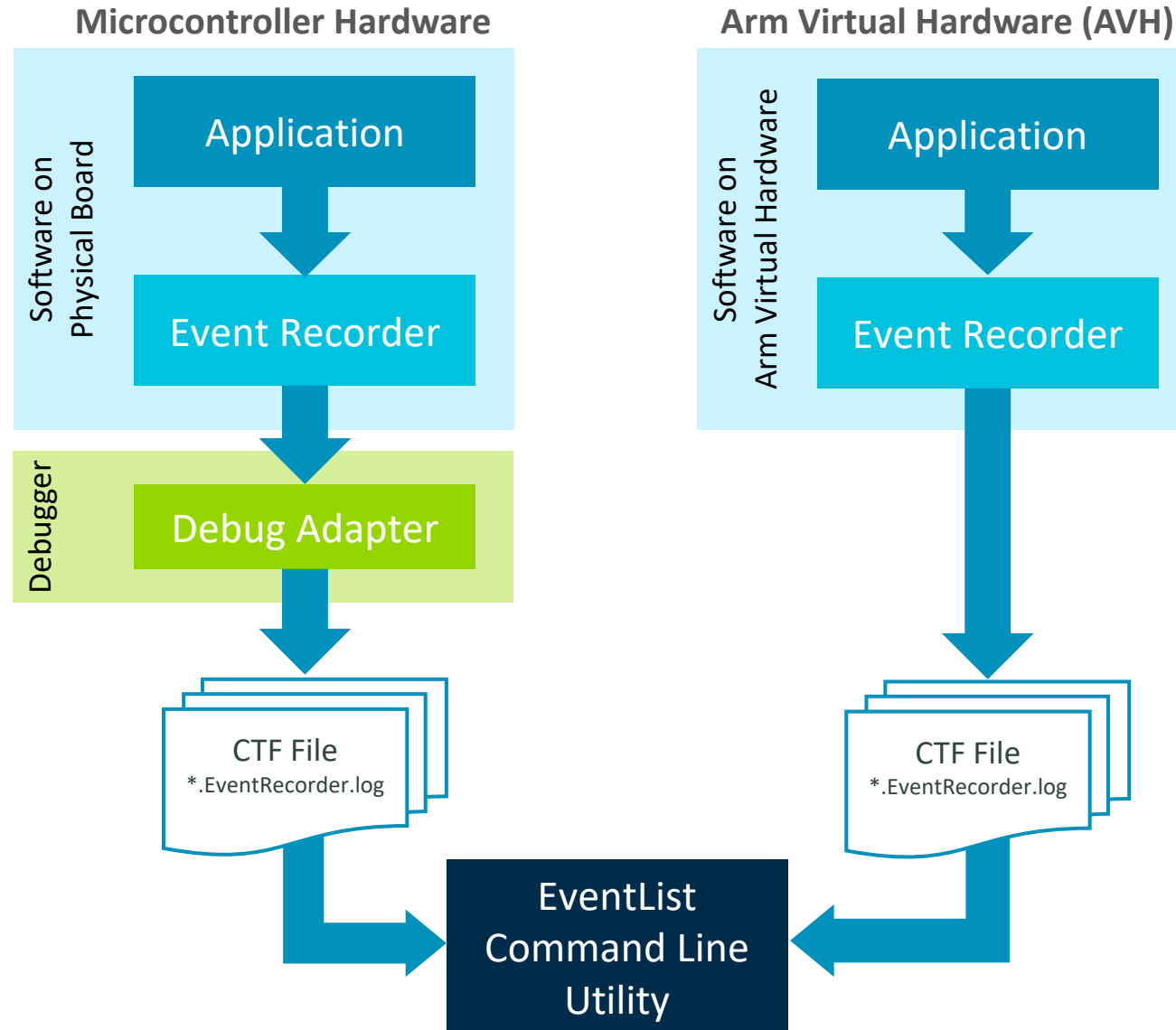


arm

Demo

Fault Analysis

Processing of Event Recorder Data

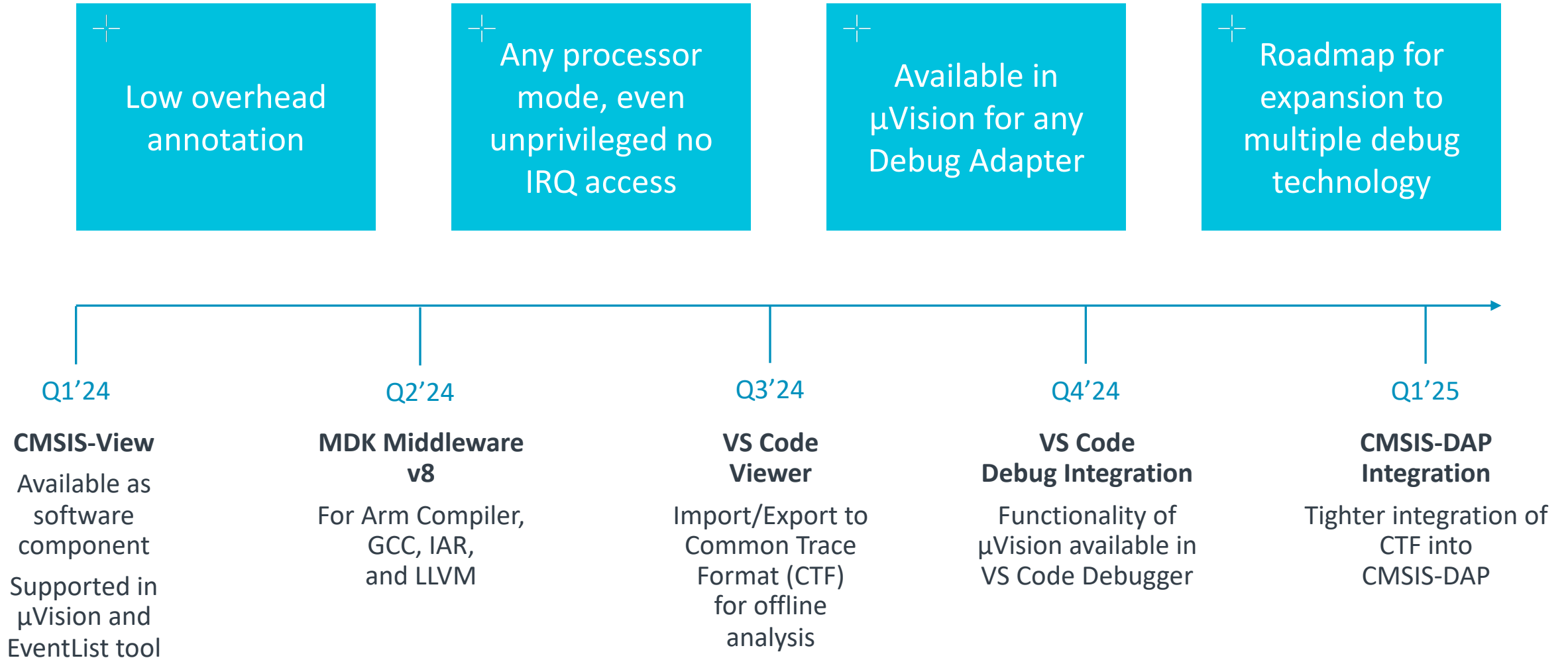


A CI/MLOps Example

✓ Get timings from Event Recorder Log

```
1 ▶ Run su - arm_mlops_docker -c "/home/arm_mlops_docker/cmsis-toolbox-linux-amd64/bin/eventlist -s ./EventRecorder.log"
4   Start/Stop event statistic
5   -----
6
7   Event count      total      min      max      average      first      last
8   -----
9   C(0)          1      2.05961s      2.05961s      2.05961s      2.05961s      2.05961s
10      Min: Start: 0.70457864 val1=0x00000000, val2=0x00000000 Stop: 2.76419180 val1=0x00000000, val2=0x00000000
11      Max: Start: 0.70457864 val1=0x00000000, val2=0x00000000 Stop: 2.76419180 val1=0x00000000, val2=0x00000000
12
13   C(1)          1      1.86834s      1.86834s      1.86834s      1.86834s      1.86834s
14      Min: Start: 2.76419892 val1=0x00000000, val2=0x00000000 Stop: 4.63253560 val1=0x00000000, val2=0x00000000
15      Max: Start: 2.76419892 val1=0x00000000, val2=0x00000000 Stop: 4.63253560 val1=0x00000000, val2=0x00000000
16
17   C(2)          1      414.96000µs      414.96000µs      414.96000µs      414.96000µs      414.96000µs
18      Min: Start: 4.63254272 val1=0x00000000, val2=0x00000000 Stop: 4.63295768 val1=0x00000000, val2=0x00000000
19      Max: Start: 4.63254272 val1=0x00000000, val2=0x00000000 Stop: 4.63295768 val1=0x00000000, val2=0x00000000
20
```

CMSIS-View: Roadmap for adoption



arm

CMSIS-Compiler

CMSIS-Compiler

Standard C Library File, I/O and OS Retargeting

What?

