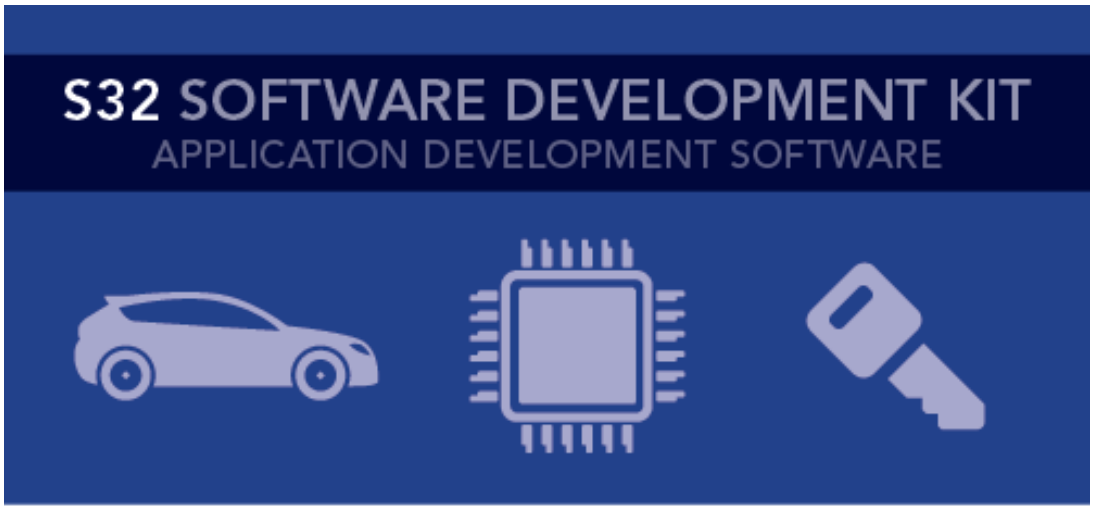

S32 SDK for S32V23x
Release Notes
Version 1.0.0 RTM

Contents

1. Description

The S32 Software Development Kit (S32 SDK) is an extensive suite of peripheral drivers, RTOS, stacks and middleware designed to simplify and accelerate application development on NXP S32V23x ARM based microcontrollers.

This release has RTM quality status in terms of testing and quality documentation.
RTM releases contain all planned features implemented and tested.
RTM releases are candidates that can be used in production.

This SDK can be used as is (see Documentation) or it can be used with S32 Design Studio IDE.

Refer to *License(License.txt)* for licensing information and *Software content register(SW-Content-Register-S32-SDK.txt)* for the Software contents of this product. The files can be found in the root of the installation directory.

For support and issue reporting use the following ways of contact:

- NXP Support to https://www.nxp.com/support/support:SUPPORTHOME
- NXP Community
  https://community.nxp.com/community/s32/s32sdk?tid=communitys32

2. New in this release

2.1 Drivers

PINS

- Added configurator support for generating DDR pins settings.

ADC_SAR

- Improved configurator UI channel chains configuration – use list of channels instead of bitmask.

EIM, ERM

- Improved all available code documentation.

2.2 Examples

AMMClib

- Added example.

SCST

- Added example.

2.3 RTOS

FreeRTOS

- Updated to v10.1.1

2.4 Middleware

TCP/IP

- Stack using either bareboard or FreeRTOS services.

SDHC

- Stack using either bareboard or FreeRTOS services.

2.5 Libraries

AMMClib

- Library added.

SCST

- Library added.

