S32 SDK for Power Architecture Release Notes

Version 2.0.0 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 MPC574x-B-C-G, MPC574x-P, S32R274 and S32R372 Power Architecture 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/

2. New in this release

2.1 Fixed from BETA 1.9.0

Component	Description
adc_pal	ADC_MPC574xC_G_R: conversions were being executed and results were written beyond the result buffer boundaries, when HW triggers occurred after ADC_Init() for HW triggered groups for which neither ADC_EnableHardwareTrigger/ADC_DisableHardwareTrigger was called.
adc_pal	There was no DEV_ASSERT and configurator warning when multiple HW triggered groups were configured with 'No delay' or same delay value.
adc_pal	When multiple HW triggered groups were enabled, only the first one actually executed when the trigger event occurred. The rest were not starting execution.
adc_sar	The analog watchdog was not reset by ADC_DRV_Reset() function
can_pal	CAN_PAL & FLEXCAN configuration components would not work in case there were multiple configurations and at least one was disabled.
clock_manager	FXOSC was incorrectly displayed as the source of external clock for ENET module (the actual external clock source is received from the PHY).
clock_manager	On MPC5748G, CLKOUT gating from the configuration component did not work properly.
CPU	customSectionBlock was not located at a fixed memory address
CPU	Build warning were issued when using MPC5744B, MPC5744C, MPC5745B and MPC5746B
CPU	Components Library showed driver components which were not applicable for the active device
CPU	SystemCoreClockUpdate was not calculating the correct value for PLL1
CPU, fccu	FCCU configuration component did not check NCF timeout to be in the correct range
crc	CRC driver could be used with both Bit transpose and Byte transpose enabled, which is not a valid configuration
dspi	When the spi bus frequency was too low the driver was not able to generate it
eee	The EEE_DRV_ReadEeprom function will not read the record ID of the older block when a swapping occurs before that. This issue is fixed to check the blank space point to a address aligned which is not end of EEE block.
eee	When a swapping occurs the driver will copy data from oldest active block. It is missing some records. For example the record 1,2,3 when cache is enabled.
eee	The user need to make sure that the DeInit() function return STATUS_SUCCESS. If not it must re-call again the DeInit() function.
eee	The EEE_DRV_ReadEeprom function would not read the record ID of the older block when a swapping occurred before that

eee	When a swapping occurred the driver would copy data from the oldest active block when cache was enabled thus missing some blocks
eee	The user needed to make sure that the DeInit() function return STATUS_SUCCESS
eee, flash	When checking PGOOD (PEG) was low FLASH_DRV_CheckProgramStatus() did not return error
emios	The eMIOS_PWM output signal follow active high while PWM_PAL follow active low and active high.
emios	EMIOS_IC: Get period measurement in internal timebase was not stable
emios	EMIOS_DRV_OutputDisable disabled the selected channel, but enabled all others
emios	EMIOS_DRV_PWM_GetDutyCycle returned an incorrect value
emios	Output pwm signal was inverted in center align mode
emios	The eMIOS_PWM output signal follow active high while PWM_PAL follow active low and active high
examples	Doxygen documentation for hsm_keyconfig example contained some innacurate information about flash partitioning on both MPC5746R and MPC5748G.
examples	The FreeRTOS example on S32R372 was issuing an error saying that the selected target processor was wrong
examples	FreeRTOS example on S32R274: debug settings were missing (no core selected)
fccu	FCCU_DRV_IrqHandler had unreacheable code
fccu	FCCU_DRV_DisableFault returned STATUS_SUCCESS even if the module was in FAULT state
flexcan	FLEXCAN_DRV_RxFifoBlocking did not stop the dma channel, in case of timeout.
flexcan	FLEXCAN transfer abort mechanism did not work properly.
flexcan	FLEXCAN on S32R274 missed IRQ handlers for mailboxes greater than 63.
flexcan	FLEXCAN driver would not remove references to previous callbacks after deinitialization; subsequent driver initializations would keep the old callbacks installed.
flexcan	FLEXCAN driver would send a remote frame immediately calling FLEXCAN_DRV_ConfigTxMb function to configure a tx message buffer for remote frames.
flexcan	In Rx FIFO mode, flexcan driver could report error events even after the receive operation was complete.
flexcan	After initialization the driver would always get the module out of freeze mode, even if freeze mode was selected in the user configuration.
flexpwm	FLEXPWM examples: the LED intensity was only increasing, instead of alternatively increasing to max then decreasing to 0.

