



Elektrobit

# EB tresos<sup>®</sup> AutoCore OS release notes CORTEXM S32K14X

product release 6.1





Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2023, Elektrobit Automotive GmbH.



# Table of Contents

- 1. Release Notes ..... 4
  - 1.1. Supported Compilers & Hardware used for testing ..... 4
  - 1.2. New features ..... 4
  - 1.3. Obsolete features ..... 5
  - 1.4. EB-specific enhancements ..... 5
  - 1.5. Deviations ..... 5
  - 1.6. Open-source software ..... 7
    - 1.6.1. Open-source software in software executed on the ECU ..... 8
    - 1.6.2. Open-source software in software used for the development infrastructure ..... 8
  - 1.7. Change log ..... 8

# 1. Release Notes

Release:

EB tresos AutoCore OS 6.1.144 [revision 000]

Date:

2023-03-08

Supported architecture:

CORTEXM/S32K14X

Supported AUTOSAR versions:

Release 4.0 revision 3, SWS document version 5.0.0

Release R19-11

Supplier:

Elektrobit Automotive GmbH

## 1.1. Supported Compilers & Hardware used for testing

For an updated list of supported compilers including supported compiler options and hardware used for testing, see the quality statement provided with the delivery.

## 1.2. New features

The following new features have been implemented in EB tresos AutoCore OS 6.1:

- ▶ The module now implements the *Specification of Operating System*, AUTOSAR release R19-11.
- ▶ The module provides the asynchronous versions *ActivateTaskAsyn()* and *SetEventAsyn()* for the API functions *ActivateTask()* and *SetEvent()*.
- ▶ The module provides the API function *GetCurrentApplicationID()*, which returns the calling application when called from a trusted function.
- ▶ The module provides the API function *ControlIdle()* so that the application can select the behavior of the idle loop.
- ▶ The module provides the API function *ClearPendingInterrupt()*, which clears the specified pending interrupt source.

## 1.3. Obsolete features

The following features have been removed from EB tresos AutoCore OS 6.1:

- ▶ support for the CLZ scheduling algorithm
- ▶ support for CPU load measurement
- ▶ support for faster versions of *SuspendAllInterrupts()* and *ResumeAllInterrupts()*

## 1.4. EB-specific enhancements

The following enhancements are implemented in EB tresos AutoCore OS 6.1:

- ▶ Code Size and Execution Time Optimizations using
  - ▶ Configuration-dependent source-level optimization
  - ▶ Customizable AUTOSAR support
  - ▶ Use of a single hardware timer for all timing protection and timebase counters
  - ▶ Optional small auto-incrementing software counter
- ▶ API functions for run-time measurement of tasks and ISRs
- ▶ At least 32-bit counters on every supported architecture, including those with 16-bit hardware timers.
- ▶ OS availability for both 64-bit and 32-bit MCU platforms

## 1.5. Deviations

This section lists the deviations of EB tresos AutoCore OS 6.1 from the supported AUTOSAR versions.

The AUTOSAR requirement IDs are given in the style of the AUTOSAR R19-11 documents (e.g. SWS\_Os\_00123) unless the deviation only applies to the AUTOSAR 4.0.3 standard.

To convert R19-11 IDs to the 4.0.3 style, take the last three digits of the number and prefix the module short name in capitals. For example, [SWS\_Os\_00123] is equivalent to [OS123].

- ▶ [SWS\_Os\_91010 to SWS\_Os\_91018]: The peripheral read/write/modify API is not implemented.
- ▶ [All IOC requirements]: IOC is provided as a separate module for EB tresos Safety OS only.
- ▶ [ECUC\_Os\_00032]: The `OsGptChannelRef` configuration parameter is ignored.

EB tresos AutoCore OS provides its own drivers for hardware timers. The AUTOSAR SWS has no requirement to use the GPT module.



- ▶ [ECUC\_Os\_00402]: The `OsMemoryMappingCodeLocationRef` configuration parameter is ignored.

Memory protection for code sections is described in SWS\_Os\_00027 and is optional. Furthermore, there is no conceivable use case for preventing non-trusted OS applications from executing trusted code, because any data that the trusted code accesses is protected.

- ▶ [ECUC\_Os\_00394]: The `OsTrustedApplicationWithProtection` configuration parameter is ignored.

