ARM KNOWLEDGE LAB: DESIGNING THE FUTURE

DAY 3

Simplifying Functional Safety with Keil FuSa RTS

Trevor Martin, Hitex UK

Arm Knowledge Lab:
Three short and compact sessions

Day 1

Secure by Design – From Threats to Trusted Devices

Day 2

AI at the Edge – Building Smarter Embedded Systems

Day 3

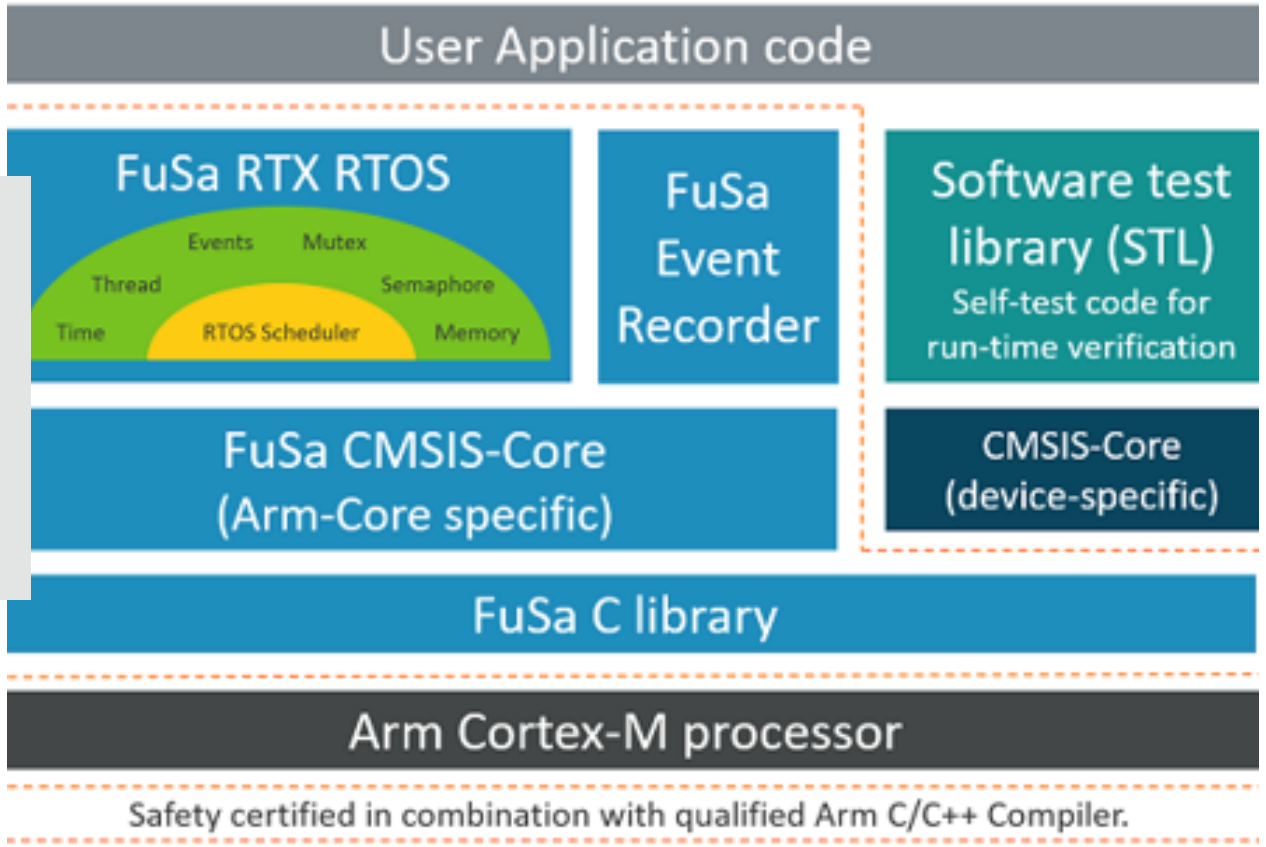
Safety-Critical Systems – Engineering Reliability