flexpwm	FLEXPWM_DRV_GetDefaultConfig() was setting reload logic to "FlexPwmReloadImmediate" instead of "FlexPwmReloadPwmFullCycle"
i2c	In case errors occurred on DMA channel the status returned by I2C_DRV_MasterSendData was still STATUS_SUCCESS
i2c	If I2C address or baudrate were out of range PEx component didn't generate any error
i2c_pal	For I2C_DRV_MasterSendData, in case DMA mode was used and the baudrate was higher than 100KHz, the DMAEN bit was de-asserted after the acknowledge was received from the second last byte. Because of this, the interrupt for transfer complete was not triggered and the last byte was not sent. The problem was solved by sending the last byte in the complete dma transfer subroutine, after the transfer complete flag is checked.
i2c_pal	For I2C_DRV_MasterSendData, in case DMA mode was used and the baudrate was higher than 100KHz, the DMAEN bit was de-asserted after the acknowledge was received from the second last byte. Because of this, the interrupt for transfer complete was not triggered and the last byte was not sent
power_manager	On S32R274, S32R372: An invalid mode (Clock Usage) configuration error could occur if the Sigma-Delta PLL and PLL1 were switched on or off independently and mode transition failed
pwm_pal	PWM_PAL configurator was generating wrong instance index when the selected device was not the first type available in the drop-down list
pwm_pal	PWM_PAL configurator was not issuing error when the same timebase was selected for multiple emios channels, with different channel type, on the same instance. PWM_PAL configurator was not issuing error when the same channel was selected multiple times. PWM_PAL configurator was not issuing error when the duty cycle was greater than the period.
pwm_pal	Compilation error for multiple definitions of variable occurred on PWM_PAL generated code, when the same Bus Timebase was selected for multiple PWM_PAL instances over EMIOS
rtc	Refactored: RTC_DRV_ConfigurePeriodicInterrupt, RTC_DRV_ConfigureComparatorTrigger and hardware access functions
rtc	RTC PEx component was not reporting when tick value is not an integer
rtc	Refactored RTC_DRV_ConfigurePeriodicInterrupt, RTC_DRV_ConfigureComparatorTrigger and hardware access functions
rtc	RTC PEx component was not reporting when tick value is not an integer
smpu	Some region information was not correct in smpu documentation
timing_pal	Default configuration generated for timing_pal was causing DIAB compiler error because the default interrupt handler was undefined
uart_pal	UART_PAL configuration component allowed callback function names that were not valid C identifiers.

3. Software Contents

3.1 Drivers

- ADC_SAR
- BCTU
- CLOCK MANAGER
- CMP
- CPU
- CRC
- CTU
- CSE
- DSPI
- EDMA
- EIM
- EMIOS
- ENET
- ETIMER
- FCCU
- FLASH
- FLEXCAN
- FLEXPWM
- HEADER
- HSM (SHE firmware v.1.0.5)
- I2C
- INTERRUPT MANAGER
- LINFLEX (UART)
- OSIF
- PASS
- PHY
- PINS
- PIT
- POWER MANAGER
- RTC_API
- SAI (I2S)
- SEMA42
- SMPU
- SRX
- STM
- SWI2C
- SWT
- TDM
- USDHC
- WKPU

3.2 PAL

- ADC_PAL
- CAN_PAL
- I2C_PAL
- I2S_PAL
- IC_PAL
- MPU_PAL
- OC_PAL
- PWM_PAL
- SECURITY_PAL
- SPI_PAL
- TIMING_PAL
- UART_PAL
- WDG_PAL

