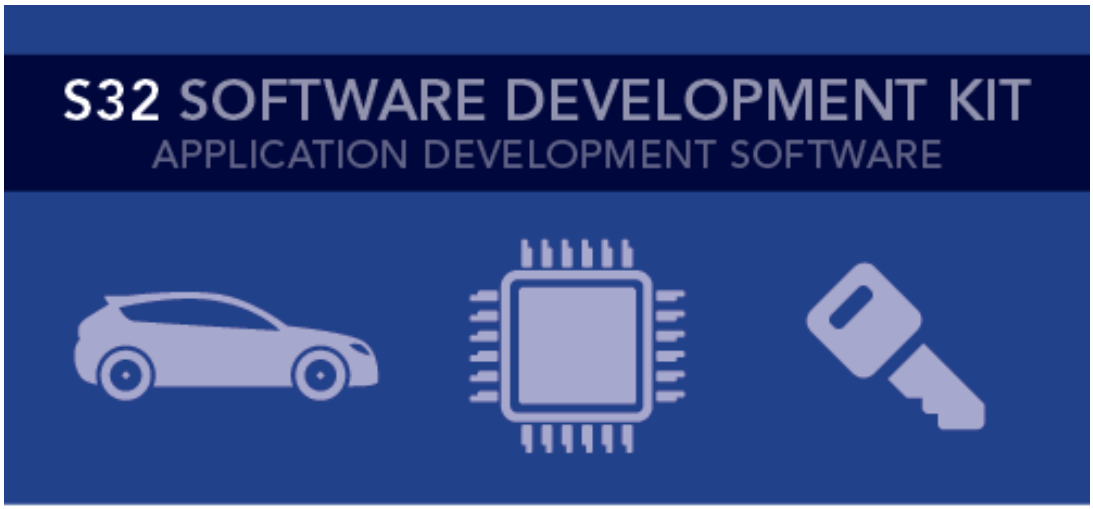

S32 SDK for Power Architecture Release Notes

Version 3.0.0 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, MPC577xB-E-C, MPC574xR, 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 Drivers

EQADC:

- Added DEV_ASSERT to check on chip ADC clock frequency is in allowed range.

ESCI:

- Added separate callback methods for TX/RX required for UART_PAL over ESCI support.
- Improved driver manual, code and example documentation.

OSIF:

- Added feature to allow configuration of PIT channel in OSIF baremetal mode by user.

OSIF, PIT, FreeRTOS:

- Added validation when enabling PIT channel to check if the timer is already configured and running.

EIM, ERM:

- Improved all available code documentation.

PINS:

- Added DEV_ASSERTs for validating input parameters.
- Restricted configuring of input buffer in PINS_DRV_SetInputBuffer to allow only the configuration of the pins that are routed from different modules.

SBC, PSI5

- Support was removed from this release, will be included in the next service release.

SDADC:

- Added SDADC_DRV_CalibrateDataSet function to support calibrating the converted data copied via DMA.

2.2 Examples

FreeRTOS:

- Added MPC5777C and MPC5746R FreeRTOS examples.

TIMING_PAL:

- Added MPC5777C and MPC5746R timing_pal examples.

2.3 Fixed from BETA 2.9.0

Component	Description
adc_pal	On ADC PAL over EQADC SW triggered groups could not be re-started after ADC_StopGroupConversion was called
adc_pal	On ADC PAL over EQADC, hardware triggered groups with more than 4 conversions, were executing only on the first occurrence of the trigger
adc_pal	On ADC PAL over EQADC, HW triggered conversion groups with more than 4 conversions in the group were not working correctly when re-enabling after ADC_Deinit() or ADC_DisableHardwareTrigger().

