S32K3_S32M27x Real-Time Drivers AUTOSAR R21-11

Version 6.0.0 QLP01

30 June 2025 **Release notes**

1 Introduction

1.1 Revision History

Table 1. Revision History

Revision number	Date	Substantive changes
1.0	06/30/2025	Release notes of S32K3xx_S32M27x Real-Time Drivers release 6.0.0 QLP01.

1.2 Package content

This release contains the NXP S32K3_S32M27x Real-Time Drivers Version 6.0.0 QLP01:

- "eclipse/plugins/<mod>_TS_T40D34M60I0R0" directories - Tresos Plugins, 1 per module.
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506.exe"
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506_ReleaseNotes.pdf" - This file.
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506_SBOM.spdx.json"
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506_DesignStudio_updatesite.zip"
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506_SafetyPackage.zip" - contains FMEA reports and Safety Manual.
- "SW32K3_S32M27x_RTD_R21-11_6.0.0_QLP01_D2506_QualityPackage.zip"
- Various other files: GettingStarted.htm start page and associated images, the license.txt EULA file and the Uninstall.exe utility for removing the RTD installation.

1.3 Installation Design Studio

1.3.1 Bundled in S32 Design Studio

S32 RTD is delivered bundled in the S32 Design Studio. In this case, it's already configured and ready to use. New S32DS project wizard and New S32DS project from Example can be used to create S32 RTD based projects.

1.3.2 Delivered as an extension for S32 Design Studio

S32 RTD is delivered as an Update Site for S32 Design Studio "*S32 Design Studio 3.6.2*". In this case, it must be installed by opening Help -> S32 Design Studio Extensions and Updates -> Add Update Sites and selecting the archive file containing the S32 RTD software and then check the S32 RTD software package to be installed and continue the installation process. After it is installed, New S32DS project wizard and New S32DS project from Example can be used to create S32 RTD based projects.

1.4 Installation EB Tresos

Follow the installer steps. By default the installer will create a link between the installation target directory and a selected EB Tresos installation. If you choose not to create a link, you can later create one manually or you can copy all "<mod>_TS_T40D34M60I0R0" directories and .JAR files to the "<Tresos Install Path>\plugins" directory.

2 Release Specifics

The S32K3_S32M27x Real-Time Drivers Version 6.0.0 QLP01 is AUTOSAR R21-11 compliant. This release contains Crypto Driver. The AUTOSAR Configuration ARXML specification takes precedence over AUTOSAR SWS PDF Specifications if there are discrepancies.

The S32K3_S32M27x Real-Time Drivers Version 6.0.0 QLP01 can be used also in non-AUTOSAR environment, as a collection of peripheral drivers designed to simplify and accelerate application development on NXP microcontrollers.

2.1 Release Details

This is the NXP S32K3_S32M27x Real-Time Drivers Release Version 6.0.0 QLP01 contains Crypto Driver, qualified as RFP release in terms of quality, supporting AUTOSAR R21-11 and non AUTOSAR, with functionalities and tests on S32K358(Rev 1.1), S32K396(cut 1.1), S32K311, S32K342, S32K312, S32K344, S32K388, and S32M276 (cut 2.0).

This release contains a deviation from AUTOSAR recommended version check inside source files for more details.

For S32K389, this release is of Code Drop quality in terms of testing and quality documentation. Code Drop Release is not intended for production, it must be replaced by the next official standard release (EAR, PRC or RFP).