3.3 RTOS

- FreeRTOS version 9.0.0

3.4 Middleware

- EEE
- FATFS
- SDHC
- TCP/IP
- USB

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

Type	Name	Description
Driver examples	adc_pal	Shows the usage of the ADC_PAL
	adc_swtrigger	Shows the usage of the ADC MPC574xx
	bctu_trigger	Shows the usage of BCTU cross triggering
	can_pal	Shows the usage of the CAN_PAL
	cmp_dac	Shows how to use CMP with the internal DAC
	crc_checksum	Calculates CRC using the peripheral driver for multiple standards.
	ctu_trigger	Shows the usage of the CTU module
	dspi_master	Shows the usage of the DSPI/SPI module in master mode
	dspi_slave	Shows the usage of the DSPI/SPI module in slave mode
	edma_transfer	Show multiple usage scenarios of DMA.
	emios_ic	Shows the usage of the eMIOS IC functionality
	emios_oc	Shows the usage of the eMIOS OC functionality
	emios_pwm	Shows the usage of the eMIOS PWM functionality
	enet_loopback	Shows the usage of the ENET module configured in loopback
	enet_ping	Shows the usage of the ENET module by implementing an application which responds to ping requests.
	etimer	Shows the usage of the ETIMER module
	fccu_fault_injection	Show the usage of FCCU driver.
	flash_program_erase	Shows the usage of the flash driver how to program or erase the flash memory
	flexpwm_pwm	Shows the usage of the PWM functionality of FlexPWM
	hsm_key_config	Demonstrates the non-volatile key update procedure
	i2c_pal	Shows the usage of the I2C_PAL
	i2c_pal	Shows the usage of the I2C_PAL
	i2c_transfer	Shows the usage of the I2C driver in both master and slave modes.
	i2s_pal	Shows the usage of the I2S_PAL
	ic_pal	Shows the usage of the I2C_PAL
	interrupt_control_multicore	Shows the usage of the Interrupt Manager in a multicore environment
	linflexd_uart	Shows the usage of LINFlexD_UART driver in interrupt based mode
	mpu_pal_memory_protection	Shows the usage of the MPU_PAL
	oc_pal	Shows the usage of the OC_PAL
	pass_lock_unlock	Shows the usage of the PASS module

	phy_autoneg	Shows the usage of the PHY module with autonegociation
	pit_periodic_interrupt	Shows the usage of the PIT
	power_mode_switch	Transitions the MCU into all available power modes.
	pwm_pal	Shows the usage of PWM_PAL
	rtc_alarm	Shows the usage of the RTC
	sai_transfer	Shows the usage of the SAI driver in both master and slave modes
	security_pal	Shows the usage of the SECURITY_PAL
	sema42_multicore	Shows the usage of SEMA42 driver simultaneous over all available cores
	smpu_protection	Shows how to configure SMPU to protect a region of memory
	spi_pal	Shows the usage of the SPI_PAL
	spi_pal_master	Shows the usage of the SPI_PAL in master mode
	spi_pal_slave	Shows the usage of the SPI_PAL in slave mode
	srx_fast_dma	Shows the usage of SRX in DMA based mode
	stm_periodic_interrupt	Shows the usage of the STM
	swi2c_master	Shows the usage of the SWI2C
	swt_interrupt	Shows the usage of the SWT
	timing_pal	Shows the usage of the TIMING_PAL
	uart_pal	Shows the usage of UART PAL over LinFlexD
	wdg_pal_interrupt	Shows the usage of the WDOG_PAL
	wkpu_interrupt	Shows the usage of the WKPU driver
Demos	eeprom_emulation	Shows basic use cases of the EEPROM Emulation middleware
	flexcan	Shows the usage of FlexCAN driver configured as both bus master and slave
	freertos	Shows the usage of the FreeRTOS MPC574xx
	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
	hsm_freertos	Shows the usage of HSM driver using two tasks (one for encryption, one for decryption)
	lwip	Shows the usage of TCP IP stack
	sdhc_fatfs	Shows the usage of FAT FS over uSDHC driver
	sdhc_freertos	Shows the usage of SHDC with FATFS over FreeRTOS
	usb_msd_fatfs	This is a simple FATFS application created to access USB mass storage device
	usb_cdc_lwip	Shows the usage of the TCP/IP stack over USB ethernet

