

TRAVEO™ T2G family AUTOSAR FEE release notes

SRN223768 Version 1.13

About this document

Scope and purpose

Thank you for your interest in the TRAVEO™ T2G family AUTOSAR FEE version 1.13. This document lists the installation requirements, software changes, limitations, and known issues.

Intended audience

This document is intended for anyone who uses the flash EEPROM emulation (FEE) software of the TRAVEO™ T2G family.

Table of contents

About this document.....	1
Table of contents.....	1
1 System requirements and recommendations	3
1.1 Supported compilers	3
1.2 Supported MCAL versions (In case of install FEE with MCAL)	3
1.3 Compiler options.....	3
1.4 Library compiler options.....	4
1.5 Memory consumption(GHS)	4
1.6 Memory consumption(IAR)	5
1.7 Explanatory notes for this section	5
1.8 Stack consumption(GHS):.....	5
1.9 Stack consumption(IAR):	6
1.10 Release details.....	7
2 Installation.....	8
3 Deviations from AUTOSAR.....	9
4 Limitations.....	10
5 Documentation	11
6 Technical support	12
7 Version history.....	13
7.1 Module SW-Version 1.1.....	13
7.2 Module SW-Version 1.2.....	13
7.3 Module SW-Version 1.3.....	13
7.4 Module SW-Version 1.4.....	13
7.5 Module SW-Version 1.5.....	13
7.6 Module SW-Version 1.6.....	14
7.7 Module SW-Version 1.7-1.10	14
7.8 Module SW-Version 1.11.....	14
7.9 Module SW-Version 1.12.....	15

Table of contents

7.10	Module SW-Version 1.13.....	15
------	-----------------------------	----

System requirements and recommendations

1 System requirements and recommendations

Software prerequisites	Supported version
EB tresos Studio package for Infineon	26.2.0

1.1 Supported compilers

Green Hills Software, compiler v2017.1.4

IAR Embedded Workbench 8.0 EWARM FS 8.22.3

1.2 Supported MCAL versions (In case of install FEE with MCAL)

Only same release

1.3 Compiler options

This section summarizes the compiler options used to build and test the module. When changing the compiler options, the module must be considered untested.

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	<code>-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle</code>

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	<code>-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard</code>

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0 EWARM FS 8.22.3	<code>--cpu=Cortex-M4 -e --debug -Ohs --endian=little --fpu=VFPv4_sp --no_size_constraints</code>

Compiler	Option (Cortex®-M7 core)
IAR Embedded Workbench 8.0 EWARM FS 8.22.3	<code>--cpu=Cortex-M7 -e --debug -Ohs --endian=little --fpu=VFPv5_d16--no_size_constraints</code>

System requirements and recommendations

1.4 Library compiler options

If a binary library is delivered with this module, it is built using the following options.

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0 EWARM FS 8.22.3	--cpu=Cortex-M4 -e --debug -Ohs --endian=little --fpu=VFPv4_sp --no_size_constraints

Compiler	Option (Cortex®-M7 Core)
IAR Embedded Workbench 8.0 EWARM FS 8.22.3	--cpu=Cortex-M7 -e --debug -Ohs --endian=little --fpu=VFPv5_d16 --no_size_constraints

1.5 Memory consumption(GHS)

GHS (Fee_lib.lib) section	Size (in bytes)
.text	39082
.bss	404
Combined:	39486

GHS (Fee_src.lib) section	Size (in bytes)
.text	2964
.bss	9324
.rodata	192
Combined:	12480

Note: The memory consumption of Fee_src.lib depends on the configuration.

Note: The listed memory consumption will vary depending on customer configuration.

System requirements and recommendations

1.6 Memory consumption(IAR)

IAR (Fee_lib.lib) section	Size (in bytes)
.text	38368
.bss	349
Combined:	38717

IAR (Fee_src.lib) section	Size (in bytes)
.text	3004
.bss	9404
.rodata	16
Combined:	12424

Note: The memory consumption of Fee_src.lib depends on the configuration.

Note: The listed memory consumption will vary depending on customer configuration.

1.7 Explanatory notes for this section

Section	Description
.text	Program code
.bss	Variables that are not explicitly initialized
.rodata	Read-only data
.data	Variables with explicitly initialized values

1.8 Stack consumption(GHS):

Function	Max stack usage (in bytes)
Fee_Init	52
Fee_SetMode	56
Fee_Read	84
Fee_Write	68
Fee_Cancel	24
Fee_GetStatus	12
Fee_GetJobResult	52
Fee_InvalidateBlock	64
Fee_GetVersionInfo	48
Fee_EraseImmediateBlock	64
Fee_MainFunction	112
Fee_Clear / Fee_ClearEx	56
Fee_GetRemainingPages / Fee_GetRemainingPagesEx	52

System requirements and recommendations

Function	Max stack usage (in bytes)
Fee_CleanupAndErase / Fee_CleanupAndEraseEx	56
Fee_SetCycleMode	64

Note: Stack consumption is evaluated using the *gstack* utility program, which is part of the Green Hills release package. To enable the measurement of stack consumption in your project, build the source code according to the instructions given in the "Measuring stack consumption" section of the FEE user guide.