adc_pal	On MPC5777C ADC PAL over SDADC the notification callback only occurred on the most recent instance which called ADC_StartGroupConversion
adc_pal	On ADC PAL over EQADC, callback notification was being called after each conversion complete, instead of after the completion of the last conversion in the group.
adc_pal, ctu	On ADC_PAL_TYPE_ADC_SAR_CTU, when multiple hardware triggered groups were enabled simultaneously and compiler optimizations enabled, some groups were not executing on each occurrence of the trigger. CTU_DRV_DisableGeneralReload() was not waiting for GRE bit to be cleared, which was creating a race condition for some clock and compiler optimization settings.
adc_pal, eqadc	EQADC_DRV_Reset function was not resetting all on chip ADC registers
can_pal	CAN_PAL over MCAN would allow configuring an invalid tx mailbox number.
can_pal	CAN PAL over MCAN - wrong verification of RX FIFO number.
can_pal	CAN PAL over MCAN - a DEV_ASSERT checking for module instance number validity was wrong.
can_pal	CAN_PAL - CAN_SetBitrate function would not work when used over MCAN.
can_pal, flexcan	The CAN_PAL configuration component allowed incorrect time segments for flexible datarate.
can_pal, mcan	The time segments computed in the configuration component for input clock higher than 2MHz were wrong.
clock_manager	CLOCK - frequencies range for ADCSD instances were not checked.
clock_manager	CLOCK - default frequency for MOTC_CLK was incorrect.
clock_manager	CLOCK - CLOCK_DRV_GetFreq function would return wrong values for several input clocks.
clock_manager	CLOCK - On MPC5746R, PLL1 sources lacked IRCOSC.
clock_manager	CLOCK - ENET_RMII_CLK was not supported on MPC5746R.
clock_manager, power_manager	CLOCK - On MPC574xG the program would hang after switching power to SAFE mode.
cpu	CPU configurator did not properly change startup and linker files
cpu	CPU_EXT_FAST_CLK_HZ was not defined correctly for MPC5746R
cpu	SystemCoreClockUpdate contained dead code
cpu	SPR531 for IVOR35 was not initialized
cpu	Macros from startup.h were interpeted wrongly as being placed in sdata instead of data
dspi	Dspi driver code has updated DSPI_SetFrameSize function to fix the issue which can not update 32 bits frame size on slave mode with device supports extended mode
emios	EMIOS_DRV_PWM_GetCenterAlignIdealDutyCycle returned and incorrect value
emios	EMIOS_DRV_OutputDisable was not working when channels were initialized OPWMB mode with active level is HIGH
emios	False errors were reported when configuring input capture channels with internal bus counter on MPC5777C and MPC5746R

emios	EMIOS_PWM was not compiling on MPC5744B
emios, oc_pal	Output compare mode couldn't be initialized in SAOC under certain conditions
enet	ENET_DRV_TimerSetPulseWidth might not work if the channel is enabled.
eqadc	The EQADC_DRV_DoCalibration was using incorrect RFIFO index for the calibration sequence.
eqadc	The EQADC_DRV_ClearFifoStatus function was clearing all flags.
esci	Setting the receiver wakeup mode is not supported by the driver. This option was removed from PEx interface, as the configuration structure is now generated with a fixed default option.
esci	ESCI driver send and receive blocking methods no longer get stuck in timeout before reporting STATUS_ERROR if a DMA channel error is reported.
etimer	ETIMER_DRV_InitChannel function was not configuring the input filter.
examples	Empty *.lnt file was generated
examples	Some MPC5746R examples were using Z4_0 core instead of Z4_1
examples	Examples were not using C99 dialect
examples	Analog examples were not properly displayed for MPC5777C
examples	SPI_PAL examples for S32R372 and S32R274 were updated to facilitate the running of the exmples, without timing issues.
examples	EMIOS_OC examples displayed a warning on MPC5746C and MPC5746R
examples	Warnings were present in MPU_E200 examples
fccu	Configurator included RCCU features which were not support on these MPC5746C, MPC5748G, S32R274 and S32R372.
flash	STATUS_ERROR returnd code from FLASH_DRV_Suspend and FLASH_DRV_Resume was removed.
flexcan	Calling FLEXCAN_DRV_ConfigRxMb for a mailbox already filled with an older frame would result in the incorrect behavior of notifying the reception of a message with payload zero.
flexcan	FLEXCAN - A new transfer could not be triggered from the callback for RX FIFO DMA complete event.
flexcan	The configuration for CAN_PAL over flexcan would not allow setting the peripheral clock as the protocol clock source.
flexpwm	Function FLEXPWM_DRV_DisableModuleInterrupts was disabling FLEXPWM CAPTURE interrupt for both channels from Interrupt Manager even though one of them was still enabled in the module.
i2c	The EQADC_DRV_ClearFifoStatus function was clearing all flags.
igf	Add extern "C" in driver code to publing C++
igf	The IGF_DRV_GetRisingFilterModeChannel was update to get status of channel 32
interrupt_manager	Interrupt priority defaults to "1" on all C55 platforms. Any call to Enable/Enable_MC API assigns the IRQ to the current/given cores and removes it from others. Any call to Disable/Disable_MC API removes the IRQ from the current/given cores.