6. Supported hardware and compatible software

6.1 CPUs

- MPC5744B
- MPC5745B
- MPC5746B
- MPC5744C
- MPC5745C
- MPC5746C - 1N84S (Cut 2.1)
- MPC5747C
- MPC5748C
- MPC5746G
- MPC5747G
- MPC5748G - 0N78S (Cut 3.0)
- MPC5741P
- MPC5742P
- MPC5743P
- MPC5744P - 1N15P (Cut 2.2B)
- S32R274 - 2N58R (Cut 1.2)
- S32R372 - 0N36U (Cut 1.0)

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

- MPC5748GRM Rev. Rev. 6, 10/2017
- MPC5746CRM Rev. 5, 10/2017
- MPC5744PRM Rev. 6, 06/2016
- S32R274RM Rev. 4, 05/2018
- S32R372 RM Rev. 3.1, 05/2018

6.2 Boards

- DEVKIT-MPC5744P PCB RevX1 SCH RevB
- DEVKIT-MPC5748G PCB RevA SCH RevB
- Daughter Card MPC574XG-256DS Rev B
- Daughter Card X-MPC574XG-324DS Rev A
- Motherboard X-MPC574XG-MB Rev D
- Daughter Card MPC5744P-257DS Rev B1
- Motherboard MPC57XX Rev C
- S32R274RRUEVB 700-28921 REV B SCH-28921 REV D

6.3 Compiler and IDE versions:

- GCC E200 VLE GNU Compiler 4.9.4
  - 20160726 (release_g738c595_build_Fed_ELe200_ML3)
  - included in S32DS for Power Architecture 2017 R1
- Green Hills Multi 7.1.4 / Compiler 2015.1.6
- 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

- Some warnings might be observed after project creation or import.
- Project creation takes a considerable amount of time.
- On multicore projects, it might take a greater amount of time to debug the projects in FLASH target.

7.2 Drivers

CPU

- Special care must be taken when using core exceptions. The defined handlers must save/restore all the registers compiler might use. See core reference manual for more details. E.g. IVOR3_Handler must save the context so that it will work in all possible cases. This is not applicable for IVOR1, IVOR4 and IVOR8.

CRC

- When generating CRC-32 for the ITU-T V.42 standard the user needs to set SWAP_BYTEWISE together with INV and SWAP.
- When generating CRC-16 the user needs to set SWAP_BITWISE bit.

CSE

- For CSE driver and SECURITY PAL compiled with GHS, the key update procedure provided as example may incorrectly set the 'write protected' flag, hence subsequent erasing of the non volatile keys becomes impossible.

EEE

- The EEE_DRV_ReportEepromStatus() function will return the erasing cycles of the current ACTIVE block. This number is not an accurate value. Because if brownout occurs during updating erase cycle, this erasing cycle will be re-counted from the erase cycle value of the other block.
- The user needs to ensure that EEE_DRV_MainFunction() function is called after every write operation. The user can check the status of g_eraseStatusFlag global variable after writing data record to decide when needs to call this function.
- When ECC errors occur during the read operation from flash, the driver only supports to get the failing address in the C55FMC_ADR register.

ENET

- Code cannot be generated by the configuration component if at least one configuration is disabled.

ETIMER

- When using eTimer in PULSE_OUTPUT mode the generated pulse is of length one clock tick (not a square wave).

FLASH

- It is recommended that the D-cache of the core should be disabled at the initialization code to make sure the program or erase functions work properly.
- Flash controller buffer shall be disabled in the beginning of application for reading and writing to flash.