This release was tested using:

- Silicon P32K358GHMJBS 1P14E CTAA2336A (Rev1.1).
- Silicon P32K396EHMJBS 1P40E QAA2331A (E5) (cut 1.1).
- Silicon P32K311HVS 0P98C CTZA2242B (E5).
- Silicon P32K344EHVMMS 1P55A CTSB2128B
- Silicon P32K312NHVPBS 0P09C CTZM2132B.
- Silicon P32K342EHVPBE 0P97C CTZM2139A.
- Silicon P32K388HMS 0P39J CTAA2332D (E5).
- Silicon P32M276CC MKHS 0P69K 20230904 (E5-B0) (Use for CANTRCV).
- Silicon P32M276LC MKHS 0P69K 20230904 (E5-B0) (Use for LINTRCV).
- Mini Module XS32K3X8CVB-Q289 PCB 53108 RevX3 SCH RevA1.
- Mini Module XS32K396-BGA-DC PCB 54614 RevX1 SCH RevA.
- Mini Module XS32K3X2CVB-Q100 PCB 48306 RevX1 SCH RevA.
- Mini Module XS32K3XXCVB-Q257 PCB 47020 RevA1 SCH RevX2.
- Mini Module XS32K3X2CVB-Q172 PCB 48307 RevX1 SCH RevA.
- EVB S32K3X8EVB-Q289 PCB 54870 RevX2 SCH RevB2
- EVB S32K388EVB-Q289 PCB 88925 RevX1 SCH RevA
- CVB XS32M2XXCVB-Q064 PCB 53099 RevX1 SCH RevA.
- CVB XS32K388CVB-Q289 PCB RevA SCH RevX3.
- HSE firmware release: S32K344: hse_fw_s32k344 0.2.55.0 (RFP).
- HSE firmware release: S32K312: hse_fw_s32k312 0.2.55.0 (RFP).

- HSE firmware release: S32K342: hse_fw_s32k342 0.2.55.0 (RFP).
- HSE firmware release: S32K358: hse_fw_s32k358_0.2.55.0 (RFP).
- HSE firmware release: S32K311: hse_fw_s32k311_0.2.55.0 (RFP).
- HSE firmware release: S32K396: hse_fw_s32k396_0.2.50.0 (RFP).
- HSE firmware release: S32M276: hse_fw_s32m276_0.2.55.0 (RFP).
- HSE firmware release: S32K388: hse_fw_s32k388_0.2.49.0 (RFP).
- HSE firmware release: S32K389: hse_fw_s32k389_0.2.70.0 (CD).

In all source files, Software Version values are checked (major, minor, patch). AUTOSAR release or SWS versions are not checked during preprocessing/template generation.

The correct SWS versions are exported by each module.

The functions contained in the CSM, CryIf and NvM plugins are sample stub functions.

These functions should be replaced by the user developed code during integration.

The Resource module is needed to select the MCU derivative.

The derivatives supported can be found in the Resource module definition file, parameter 'ResourceSubderivative'.

The following limitations are present in this release:

Known limitation:

- Due to low SRAM memory space on S32K310, S32K311, S32K312, S32K322, S32K341, S32K342, S32M274 and S32M276 derivatives, users should select FLASH profile to create new project on S32 Design Studio.

Known issue with IAR compiler, all RTD drivers:

- Warning regarding stack usage is thrown for reference implementations of core exceptions in startup files when drivers are compiled with IAR. These functions are provided as reference code and can be replace/ modified by the application.
- Usage of IAR compiler option "-enable_stack_usage " will issue warnings regarding uncalled functions (eg : interrupt handlers). This should be disregarded.
- IAR cannot analyze stack usage for function in .s file and the function with indirect call(function pointer).

Known issue with GCC compiler, all RTD drivers:

- Warning regarding enum size is thrown by the linker due to usage of "-fno-short-enums" option: "use of enum values across objects may fail". The drivers do not use any library enum types - no functional impact.

Known issue for PE Micro debug plugins in Design Studio:

- Debugging an application in RAM might fail if another application is present in target flash (debug communication lost during startup, MCU boots the application in flash). Workaround proposed: erase flash before loading an application in RAM; this can be done with the following command launched in the PE for example GDB Server console: "pegdbserver_console.exe -device=NXP_S32K3xx_S32K344 -programmingtype=3 -runafterprogramming=0 -interface=USBMULTILINK -startserver -singlesession -quitafterprogramming -flashobjectfile=path_to_a_srec_or_elf_file".