The tasks and ISRs of a trusted OS application execute without memory access restrictions.

- ▶ [SWS\_Os\_00265, SWS\_Os\_00266, SWS\_Os\_00312, SWS\_Os\_00563 to SWS\_Os\_00565, SWS\_Os\_00364, SWS\_Os\_00365, ECUC\_Os\_00395]: The `OsTrustedApplicationDelayTimingViolationCall` configuration parameter is ignored.

AutoCore OS retains the trusted function semantics specified by AUTOSAR 4.0.3. The AUTOSAR R19-11 requirements for trusted functions contain contradictions and inconsistencies. See <https://jira.autosar.org/browse/AR-101130> for details.

- ▶ [SWS\_OS\_00672]: When called from a permitted context, `GetNumberOfActivatedCores()` returns the number of cores configured to run AutoCore OS, as specified by SWS\_OS\_00626.

SWS\_OS\_00672 requires the return value to be the number of cores activated by the `StartCore()` function, which by definition would be one fewer than the number of cores running AutoCore OS and therefore contradicts SWS\_OS\_00626. See <https://jira.autosar.org/browse/AR-102329> for details.

- ▶ [SWS\_Os\_00505]: `NextScheduleTable()` does not continue synchronization.

There is no configured relationship between the synchronization requirements of the two schedule tables in a call to `NextScheduleTable()`. The relationship is defined by the application at run-time. The implementation therefore follows the AUTOSAR 3.1 specification, so synchronization does not continue. To ensure correct synchronization, the application must call `SyncScheduleTable()` after the next schedule table starts scheduling tasks.

- ▶ [SWS\_Os\_00566]: The error code `E_OS_PARAM_POINTER` is never returned by AutoCore OS.

AutoCore OS checks pointer parameters for write access against the memory protection configuration and returns `E_OS_ILLEGAL_ADDRESS` as specified by SWS\_Os\_00051. It is assumed that write access to address 0 is not permitted by the memory protection configuration.

- ▶ [SWS\_Os\_00557]: If `ProtectionHook()` returns `PRO_TERMINATEAPPL_RESTART` but the OS application does not have a restart task, AutoCore OS does not shut down.

Instead, the OS terminates the OS application. The implementation is consistent with a call to `TerminateApplication()`.

- ▶ [SWS\_Os\_00506]: If `ProtectionHook()` is called with `E_OS_PROTECTION_ARRIVAL` and returns a value other than `PRO_IGNORE` or `PRO_SHUTDOWN`, AutoCore OS does not shut down.

AutoCore OS handles the return value from `ProtectionHook()` as specified by AUTOSAR 3.1.



- ▶ [OS560]: `GetActiveApplicationMode()` is not provided as a software component service.

Note that `GetActiveApplicationMode()` is no longer required as a service interface by AUTOSAR R19-11.

- ▶ [SWS\_Os\_00560]: The `GetCounterValue()` and `GetElapsedValue()` software component services return `TickType` values instead of `TimeInMicroSecondsType`. Consequently, `TimeInMicroSecondsType` is not declared by the OS.

`TickType` is the data type that is used by the underlying OS API. On most hardware the maximum value of a hardware counter is limited to 32 bits. See <https://jira.autosar.org/browse/AR-101102> for details.

- ▶ [SWS\_Os\_00560]: In the software component services, `CounterType` is not `uint32`.

AutoCore OS declares `CounterType` as `sint32` to match the underlying implementation of the API.

- ▶ [SWS\_Os\_00584, SWS\_Os\_00585, SWS\_Os\_00680, SWS\_Os\_00682, SWS\_Os\_00683, SWS\_Os\_00684, SWS\_Os\_00685]: `StartNonAutosarCore()` is not implemented.

The ability to start cores that are not under the control of the OS depends on system-dependent information that is unavailable to the OS. If the functionality of `StartNonAutosarCore()` is required, the API must be supplied by the system integrator using information about the bootstrap address for the non-AUTOSAR core. A generic implementation is not possible.

- ▶ [SWS\_Os\_00313, SWS\_Os\_00314] `CheckTaskMemoryAccess()` and `CheckISRMemoryAccess()` return 0 (i.e. no access rights), if you call them with a memory area that straddles two or more memory regions.

