	RadarSDK	
RELEASE MANAGEMENT	Revision 1.0	Page i of 15

RSDK 1.2.1 patch release for S32R274 and S32R372 Release Notes


	RadarSDK	
RELEASE MANAGEMENT	Revision 1.0	Page ii of 15

Table of Contents

RSDK 1.2.1 for S32R274 and S32R372 Release Notes	i
1 Read Me First	1
1.1 Environment Requirements	1
1.2 Support information	1
2 Release Description	3
2.1 List of Modules	3
2.2 What's New.....	3
2.3 List of Supported Features	4
2.4 Known Issues and Limitations	8
3 Upgrading from the Previous Release	10
4 Appendix A: Document History	11
5 Appendix B: Release History	12

1 Read Me First

We are glad to announce 1.2.1 patch release of **RadarSDK** for S32R274 and S32R372. The purpose of this release is to provide software enablement for the NXP S32R274 and S32R372 SPT accelerators and MR3003 and TEF810x radar transceivers.

This release can be used by any party interested in accessing SPT and radar transceiver functionalities from the Power Architecture Z4/Z7 host processor of S32R274 and S32R372, in OS-free environment.

If you are new to RadarSDK, please start by getting familiarized with the User Manual (html format, located in /Docs folder [RSDK User Manual](#)) and example applications (located in /Apps folder: /Apps/1RF_4Antennas_demo and /Apps/SPT_Example). For details about applications provided, please consult [Sample Apps UserGuide](#).

A **Troubleshooting Guide** is also included, on main page of **RSDK User Manual**.

1.1 Environment Requirements

The following tools are used for building and executing the binaries in the software package:

Software:

- S32 Design Studio for Power Architecture, Version: 2017.R1 Build id: 171018 (gcc for PPC version 4.9.4 20160726) with all online updates.
- Lauterbach Trace32 PowerView for PowerPC™, Version: R.2018.09.000104140, Release Sep 2018 (64 bit)
- Matlab R2016b 64bit
- Matlab Signal Processing Toolbox v7.3 and Model-Based Design SPT Toolbox v1.0_b171214, used by the Matlab bit-exact model – applicable to S32R274 only

Hardware:

- Host PC running Windows 10
- S32R274/S32R372 RRU EVB development board with
 - (for SPT_Example demo application) PS32R274KAMM, chip cut 1.2, 2N58R
 - (for both SPT_Example and 1RF_4Antennas demo application) PS32R372SAMMM, 0N36U
- (for 1RF_4Antennas demo application)
 - RFBeam Microwave MR3003_RD-NXP device integrating S32R274 (chip cut 1.2) and MR3003 (ES2+RF) Radar Transceiver, or
 - NXP TEFL810x DCC integrating BEST3-S1 ES3 and S32R274 (chip cut 1.2)
- Debugger
 - Lauterbach Power Debug Interface and DEBUG-MPC5XXX-AUTO debug cable
 - (for 1RF_4Antennas demo application) PEmicro USB Multilink/Multilink FX

1.2 Support information

Technical Contact:

Please contact your friendly sales representative or access <https://www.nxp.com/> support page.

When reporting problems (we thank you for that!), please include:

- RSDK release ID
- tools used for building the RSDK artefacts
- target hardware platform (e.g. S32Rxxx and radar front-end revision/versions, board revision)
- any dependencies or modifications you may have added or applied to create your environment

2 Release Description

2.1 List of Modules

DELIVERABLE	LOCATION	STATUS
Example applications	/Apps/SPT_example	Unchanged
	/Apps/1RF_4Antennas_demo	Updated
CPU algos example library	/CPU_algos/PPC_algo	Unchanged
		Unchanged
User manual	/Docs/Doxygen/RSDK_User_Manual/index.html	Updated
SPT driver	/SPT/SPT_driver	Unchanged
SPT example kernels	/SPT/SPT_kernels	Unchanged
SPT bit-exact matlab kernel models	/Tools/SPT_bitexact_model	Unchanged
Radar Front-End (RFE) abstraction driver	/RFE_abstract/RFE_driver	Updated
	/RFE_abstract/CSI2	Unchanged
	/RFE_abstract/SPI	Unchanged

A detailed description of release content and incremental changes is given in 2.3 and 2.2.3.

2.2 What's New

2.2.1 New Features

RFE abstract:

- Support for the TEF810x ES3 front-end. See chapter Upgrading from the Previous Release for details on changes from bare low level driver of TEF810X front end.
- Provided API for monitoring errors of TEF810X front-end

All modules:

- Compile S32R274 libs and apps with diab for PPC 5.9.6.4

This package contains interface and functional changes in respect to the previous release. Section 2.3 includes the detailed feature list. Please inspect the RSDK User Manual to introduce yourself to these new features.

2.2.2 Regressions

There are no known regressions in this release.