FLEXCAN

- On MPC5746C and MPC5748G the feature of selfwake-up is not supported. (FLEXCAN_DRV_ConfigSelfWakeUp function)

FLEXPWM

- When using more than one submodule and configuring any of the sub-modules from 1 to 3 to have the reload signal from Master Reload, the LDOK bit for sub-modules 1 to 3 is not cleared.

HEADER FILES

- Not all interrupts are declared in the header file.

I2C

- Aborting a transfer with the function I2C_DRV_MasterAbortTransferData() can't be done safely due to device limitation; there is no way to know the exact stage of the transfer, and if we disable the module in the middle of the transfer of a character the slave may hold the SDA line forever low and block the I2C bus. Same situation may happen if a blocking transfer function is used and TIMEOUT occurs.

IC_PAL over FLEXPWM

- When IC PAL is used over FlexPWM with multi-channels combined in mutil-capture modes, the measured result will not work as expected with high frequency as input signal for capture modes.

IC_PAL and OC_PAL

- PEx component limitation: Multiple PEx components, either IC_PAL or OC_PAL, cannot share the same EMIOS module instance.

LINFLEXD UART

- In DMA mode, bytesRemaining parameter is not always 0 after calling LINFLEXD_UART_GetReceive/TransmittStatus, although all data is successfully transferred.

OSIF

- Current bare metal implementation uses the last PIT channel (3 or 15 depending on platform) for internal timing.

PINS

- Generation of the pin configuration using the PEx component is slow.

POWER MANAGER

- The core must execute code from RAM memory when switching to a mode in which the flash is in power down or low power state.
- MCU cannot enter STOP0, STANDBY, HALT0 mode while debugger is connected. In addition, STOP0, HALT0, STANBY are not supported while CPU executes code from RAM.
- User does not use the internal ballast in applications using any of the STOP, HALT or STANDBY mode. User must ensure the device is set up to use external ballast (INT_BAL_SELECT pin tied to ground on board). Because the interrupt or wakeup event is activated while transition to HALT, STOP or STANDBY mode, the transition is aborted.
- LPU modes are not supported.
- The driver code will switch to RUN0 mode before entering STOP0 or HALT0 mode from DRUN mode. If CPU is in STOP0, HALT0 and a wake-up signal is detected it will switch to RUN0 mode.

PWM_PAL over ETIMER

- When generating signals with 0% or 100% duty cycle, a pulse of length equal to one clock tick is generated on the output which has inverted polarity. Consequently, true 0% or 100% duty cycles cannot be achieved.

PWM_PAL

- PWM_PAL does not check with DEV_ASSERTs that the EMIOS instance index is in the correct range.

RTC

- Driver does not support using all prescalers when 32KHz clock is selected. The application should either disable prescalers, or use a higher clock frequency.

UART PAL, LINFLEXD UART

- For S32R274 & S32R372 (S32R274RRUEVB), LIN1_RX pin on daughter board will not work if J13 and J14 are connected on motherboard.

SRX

- If the driver is reinitialized, the peripheral can't resynchronize with the clock.

SWI2C

- The SWI2C driver doesn't support multi-master mode.
- Detection of bus busy is not supported.
- Baud rate of SWI2C depends on CPU frequency, optimizations, compiler, pull-up resistors that are used, so user should check the baud rate and timing of the SCL and SDA for his application.
- The driver can't ensure a fix baud rate.

SWT

- The driver does not support timer reset in Fixed Execution Address mode and Incremental Execution Address mode (The watchdog is serviced by executing code at the address loaded into the designated IAC register).

7.3 Examples

- WKPU example runs in FLASH only if the reset button in pressed after the download to the target.
- Some examples may display warning messages with unresolved includes.
- PASS example can only be run using Lauterbach debug support. S32 Design Studio debug plugins do not support flash access unlocking on secured chips.