In most cases there is a physical alignment gap between any two memory regions. The gap is not part of either region, but the alignment of the boundary addresses means that the OS cannot determine the existence of the gap. The implementation of the API in AutoCore OS ensures that the behavior of the API is consistent, regardless of the hardware alignment requirements and the exact sizes of the regions.

- ▶ Assigning an OS application to cores is done via the `OsApplicationCoreAssignment` parameter rather than the `OsApplicationCoreRef` parameter.
- ▶ For multi-core related APIs, the service IDs used during error reporting may not match the AUTOSAR SWS. For example, `ActivateTaskAsyn()`.
- ▶ [SWS\_Os\_00058, SWS\_Os\_00198] The OS runs all trusted and non-trusted ISRs and hook functions in the privileged handler mode of the Cortex-M architecture. Consequently, the OS does not protect the kernel stack from unwanted modifications by non-trusted ISRs and hooks.

## 1.6. Open-source software

The software that is delivered with EB tresos AutoCore OS can be classified into the following two categories:

- ▶ Software that is executed on the electronic control unit (ECU).



- ▶ Software that is used for the development infrastructure (configuration, generation, building) and thus executed on the development platform.

All license text files are located in your module delivery in the sub-folder

`<installed plugin location>\Os_<variant_string>\licenses`

## 1.6.1. Open-source software in software executed on the ECU

No open-source software that runs on the ECU is delivered with EB tresos AutoCore OS

## 1.6.2. Open-source software in software used for the development infrastructure

The following list of open-source software that is used in development is delivered with EB tresos AutoCore OS:

- ▶ The Apache Velocity Engine  
1.7  
<https://archive.apache.org/dist/velocity/engine/1.7>

List of licenses:

- ▶ Apache License Version 2.0  
VTE17\_Apache20\_license.txt

List of copyrights:

- ▶ Copyright (C) The Apache Software Foundation.

## 1.7. Change log

This chapter lists the changes between different versions for the CORTEXM architecture. This list includes those changes that affect all architectures together with those affecting the CORTEXM architecture.



**WARNING****Your delivery may not be suitable for use in production**

The final quality level assigned to your release may not be used for production if it has not been assigned **ready for manufacturing (RFM)**. Check your quality statement for details. The information provided in these release notes is the expected quality level and is for guidance only.

## EB tresos AutoCore OS version 6.1.144

CORTEXM S32K14X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32K14X derivative

## EB tresos AutoCore OS version 6.1.141

CORTEXM S32R45X release (including EB tresos Safety OS support)

Quality level: ready for manufacturing.

- ▶ Corrected parameter multiplicities published in the user documentation references section
- ▶ Updated support for S32R45X derivative

## EB tresos AutoCore OS version 6.1.138

CORTEXM S32G399 release (including EB tresos Safety OS support)

Quality level: ready for manufacturing.

- ▶ Updated support for S32G399 derivative
- ▶ Fixed several API names and prototypes in the user documentation

## EB tresos AutoCore OS version 6.1.132

CORTEXM S32Z27XM33 release

Quality level: development drop. **Do not use for production.**

- ▶ Updated support for S32Z27XM33 derivative



## EB tresos AutoCore OS version 6.1.131

CORTEXM S32G399 release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Updated support for S32G399 derivative

## EB tresos AutoCore OS version 6.1.130

CORTEXM S32G27X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32G27X derivative

## EB tresos AutoCore OS version 6.1.129

CORTEXM SR6P6M4 release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Added support for SR6P6M4 derivative
- ▶ Updated memory protection support

## EB tresos AutoCore OS version 6.1.127

CORTEXM S32R45X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32R45X derivative

## EB tresos AutoCore OS version 6.1.126

CORTEXM S32K14X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32K14X derivative



## EB tresos AutoCore OS version 6.1.124

CORTEXM S32K31X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32K31X derivative

## EB tresos AutoCore OS version 6.1.123

CORTEXM SAF85XXM7 release

Quality level: ready for manufacturing.