2.6 Fixed from BETA 0.9.0

Component	Description
adc_pal	New project created with ADC PAL component was generating linker error because of undefined symbol for the default ADC callback name
adc_pal	ADC_PAL generated code was issuing compilation error when adcPalGroupConfig was not selected as "Read-only"
adc_sar	ADC_DRV_ConfigWdg() was triggering DEV_ASSERT when numThresholds and numChannels were equal with the maximum allowed value.
adc_sar	Generated code for ADC_SAR was issuing compilation error when watchdog threshold was not used

can_pal	The driver used extended time segments for arbitration phase when FD capabilities were enabled.
can_pal	The clock source selection should not be present in FlexCAN configuration component because it is not supported by driver. The clock selection for flexcan module should be done from clocks tool.
can_pal	The default names for the FlexCAN configuration structures generated by the configuration tool were identical, resulting in duplicated symbols at compile time, unless modified by the user prior to code generation.
clock_manager	CLOCK_DRV_GetFrequency would return invalid frequencies for some peripheral clocks: LFAST0_CLK, ENET0_CLK, FTIMER1_EXT_CLK, DDR4_CLK, LINx_MODULE_CLK.
clock_manager	CLOCK driver would allow more than 8 peripheral clock configurations (maximum number allowed by hardware).
clock_manager	Default value for fixed VIDEO_PLL_PHI was wrong.
clock_manager	Clock configuration would not allow the selection of MMDC source clock.
cpu	SystemCoreClockUpdate did not retrieve the correct clock frequency.
crc	CRC CT component included some non present files.
dspi	When frame size is 32 bits and interrupt mode is used to receive data, the alignment is wrong. First 2 bytes are swapped by last 2 bytes.
examples	hello_world example did not include sdk_project_config.h.
examples	Compiler options were not aligned.
examples	Typos were present in example documentation.
examples	FCCU example documentation contained some invalid references.
examples	Fixed freeRTOSConfig.h to upper case FreeRTOSConfig.h in example.
examples	The imported freertos example was not containing the freertos configuration component and the settings would not be preserved when updating the configuration.
flexray	Driver did not check the validity of the FIFO configuration pointer (could be NULL).
flexray	The configuration component lacked valid default values for preset bitrates.

hyperflash	The flags from Interrupt Status Register were not cleared after they were checked. The problem was fixed by clearing the corresponding flags after the event was checked.
hyperflash	Locking the Secure Silicon Region(SSR) could have been done only if the SSR overlaid the first sector. The issue was fixed, the lock for the 32 regions could now be done if SSR overlays any sector.
i2c	Aborting a transfer using I2C_DRV_MasterAbortTransfer function could not have been done safely, as master module could have been disabled in the middle of the transfer, which caused the slave to hold the SDA line forever low and block the I2C bus. The issue was fixed by aborting the transfer at byte boundary. Same issue was fixed for blocking functions in case TIMEOUT occurred.
pins	PINS generated code was issuing build error when no pin was configured
pins	Default values for some pins were not generated correctly even though they where shown correctly in the UI
pins	The "sdk_project_config.h" file would include configuration for pins and clocks even if the tools were disabled.
pwm_pal	Constraints on the relation between period and duty were not mentioned in the documentation and not guarded by DEV_ASSERTs, for PWM_Init, PWM_UpdateDuty and PWM_UpdatePeriod.
pwm_pal	PWM_PAL configurator was not issuing warning when different values for deadtime were selected
s32ct	Misra violations were present in generated code.
timing_pal	For TIMING_PAL over STM, TIMING_GetRemaining and TIMING_GetElapsed were returning wrong value from the second period, if the callback was NULL or notification disabled.

3. Software Contents

3.1 Drivers

- ADC_SAR
- CLOCK MANAGER
- CPU
- CRC
- CSE3
- DSPI
- EDMA
- EIM
- ENET
- ERM
- FCCU
- FLEXCAN
- FLEXRAY
- FTM
- HEADER
- HYPERFLASH
- I2C
- INTERRUPT MANAGER
- LINFLEXD_UART
- OSIF
- PHY
- PINS
- PIT
- POWER MANAGER
- QSPI
- SEMA42
- STM
- SWT
- USDHC
- WKPU
- XRDC

3.2 PAL

- ADC_PAL
- CAN_PAL
- I2C_PAL
- IC_PAL
- MPU_PAL
- OC_PAL
- PWM_PAL
- SECURITY_PAL
- SPI_PAL
- TIMING_PAL
- UART_PAL
- WDOG_PAL

3.3 RTOS

- FreeRTOS version 10.1.1

3.4 Middleware

- SDHC
- TCP/IP

3.5 Libraries

- AMMClib version 1.1.16
- SCST version 1.0.2

4. Documentation

- Quick start guide available in “doc” folder.
- User and integration manual available at “doc\Start_here.html”.
- Driver user manuals available in “doc” folder.
- Release notes for Middleware available in “doc” folder.
- Documentation for the Middleware can be found in the respective folder.