Find more presentations from our Arm Knowledge Lab:
https://www2.hitex.com/arm-knowledge-lab-2026-download

Keil Functional Safety Run Time System

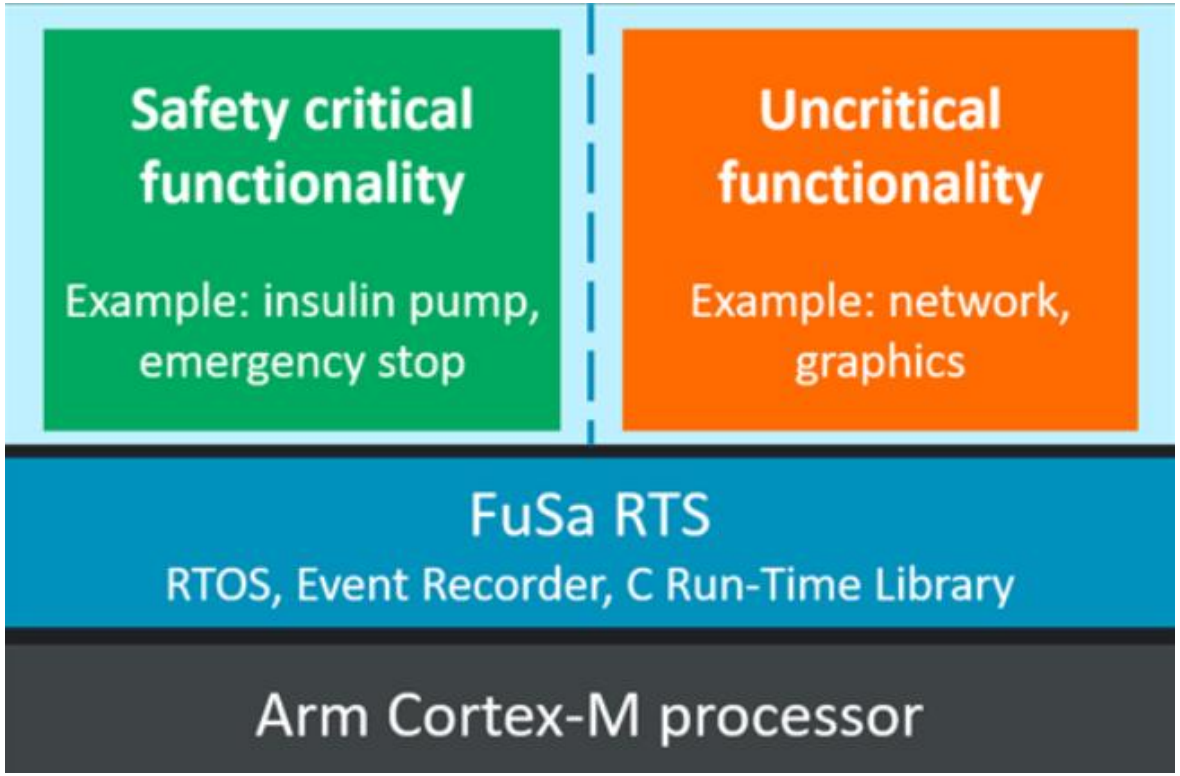
A set of pre-certified software components

- FuSa RTX provides additional safety features over the standard version



Functional isolation

Functional Isolation is intended to protect the safety elements of a system from a failure or malfunction if there is a failure or malfunction in the non safety elements of the system.