The following limitations are present in this release, beside the ones presented in Drivers Manuals:

- During this SW product development and validation, the achieved cumulative statement coverage is 95.98% for HSE_IP and 94.74% for ACE_IP. Consequently the Decision coverage and MC/DC coverage are less than 100%. It is also observed that some of the functions did not achieve the target statement coverage of 80%. Nevertheless, the whole code went through a diligently inspection-based review to ensure there are no functional issues. For details on the coverages, refer to the module's Code Coverage Summary reports.

For the following functions the achieved code coverage is less than 80%:

- Crypto_43_ACE_Init
- Crypto_43_ACE_KeyElementSet
- Crypto_43_HSE_Ipw_ImportKey
- Hse_Ip_GeneralPurposeIrqHandler

RTD Header Files for main derivatives reference

Table 2. RTD Header Files for main derivatives reference

Derivatives	RTD Header files
S32K344	S32K3 Reference Manual Rev. 3, 10/2021
S32K324	S32K3 Reference Manual Rev. 3, 10/2021
S32K314	S32K3 Reference Manual Rev. 3, 10/2021
S32K312	S32K3 Reference Manual Rev. 4, 04/2022
S32K311	S32K3 Reference Manual Rev. 7, 05/2023
S32K341	S32K3xx Reference Manual Rev.5, 09/2022
S32K342	S32K3xx Reference Manual Rev.5, 09/2022
S32K322	S32K3xx Reference Manual Rev.5, 09/2022
S32K358	S32K3xx Reference Manual Rev. 8_Update, 11/2024
S32K388	S32K3 Reference Manual Rev. 9, 08/2024
S32K389	S32K3xx Reference Manual Rev. 10, 04/2025
S32K39, S32K37, S32K36	S32K39,S32K37 and S32K36 Reference Manual Rev.4, 11/2024
S32M276	S32M27x Reference Manual, Rev.4, 09/2024

Note: The header file of sub-derivatives use header file of main derivatives

2.2 Used Documentation

This release was developed and tested based on the following documents:

Table 3. Reference Manuals

Document Title	Version and Date
S32K3xx Reference Manual	Rev.9, 07/2024
S32K3xx Data Sheet	Rev. 10, 07/2024
S32M27x Reference Manual	Rev.5, 02/2025
S32M2xx Data Sheet	Rev. 8 — 02/2025
S32K39, S32K37 and S32K36 Reference Manual	Rev. 4, 11/2024
S32K39 and S32K37 Data Sheet	Rev.4 — 06/2024

Table 4. Implemented Errata

Document Title	Maskset	Date
S32K358 Mask Set Errata for Mask	1P14E	Rev. 4, 6/2024

Table 4. Implemented Errata*...continued*

Document Title	Maskset	Date
S32K311 Mask Set Errata	0P98C	Rev. 11/ January/2024, 1/2024
S32K396 Mask Set Errata for Mask	0P40E	Rev. FEB2024, 2/2024
S32K396 Mask Set Errata for Mask	1P40E	Rev. MAR2024, 3/2024
S32K312 Mask Set Errata for Mask	0P09C	Rev. 14/Dec/2023, 12/2023
S32K342 Mask Set Errata for Mask	0P97C	Rev. 14, 12/2023
S32K344 Mask Set Errata for Mask	0P55A/1P55A	Rev. 12/Dec/2023, 12/2023
S32K388 Mask Set Errata for Mask	0P39J/0P24N	Rev. 1, 7/2024
S32M276 Mask Set Errata for Mask	P98C+P69K	Rev. 1, 8/2024
S32K389 Mask Set Errata for Mask	mask_id	Rev. 1.1 — 18 September 2024

2.3 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices from NXP:

