



S32 SDK for Power Architecture Release Notes

Version 0.9.0 BETA



Contents

1.	DESCRIPTION	3
2.	NEW IN THIS RELEASE	4
2.1	<i>Examples</i>	4
2.2	<i>Drivers</i>	4
2.3	<i>Middleware</i>	4
2.4	<i>PAL</i>	4
3.	SOFTWARE CONTENTS.....	5
3.1	<i>Drivers</i>	5
3.2	<i>PAL</i>	5
3.3	<i>RTOS</i>	6
3.4	<i>Middleware</i>	6
4.	DOCUMENTATION.....	7
5.	EXAMPLES.....	8
6.	SUPPORTED HARDWARE AND COMPATIBLE SOFTWARE	10
6.1	<i>CPUs</i>	10
6.2	<i>Compiler and IDE versions</i>	10
6.3	<i>Debug Probes</i>	10
7.	KNOWN ISSUES AND LIMITATIONS.....	11
7.1	<i>S32 Design Studio integration</i>	11
7.2	<i>Drivers</i>	11
7.3	<i>Examples</i>	13
8.	COMPILER OPTIONS	14
8.1	<i>GCC Compiler/Linker/Assembler options</i>	14
8.2	<i>GHS Compiler/Linker/Assembler options</i>	16
8.3	<i>DIAB Compiler/Linker/Assembler options</i>	18
9.	ACRONYMS	20
10.	VERSION TRACKING.....	21



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 Power MPC574x-B-C-G and MPC574x-P microcontrollers.

All software included in this release has BETA quality level in terms of features, testing and quality documentation, according to NXP software release criteria.

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:

- Email to support@nxp.com
- NXP Community <https://community.nxp.com/>



2. New in this release

2.1 Examples

- Added new driver examples for: ic_pal, oc_pal, security_pal, i2c_pal(for MPC5744P), flexpwm_pwm, mpu_pal_memory_protection, phy_autoneg, pwm_pal, sdhc_freertos, srx_fast_dma, usb_cdc_lwip

2.2 Drivers:

- PHY
- HSM: Driver was tested using HSM Security Firmware MPC574XX RTM 1.0.3. Functionality is guaranteed only if the same firmware version is flashed on the end-user device.
- Power_manager: Added function POWER_SYS_GetPreviousMode return previous mode where the chip jump from latest mode to current mode
- Flash_C55: Added functions FLASH_DRV_EnableCmdCompleteInterrupt, FLASH_DRV_DisableCmdCompleteInterrupt, enable and disable the command complete interrupt is generated when an program erase complete or suspend program, erase. User must enable the flash interrupt in interrupt controller when uses this function.
- DSPI: The support for RX overflow and TX underflow was added in driver and the driver won't be stuck in STATUS_BUSY if one of these flags is triggered.
- DSPI and SPI_PAL: drivers support DMA mode on MPC574xP devices.
- ADC C55: during calibration, ADC driver will set Half Bus Speed as clock and restore after calibration finish.

2.3 Middleware

2.4 PAL

- MPU_PAL
- PWM_PAL
- SECURITY_PAL
- IC_PAL: The input capture is available over the ETIMER.
- OC_PAL: All PAL functions are available on Panther over the ETIMER
- IC_PAL: IC_PAL is integrated with ETIMER. Currently it can be used over ETIMER and eMIOS.
- OC_PAL: OC_PAL is integrated with ETIMER. Currently it can be used over ETIMER and eMIOS.
- ADC_PAL: Added support for MPC574xP devices
- I2C_PAL: available over the SWI2C.



3. Software Contents

3.1 Drivers

- ADC
- CLOCK MANAGER
- CPU
- DSPI
- EDMA
- EMIOS
- ENET
- FLASH
- FLEXCAN
- HEADER
- INTERRUPT MANAGER
- LINFLEXD (UART)
- OSIF
- PINS
- PIT
- RTC
- SEMA42
- STM
- SWT
- WKPU
- BCTU
- CMP
- CRC
- FCCU
- HSM (SHE)
- I2C
- POWER MANAGER
- PASS
- PHY
- SAI (I2S)
- SMPU
- TDM
- USDHC
- CTU
- FLEXPWM
- ETIMER
- SRX
- SWI2C

3.2 PAL

- UART_PAL
- SPI_PAL



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

3.3 RTOS

- FreeRTOS version 9.0.0

3.4 Middleware

- TCP/IP
- EEE
- SDHC
- FATFS
- 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.



5. Examples

Type	Name	Description
Driver examples	adc_swtrigger	Shows the usage of the ADC MPC574xx
	bctu_trigger	Shows the usage of BCTU cross triggering
	cmp_dac	Shows how to use CMP with the internal DAC
	dspi_master	Shows the usage of the DSPI/SPI module
	enet_ping	Shows the usage of the ENET module by implementing an application which responds to ping requests.
	i2c_transfer	Shows the usage of the I2C driver in both master and slave modes.
	linflexd_uart	Shows the usage of LINFlexD_UART driver in interrupt based mode.
	sai_transfer	Shows the usage of the SAI driver in both master and slave modes.
	uart_pal	Shows the usage of UART PAL over LinFlexD.
	crc_checksum	Calculates CRC using the peripheral driver for multiple standards.
	edma_transfer	Show multiple usage scenarios of DMA.
	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.
	power_mode_switch	Transitions the MCU into all available power modes.
	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.
	swt_interrupt	Shows the usage of the SWT
	wkpu_interrupt	Shows the usage of the WKPU 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
	pit_periodic_interrupt	Shows the usage of the PIT
	rtc_alarm	Shows the usage of the RTC
	stm_periodic_interrupt	Shows the usage of the STM
	adc_pal	Shows the usage of the ADC_PAL
	can_pal	Shows the usage of the CAN_PAL
	i2c_pal	Shows the usage of the I2C_PAL
	i2s_pal	Shows the usage of the I2S_PAL
	spi_pal	Shows the usage of the SPI_PAL



	interrupt_control_multicore	Shows the usage of the Interrupt Manager in a multicore environment
	pass_lock_unlock	Shows the usage of the PASS module
	wdg_pal_interrupt	Shows the usage of the WDOG_PAL
	timing_pal	Shows the usage of the TIMING_PAL
	ctu_trigger	Shows the usage of the CTU module
	enet_loopback	Shows the usage of the ENET module configured in loopback
	swi2c_master	Shows the usage of the SWI2C
	etimer	Shows the usage of the ETIMER module
	ic_pal	Shows the usage of the I2C_PAL
	oc_pal	Shows the usage of the OC_PAL
	security_pal	Shows the usage of the SECURITY_PAL
	i2c_pal	Shows the usage of the I2C_PAL
	flexpwm_pwm	Shows the usage of the PWM functionality of FlexPWM
	mpu_pal_memory_protection	Shows the usage of the MPU_PAL
	phy_autoneg	Shows the usage of the PHY module with autonegociation
	pwm_pal	Shows the usage of PWM_PAL
	srx_fast_dma	Shows the usage of SRX in DMA based mode
Demos	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
	flexcan	Shows the usage of FlexCAN driver configured as both bus master and slave.
	freertos	Shows the usage of the FreeRTOS MPC574xx.
	eeeprom_emulation	Shows basic use cases of the EEPROM Emulation middleware.
	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.
	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
	sdhc_freertos	Shows the usage of SHDC with FATFS over FreeRTOS



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)

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

6.2 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

6.3 Debug Probes

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



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_C55

- For MPC5744P, examples use an ADC frequency lower than the minimum value stated in DataSheet. The minimum value is 20MHz.

ADC_PAL

- When a group (either sw or hw) without callback (callback pointer set to NULL) is enabled/started after another group with callback enabled, the NULL callback pointer is dereferenced from the interrupt handler registered for the end of group conversion

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.

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 occurred during the read operation from flash, the driver only support to get the failing address in the C55FMC_ADR register.

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

- Wake Up interrupt is not issued upon detecting a wake-up event when Pretended Networking / Self Wake Up feature is enabled.

HEADER FILES

- Not all interrupts are declared in the header file.



HSM

- *Cancel* command erases the RAM key slot; therefore, blocking operations ended with timeout render the RAM key slot unusable for subsequent commands (issue was fixed starting from security firmware version RTM 1.0.4).

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.

I2S_PAL

- DMA transfer is not working

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

- Initialization function does not clear the contents of the RX FIFO in DMA mode; therefore, some invalid data may be caught from the bus while reception is disabled.
- In DMA mode, bytesRemaining parameter is not always 0 after calling `LINFLEXD_UART_GetReceive/TransmittStatus`, although all data is successfully transferred.
- In DMA mode, after successfully receiving a buffer, the FIFO may not be empty, which results in corrupted data on subsequent receive operations.

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.
- Generating configuration for external interrupt is not working.
- Some pins do not appear in configuration file - GPI pin/ ADC.

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.

PIT

- The `PIT_DRV_Deinit` will not disable the timer and one PIT channel. One PIT channel is used by either FreeRTOS or OSIF (in baremetal case). For applications that do not use FreeRTOS or OSIF baremetal the timer is left running.

**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.

SAI

- DMA transfer is not working

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)

WKPU

- Wakeup unit is not successfully run when enable WIPER[25] and WIPER[26] (one of them or both) in STANDBY mode

7.3 Examples

- WKPU example runs in FLASH only if the reset button is 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=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/-mhard-float	Select between hardware and software FP
-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_CPUX	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=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/-mhard-float	Specify the type of FP to be used (Z2 core supports only software FP)
-g3	Generate debug information
-O1	Optimization level
-DTURN_ON_CPU0	Mandatory define for the boot core(Z4_0)
-DTURN_ON_CPUX	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	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
-sda=0	Enables the Small Data Area optimization with a 0 threshold
-dual_debug	Generates the DWARF debugging information in the object file
-O1	Optimization level
-D<cpu_name>	Define a preprocessor symbol for MCU
-DTURN_ON_CPUX	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



Table 8-6 GHS Assembler options

Option	Description
-cpu=ppc574xcz4204/-cpu=ppc574xgz210	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
-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_CPUX	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
- -tPPCE200Z210N3VES:simple /-tPPCE200Z4VEN:simple	Selects target processor
-Xdialect-c99	Use C99 standard
-Xsmall-data=8	Set size limit for “small data” variables
-Xsmall-const=8	Set size limit for “small const” variables
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_CPUX	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
- tPPCE200Z210N3VES:simple/ tPPCE200Z4VEN: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
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/- tPPCE200Z4VEN: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_CPUX	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



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.