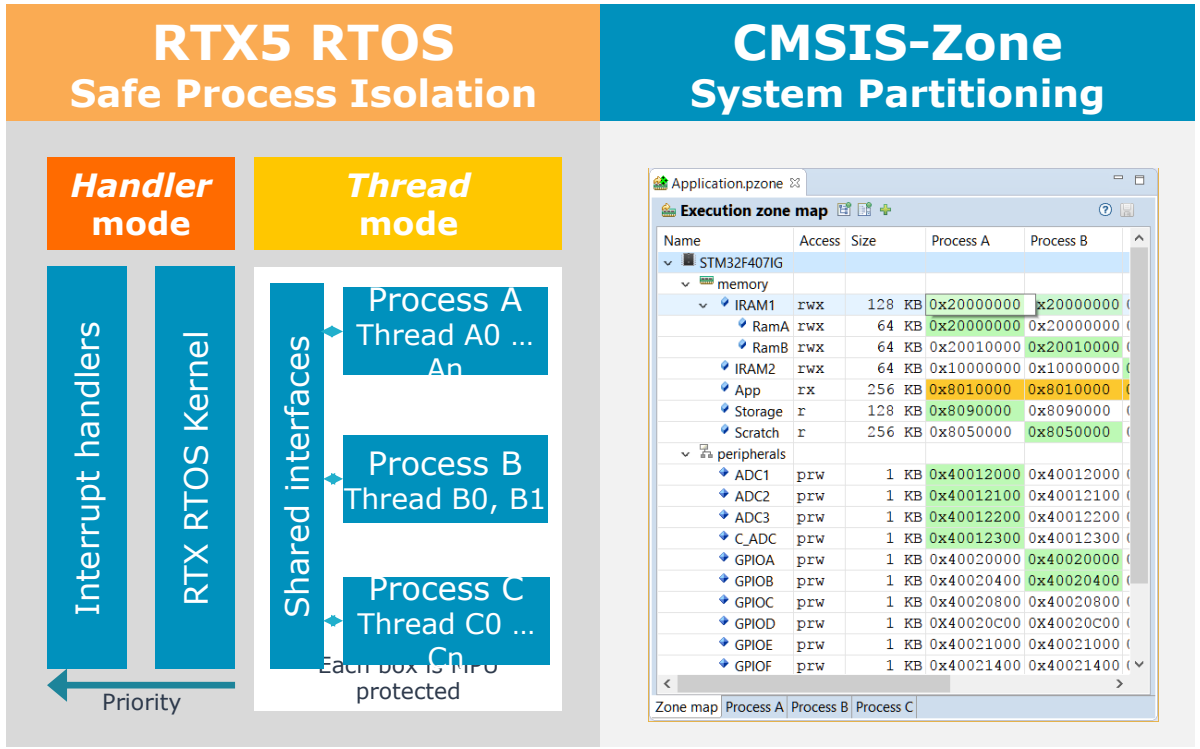


Functional Isolation

Feature	Description
Spatial Isolation	The ability to control access to the microcontroller resources depending on which software component is running within the system.
Temporal Isolation	Ability to monitor the timing constrains within the system
Controlled System Recovery	The ability to control system operation in the case of failures. This may involve proceeding to a safe operation state or blocking execution of non-safety components.

- In practice functional Isolation by partitioning the system using spatial and temporal Isolation
- Improved design
- Lower certification effort

RTOS with Functional Isolation



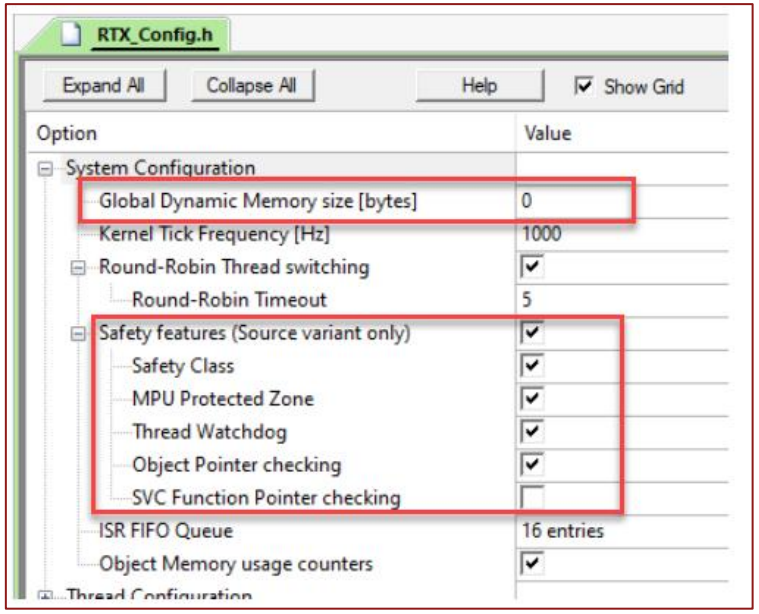
- Simplifies implementation of temporal independence (reducing SIL for parts of application)
- Memory Protection Unit (MPU) and processor features enable separation of RTOS kernel and application code processes
- Easy MPU setup with CMSIS-Zone utility