- ▶ Added SAF85XXM7 derivative to architecture notes
- ▶ Updated support for SAF85XXM7 derivative

## EB tresos AutoCore OS version 6.1.120

CORTEXM CYT4BB release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Updated support for CYT4BB derivative
- ▶ ASCOS-8036 Fixed known issue: OS enters endless exception loop when user tries to call a function in ARM mode
- ▶ Added support for the Traveo II family's interrupt multiplexer
- ▶ ASCOS-8144 Fixed known issue: Atomics functions may lose updates of atomic objects
- ▶ ASCOS-8012 Fixed known issue: Atomics functions may overwrite nearby variables
- ▶ ASCOS-7788 Fixed known issue: Possible compilation warnings in generated IOC files

## EB tresos AutoCore OS version 6.1.119

CORTEXM SAF85XXM7 release

Quality level: development drop. **Do not use for production.**

- ▶ Changed the stack size configuration so that the generator does not add any overhead for EB tresos Safety OS



- ▶ Added support for SAF85XXM7 derivative
- ▶ Updated architecture notes with additional information for memory protection handling in ISRs and application hooks
- ▶ Improved documentation of atomics chapter including supporting glossary entries

## EB tresos AutoCore OS version 6.1.113

CORTEXM S32K34X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32K34X derivative

## EB tresos AutoCore OS version 6.1.111

CORTEXM CYT2B75 release

Quality level: ready for manufacturing.

- ▶ Added support for CYT2B75 derivative

## EB tresos AutoCore OS version 6.1.108

CORTEXM CYT4BB release

Quality level: ready for manufacturing.

- ▶ Added CYT4BB derivative to architecture notes
- ▶ Added support for CYT4BB derivative

## EB tresos AutoCore OS version 6.1.104

CORTEXM S32R45X release (including EB tresos Safety OS support)

Quality level: ready for development. **Do not use for production.**

- ▶ Updated support for S32R45X derivative
- ▶ Changed implementation of timestamp timer to use the NXP PIT module as internal timestamp supplier on derivatives that do not use the PIT module to implement Autosar counters



## EB tresos AutoCore OS version 6.1.101

CORTEXM CYT4BB release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for CYT4BB derivative

## EB tresos AutoCore OS version 6.1.99

CORTEXM S32G399 release (including EB tresos Safety OS support)

Quality level: ready for development. **Do not use for production.**

- ▶ Changed set of available timers for S32G399 derivative to STM0 to STM7

## EB tresos AutoCore OS version 6.1.97

CORTEXM RTL90XXA release

Quality level: ready for manufacturing.

- ▶ Updated support for RTL90XXA derivative

## EB tresos AutoCore OS version 6.1.95

CORTEXM S32R45X release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Added support for S32R45X derivative

## EB tresos AutoCore OS version 6.1.92

CORTEXM S32G399 release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Updated support for S32G399 derivative
- ▶ Added support for ORTI tracing hooks in EB tresos Safety OS and general ORTI improvements
- ▶ ASCOS-7324 Fixed known issue: Multi-core Safety OS: incorrect ORTI definition for vs\_RUNNINGTH-READPRIORITY



## EB tresos AutoCore OS version 6.1.86

CORTEXM S32G27X release (including EB tresos Safety OS support)

Quality level: ready for manufacturing.

- ▶ Updated support for S32G27X derivative
- ▶ Improved support of user-configurable internal interrupts for banked interrupts in EB tresos Safety OS configurations

## EB tresos AutoCore OS version 6.1.85

CORTEXM S32G27X release

Quality level: ready for manufacturing.

- ▶ Updated support for S32G27X derivative

## EB tresos AutoCore OS version 6.1.83

CORTEXM IMX8DXLM4 release

Quality level: ready for manufacturing.

- ▶ Updated support for IMX8DXLM4 derivative
- ▶ ASCOS-7643 Fixed known issue: Hardware counter based on the LPIT timer stops working

## EB tresos AutoCore OS version 6.1.78

CORTEXM S32K14X release

Quality level: ready for manufacturing.

- ▶ Improved architecture notes for S32K14X derivative
- ▶ Updated support for derivative S32K14X

## EB tresos AutoCore OS version 6.1.77

CORTEXM S32G399 release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**



- ▶ Updated support for derivative S32G399

## EB tresos AutoCore OS version 6.1.76

CORTEXM S32Z27XM33 release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative S32Z27XM33

## EB tresos AutoCore OS version 6.1.75

CORTEXM S32G27X release (including EB tresos Safety OS support)

Quality level: development drop. **Do not use for production.**

- ▶ Updated support for derivative S32G27X
- ▶ Changed the severity of the optional EcuC core ID consistency check from error to warning

## EB tresos AutoCore OS version 6.1.68

CORTEXM CYT4BF release

Quality level: ready for manufacturing.