- s32k310_hdqfp100
- s32k310_lqfp48
- s32k311_hdqfp100 / MWCT2015S_hdqfp100
- s32k311_lqfp48
- s32k312_hdqfp100 / MWCT2016S_hdqfp100
- s32k312_lqfp100
- s32k312_hdqfp172 / MWCT2016S_hdqfp172
- s32k314_hdqfp172
- s32k314_mapbga257
- s32k322_hdqfp100 / MWCT2D16S_hdqfp100
- s32k322_hdqfp172 / MWCT2D16S_hdqfp172
- s32k324_hdqfp172 / MWCT2D17S_hdqfp172
- s32k324_mapbga257
- s32k341_hdqfp100
- s32k341_hdqfp172
- s32k342_hdqfp100
- s32k342_hdqfp172
- s32k344_hdqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k394_lqfp176
- s32k396_mapbga289
- s32k396_lqfp176

- s32k374_mapbga289
- s32k374_lqfp176
- s32k376_mapbga289
- s32k376_lqfp176
- s32k364_mapbga289
- s32k364_lqfp176
- s32k366_mapbga289
- s32k366_lqfp176
- s32k358_hdqfp172
- s32k358_mapbga289
- s32k356_hdqfp172
- s32k356_mapbga289
- s32k328_hdqfp172
- s32k328_mapbga289
- s32k338_hdqfp172
- s32k338_mapbga289
- s32k348_hdqfp172
- s32k348_mapbga289
- s32m274_lqfp64
- s32m276_lqfp64
- s32k388_mapbga289
- s32k389_mapbga437

The mapping between MWCT2xxxS parts and S32K3XX is showed in the table below:

Table 5. Derivatives mapping

MWCT2xxxS derivative	S32K3 derivative
MWCT2D17S_MQFP172	S32K324_MQFP172
MWCT2D16S_MQFP100	S32K322_MQFP100
MWCT2D16S_MQFP172	S32K322_MQFP172
MWCT2016S_MQFP172	S32K312_MQFP172
MWCT2016S_MQFP100	S32K312_MQFP100
MWCT2015S_MQFP100	S32K311_MQFP100

Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power and they tested with:

- P32K344EHVMMS 1P55A CTSB2128B (and in S32K324 configuration).
- P32K312NHVPBS 0P09C CTZM2132B.
- P32K311HVS 0P98C CTZA2242B.

2.4 Modules Configuration

2.4.1 EB Tresos

Modules configurations were developed and tested using the Tresos Configuration Tool version "*EB Tresos Studio 29.0.0 b220329-0119*"

Configuration definition files were developed according to AUTOSAR R21-11, AUTOSAR_EcucParamDef.arxml

A folder named "<mod>_TS_TtDdMmIiRr" exists for each delivered module (<mod>). It is called a Tresos plugin for the module. A plugin contains the AUTOSAR module definition file (epd), the Tresos Xpath Data Model module definition file (xdm), the module user and integration manuals, the module configuration generation template source files, and the module driver static source files. Additional necessary Tresos specific tooling files are also included.

Plugin Encoding: <mod>_TS_TtDdMmIiRr

Important change related to the plugin notation:
- "m" = coding major and minor version number, can contain 1 or more digits
- "i" = patch number.
The major version number will be left out, if it is "0", in this case "m" contains 1 digit only, otherwise it contains 2 digits

For this release:

- t=40, CortexM Architecture
- d=34, S32K3XX (derivative)
- m=60, Release major and minor version
- i=0, Release patch version
- r=0, Reserved

2.4.2 S32 Design Studio

Configuration components were developped using "*S32 Design Studio 3.6.2*".

The components are split in three tools inside S32 Design Studio: Pins Tool, Clocks Tool, Peripherals tool which enable the generation of configuration structures to be used by both Autosar and low-level drivers.