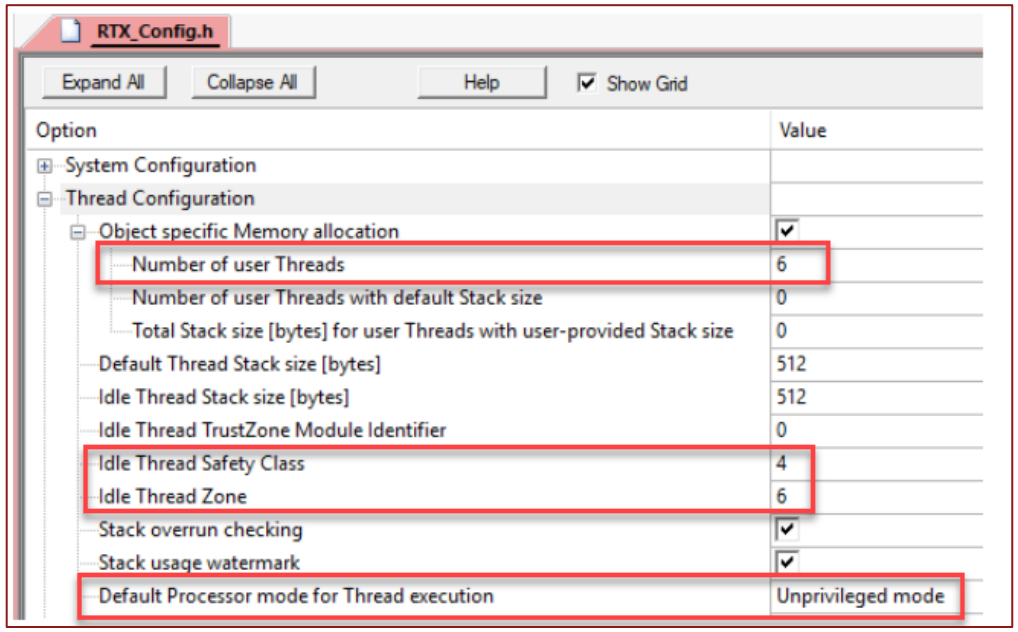
CMSIS COMPLIANT
ARM® Cortex™ Microcontroller Software Interface Standard

RTOS Safety features

Feature	Description
Kernel executes in Privileged mode	The RTOS code has full privileged access the thread code is limited to unprivileged
Static memory allocation	The thread stacks are statically allocated. The global Dynamic memory pool is disabled
MPU protection zone	The MPU is dynamically configured to ‘sandbox’ each thread and its required resources (RAM and Peripherals)
Safety Class	Communication between threads using RTOS objects is limited by defined safety classes
Thread watchdogs	The run time execution of each thread is monitored by thread watchdogs
Fault Handling	Additional RTOS support is provided to enter safe operating modes