5. Examples

	Name	Description
Driver examples	adc_swtrigger	Simple application using ADC to read converted values.
	adc_pal	Configures a group of channels formed of only the internal analog channels that can be monitored by the ADC.
	can_pal	The example listens for CAN frames, displays the received data to the semihosting console in S32 Design Studio and replies back with the value received.
	crc_checksum	Calculates CRC using the peripheral driver for multiple standards.
	edma_transfer	Shows the usage of eDMA.
	eim_injection	The EIM module enables the user to inject 1 bit error or 2 bit errors into bus data.
	enet_ping	Shows the usage of ENET.
	erm_report	The ERM module reports any detected memory error.
	fccu_fault_injection	Show the usage of FCCU driver.
	flexcan	Shows the usage of FlexCAN driver configured as both bus master and slave.
	flexray	Shows the functionality of FLEXRAY.
	ftm	The example toggles LED2 at 1HZ using FTM_MC driver
	linflexd_uart	Shows the functionality of LINFLEXD module in UART mode.
	mpu_pal_memory_protection	Shows the usage of the MPU_PAL.
	oc_pal	Shows the usage of the OC_PAL over FTM.
	phy_autoneg	Shows the functionality of PHY.
	pit_periodic_interrupt	The demo is configured to trigger an interrupt every second, which toggles a LED.
	power_mode_switch	Transitions the MCU into all available power modes.
	stm_periodic_interrupt	Shows the usage of the System Timer Module.
	swt_interrupt	Shows the usage of the Software Watchdog Timer.
	timing_pal	Shows the usage of the TIMING_PAL over PIT and FTM
	uart_pal	Shows the usage of UART PAL over LinFlexD
	wdg_pal_interrupt	Shows the usage of the WDOG_PAL
	xrdc_memory_protection	Shows how to use Extended Resource Domain Controller
Demos	FreeRTOS	Shows the usage of FreeRTOS
	hello_world	This is a simple application created to show the basic configuration with S32DS
	hello_world_mkf	This is a simple application created to show the basic configuration with makefile for the supported compilers
	lwip	Shows the usage of TCP IP stack

	sdhc_fatfs	Shows the usage of SDHC stack
	sdhc_freertos	Shows the usage of SDHC stack with FreeRTOS
	ammclib	Shows integration of AMMClib.
	scst	Shows integration of core self tests.

6. Supported hardware and compatible software

6.1 CPUs

- S32V234 - 1N81U
- S32V232

The following processor reference manual has been used to add support:

- S32V234RM Rev. 3 10/2017

6.2 Boards

- EVB SBC-S32V234 Microsys
- X-TR-DVAL-625 PCB RevX2

6.3 Compiler and IDE versions:

- GCC Compiler for ARM NXP GCC 6.3.1
  - 20170509 (BLD = 1574 rev=g924fb68)
  - included in S32 Design Studio v3.1
- Green Hills Multi 7.1.4 / v.2017.1.4
- Windriver DIAB Compiler v5.9.6.2

6.4 Debug Probes

- Lauterbach TRACE32 JTAG Debugger
- P&E Multilink (with P&E GDB Server)

7. Known issues and limitations

7.1 S32 Design Studio integration

- An error is returned when a new component is added to the project.
- Attach / Detach SDK functionality does not work at the moment, therefore the user cannot create a project without the SDK and add it afterwards. Workaround: Create a project with SDK enabled from the beginning with New Project Wizard or start from an example from the SDK release.

7.2 S32 Configuration Tool integration

- If the same configuration component is enabled over multiple module instances, the according generated structures will have the same name. It is user's responsibility to make sure different names are used for different structures.