2.5 Support and Driver Plugins Delivered

Table 6. Support and Driver Plugins Delivered

Plugin	Low level interface	SW Version	Description
Crypto	HSE_Ip, ACE_Ip	sw version 6.0.0 QLP01	Driver, Crypto Driver
CSM	N/A	sw version 6.0.0 QLP01	Support stub, Crypto Service Manager
CryIf	N/A	sw version 6.0.0 QLP01	Support stub, Crypto Interface
NVM	N/A	sw version 6.0.0 QLP01	Support stub, Non-Vloatile Memory

2.6 Module Plugin Folder Structure

Table 7. Module Plugin Folder Structure

Folder or file	Description
<mod>_TS_TtDdMmIiRr\anchors.xml	Tresos Configuration tooling documentation data file
<mod>_TS_TtDdMmIiRr\plugin.xml	Tresos Configuration tooling data file
<mod>_TS_TtDdMmIiRr\ant_generator.xml	Tresos Configuration tooling data file
<mod>_TS_TtDdMmIiRr\autosar\<mod>.epd	Module Parameter Definition in AUTOSAR format
<mod>_TS_TtDdMmIiRr\config\<mod>.xdm	Module Parameter Definition in Tresos XDM format
<mod>_TS_TtDdMmIiRr\config_ext\<mod>PreConfiguration.xdm	Module Parameter Default Configuration in Tresos XDM format[1]

Table 7. Module Plugin Folder Structure*...continued*

Folder or file	Description
<mod>_TS_TtDdMmIiRr\doc\AUTOSAR_RTD_<mod>_IM.pdf	Module Integration Manual
<mod>_TS_TtDdMmIiRr\doc\AUTOSAR_RTD_<mod>_UM.pdf	Module User's Manual
<mod>_TS_TtDdMmIiRr\generate_PB	Post-build source files (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_PB\src	Post-build source file templates (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_PB\include	Post-build source file header templates (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_LT	Link-time source files (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_LT\src	Link-time source file templates (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_LT\include	Link-time source file templates (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_PC\	Pre-compile source files
<mod>_TS_TtDdMmIiRr\generate_PC\src	Pre-compile source files templates
<mod>_TS_TtDdMmIiRr\generate_PC\include	Pre-compile source files templates
<mod>_TS_TtDdMmIiRr\generate_swcd	Module BSWMD file
<mod>_TS_TtDdMmIiRr\include\	Module driver header files
<mod>_TS_TtDdMmIiRr\META-INF	Tresos Configuration tooling data and signature files
<mod>_TS_TtDdMmIiRr\src\	Module driver source files[2]

Notes:

[1] Not available for all plugins.

[2] The Support Stub Resource contains the "resource" folder instead of the "src" folder.

2.7 Compiler versions and options

This release was developed and tested with:

- NXP GCC 10.2.0 20200723
- Green Hills Multi 7.1.6d / Compiler 2021.1.4
- IAR ANSI C/C++ Compiler v.8.50.10 (safety version)

2.7.1 GCC Compiler/Linker/Assembler Options

Table 8. Compiler Options

Option	Description
-mcpu=cortex-m7	Targeted ARM processor for which GCC should tune the performance of the code
-mthumb	Generates code that executes in Thumb state
-mlittle-endian	Generate code for a processor running in little-endian mode
-mfpu=fpv5-sp-d16	Specifies the floating-point hardware available on the target
-mfloat-abi=hard	Specifies the floating-point ABI to use. "hard" allows generation of floating-point instructions and uses FPU-specific calling conventions
-std=c99	Specifies the ISO C99 base standard