- ▶ Updated support for derivative CYT4BF

## EB tresos AutoCore OS version 6.1.67

CORTEXM S32K31X release

Quality level: ready for manufacturing.

- ▶ Added support for derivative S32K31X

## EB tresos AutoCore OS version 6.1.66

CORTEXM S32G399 release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative S32G399

## EB tresos AutoCore OS version 6.1.64

CORTEXM BCM89107 release

Quality level: ready for manufacturing.

- ▶ Improved architecture notes for all supported derivatives
- ▶ Updated support for derivative BCM89107
- ▶ ASCOS-7529 Fixed known issue: The OS fails to call ProtectionHook() for a non-maskable interrupt
- ▶ ASCOS-7531 Fixed known issue: OS fails to schedule upcoming tasks when the critical section of the OS dispatcher is interrupted by a Category 1 ISR
- ▶ Updated support for derivative CYT4BF
- ▶ Added support for MkMemoryRegionPrepend parameter to the EB tresos Safety OS memory-region configuration
- ▶ Updated error handling for StartOS() API

## EB tresos AutoCore OS version 6.1.60

CORTEXM BCM89107 release

Quality level: development drop. **Do not use for production.**

- ▶ Removed the obsolete parameter MkFastInterruptLocking and its container MkOptimization for those architectures that support EB tresos Safety OS
- ▶ Removed the release clearance file and its handling in the generator
- ▶ Added support for derivative BCM89107

## EB tresos AutoCore OS version 6.1.58

CORTEXM S32K34X release

Quality level: ready for manufacturing.

- ▶ Updated support for derivative S32K34X
- ▶ Updated schema to fix VSMD violations
- ▶ Changed the generator so that it no longer produces unnecessary log files





- ▶ Updated support for coverage analysis tooling in source code
- ▶ Added the new parameter `MkInitializeDataSections` for those architectures that support EB tresos Safety OS
- ▶ Removed extern declarations for `OS_NullFunction` in the generated file `MK_gen_user.h` for those architectures that support EB tresos Safety OS
- ▶ Added initial generator support for those architectures that support EB tresos Safety OS 2.1
- ▶ ASCOS-7413 Fixed known issue: Manually configured memory regions with enabled kernel access can only be accessed from the kernel on core 0
- ▶ Updated the table for the permitted calling context of API functions in the EB tresos AutoCore OS documentation
- ▶ Added support for microkernel v2.1 for those architectures that support EB tresos Safety OS
- ▶ Updated the generator support for shared timer interrupts
- ▶ ASCOS-7356 Fixed known issue: External tools can report errors or warnings based on the VSMD (`Os.-bmd`)
- ▶ Added the new interrupt ISR category `INTERNAL` for those architectures that support EB tresos Safety OS
- ▶ Improved generator log messages
- ▶ Improved description of the technical references in the EB tresos AutoCore OS documentation

## EB tresos AutoCore OS version 6.1.47

CORTEXM S32G27X release (including EB tresos Safety OS support)

Quality level: ready for manufacturing.

- ▶ Updated MSCM driver implementation to use the hardware's locking feature for configured interrupts. This change affects all multi-core derivatives that use the MSCM as an interrupt router.
- ▶ Updated support for derivative S32G27X
- ▶ Changed handling of unknown interrupts. If an interrupt is triggered that has no configured ISR, EB tresos AutoCore OS will shut down.
- ▶ Updated the exception handling code for CortexM exceptions
- ▶ Changed the vendor specific module definition (VSMD) file, the parameter `POST_BUILD_VARIANT_USED` is now specified by AUTOSAR

## EB tresos AutoCore OS version 6.1.44

CORTEXM RTL90XXA release



Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative RTL90XXA
- ▶ Added the open-source software section to the release notes to document the use and license information for open-source software delivered with EB tresos AutoCore OS
- ▶ Changed the generator and added velocity template engine support for code generation

## EB tresos AutoCore OS version 6.1.39

CORTEXM IMX8DXLM4 release

Quality level: ready for manufacturing.

- ▶ Updated support for derivative IMX8DXLM4
- ▶ Improved support for GHS compiler option "-align4"
- ▶ ASCOS-7264 Fixed known issue: Generation of MPU configuration can lead to unpredictable hardware behavior for Safety OS configurations

## EB tresos AutoCore OS version 6.1.34

CORTEXM IMX8QXPM4 release

Quality level: ready for manufacturing.

- ▶ Updated support for derivative IMX8QXPM4
- ▶ Removed generator error codes from the user documentation. These can be seen if there are errors during generation in EB tresos Studio.
- ▶ ASCOS-7236 Fixed known issue: Error information for BusFault and MemManage exceptions contains incorrect fault-address value
- ▶ Updated architecture notes for all derivatives

## EB tresos AutoCore OS version 6.1.30

CORTEXM IMX8DXLM4 release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative IMX8DXLM4
- ▶ Fixed known issue: IRQs are re-enabled during exception handling



- ▶ Added fast interrupt locking for trusted applications (EB\_FAST\_LOCK)
- ▶ Changed FPU handling to optimize performance
- ▶ ASCOS-7234 Fixed known issue: Error information for exceptions contains incorrect program location

## EB tresos AutoCore OS version 6.1.29

CORTEXM SAME5X release

Quality level: ready for manufacturing.

- ▶ Improved code quality and fixed names of board macros for derivative SAME5X

## EB tresos AutoCore OS version 6.1.28

CORTEXM IMX8QXPM4 release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative IMX8QXPM4
- ▶ Added general information about interrupt handling to user's guide and API descriptions
- ▶ Removed general information about interrupt handling from the architecture notes
- ▶ Updated architecture notes to include derivative SAME5X

## EB tresos AutoCore OS version 6.1.20

CORTEXM S32G27X release (including EB tresos Safety OS support)

Quality level: ready for development. **Do not use for production.**

- ▶ Updated architecture notes to include derivative S32G27X
- ▶ Updated support for derivative S32G27X

## EB tresos AutoCore OS version 6.1.19

CORTEXM SAME5X release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative SAME5X
- ▶ Removed redundant OIL translation information from documentation references



- ▶ Corrected the VSMD to ensure that AUTOSAR parameters `OsUseArti`, `OsTrustedApplication-WithProtection`, and `OsTrustedApplicationDelayTimingViolationCall` are only included in the model version 4.5.0. Note that these parameters are not supported by EB tresos AutoCore OS.
- ▶ Added atomics user's guide to EB tresos AutoCore OS documentation. This is no longer a separate document. Architecture specific restrictions for using atomic functions are now given in the architecture notes.
- ▶ Updated deviations list with regard to `OsApplicationCoreAssignment` and multi-core error handling
- ▶ Updated EB tresos AutoCore OS safety application guide to align with OS versions 6.1.x

## EB tresos AutoCore OS version 6.1.3

CORTEXM S32K34X release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative S32K34X

## EB tresos AutoCore OS version 6.1.2

CORTEXM CYT3DL release

Quality level: development drop. **Do not use for production.**

- ▶ Added support for derivative CYT3DL
- ▶ Disabled advanced logical core mapping support
- ▶ Updated `EnableInterruptSource()` and `DisableInterruptSource()` as described in AUTOSAR 19-11
- ▶ Removed support for the `OS_EXCLUDE_APPLICATION` optimization feature
- ▶ Removed support for Inter-OS application Communicator (IOC)
- ▶ Added `OsTaskPeriod` to `OsTask` container and `OsIsrPeriod` to `OsIsr` container
- ▶ Removed support for GPT drivers. It is no longer possible to generate a GPT notification function to increment a counter. You can implement this functionality in the application without the need for generator support.
- ▶ Updated schedule table handling to ensure that the schedule table stays in running state until the duration is reached
- ▶ Removed support for the CLZ scheduling algorithm
- ▶ Removed support for the `OsFastInterruptLocking` feature
- ▶ Removed support for CPU load measurement. The APIs `GetCpuLoad()`, `GetMaxCpuLoad()`, `GetCpuLoadOnCore(coreId)`, `GetMaxCpuLoadOnCore(coreId)`, `InitCpuLoad()`, and



`InitCpuLoadOnCore(coreId)` are no longer available. You can implement CPU load measurement in a low-priority task without dedicated support in the OS.