7.3 Drivers

CLOCK_MANAGER

- Clock sources can't be enabled/disabled per power mode. A clock source is enabled or is disabled in all power modes. Module clock gate can't be configured from "Peripheral clocks". As a workaround module clock gate must be configured from clock diagram.

FTM_MC

- The hardware trigger is not work as expected when the source is ENET module from MAC0_TIMER3 to trigger0 of FTM.

FLEXRAY

- Flexray can only operate stably when system and FIFO memory is allocated into TCM area (start address: 0x3E000000); the size of all message buffers is limited to maximum 32KB.

LINFLEXD_UART

- In DMA mode, a new reception may contain junk data received previously; the FIFO cannot be flushed before receiving a new buffer.

PIT/STM

- Module cannot run in Debug Mode (counter not count).

PINS

- When calling PINS_DRV_Init, the configuration for EIRQ0 pin will be overridden if there is another pin with disabling interrupt field that is configured after EIRQ0 pin. The workaround is to configure EIRQ0 at the last index of the array.

POWER_MANAGER

- User must enable clock source in other mode of the clock configuration which correspond with peripheral clock source. This one is changed before user calls the API CLOCK_DRV_Init.

PWM_PAL

- For PWM_PAL over FTM the deadtime value is set only from first channel config struct element.

SWT

- Module does not return a bus error when accesses are invalid and the module is configured to not reset the CPU on invalid accesses.

7.4 Stacks

TCP/IP

- No FreeRTOS support (i.e. only bareboard version is available).

SDHC

- File system timestamp is not available.

7.5 Examples

- Some examples may display warning messages with unresolved includes.

8. Compiler options

8.1 GCC Compiler/Linker/Assembler options

Table 8-1 GCC Compiler options

Option	Description
-mcpu=cortex-m4	Selects target processor: Arm Cortex M4
-mthumb	Selects generating code that executes in Thumb state.
-std=c99	Use C99 standard
-DCPU_S32V234	Define a preprocessor symbol for MCU
-L$(<library_path>)	Add specific library used in the compiler options. V23X : ../arm-none-eabi/newlib/lib/thumb/v7e-m/fpv4-sp/(softfp or hard)
-g	Generate debug information
-mfpu=fpv4-sp-d16 -mfloat-abi=hard	Use single precision FPU instructions
-O1	Optimize option
-Werror	Treat warnings as errors
-Wall	Produce warnings about questionable constructs
-Wextra	Produce extra warnings that -Wall
-Wstrict-prototypes	Warn if a function is declared or defined without specifying the argument types.
-pedantic	Issue all the warnings demanded by strict ISO C
-Wunused	Produce warnings for unused variables
-Wsign-compare	Produce warnings when comparing signed type
-funsigned-char	Let the type char be unsigned, like unsigned char
-funsigned-bitfields	Bit-fields are signed by default
-fshort-enums	Allocate to an enum type only as many bytes as it needs for the declared range of possible values.
-ffunction-sections	Place each function into its own section in the output file
-fdata-sections	Place data item into its own section in the output file
-fno-common	The -fno-common option specifies that the compiler should place uninitialized global variables in the data section of the object file.
-fno-jump-tables	Do not use jump tables for switch statements

Table 8-2 GCC Linker options

Option	Description
-mcpu=cortex-m4	Selects target processor
-mthumb	Selects generating code that executes in Thumb state
--entry=<entry_symbol>	Make the symbol Reset_Handler be treated as a root symbol and the start label of the application
-T <linker_script_file.ld>	Use the specified linker file
-Xlinker --gc-sections	Remove unused sections
-lc	Link C library
-lm, -lgcc	Link Math library, Link libgcc
-Wl, -Map=<map_file_name>	Produce a map file
-mfpu=fpv4-sp-d16 -mfloat-abi=hard	Use single precision FPU instructions

Table 8-3 GCC Assembler options

Option	Description
-mcpu=cortex-m4	Selects target processor
-mthumb	Selects generating code that executes in Thumb state
-mfpu=fpv4-sp-d16 -mfloat-abi=hard	Use single precision FPU instructions
-x assembler-with-cpp	Preprocess assembly files