Table 8. Compiler Options*...continued*

Option	Description
-Os	Optimize for size. Enables all -O2 optimizations except those that often increase code size
-ggdb3	Produce debugging information for use by GDB using the most expressive format available, including GDB extensions if at all possible. Level 3 includes extra information, such as all the macro definitions present in the program
-Wall	Enables all the warnings about constructions that some users consider questionable, and that are easy to avoid (or modify to prevent the warning), even in conjunction with macros
-Wextra	This enables some extra warning flags that are not enabled by -Wall
-pedantic	Issue all the warnings demanded by strict ISO C. Reject all programs that use forbidden extensions. Follows the version of the ISO C standard specified by the aforementioend -std option.
-Wstrict-prototypes	Warn if a function is declared or defined without specifying the argument types
-Wundef	Warn if an undefined identifier is evaluated in an #if directive. Such identifiers are replaced with zero
-Wunused	Warn whenever a function, variable, label, value, macro is unused
-Werror=implicit-function-declaration	Make the specified warning into an error. This option throws an error when a function is used before being declared
-Wsign-compare	Warn when a comparison between signed and unsigned values could produce an incorrect result when the signed value is converted to unsigned.
-Wdouble-promotion	Give a warning when a value of type float is implicitly promoted to double
-fno-short-enums	Specifies that the size of an enumeration type is at least 32 bits regardless of the size of the enumerator values.
-funsigned-char	Let the type char be unsigned by default, when the declaration does not use either signed or unsigned
-funsigned-bitfields	Let a bit-field be unsigned by default, when the declaration does not use either signed or unsigned
-fomit-frame-pointer	Omit the frame pointer in functions that don’t need one. This avoids the instructions to save, set up and restore the frame pointer; on many targets it also makes an extra register available.
-fno-common	Makes the compiler place uninitialized global variables in the BSS section of the object file. This inhibits the merging of tentative definitions by the linker so you get a multiple-definition error if the same variable is accidentally defined in more than one compilation unit
-fstack-usage	Makes the compiler output stack usage information for the program, on a per-function basis
-fdump-ipa-all	Enables all inter-procedural analysis dumps
-c	Stop after assembly and produce an object file for each source file
-DUSE_SW_VECTOR_MODE	Predefine USE_SW_VECTOR_MODE as a macro, with definition 1. By default, the drivers are compiled to handle interrupts in Software Vector Mode.
-DD_CACHE_ENABLE	Predefine D_CACHE_ENABLE as a macro, with definition 1. Enables data cache initalization in source file system.c under the Platform driver
-DI_CACHE_ENABLE	Predefine I_CACHE_ENABLE as a macro, with definition 1. Enables instruction cache initalization in source file system.c under the Platform driver
-DMPU_ENABLE	Predefine MPU_ENABLE as a macro, with definition 1. Enable MPU default configuration from startup code under the Platform driver

Table 8. Compiler Options*...continued*

Option	Description
-DENABLE_FPU	Predefine ENABLE_FPU as a macro, with definition 1. Enables FPU initalization in source file system.c under the Platform driver
-DMCAL_ENABLE_USER_MODE_SUPPORT	Predefine MCAL_ENABLE_USER_MODE_SUPPORT as a macro, with definition 1. Allows drivers to be configured in user mode.
--sysroot= $ (NEWLIB_DIR)	Specifies the path to the sysroot, for Cortex-M7 it is $ (TOOLCHAIN_DIR)/arm-none-eabi/newlib
-specs=nano.specs	Use Newlib nano specs
-specs=nosys.specs	Do not use printf/scanf

Table 9. Assembler Options

Option	Description
-Xassembler-with-cpp	Specifies the language for the following input files (rather than letting the compiler choose a default based on the file name suffix)
-mcpu=cortex-m7	Targeted ARM processor for which GCC should tune the performance of the code
-mfpu=fpv5-sp-d16	Specifies the floating-point hardware available on the target
-mfloat-abi=hard	Specifies the floating-point ABI to use. "hard" allows generation of floating-point instructions and uses FPU-specific calling conventions
-mthumb	Generates code that executes in Thumb state
-c	Stop after assembly and produce an object file for each source file