8. Compiler options

8.1 GCC Compiler/Linker/Assembler options

Table 8-1 GCC Compiler options

Option	Description
-mcpu=e200z7/e200z4/-mcpu=e200z2	Selects target processor
-funsigned-char	Let the type char be unsigned, like unsigned char
-funsigned-bitfields	Bit-fields are by default signed
-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-jump-tables	Do not use jump tables for switch statements
-save-temps=obj	Save temp files for debugging purposes
-mbig	Big endian
-mvle	Enable variable-length encoding
-std=c99	Use C99 standard
-msoft-float/-msingle-float	Select the Floating Point type
-g3	Generate debug information
-O1	Optimization level one
-Wall	Produce warnings about questionable constructs
-D<cpu_name>	Define a preprocessor symbol for MCU
-DTURN_ON_CPU***X***	Define for turning on different CPUs. ***X*** should be replaced with the desired CPU to be turned on.
-DSTART_FROM_FLASH	Mandatory define when flash target is used

Table 8-2 GCC Linker options

Option	Description
-gc-sections	Remove unused sections
-lc	Link C library
-lgcc	Link libgcc
-lm	Link Math library
-T <linker_script_file.ld>	Use the specified linker file
--entry= Reset_Handler	Make the symbol Reset_Handler be treated as a root symbol and the start label of the application
-Wl, -Map=<map_file_name>	Produce a map file

Table 8-3 GCC Assembler options

Option	Description
-mcpu=e200z7/e200z4/-mcpu=e200z2	Selects target processor
-mregnames	Emit register names in the assembly language output using symbolic forms
-mbig	Big endian
-mvle	Enable variable-length encoding
-msoft-float/-msingle-float	Select the Floating Point type
-g3	Generate debug information
-O1	Optimization level
-DTURN_ON_CPU0	Mandatory define for the boot core(Z4_0)
-DTURN_ON_CPU***X***	Define for turning on different CPUs. ***X*** should be replaced with the desired CPU to be turned on.

8.2 GHS Compiler/Linker/Assembler options

Table 8-4 GHS Compiler options

Option	Description
-cpu=ppc574xcz4204/-cpu=ppc5748gz210/-cpu=ppc5744pz425/-cpu=ppc5744pz425/-cpu=ppc5775kz7260	Selects target processor
--gnu_asm	Enables GNU extended asm syntax support
-G	Generate debug information
-vle	Enable variable-length encoding
-C99	Use C99 standard
-noSPE	Do not generate SPE or vector floating point instructions
-nostartfiles	Do not add start-up files to link
-fno-common	Allocates uninitialized
-fnone/ -fsoft / -fsingle	Select the Floating Point type
-sda=0	Enables the Small Data Area optimization with a 0 threshold
-dual_debug	Generates the DWARF debugging information in the object file
-Ogeneral	Optimization level
-D<cpu_name>	Define a preprocessor symbol for MCU
-DTURN_ON_CPU***X***	Define for turning on different CPUs. ***X*** should be replaced with the desired CPU to be turned on.
-DSTART_FROM_FLASH	Mandatory define when flash target is used

Table 8-5 GHS Linker options

Option	Description
-nostartfiles	Do not add start-up files to link
-nostdlib	Do not use standard libraries
-entry= Reset_Handler	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
-keepmap	Controls the retention of the map file in the event of a link error
-Mn	Generates a listing of symbols sorted numerically by address
-delete -ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete

Table 8-6 GHS Assembler options

Option	Description
-cpu=ppc574xcz4204/-cpu=ppc5748gz210/-cpu=ppc5744pz425/-cpu=ppc5744pz425/-cpu=ppc5775kz7260	Selects target processor
-preprocess_assembly_files	Enable the run of the C preprocessor over the assembler files
-nostartfiles	Do not add start-up files to link
-noSPE	Do not generate SPE or vector floating point instructions
-gnu_asm	Enables GNU extended asm syntax support
-vle	Enable variable-length encoding
-C99	Use C99 standard
-gdwarf-2	Enables the generation of DWARF debugging information
-sda=0	Enables the Small Data Area optimization with a 0 threshold
-G	Generate debug information
-fnone/ -fsoft / -fsingle	Select the Floating Point type
-dual_debug	Generates the DWARF debugging information in the object file
-O1	Optimization level
-DTURN_ON_CPU0	Mandatory define for the boot core(Z4_0)
-DTURN_ON_CPU***X***	Define for turning on different CPUs. X should be replaced with the desired CPU to be turned on.