mcan	The default configuration returned by MCAN_DRV_GetDefaultConfig function would return time segments for a different bitrate than the one specified (500 kbps) for an input clock higher than 2MHz.
mcan	The MCAN configuration component would allow setting more ID filters than possbile on hw.
mcan	MCAN_DRV_AbortTransfer function did not check if the message buffer number was out of range.
mcan	MCAN_CheckIdFilter function did not work properly in case of classic filters.
mcan	The MCAN driver TX functionality would not work as expected after a driver deinitialization and re-initialization sequence.
mcan	The reception of a message would clear the interrupt flags for all other messages, resulting in only one frame being received at a time.
mcan	MCAN_DRV_RxFifoBlocking function did not return STATUS_ERROR when the Rx FIFO feature was disabled for corresponding FIFO.
mcan	MCAN_DRV_Init function did not configure the number of dedicated rx/tx transmit buffers properly, if at on of the two was equal to zero.
mpu	MPU_E200 component did not cast linker symbols to uint32_t
rtc	Duplicated if-else condition in driver.c and hw_access.c
sdadc	SDADC wraparound mode was not working correctly after re-initialization
srx	The driver checks for out of range FIFO sizes when performing DMA transfers.
timing_pal	Disabled stop in debug mode for TIMING PAL over PIT and STM, because timer was not restarting after debug mode.
zipwire	ZIPWIRE - status of DMA channels used for zipwire was not correctly updated, resultin in errors reported on subsequent transfers over same channel.

3. Software Contents

3.1 Drivers

- ADC_SAR
- BCTU
- CLOCK MANAGER
- CMP
- CPU
- CRC
- CTU
- CSE
- DSPI
- EDMA
- EIM
- EMIOS
- ENET (FEC)
- EQADC
- ERM
- ESCI
- ETIMER
- FCCU
- FLASH
- FLEXCAN
- FLEXPWM
- HEADER
- HSM (SHE FIRMWARE V.1.0.5)
- I2C
- IGF
- INTERRUPT MANAGER
- LINFLEX (UART)
- MCAN
- OSIF
- MPU
- MPU_E200
- PASS
- PHY
- PINS
- PIT
- POWER MANAGER
- RTC_API
- SAI (I2S)
- SDADC
- SEMA42
- SMPU
- SRX
- STM
- SWI2C