RTOS Safety Configuration Options

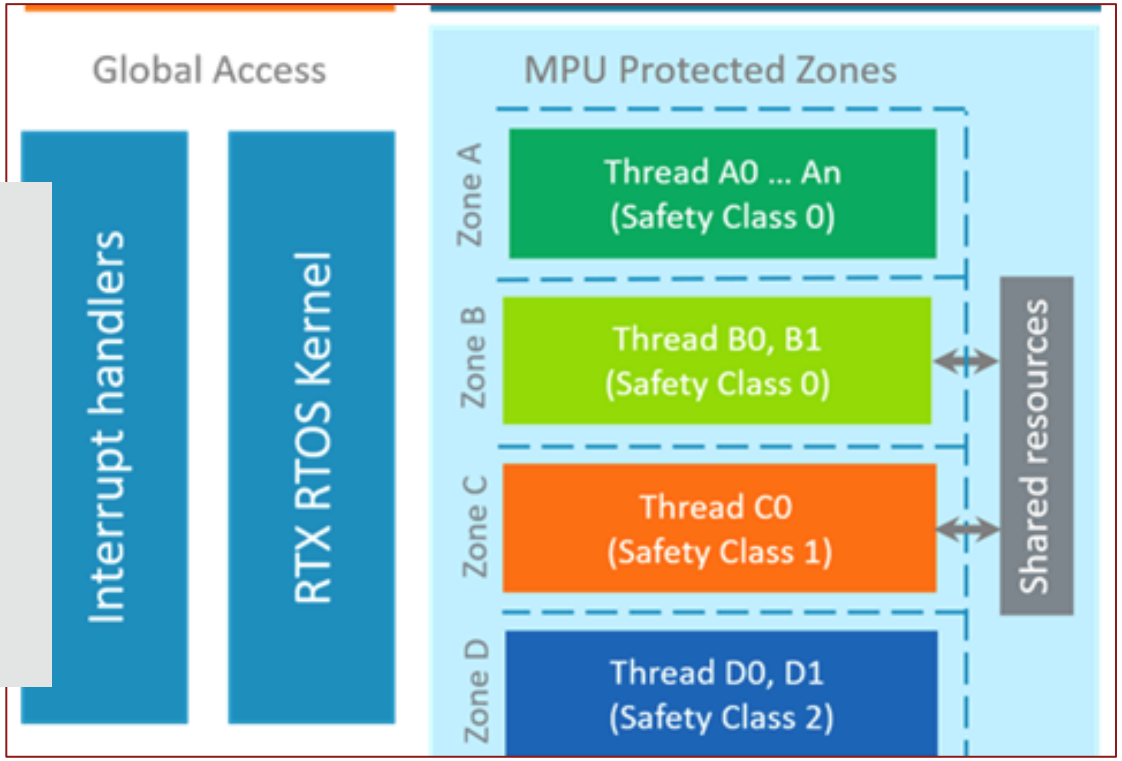




Spatial Isolation with Memory Protection Zones

A memory Protection Zone ‘sandboxes’ a thread and resources

- Done at the processor level using the Memory Protection Unit MPU
- Illegal memory access will generate an exception



