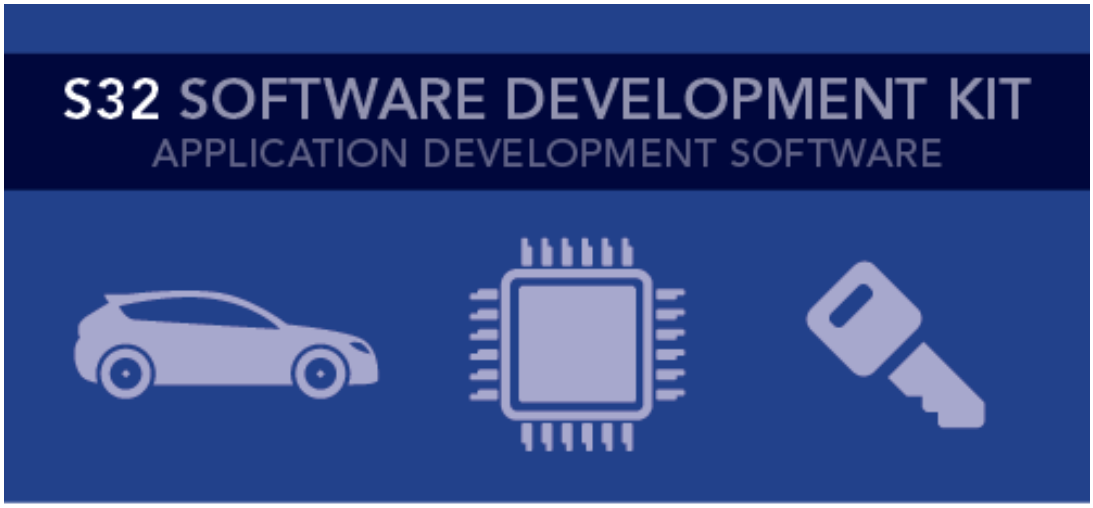

S32 SDK for Power Architecture Release Notes

Version 0.8.1 EAR

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 Power MPC5748G and MPC5746C microcontrollers.

All software included in this release has EAR 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: FATFS, TCP-IP, EEPROM Emulation, HSM, BCTU, CMP, I2C, SAI, UART PAL, EDMA, FCCU, POWER, SEMA42(Multicore), SMPU, EMIOS IC and EMIOS OC

2.2 Drivers:

2.2.1 New drivers:

- BCTU
- CMP
- CRC
- FCCU
- HSM (SHE)
- I2C
- POWER MANAGER
- PASS
- SAI (I2S)
- SMPU
- TDM
- USDHC

2.2.2 Updates on existing drivers:

- ADC – Added API for presampling and DMA support
- CLOCK - Implemented support for PLL clock source, peripheral clock gating and clock source enablement in all modes.
- DSPI – Updated Callback type to comply with other SPI drivers and added DMA support
- EDMA – Merged HAL and PD
- EMIOS – Added function for clearing flags, simplified PWM API, renamed header and source files
- ENET - Implemented unicast and multicast filtering features. Added transmission confirmation function. Simplified transmission and reception API and added mechanism for reusing Rx buffers for Tx.
- FLASH – Added function for retrieving the address where an operation has failed
- FLEXCAN – Implemented support for Self Wake Up and Transceiver Delay Compensation features
- INTERRUPT – Added multicore support
- LINFLEXD UART - Updated Callback type to comply with other UART drivers
- PINS – Added API for Pin interrupt and GPIO manipulation
- WKPU – Split API for different distinct configurations(Interrupt/NMI/Reset)

2.3 Middleware

- TCP/IP
- EEE
- SDHC
- FATFS

2.4 PAL

- SPI
- UART

2.5 RTOS

- Updated FreeRTOS version to 9.0.0

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
- SAI (I2S)
- SMPU
- TDM
- USDHC

3.2 PAL

- UART_PAL
- SPI_PAL

3.3 RTOS

- FreeRTOS version 9.0.0

3.4 Middleware

- TCP/IP
- EEE
- SDHC
- FATFS

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.

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
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.

	eeprom_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.

6. Supported hardware and compatible software

6.1 CPUs

- MPC5748G - 0N78S (Cut 3.0)
- MPC5746C - 1N84S (Cut 2.1)

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

- MPC5748GRM Rev. 5, 12/2016
- MPC5746CRM Rev. 4.1, 02/2017

6.2 Compiler and IDE versions:

- GCC E200 VLE GNU Compiler 4.9.4
    - 20160726 (release_gd8b6c20_build_Fed_ELe200_ML0)
    - included in S32 Design Studio for Power Architecture v1.2
- 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

CLOCK

- Interface clock for FLEXCAN0 is not configurable. FS80 is used as interface clock.
- When component is added in Design Studio project, internal script is not run. Modify at least one of the properties and the internal script will run.

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. NMI handler (IVOR1_Handler) must save the context so that it will work in all possible cases.

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.

FCCU

- FCCU Module used on Bistable Configuration for EOUT signals and enable the EOUT Signals for Non Critical Fault will drive the signals correctly in Normal Configuration and in case of triggered fault source with enabled EOUT signals will reset the chip

FLASH

- The driver is designed to support asynchronous for program or erase operation.
- 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.
- The driver does not support an interrupt request when the high voltage operation is set.

FLEXCAN

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

FREERTOS

- The symbol 'USING_OS_FREERTOS' must be added manually to project settings for projects using FreeRTOS to work

HEADER FILES

- Not all interrupts are declared in the header file.

HSM

- MAC verification function does not work (VERIFY_MAC command always returns true)
- The busy flag may remain asserted for several clock cycles after command completion, so the subsequent command returns STATUS_BUSY
- HSM driver is sending SHE commands to the MPC574x security firmware. The firmware must be flashed and the HSM module must be enabled through DCF scripts before using the HSM driver

I2C

- After using the I2C_DRV_MasterAbortTransferData function, the application should de-initialize and re-initialize the driver by calling I2C_DRV_MasterDeinit and I2C_DRV_MasterInit, otherwise subsequent transfers will fail.
- The same procedure should be applied in the case of a blocking transfer ending in STATUS_TIMEOUT error code.
- Driver status is not correctly updated after a slave transmission in non-listening mode

LINFLEXD UART

- LINFLEXD_UART_DRV_GetBaudRate function ignores the fractional part of the prescaler, which causes a deviation in the computed baudrate
- 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

OSIF

- Current bare metal implementation uses PIT channel 15 for internal timing. Applications should not attempt to use this channel

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

- When flash power down control component select 'IN POWER DOWN MODE' in PEx, program does not run with flash power down mode.

SAI

- DMA transfer is not working

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 in pressed after the download to the target.
- Some examples do not use Pins component and driver
- Some examples may display warning messages with unresolved includes
- EDMA example may remain trapped into DEV ASSERT due to LINFLEXD known issue.

8. Compiler options

Please refer to the compiler options used in the examples.

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.