Note: The listed stack consumption will vary depending on customer configuration

Note: The GHS stack consumption listed in the release notes was measured using the additional compile option `"-gs"`. The GHS compiler cannot measure stack consumption for the selected optimization level (see Compilation options). Green Hills cannot exclude possible effects of `"-gs"` on optimization and stack consumption. Therefore, Infineon cannot guarantee the accuracy of these values. For more information on measuring GHS stack consumption, see the section *gstack* utility program in *build_arm.pdf*.

1.9 Stack consumption(IAR):

Function	Max stack usage (in bytes)
Fee_Init	216
Fee_SetMode	40
Fee_Read	60
Fee_Write	52
Fee_Cancel	208
Fee_GetStatus	0
Fee_GetJobResult	24
Fee_InvalidateBlock	44
Fee_GetVersionInfo	16
Fee_EraseImmediateBlock	44
Fee_MainFunction	608
Fee_Clear / Fee_ClearEx	24
Fee_GetRemainingPages / Fee_GetRemainingPagesEx	24
Fee_CleanupAndErase / Fee_CleanupAndEraseEx	36
Fee_SetCycleMode	24

Note: To enable the measurement of stack consumption in your project, build the source code with the linker option `"--enable_stack_usage --log call_graph"`. See stack usage analysis of the IAR C/C++ development guide for details.

Note: The listed stack consumption will vary depending on customer configuration.

System requirements and recommendations**Note on “*_Bswmd.arxml”**

Note that the <Module>_Bswmd.arxml files are templates that can be freely modified by the customer or RTE vendor.

These are in the *output\generated\swcd* subfolder of your *project* folder.

Named files are not tested.

1.10 Release details

Module software version

1.13.x

(x=software patch version; see the delivery notes for details)

AUTOSAR specification version (ASR)

4.2.2

Target

MXS40

MCAL configuration setting	Supported derivatives
See the resource release notes	See the resource module release notes

Corresponding Fee_MemMap.h stub file version

1.0.1

Installation

2 Installation

See the installation manual for EB tresos Studio for Infineon AUTOSAR software products and installation manual for FEE42-TRAVEO.

3 Deviations from AUTOSAR

The following configuration parameters have properties that differ from AUTOSAR specifications.

- The value of `FeeNumberOfWriteCycles` cannot be changed. This value is automatically determined by `FeeCallCycle`.
- `FeePollingMode` is fixed value(true), so its value cannot be changed.
- `FeeNvmJobEndNotification` is fixed name, and cannot be renamed.
- `FeeNvmJobErrorNotification` is fixed name, and cannot be renamed.
- `FeeVirtualPageSize` is fixed value (4).
- The maximum value of `FeeBlockSize` is configured by `FeeBlockMaxsize`, which is a vendor-specific parameter.
- `FeeDeviceIndex` is fixed value (0).
- `FeeBlockOverhead` is fixed value (20).
- `FeePageOverhead` is fixed value (16).

Limitations

4 Limitations

- Configuration parameter: `FeeBlockNumber`

When `ConfigEx('FeeBlockConfigurationEx')` is used, the same value of `FeeBlockNumber` cannot be used between `Config('FeeBlockConfiguration')` and `ConfigEx('FeeBlockConfigurationEx')`.

5 Documentation

User guide for FEE is in the `\doc` subdirectory of the *installation* directory. The default location is:

`C:\INFINEON_ESDB\Tresos26_2_0\doc`

6 Technical support

If you have questions related to the TRAVEO™ T2G family AUTOSAR FEE, contact the local support application engineer.

Version history

7 Version history

7.1 Module SW-Version 1.1

Initial module setup.

7.2 Module SW-Version 1.2

- Supports TRAVEO™ T2G-B-E-2M.
- Supports TRAVEO™ T2G-B-E-8M.
- Add Block check function.
- Target name is changed from CYT2 to MXS40.
- Remove MCAL modules from EB tresos help contents.

7.3 Module SW-Version 1.3

- Improved performance.
- Added automatic calculation of sector size.
- Updated Sections "Supported compilers", "Compiler options", "Library compiler options", "Memory consumption", and "Stack consumption" to add IAR compiler information.

7.4 Module SW-Version 1.4

- Supports TRAVEO™ T2G-C-2D-6M support.
- Supports EB tresos V26.2.0

[Impact]

Strict AUTOSAR specification and check for parameter configuration errors are implemented in EB tresos V26.2.0.

In addition, handling of reference paths (relative paths) such as system description file (ARXML) is changed in EB tresos V26.2.0.

Therefore, if the current ECUC configuration definitions XML file contains deviations or errors, you may find errors during import to tresos26. In that case, the ECUC configuration definitions XML file must be modified appropriately.

In addition, if the current ARXML file contains unresolvable paths, you may find errors during import to tresos26. In that case, ARXML file must be modified.

The SW has been tested; no risks except for the low-level risk listed above were found.

7.5 Module SW-Version 1.5

- Supports MISRA C:2012 coding rule

The MISRA C:2012 coding rule checks the source code.

If a deviation from the rules is required, add the deviation comment to the code and report the result.