CMSIS Zone

CMSIS Zone is intended to help you **manage complex memory maps**

- Safety
- Security
- Multicore

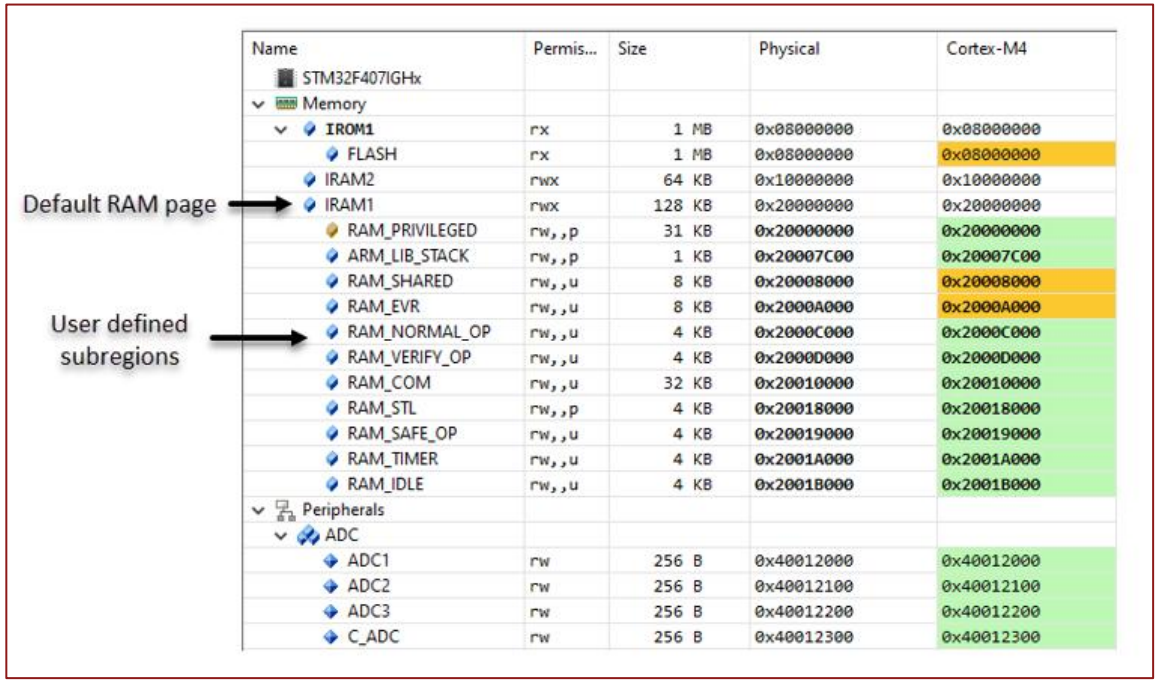
A Memory Zone is a **flexible concept**

- Project Zone
- Execution Zone

CMSIS-Zone **Utility**

- Eclipse plugin
- Allows you to define and manage Zones
  - Autogenerate configuration files

CMSIS Zone Utility



Name	Permis...	Size	Physical	Cortex-M4
STM32F407IGHx				
Memory				
IROM1	rx	1 MB	0x08000000	0x08000000
FLASH	rx	1 MB	0x08000000	0x08000000
IRAM2	rwx	64 KB	0x10000000	0x10000000
IRAM1	rwx	128 KB	0x20000000	0x20000000
RAM_PRIVILEGED	rw,,p	31 KB	0x20000000	0x20000000
ARM_LIB_STACK	rw,,p	1 KB	0x20007C00	0x20007C00
RAM_SHARED	rw,,u	8 KB	0x20008000	0x20008000
RAM_EVR	rw,,u	8 KB	0x2000A000	0x2000A000
RAM_NORMAL_OP	rw,,u	4 KB	0x2000C000	0x2000C000
RAM_VERIFY_OP	rw,,u	4 KB	0x2000D000	0x2000D000
RAM_COM	rw,,u	32 KB	0x20010000	0x20010000
RAM_STL	rw,,p	4 KB	0x20018000	0x20018000
RAM_SAFE_OP	rw,,u	4 KB	0x20019000	0x20019000
RAM_TIMER	rw,,u	4 KB	0x2001A000	0x2001A000
RAM_IDLE	rw,,u	4 KB	0x2001B000	0x2001B000
Peripherals				
ADC				
ADC1	rw	256 B	0x40012000	0x40012000
ADC2	rw	256 B	0x40012100	0x40012100
ADC3	rw	256 B	0x40012200	0x40012200
C_ADC	rw	256 B	0x40012300	0x40012300