Table 10. Linker Options

Option	Description
-Wl,-Map,filename	Produces a map filer
-T linkerfile	Use linkerfile as the linker script. This script replaces the default linker script (rather than adding to it)
--entry=Reset_Handler	Specifies that the program entry point is Reset_Handler
-nostartfiles	Do not use the standard system startup files when linking
-mcpu=cortex-m7	Targeted ARM processor for which GCC should tune the performance of the code
-mthumb	Generates code that executes in Thumb state
-mfpu=fpv5-sp-d16	Specifies the floating-point hardware available on the target
-mfloat-abi=hard	Specifies the floating-point ABI to use. "hard" allows generation of floating-point instructions and uses FPU-specific calling conventions
-mlittle-endian	Generate code for a processor running in little-endian mode
-ggdb3	Produce debugging information for use by GDB using the most expressive format available, including GDB extensions if at all possible. Level 3 includes extra information, such as all the macro definitions present in the program
-lc	Link with the C library
-lm	Link with the Math library
-lgcc	Link with the GCC library

Table 10. Linker Options *...continued*

Option	Description
-specs=nano.specs	Use Newlib nano specs
-specs=nosys.specs	Do not use printf/scanf
--sysroot=$ (LIB_DIR)	Specifies the path to the sysroot, for Cortex-M7, it is $ (TOOLCHAIN_DIR)/arm-none-eabi/newlib

2.7.2 GHS Compiler/Linker/Assembler Options

Table 11. Compiler Options

Option	Description
-cpu=cortexm7	Selects target processor: Arm Cortex M7
-thumb	Selects generating code that executes in Thumb state
-fpu=vfpv5_d16	Specifies hardware floating-point using the v5 version of the VFP instruction set, with 16 double-precision floating-point registers
-fsingle	Use hardware single-precision, software double-precision FP instructions
-c99	Specifies the ISO C99 base standard
--ghstd=last	Use the most recent version of Green Hills Standard mode (which enables warnings and errors that enforce a stricter coding standard than regular C and C++)
-Osize	Optimize for size
--gnu_asm	Enables GNU extended asm syntax support
-dual_debug	Generate DWARF 2.0 debug information
-G	Generate debug information
-keeptempfiles	Prevents the deletion of temporary files after they are used. If an assembly language file is created by the compiler, this option will place it in the current directory instead of the temporary directory
-Wimplicit-int	Produce warnings if functions are assumed to return int
-Wshadow	Produce warnings if variables are shadowed
-Wtrigraphs	Produce warnings if trigraphs are detected
-Wundef	Produce a warning if undefined identifiers are used in #if preprocessor statements
--unsigned_chars	Let the type char be unsigned, like unsigned char
--unsigned_fields	Bitfelds declared with an integer type are unsigned
--no_commons	Allocates uninitialized global variables to a section and initializes them to zero at program startup
--no_exceptions	Disables C++ support for exception handling
--no_slash_comment	C++ style // comments are not accepted andgenerate errors
--prototype_errors	Controls the treatment of functions referenced or called when no prototype has been provided
--incorrect_pragma_warnings	Controls the treatment of valid #pragma directives that use the wrong syntax
-c	Stop after assembly and produce an object file for each source file
-DUSE_SW_VECTOR_MODE	Predefine USE_SW_VECTOR_MODE as a macro, with definition 1. By default, the drivers are compiled to handle interrupts in Software Vector Mode.

Table 11. Compiler Options*...continued*

Option	Description
-DD_CACHE_ENABLE	Predefine D_CACHE_ENABLE as a macro, with definition 1. Enables data cache initalization in source file system.c under the Platform driver
-DI_CACHE_ENABLE	Predefine I_CACHE_ENABLE as a macro, with definition 1. Enables instruction cache initalization in source file system.c under the Platform driver
-DMPU_ENABLE	Predefine MPU_ENABLE as a macro, with definition 1. Enable MPU default configuration from startup code under the Platform driver
-DENABLE_FPU	Predefine ENABLE_FPU as a macro, with definition 1. Enables FPU initalization in source file system.c under the Platform driver
-DMCAL_ENABLE_USER_MODE_SUPPORT	Predefine MCAL_ENABLE_USER_MODE_SUPPORT as a macro, with definition 1. Allows drivers to be configured in user mode.