- + Simplify retargeting of standard C library functions.
- + Enables thread-safe operations when using a CMSIS-RTOS based real-time operating system.

Why?

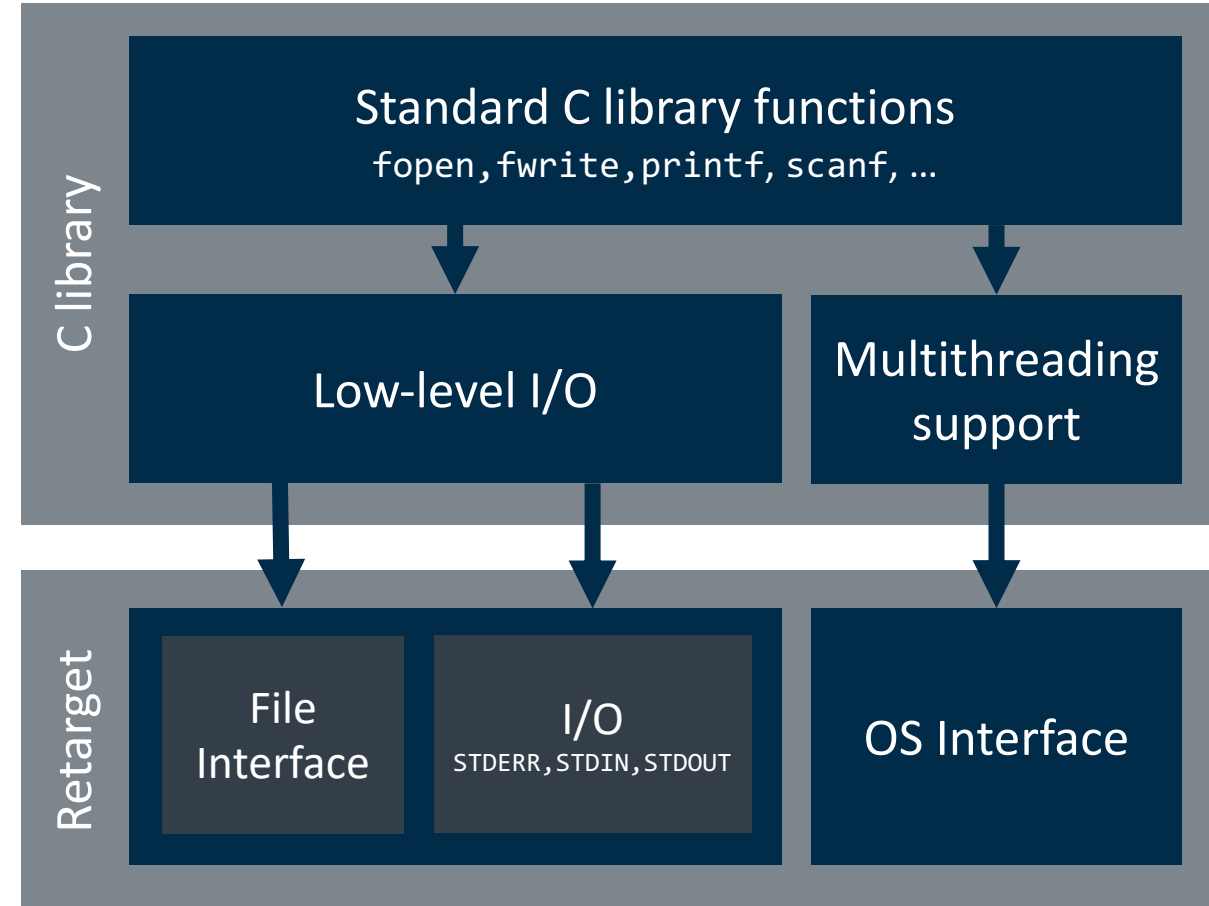
- + Makes it easy to port code across platforms and compilers.
- + File operations need to ensure that they get a file lock from the RTOS.

CMSIS-Compiler

Software components for retargeting I/O operations in standard C run-time libraries

Supported retargeting interfaces:

- + File interface for reading and writing files.
- + I/O interface for standard I/O stream retargeting (stderr, stdin, stdout).
- + OS interface for multithread safety using an arbitrary RTOS.



arm

Demo

Using CMSIS-Compiler

Upcoming webinars

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

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكراً

ধন্যবাদ

תודה

ధన్యవాదములు



The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks