Elektrobit

EB tresos® AutoCore OS release notes TRICORE TC38XQ

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

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Release Notes

Release:
EB tresos AutoCore OS 6.1.31 [revision 000]

Date:
2021-08-23

Supported architecture:
TRICORE/TC38XQ

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 runtime. 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()`.

1.6. Change log

This chapter lists the changes between different versions for the TRICORE architecture. This list includes those changes that affect all architectures together with those affecting the TRICORE 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 6.1.31

TRICORE TC38XQ release

Quality level: ready for manufacturing.

- Updated support for derivative TC38XQ.
- Improved architecture notes for TC32XL and TC38XQ.
- Added general information about interrupt handling to user's guide and API decriptions.
- Removed redundant OIL translation information from documentation references.

EB tresos AutoCore OS version 6.1.13

TRICORE TC32XL release (including EB tresos Safety OS support)

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

- Corrected the VSMD to ensure that AUTOSAR parameters `OsUseArti`, `OsTrustedApplicationWithProtection`, and `OsTrustedApplicationDelayTimingViolationCall` are only included in the model version 4.5.0. Note that these parameters are not supported by EB tresos AutoCore OS.

EB tresos AutoCore OS version 6.1.8

TRICORE TC32XL release (including EB tresos Safety OS support)

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

- Added support for derivative TC32XL.
- 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.
- Adapted build to remove multi-core files not required for single-core derivative TC33XL.

- 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.