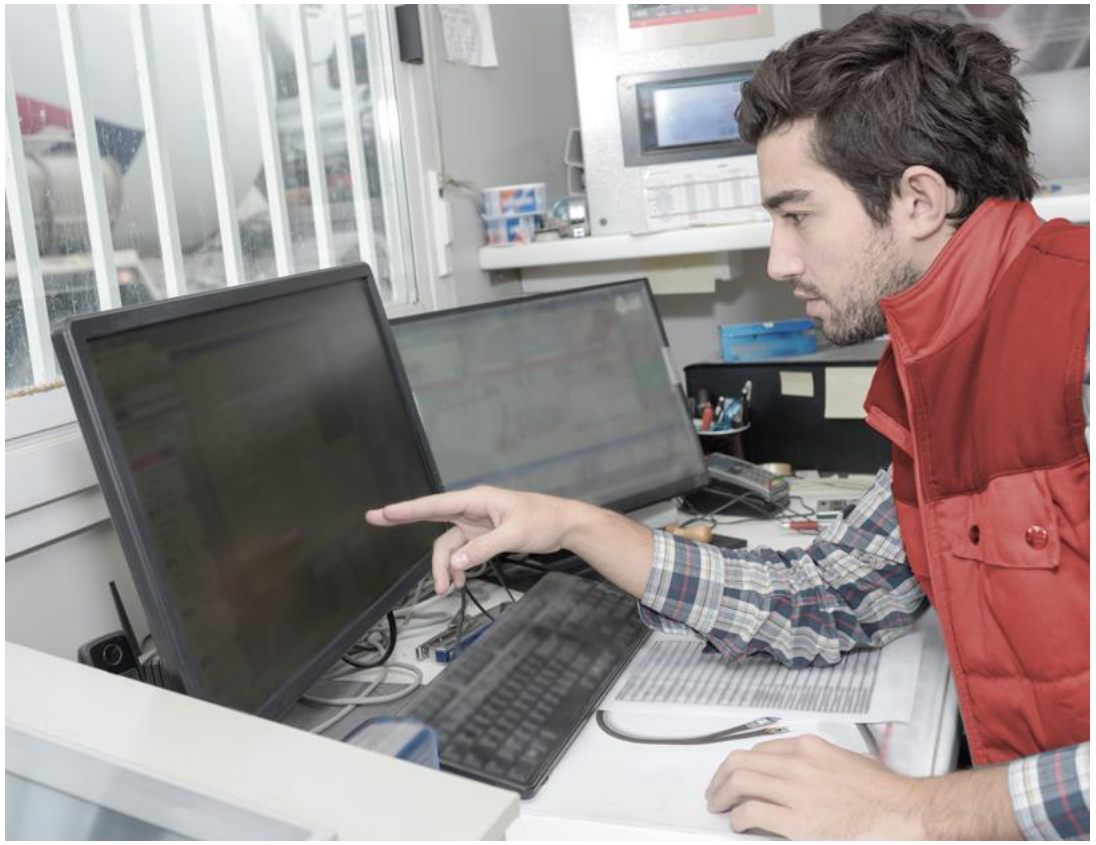
Import the memory map of a microcontroller from its CMSIS Pack file