Table 12. Assembler Options

Option	Description
-cpu=cortexm7	Selects target processor: Arm Cortex M7
-fpu=vfpv5_d16	Specifies hardware floating-point using the v5 version of the VFP instruction set, with 16 double-precision floating-point registers
-fsingle	Use hardware single-precision, software double-precision FP instructions
-preprocess_assembly_files	Controls whether assembly files with standard extensions such as .s and .asm are preprocessed
-list	Creates a listing by using the name and directory of the object file with the .lst extension
-c	Stop after assembly and produce an object file for each source file

Table 13. Linker Options

Option	Description
-T linker_script_file.ld	Use linker_script_file.ld as the linker script. This script replaces the default linker script (rather than adding to it)
-e Reset_Handler	Make the symbol Reset_Handler be treated as a root symbol and the start label of the application
-map	Produce a map file
-keepmap	Controls the retention of the map file in the event of a link error
-Mn	Generates a listing of symbols sorted alphabetically/numerically by address
-delete	Instructs the linker to remove functions that are not referenced in the final executable. The linker iterates to find functions that do not have relocations pointing to them and eliminates them
-ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete. DWARF debug information will contain references to deleted functions that may break some third-party debuggers
-Llibrary_path	Points to library_path (the libraries location) for thumb2 to be used for linking
-larch	Link architecture specific library

Table 13. Linker Options*...continued*

Option	Description
-lstartup	Link run-time environment startup routines. The source code for themodules in this library is provided in the src/libstartup directory
-lind_sd	Link language-independent library, containing support routines for features such as software floating point, run-time error checking, C99 complex numbers, and some general purpose routines of the ANSI C library
-v	Prints verbose information about the activities of the linker, including the libraries it searches to resolve undefined symbols
-keep=C40_Ip_AccessCode	Avoid linker remove function C40_Ip_AccessCode from Fls module because it is not referenced explicitly
-nostartfiles	Controls the start files to be linked into the executable

2.8 Debugger

• Lauterbach TRACE32 JTAG Debugger

The above mentioned debug probes are supported plug and play by the example applications. Other debug probes compatible with S32K3 family can be used.

2.9 Examples and Demos

The drivers provide a set of examples. For details, please refer to Examples\...\readme.txt file from each driver folder.

3 Known Issues for S32K3_S32M27x RTD 6.0.0 QLP01

3.1 Known Issues for 6.0.0 QLP01

ID	Headline
ARTD-192451	Issue with Software Sempahore in multicore derivatives

4 Changes List for S32K3_S32M27x RTD 6.0.0 QLP01

4.1 Change List for 6.0.0 QLP01

ID	Subtype	Headline
ARTD-184706	New Feature	Synchronizing flash read/write access between Crypto and Flash driver for Standard RTD release
ARTD-177555	Bug	CLONE - [Crypto_Sec_ng] Wrongly handled memory mapping-Static variables in function scope
ARTD-177528	Bug	[Crypto_Sec_Ng] [S32K3XX] GetCoreID() calling issue in the single core MCU when CryptoEnableTCMSupport is true
ARTD-176446	Bug	[CRYPTO_NG] keyFormat.eccKeyFormat in hseExportKeySrv not initialized

5 Acronyms

Table 14. Acronyms

Acronym	Description
CD	Code Drop
EAR	Early Access Release
PRC	Production Release Candidate
RFP	Ready For Production
RTD	Real-Time Drivers
HSE FW	Hardware Security Engine - Firmaware
S32DS	S32DS Design Studio-IDE

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents