S32 SDK for Power Architecture Release Notes

Version 2.9.0 BETA

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 BETA and EAR quality status in terms of testing and quality documentation. BETA releases are not fully qualified and tested. BETA releases are release candidates that can be used by customer for development and qualification. It is not recommended to be used in production. EAR releases are not fully tested and contain partial feature set. It is not recommended to 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

New drivers:

- PSI5: Initial PSI5 peripheral driver support for MPC5777C – EAR quality.
- SBC FS65: Initial driver support for MPC5777C and MPC5744P – EAR quality.
- ZIPWIRE
- IGF - peripheral driver support for MPC5777C
- MPU_E200 peripheral driver support for S32R274 and S32R372
- EMIOS_QD

FreeRTOS updated to v10.0.1.

ADC_PAL:

- Added support for ADC PAL over EQADC.
- Added support for ADC PAL over SDADC.
- Added support for multiple ADC PAL types enabled simultaneously (for platforms that support multiple types):
  - ADC PAL TYPE SAR BCTU and ADC PAL TYPE SDADC on MPC5746R;
  - ADC PAL TYPE SDADC and ADC PAL TYPE EQADC on MPC5777C.

CPU:

- Added support for changing the CPU in configurator. After selecting another CPU from Components Library, the configurator will add the proper linker, startup files and CPU define. When selecting a new device, the desired core must be selected from the CPU component.

  The debug configurations must be manually updated after the CPU/core is changed.

  High care must be taken when a CPU with different core type is added. In this case, the target core in toolchain settings must be updated too with the new core type.

  Linker files will be added in the projects with the following names: **linker_flash.ld** and **linker_ram.ld**. The linker names must be updated in projects created with a previous version of S32 SDK.
- Added support for calling custom function before main() is called. To achieve this *CUSTOM_INIT* symbol must be defined in assembler preprocessor define as follows *CUSTOM_INIT=custom_func*, where *custom_func* is the target function
- Modified data and bss initialization mechanism. Regions that must be copied at startup or bss(zero initialized) regions are now grouped into two tables: zero_table and copy_table.
- Added support to modify flash/sram sizes and bases in linker files. The origins and dimensions of this memory sections can be defined using linker symbols: __<*region*>_*base_addr*__=<*addr*> and __<*region*>_*size*__=<*size*>, where region can be replaced with *sram* or *flash*.
- Added support to specify secondary cores entry points via C preprocessor defines.
- Implemented code for all core exceptions.
- Added MPC5746R_176.
- Added support for sdata, sdata2 and sbss sections

EMIOS:

- Changed return value of the function *EMIOS_DRV_PWM_SetLeadingEdgePlacement* from *void* to *status_t*.
- Function EMIOS_DRV_PWM_SetCenterAlignIdealDutyCycle is now private.
- Added set duty cycle and set period use corresponding to function *EMIOS_DRV_PWM_SetDutyCycle* and *EMIOS_DRV_PWM_SetPeriod* for some mode:
  - EMIOS_MODE_OPWMCB_TRAIL_EDGE_DEADTIME_FLAGX2;
  - EMIOS_MODE_OPWMCB_LEAD_EDGE_DEADTIME_FLAGX1;
  - EMIOS_MODE_OPWMCB_LEAD_EDGE_DEADTIME_FLAGX2.
- Added const in the input parameter for some functions.

EQADC:

- Added new EQADC driver functions: *EQADC_DRV_DoCalibration*, *EQADC_DRV_SetCfifoOpMode*, *EQADC_DRV_GetCfifoOpMode*, *EQADC_DRV_InvalidateCfifo*, *EQADC_DRV_FlushRfifo.*

ETIMER:

- Added *coChannelVal* member in the configuration structure.

FLEXPWM:

- Added fault protection functionality with the following functions to use the feature: *FLEXPWM_DRV_SetupFaultProtection*, *FLEXPWM_DRV_SetFaultFilterPeriod*, *FLEXPWM_DRV_SetFaultFilterCounter*, *FLEXPWM_DRV_EnableFaultGlitchStretch*, *FLEXPWM_DRV_DisableFaultGlitchStretch*, *FLEXPWM_DRV_EnableFaultInterrupt*, *FLEXPWM_DRV_DisableFaultInterrupt*, *FLEXPWM_DRV_ClearFaultFlags*, *FLEXPWM_DRV_GetFaultFlags*, *FLEXPWM_DRV_SimulateFault*.
- Added a new function for cleating the LDOK bits: *FLEXPWM_DRV_ClearLDOK.*

OSIF:

- Each core uses different PIT channel, for improved multicore support.

PWM_PAL:

- Added deadtime support over eMIOS.

sBoot:

- Integrated sBoot RTM 1.0.1 for MPC5744P.

SDADC:

- Added DMA support for transferring conversion results.
- Added new functions:
- *SDADC_DRV_GetRawConvDataFifo*, *SDADC_DRV_SetOutputSettlingDelay*.
- Updated functions: *SDADC_DRV_DisableEvents*, *SDADC_DRV_EnableInterruptEvents*, *SDADC_DRV_ConfigConverter*.
- Removed function *SDADC_DRV_SetConvDataValidEvent* because it is covered by *SDADC_DRV_EnableInterruptEvents*.

2.2 Examples

ADC_PAL:

- Added examples for MPC5777C: *adc_pal_eqadc* and *adc_pal_sdadc.*

EEE:

- Added examples for MPC5777C and MPC5746R.

EMIOS:

- Added example for MPC5746R: *emios_pwm*, *emios_ic*, *emios_oc*.
- Added example for MPC5777C: *emios_pwm*, *emios_ic*, *emios_oc.*

EQADC:

- Added example for EQADC.

ETPU:

- Added example for eTPU on MPC5746R.

IC_PAL:

- Added examples for MPC5777C and MPC5746R.

IGF:

- Added *input_glitch_filter_MPC5777C* example.

MCAN:

- Added example for MCAN.

PSI5:

- Added example for MPC5777C.

SBC FS65:

- Added example for MPC5777C and MPC5744P.

ZIPWIRE:

- Added master/slave examples for ZIPWIRE.

2.3 Fixed from RTM 2.0.0 and EAR 1.8.0

Component	Description
Adc	ADC_SAR watchdog interrupt flags were not supported
cmp	The same configuration structure name was generated for all CMP instances. This generated a compile error when multiple CMP instances were used.
clock_manager	The default clock manager configuration for S32R/MPC574XC/G issued a warning.
clock_manager	On MPC574xG, no warning was shown when an invalid value for the FXOSC was configured.
clock_manager	On S32R274, PIT clock could not be configured (was read-only).
clock_manager	On MPC5744P, the frequency returned by CLOCK_SYS_GetFreq function for FLEXCAN and FLEXRAY peripheral clocks was wrong.
clock_manager	On MPC5746R, eTPU peripheral clock gating was not working as expected.
clock_manager	On MPC5777C and MPC5746R, some invalid warnings could appear in the CMU tab of the clock manager configuration component.

CPU	Some errors were present when MPC5775B/E and MPC5745R/MPC5743R were compiled
CPU	Some warnings were present when creating a project with MPC5743R
CPU	Build fail for projects that have use printf function with DIAB compiler.
cse	The sreg output parameter value of the CSE_GetID function was not correct.
dma	Fixed incorrect DMA channel assignment for SIPI_CH3
eee	The EEE_DRV_DeleteRecord() function has added the new enum in the eee_request_type_t which the request is to delete immediate record.
emios	Nesting Level for eMIOS_PWM driver was above 4
emios	Nesting Level for eMIOS_IC driver was above 4
emios	Initial CNT register value in EMIOS OPWFMB was not correct
emios	EMIOS_DRV_PWM_SetPeriod function did not support lead edge deadtime
emios	Warnings were shown on EMIOS_IC on MPC5777C and MPC5743R when DEV_ASSERT was disabled. Configurator for emios_ic was not creating a valid default configuration.
emios	Information about MPC5746C EMIOS instance number in the doxygen files was incorrect
emios	Output signal was not Low level when Initial Output compare in EMIOS_OUTPUT_ACTIVE_DISABLE mode.
emios	EMIOS_DRV_EnableAllChannelClk did not re-enable EMIOS channel clock after they were disabled
emios	EMIOS driver caused build failure when the configuration structures were configured as read-only(const)
emios, ic_pal	Counter channel did not start after call IC_StartChannel function
eqadc	EQADC_DRV_Reset() was not resetting CFTCRx and not flushing RFIFOs
eqadc	"The PEx component was generating a single command in the conversion command array, no matter how many elements are configured in GUI "
eqadc	EQADC_DRV_Init() was not clearing the status flags from configuration commands issued by the function
eqadc	DMA virtual channels were released in EQADC reset function, triggering DEV_ASSERT during re-initialization of EQADC
esci	Changed callback method type from esci_callback_t to uart_callback_t in order to be consistent with other UART-specific drivers.
esci	Fixed issue with driver triggering IVOR1 when calling ESCI_DRV_Init with the clock source disabled
esci	The break character parameter has been removed from the ESCI_DRV_Send/SendBlocking methods
esci, uart_pal	The driver no longer allows for baud rate updating during a transfer.

etimer	Output polarity was not matching configuration of inverted polarity when calling ETIMER_DRV_ForceOutputLogicLevel
examples	crc_checksum example was duplicated in the doxygen documentation
examples	Doxygen documentation was missing some connections for EVB board on emios_ic examples for MPC574xG/C devices
examples	EMIOS_IC example was not running properly in RAM configuration on MPC5777C
examples	Doxygen documentation was not correct for emios_oc_mpc5777c
examples	Doxygen documentation was not correct for emios_pwm_mpc5777c
examples	LED was not configured correctly in emios_oc_mpc5746r example
examples	LEDs were not configured properly in wdg_pal_interrupt
examples	LinflexD instance was not correct in fccu_fault_injection_s32r274 and fccu_fault_injection_s32r372
examples	SPT was enabled for fccu_fault_injection_mpc5746r/ mpc5777c/cse_keyconfig_mpc5777c examples
examples	One warning about clock was present memory_protection_pal_mpc5746r
fccu	Constraint for Maximun NCF timeout was not correct when using the FCCU configurator on MPC574xP devices
fccu	FCCU_DRV_DisableFault returned STATUS_SUCCESS even though FCCU was in FAULT state
fccu	FCCU could not enable RCCU
flash	"The fail address is not correct when using API getaddressfail. It occurs with low_block_16K on MPC574xR because the wrong address mask of low block_16k in the group."
flash	"The fail address is not correct when using API getaddressfail. It occurs with low_block_64K on MPC57XG,MPC574XC because the wrong address mask of low block_16k in the group."
flexcan	Driver would access unitialized memory if the transfer payload was greater than the payload configured for the module.
flexcan	For MPC5748G and MPC5746C the selfwake-up without pretended networking feature was removed because it is not supported by hardware.
flexcan	Nesting level for FLEXCAN_DRV_SendBlocking function was above the accepted threshold.
flexcan	Driver did not clear MB RAM, which could trigger the module to enter Freeze mode on parts with ecc memory detection.
FreeRTOS	On MPC5777C and MPC5746R PIT timer was not running in debug mode for OSIF and FreeRTOS
header_file	Updated base address for HFIF and IGF modules.
header_file, sdadc	Fixed header issue addressing ETPU trigger sources for SDADC
i2c	Bus busy is not checked in case the previous transfer ended with repeated start. The problem was that no other transfer could be initialized in case the previous

	transfer ended with repeated start, because the master is keeping the bus busy until stop is generated.
oc_pal	Fixed issue that made OC_EnableNotification invalid for channel > 4
oc_pal	Fixed issue onFlexPWM with toggle on match mode not correctly setting the GPIO
oc_pal	Fixed OC_StartChannel behavior when using only some of the channels.
oc_pal	When OC_PAL is used over eMIOS, in PEX, the "Period value" field has no functionality while "Counter bus" field is configured as "Local counter" or "Global counter".
phy	PHY_GetState returned active state even when the PHY was powered down.
phy	The value of the OUI field returned by PHY_GetID was incorrect.
pins	Pins hysteresis was default enabled for GPIO output pins
pins	Pins PEx Input Buffer could not be disabled on ADC pins resulting in increased operating current
pins	External interrupt could not be disabled when using overrun interrupt
pit	PIT was using PIT RTI clock sources for all channels, instead of the correct source for regular channels.
pit	PIT_DRV_Reset() was not reseting MCR register for some instance.
pwm_pal	Complementary channel for PWM_PAL over FlexPWM was not initialized
pwm_pal	PWM_OverwriteOutputChannels function was not returning STATUS_UNSUPPORTED when used over FLEXPWM
pwm_pal	Invalid condition in DEV_ASSERT for PWM_PAL in PWM_Init
sdadc	SDADC component presented SDADC peripheral instances that were not present on the hardware
sdadc	Fixed wrong peripheral base address for ETPU_CCR
srx	Enabled missing workaround for errata issue on MPC5777C / MPC5746R.
srx	Errata behavior in case of spurious ERR_NUM_EDGES_ERR fixed to comply with the errata workaround.

3. Software Contents

3.1 Drivers

DRIVER	QUALITY LEVEL
ADC_SAR	BETA
BCTU	BETA
CLOCK MANAGER	BETA
CMP	BETA
CPU	BETA
CRC	BETA
CTU	BETA
CSE	BETA
DSPI	BETA
EDMA	BETA
EIM	BETA
EMIOS	BETA
ENET (FEC)	BETA
EQADC	BETA
ERM	BETA
ESCI	BETA
ETIMER	BETA
FCCU	BETA
FLASH	BETA
FLEXCAN	BETA
FLEXPWM	BETA
HEADER	BETA
HSM (SHE FIRMWARE V.1.0.5)	BETA
I2C	BETA
IGF	BETA
INTERRUPT MANAGER	BETA
LINFLEX (UART)	BETA
MCAN	BETA
OSIF	BETA
MPU	BETA
MPU_E200	BETA
PASS	BETA

PHY	BETA
PINS	BETA
PIT	BETA
POWER MANAGER	BETA
PSI5	EAR
RTC_API	BETA
SAI (I2S)	BETA
SDADC	BETA
SEMA42	BETA
SMPU	BETA
SRX	BETA
STM	BETA
SWI2C	BETA
SWT	BETA
TDM	BETA
USDHC	BETA
WKPU	BETA
ZIPWIRE	BETA

3.2 PAL (BETA QUALITY)

- 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
- SBC_FS65 (EAR)

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_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.
	psi5_async_rx	Shows the usage of the PSI5 in Async RX mode
	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
	sboot	Shows the integration and usage of sBoot
	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. 2.7 05/2018
- Errata MPC5777C_3N45H Rev. 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
- 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 (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

ADC_PAL

- For ADC_PAL over EQADC results are not written beyond the first set of result buffer.

CAN_PAL

- On MPC5777C, MPC5775E and MPC5775B, the CAN PAL configuration component does not work if peripheral clock is selected as protocol engine clock source.
- When CAN PAL is used over MCAN, *CAN_SetBitrate()* function does not work; the bitrate should be set through the configuration structure when calling *CAN_Init()*.

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

- After *EQADC_DRV_DoCalibration* is used, conversion results for commands with sign enabled will not be calibrated accurately
- Streaming mode is not working correctly when trigger source is selected as ETPU or EMIOS

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.

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.*

FreeRTOS

- Missing FreeRTOS examples for MPC5777C and MPC5746R

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.

MCAN

- After initializing MCAN driver with a number of TX message buffers greater than zero, it cannot be reinitialized with zero message buffers for TX.
- After initializing MCAN driver with RX FIFO feature enabled, it cannot be reinitialized with the same feature disabled, as the FIFO receive functions will return invalid error codes.
- Multiple receive operations performed on different message buffers in parallel may result in loss of data, in case data for one buffer is completely received before the driver clear the flag for previous buffer data.

MPU_E200

- MPU Core module will not detect an access violation for unaligned instruction execution on GreenHills and Windriver Diab compilers. The instructions need to be aligned to 8 bytes in order for the MPU to detect the violation.

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.

OSIF

- Current bare metal implementation uses the last *n* PIT channels (where n is the number of cores available 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.
- 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.

PSI5

- The driver does not generate notifications for the following events: [PSI5_EV_TX_DATA_OVR, PSI5_EV_TX_DATA_RDY]. The user has to use *PSI5_DRV_IsTransmitReady()* to poll for these events.

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.
- If UART_PAL is used over ESCI, only one callback method can be installed for both transmission and reception.

SBC FS65

- When *FS65_LIN_SetMode* is used to switch from listen only to normal mode the transmitter is not enabled immediate and first LIN frames can be missed. The workaround is to insert a delay after calling this function.
- When FCCU is not connected to SBC default configuration must be changed like in this example:
  sbc_fs651_InitConfig0.fssmReg.io23Fs = FS65_R_FS_IO_23_FS_NOT_SAFETY,
  otherwise fault event is detected and functionality of the driver will be affected;

- *FS65_Init()* returns STATUS_ERROR when called at wake up after LPOFF from LDT;
- *FS65_ReleaseFSx()* has no effect on FS0B pin on MPC5777C, the line remains low.

SDADC

- An additional DMA request generated when using watchdog threshold crossover event or data FIFO full DMA

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).

ZIPWIRE

- ZIPWIRE – On MPC5777C in high speed mode, subsequent read transfers on multiple bytes may cause global CRC error.

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
-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 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
-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 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
-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 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
-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 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
-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 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
-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 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.
21-Mar-2019	3.1	Update for BETA 2.9.0	Vlad L.