- Create user defined memory sub regions

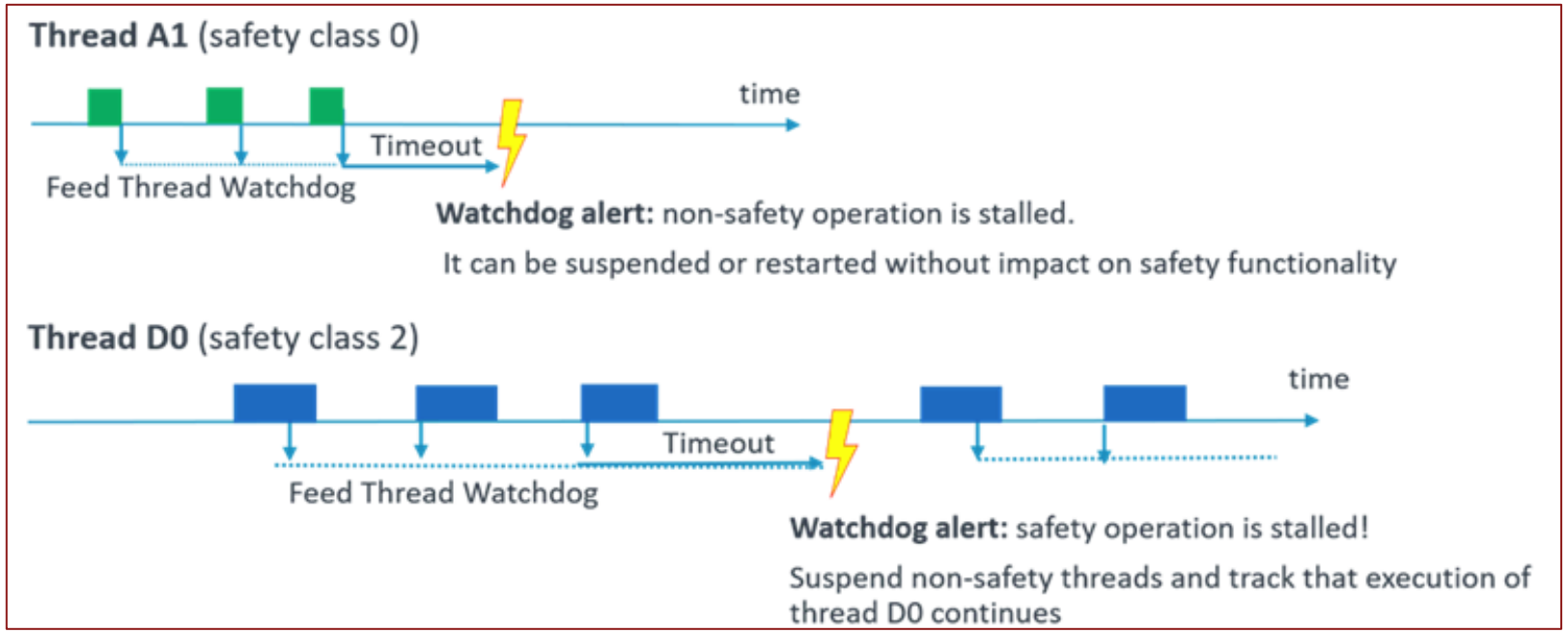
Safety Class

But we have a problem!

- RTOS Objects are pointer based
  - Can be accessed regardless of Memory Protection Zone
- Each object, including Threads are created with a **Safety Class** value
- A Thread cannot modify an RTOS object with a higher **Safety Class** value



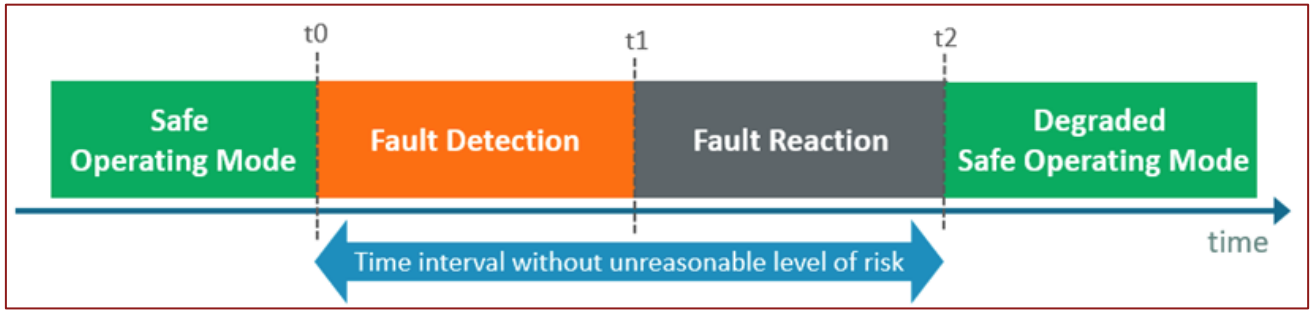
Temporal Isolation



Temporal Isolation is provided by Thread Watchdogs

- You must feed each Thread Watchdog
- A timeout triggers a callback function

Error Handling



Fault	Description
Startup fault	Hardware configuration failure at startup
STL Error	Run Time hardware CPU or memory failure
Memory Manager fault	Run time Memory Protection Zone error
Watchdog Alarm handler	Thread Temporal failure
RTX error	RTOS Kernel error
Processor error exceptions	The validating thread discovers a failure in the main control thread

Conclusion

- Process Isolation lowers the overall certification effort
- FuSa RTX can also be used to improve device security
- The FuSa RTX safety features are also included in the uncertified version of RTX

Hitex offerings …

Ask alexander.hess@hitex.de – he looks forward to receiving your inquiries regarding Hitex offerings, including:

- Development support & services
- Self-test libraries for ArmV8 microcontrollers
- Training courses: From tools and software to RTOS and Security with Arm
- Unit testing for embedded software with TESSY
- SO-DIMM compute modules for PSOC™, TRAVEO™ and more for an easy entry into the Infineon MCs.

Stay informed: webinars, training sessions, knowledge labs
Find all the information at www.hitex.de