8.3 DIAB Compiler/Linker/Assembler options

Table 8-7 DIAB Compiler options

Option	Description
-tPPCE200Z210N3VEN:simple/-tPPCE200Z4204N3VEN:simple/-tPPCE200Z4201N3VEN:simple/-tPPCE200Z7260N3VEN:simple	Selects target processor
-Xdialect-c99	Use C99 standard
-Xsection-split	Generate a separate section for each function/variable
N/S	The N in the target processor name shall be replaced with S for software FPU
-g3	Add debug information to the executable
-Xdebug-local-all	Emit debug information for unused local variables
-Xdebug-local-cie	Generate a local Common Information Entry (CIE) for each unit.
-Xdebug-struct-all	Disable debug optimization of type information
-Xdebug-dwarf2	Generate DWARF 2 debug information
-XO	Enable extra optimizations
-Xsmall-data=0	Disable small data
-Xsmall-const=0	Disable small const data
-D<cpu_name>	Define a preprocessor symbol for MCU
-DTURN_ON_CPU***X***	Define for turning on different CPUs. ***X*** should be replaced with the desired CPU to be turned on.
-DSTART_FROM_FLASH	Mandatory define when flash target is used

Table 8-8 DIAB Linker options

Option	Description
-tPPCE200Z210N3VEN:simple/-tPPCE200Z4204N3VEN:simple/-tPPCE200Z4201N3VEN:simple/-tPPCE200Z7260N3VEN:simple	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 Reset_Handler	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

-Xremove-unused-sections	remove unused section
N/S	The N in the target processor name shall be replaced with S for software FPU
-Xpreprocess-lecl	Perform pre-processing on linker scripts

Table 8-9 DIAB Assembler options

Option	Description
-tPPCE200Z210N3VEN:simple/-tPPCE200Z4204N3VEN:simple/-tPPCE200Z4201N3VEN:simple/-tPPCE200Z7260N3VEN:simple	Selects target processor
N/S	The N in the target processor name shall be replaced with S for software FPU
-g3	Add debug information to the executable
-Xdebug-local-all	Emit debug information for unused local variables
-Xdebug-local-cie	Generate a local Common Information Entry (CIE) for each unit.
-Xdebug-struct-all	Disable debug optimization of type information
-Xdebug-dwarf2	Generate DWARF 2 debug information
-D<cpu_name>	Define a preprocessor symbol for MCU
-DTURN_ON_CPU0	Mandatory define for the boot core(Z4_0)
-DTURN_ON_CPU***X***	Define for turning on different CPUs. ***X*** should be replaced with the desired CPU to be turned on.
-DSTART_FROM_FLASH	Mandatory define when flash target is used

9. Acronyms

Acronym	Description
EAR	Early Access Release
JRE	Java Runtime Environment
EVB	Evaluation board
PAL	Peripheral Abstraction Layer
RTOS	Real Time Operating System
PEx	Processor Expert Configurator
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
28-Apr-2017	1.0	Initial version for EAR 0.8.0	Iulian T.
15-Jun-2017	1.1	Updated known integration issues	Iulian T.
28-Jul-2017	1.2	Update for EAR 0.8.1	Rares V.
14-Dec-2017	1.3	Update for EAR 0.8.2	Cezar D.
28-Mar-2018	1.4	Update for BETA 0.9.0	Cezar D.
19-Jul-2018	2.0	Update for RTM 1.0.0	Cezar D.
17-Sep-2018	2.1	Update for BETA 1.9.0	Cezar D.
11-Dec-2018	3.0	Update for RTM 2.0.0	Cezar D.