8.2 GHS Compiler/Linker/Assembler options

Table 8-4 GHS Compiler options

Option	Description
-cpu=cortexm4	Selects target processor
-thumb	Selects generating code that executes in Thumb state.
-c99	Use C99 standard
--gnu_asm	Enables GNU extended asm syntax support
-DCPU_S32V234	Define CPU name
-L$(<library_path>)	Add specific library used in the compiler options. /lib/thumb2
-gdwarf-2	Generate DWARF 2.0 debug information
-G	Generate debug information
-fsingle, -fpu=vfpv4_d16	Use single precision FPU instructions
-Ogeneral	Optimize option
-Wunknown-pragmas	Produce warnings when unknown pragmas are used
-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
--quit_after_warnings	Treat warnings as errors
--unsigned_chars	Let the type char be unsigned, like unsigned char
-unsigned_fields	Bitfields declared with an integer type are unsigned
--short-enum	Store enumerations in the smallest possible type
-fno-common	Allocates uninitialized global variables to a section and initializes them to zero at program startup

Table 8-5 GHS Linker options

Option	Description
-cpu=cortexm4	Selects target processor
-entry=<entry_symbol>	Make the symbol Reset_Handler be treated as a root symbol and the start label of the application
-T <linker_script_file.ld>	Use the specified linker file
-map=<map_file_name>	Produce a map file
-larch	Link architecture specific library
-delete -ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete

Table 8-6 GHS Assembler options

Option	Description
-cpu=cortexm4	Selects target processor
-preprocess_assembly_files	Preprocess assembly files

8.3 DIAB Compiler/Linker/Assembler options

Table 8-7 DIAB Compiler options

Option	Description
-tARMCORTEXM4LV	Selects target processor
-mthumb	Selects generating code that executes in Thumb state
-Xdialect-c99	Use C99 standard
-DCPU_S32V234	Define CPU name
-Xfp-float-only	Use single precision FPU
-g	Add debug information to the executable
-O	Optimize option
-ei5388,5387,1824	ignore some specific warnings
-Xstop-on-warning	Treat warnings as error
-Xsection-split	Generate a separate section for each function/variable to remove some unused function
-Xno-common	Allocates uninitialized global variables to a section and initializes them to zero at program startup

Table 8-8 DIAB Linker options

Option	Description
-tARMCORTEXM4LV	Selects target processor
-Xremove-unused-sections	Removes unused code sections
-lc	Link the standard C library to the project in order to support elementary operations that are used by the drivers
-lm	Link the standard math library to the project in order to support elementary math operations that are used by the drivers
<linker_script_file.dld>	Use the specified linker file
-e <entry_symbol>	Make the symbol Reset_Handler be treated as a root symbol and the start label of the application
-m6 > <map_file_name>	Produce a linker map

Table 8-9 DIAB Assembler options

Option	Description
-tARMCORTEXM4LV	Selects target processor
-Xpreprocess-assembly	Preprocess assembly files

9. Acronyms

Acronym	Description
EAR	Early Access Release
JRE	Java Runtime Environment
EVB	Evaluation board
PAL	Peripheral Abstraction Layer
RTOS	Real Time Operating System
S32CT	S32 Configuration Tool
PD	Peripheral Driver
S32DS	S32 Design Studio IDE
SDK	Software Development Kit
SOC	System-on-Chip
RTM	Release To Manufacture

10. Version Tracking

Date (dd-Mmm-YYYY)	Version	Comments	Author
14-Oct-2016	1.0	First version for EAR 0.8.0	Cezar Dobromir
30-Ian-2017	1.1	First version for S32V EAR 0.8.0	Iulian T.
10-Dec-2018	1.2	First version for S32V EAR 0.8.1	Banciu Alexandru
26-Mar-2019	1.3	First version for S32V BETA 0.9.0	Banciu Alexandru
26-June-2019	1.4	First version for S32V RTM 1.0.0	Banciu Alexandru