If a deviation is for MISRA-C:2004 only, remove the deviation comment.

Version history

7.6 Module SW-Version 1.6

- Misleading comment in Module_MemMap.h

{Mip}_MemMap.h files are provided as sample template files. But, the file header comment cannot be modified, which is a contradiction. To resolve this contradiction, change the file header comment to allow user modification.

In addition, to make sure that the file is not a part of the commercial product, move the *{Mip}_MemMap.h* files to the *MemMap* stub folder.

- C/NC ECC error on 32 bits read from CM7

Read access of 32 bits on a work flash from CM7 causes C/NC ECC error on the next 32-bit address. The work flash is always accessed 64-bit-wide when it is read by AXI. Therefore, FLS/FEE may not work properly on CM7; FLS/FEE must use DMA transfer to read the data from work flash memory. In addition, this release of the software omits ECC error output from FLS in previous versions. It now acquires the ECC error in FEE and outputs an appropriate message.

7.7 Module SW-Version 1.7-1.10

- Block data is lost in case of repetitive reset during data writing

The following problems have been solved.

If all the following conditions are met, the block ID being written will be lost.

[Pre-condition]

- 1) Reset happens twice consecutively while writing data.
- 2) First reset happens during block data (block ID is written)
- 3) Second reset happens during "start writing flag" (block ID is not written yet).

Under these pre-conditions, the block is recognized as "invalid" and will be lost on rare occasions during the next recycle timing.

- Some parameters are inconsistent between XDM and ARXM

The following problems have been solved:

The following inconsistencies are found between Fee.arxml and Fee.xdm:

- 1) SHORT-NAME is duplicated in Fee.arxml.
- 2) TS_T40D13M1I0R0 has no UUID in Fee.xdm
- 3) TS_T40D13M1I0R0 (UUID) is not described in Fee.xdm.

7.8 Module SW-Version 1.11

Resolved the following issue:

In a very rare case of all the following conditions being met, the sector data may be lost if a reset occurs when the physical sector configuration (PSC) flag is being written.

1. The FEE module creates a valid sector, and it writes the "sector valid" flag followed by the PSC flag. A power failure occurs when the PSC flag is being written.

Version history

2. At the next Fee_Init, the “sector valid” flag and the PSC flag are checked and considered as “not blank” by the “blank check” operation. Normal values of the PSC flag are also read out. As a result, this sector is considered as a valid sector.
3. After these two occur, data writes to the sector and power failure occur multiple times.

During case 3, the value of the PSC flag cannot be read normally due to changes in checksum bits. As a result, the FEE module reports that no valid sector is found, which results in losing the sector data.

7.9 Module SW-Version 1.12

- Enables/disables the delay recycle operation in FEE initializing timing

The current FEE can be read after the initialization process. If a part of the data is corrupted, a recycle will occur to fix the block data in the initialization process. When this recycle occurs, sector erasing and data moving are performed; normally, these processes take longer to complete.

New configuration parameter

A new FEE configuration parameter is added to select whether to perform recycle during FEE initialization.

`FeeDelayRecycleOperation` (Boolean parameter, in `FeeGeneral` container)

- true: Recycle is not performed during initialization. Therefore, data reading will start shortly. However, when this option is selected, recycle occurs in the write process (`Fee_MainFunction` after `Fee_Write`). the first write process may take longer.
 - false: Recycle is performed during initialization. (default).
- Undefined SROM error handling

The FLS driver that works with FEE has been modified to report an undefined status from an unexpected SROM Blank Check API. FEE will output a hard error if FLS reports its status.

The following are supported in release V1.15.0.

- Update copyright notice and disclaimer statement

Copyright notice and disclaimer statement in the file header comment are updated to follow the up-to-date specifications.

7.10 Module SW-Version 1.13

- Added the threshold page function

A new FEE configuration parameter is added to determine whether to recycle after writing data when recycle is needed at immediate data writing.

New configuration parameter

`ConfigIfUseThresholdPageSize` (Boolean parameter, in `FeeGeneral` container)

This field is used for immediate data in `Config` area to determine whether to recycle after writing data when recycle.

- true(green): Do recycle after writing immediate data in `Config` area.
- false(red): Do recycle before writing immediate data in `Config` area (default).

Version history

`ConfigExIfUseThresholdPageSize` (Boolean parameter, in `ConfigEx` container)

This field is used for immediate data in `ConifgEx` area to determine whether to recycle after writing data when recycle.

- `true(green)`: Do recycle after writing immediate data in `ConfigEx` area.
 - `false(red)`: Do recycle before writing immediate data in `ConfigEx` area (default).
- Follow to the specification changes for `FlsWriteVerification` of FLS driver

FEE uses the `FlsWriteVerification` option of FLS to perform write verification after writing and the user guide of FLS was updated.

The `Fls_Write()` source address is needed to be specified for RAM area address according to the latest FLS user guide. However, FEE currently passes the Flash area address as the source address. Therefore, FEE changes the RAM area address setting as per the FLS user guide.

There is no change in the behavior of each API provided by FEE.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-10-27

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-23768 Rev. *N

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.