- SWT
- TDM
- USDHC
- WKPU
- ZIPWIRE

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 10.0.1

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_pal_eqadc	Shows the usage of the ADC_PAL over EQADC
	adc_pal_sdadc	Shows the usage of the ADC_PAL over SDADC
	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.
	cse_keyconfig	Configures CSE non-volatile keys
	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.
	eim_injection	Shows the usage of EIM driver
	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.
	eqadc	Shows the usage of EQADC driver
	erm_report	Shows the usage of ERM driver
	esci_transfer	Shows the usage of ESCI driver
	etimer	Shows the usage of the ETIMER module
	etpu_pwm	Shows the usage of the ETPU 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_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
	input_glitch_filter	Shows the usage of the IGF driver
	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
	mcan	Shows the usage of MCAN driver.
	mpu_e200_memory_protection	Shows the usage of MPU_E200 driver.
	mpu_memory_protection	Shows the usage of MPU driver.
	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
	sdadc_swtrigger	Shows the usage of the SDADC driver
	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_echo	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

	zipwire_master	Shows the usage of the zipwire driver, configured as LFAST master – works in conjunction with zipwire_slave.
	zipwire_slave	Shows the usage of the zipwire driver, configured as LFAST slave – works in conjunction with zipwire_master.
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)
- SPC5777C - 3N45H
- SPC5775B
- SPC5775E
- SPC5746R - 1N83M
- SPC5745R
- SPC5743R

The following processor reference manuals have been used:

- MPC5748G RM Rev. 6, 10/2017
- MPC5746C RM Rev. 5, 10/2017
- MPC5744P RM Rev. 6.1, 10/2017
- S32R372 RM Rev. 3.2, 11/2018
- S32R274 RM Rev. 4, 05/2018
- MPC5746R RM Rev. 6.2, 06/2017
- MPC5777C RM Rev. 8.1, 09/2018

The following errata documents were taken into consideration:

- Errata MPC5748G_0N78S.pdf Rev 2 01/2018
- Errata MPC5744P_1N15P.pdf Rev 04/2018
- Errata MPC5746C_1N84S.pdf Rev. 2, 01 2018
- Mask Set Errata for S32R274 Mask 2N58R: S32R274_2N58R.pdf Errata Rev 2
- Mask Set Errata for S32R372 Mask 0N36U: S32R372_0N36U.pdf
- Errata Rev 1 MPC5777CRM Rev. 8.1, 9/2018
- Errata MPC5746R_1N83M Rev. 2.8 10/2018
- Errata MPC5777C_3N45H Rev. 10/2018

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
- Daughter Card MPC5777C-516DS Rev D
- Daughter Card MPC5746R-252DS Rev A

6.3 Compiler and IDE versions:

- GCC E200 VLE GNU Compiler 4.9.4
  - 20160726 bld=1607 rev=gceb1328 (1 February 2019)
  - included in S32DS for Power Architecture 2017 R1
- Green Hills Multi 7.1.4 / Compiler 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

- 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

CLOCK

- The clock driver does not set the ***Flow through disable*** bit in PRAMC when the system clock frequency is above 80 MHz; the workaround is to manually set this bit in the application code for proper operation at high frequencies.
- For some PLL multiply/division factors, the *clock_manager* configuration component does not show warnings for values out of bounds; PLL setup should be done considering the constraints defined in the reference manual for each parameter.

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.

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.

EQADC

- Digital filter for external triggers cannot be bypassed

FCCU

- For S32R274, S32R372, S32V234, MPC5777C and MPC5746R devices: When injecting a fake fault to transition FCCU from Normal to Alarm then to Fault, the time configured for Alarm to Fault transition must be lower than the FOSU time. Otherwise, FOSU timeout might expire and it will trigger a CPU reset.
- For MPC5777C: The NCF 43 was present if Error Input Pin is pulled down to low signal level.
- When configuring FCCU in debug mode on MPC5746R, it will not initialize properly. Workaround: Set .debugEnable field to false.

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.

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. Workaround: The user must make sure to manually clear the LDOK bit in this situation using the dedicated function: *FLEXPWM_DRV_ClearLDOK.*

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/TransmitStatus*, although all data is successfully transferred.

MPU

- MPU driver will not report violation on the EBI Region for MPC5777C.

OC_PAL over EMIOS

- Using of internal buses, or bus B, C, D, E with channel 0, 8, 16, 24, respectively, or bus A with channel 23, or bus F with channel 22, must configure period value with the maximum available value of counter register.

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.

- FXOSC clock source must be turn on in all mode except SAFE mode when CMU is enabled. Before entering SAFE mode, the divider of clock source monitor in CMU should be reset. If the clock condition is not true, CPU will be reset when user switches 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.

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.

SDADC

The SDADC flags for data valid and watchdog crossover, do not trigger corresponding eTPU channels.

SRX

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

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).

TDM

- TDM driver is not supported on Rainier platform.

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 0-1 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
-fno-common	Allocates uninitialized global variables to a section and initializes them to zero at program startup
-msdata=eabi	Enables RAM and ROM SDA with default threshold is 8 byte
-mlra	Use local register allocation
-D__cpux_boot_addr__=<address>	Optional define when specific boot address is used. X should be replaced with the desired CPU to be updated.

Table 0-2 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
-Wl,--defsym,__sram_base_addr__=<address> -Wl,--defsym,__sram_size__=<size>	Optional define when specific RAM base address and size is used (if RAM available)
-Wl,--defsym,__flash_base_addr__=<address> -Wl,--defsym,__flash_size__=<size>	Optional define when specific Flash base address and size is used (if Flash available)
-Wl,--defsym,__local_dmem_base_addr__=<address> -Wl,--defsym,__local_dmem_size__=<size>	Optional define when specific Local Data Memory base address and size is used (if Local Data Memory available)
-Wl,--defsym,__local_imem_base_addr__=<address> -Wl,--defsym,__local_imem_size__=<size>	Optional define when specific Local Instruction Memory base address and size is used (if Local Instruction Memory available)

Table 0-3 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 0-4 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
-sda=8	Enables the Small Data Area optimization with threshold is 8 byte
-D__cpux_boot_addr__=<address>	Optional define when specific boot address is used. X should be replaced with the desired CPU to be updated.

Table 0-5 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
-D__sram_base_addr__=<address> -D__sram_size__=<size>	Optional define when specific RAM base address and size is used (if RAM available)
-D__flash_base_addr__=<address> -D__flash_size__=<size>	Optional define when specific Flash base address and size is used (if Flash available)
-D__local_dmem_base_addr__=<address> -D__local_dmem_size__=<size>	Optional define when specific Local Data Memory base address and size is used (if Local Data Memory available)
-D__local_imem_base_addr__=<address> -D__local_imem_size__=<size>	Optional define when specific Local Instruction Memory base address and size is used (if Local Instruction Memory available)
--preprocess_linker_directive_full	The C preprocessor preprocesses linker directives files

Table 0-6 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 0-7 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
-Xno-common	Allocates uninitialized global variables to a section and initializes them to zero at program startup
-Xsmall-data=8	Set size limit for “small data” variables is 8 byte
-Xsmall-const=8	Set size limit for “small const” variables is 8 byte
-D__cpux_boot_addr__=<address>	Optional define when specific boot address is used. X should be replaced with the desired CPU to be updated.

Table 0-8 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
-MD__sram_base_addr__=<address> -MD__sram_size__=<size>	Optional define when specific RAM base address and size is used (if RAM available)
-MD__flash_base_addr__=<address> -MD__flash_size__=<size>	Optional define when specific Flash base address and size is used (if Flash available)
-MD__local_dmem_base_addr__=<address> -MD__local_dmem_size__=<size>	Optional define when specific Local Data Memory base address and size is used (if Local Data Memory available)
-MD__local_imem_base_addr__=<address> -MD__local_imem_size__=<size>	Optional define when specific Local Instruction Memory base address and size is used (if Local Instruction Memory available)

Table 0-9 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.
21-Mar-2019	3.1	Update for BETA 2.9.0	Vlad L.
24-May-2019	4.0	Update for RTM 3.0.0	Vlad L.