2.2.3 Fixed Issues

ID	Module	Description
RSDK-1681	RFE_Abstract	(pertains to TEF810X only) SPI_TEF810X_xferFifoMode() does not write into its ret_cmd parameter

ID	Module	Description
RSDK-1682	RFE Abstract	Corrupted chirp data from TEF810x units
RSDK-1683	RFE_Abstract	(pertains to TEF810X only when invoked via abstraction layer) RFE Abstract does not call Tx Gain Calibration
RSDK-1684	RFE_Abstract	(pertains to TEF810X only) GPIO interaction has been disabled
RSDK-1685	RFE_Abstract	(pertains to TEF810X only) hal_es3_ISM_Init_Act_16() is NOT called (#if 0 in the source code)
RSDK-1364	RFE_Abstract	(pertains to TEF810X only) Missing CRC check on SPI transmission

2.3 List of Supported Features

Module	Features	Introduced in release	Testing approach
SPT driver	Provide an interface to access the SPT functionalities (implemented via SPT kernels), exposed on Power Architecture based host processor. Single core usage. OS free (bareboard). Blocking and non-blocking operating modes.	0.1 (0.4.0)	Review. Unit testing (no hardware). Integration testing on hardware.
	API function for reporting SDMA statistics: RsdkSptCommand()	0.4.1	Review. Unit testing (no hardware).
	API function for controlled disabling and reinitialization of SPT hardware: RsdkSptStop()	0.4.1	Review. Unit testing (no hardware). Integration testing on hardware.
	New SPT Driver library for S32R274 e200 Z7 core	0.4.2	Review. Integration testing on hardware.
	Support for SPT of S32R372	1.1.0	Review. Integration testing on hardware

Module	Features	Introduced in release	Testing approach
SPT kernels library	Example (functional and optimized) SPT assembly code to perform various flavors or range and Doppler FFT operations using the SPT accelerator. Visible to host PowerPC processor via SPT driver: Range 512 and Doppler 128 point FFT Range 256 and Doppler 256 point FFT	0.1 (0.4.0)	Review. Validation using matlab non-bitexact models.
	Now SPT kernels use WAIT instructions to sync with system events. New kernels: Magnitude calculation and non-coherent combining. 2D peak search using fixed array(input) for threshold. 3 rd FFT on azimuth axis, magnitude calculation. “3D” Peak search based on max calculation on DOA axis and 2D peak search. Fixed array for threshold.	0.4.1	Integration testing on hardware.
	Use same buffer for Doppler FFT input/output Add support for real-time SDMA data acquisition in SPT Range kernel Peak Search kernels produce histogram for threshold calculation Add binary data files containing twiddles and window coefficients Bit-exact matlab equivalents for all SPT kernels	0.4.2	Review. Unit testing
	2D and 3D Peak search kernels split into Non Coherent Combining/3DFFT and Peak Search kernels. New 128 chirps, 1024 samples/chirp kernels with CP4D PDMA compression New Digital BeamForming + Direction of Arrival kernels(with/without compression) New 256 chirps, 256samples/chirp, TD-MIMO kernels New API scaling parameter for Range and Doppler kernels(without compression) Speed optimization for Doppler/NonCoherentCombining/3DFFT/Peak Search kernels For more details consult SPT Kernels API	0.9.0	Review. Validation using matlab non-bitexact models. Integration testing on hardware.

Module	Features	Introduced in release	Testing approach
	Macro provided in SPT kernel API to return kernel size value.	1.0.0	Review.
	Support for SPT on S32R372	1.1.0	Review. Integration testing on hardware
RFE Abstract	Provides an interface to access the functionalities of a generic radar front-end device. Supporting: - NXP MR3003 ES2+RF front-end - SPI and CSI2 support for configuration and data exchange between front-end and S32R274 - Simple chirp modulation, no chirp changes during a frame	0.4.2	Review. Integration testing on hardware.
	Support for multiple chirp shapes per frame Z7 core support TD-MIMO support for MR3003 For more details consult RFE abstraction Doc	0.9.0	Review. Integration testing on hardware.
	Support for CSI2 and SPI on S32R372 Support for soft reset of MR3003 frontend Support for TEF810x radar transceiver (early access release level)	1.1.0	Review. Integration testing on hardware
	Support for TEF810x ES2 radar transceiver NB: support for ES2 has been removed starting with 1.2.1 release, when ES3 has been added	1.2.0	Review. Testing on hardware.
	Support for MR3003 ES3.1 radar transceiver	1.2.0	Review. Testing on hardware.
	MR3003 DFMEA improvements: - interrupts are configured and activated at device initialization - added function to report device status - in case a device fault is detected, the driver goes into fault state and only certain operations are permitted	1.2.0	Review. Testing on hardware.
	CSI driver monitors and reports all HW error related interrupts and offers support for spurious interrupts	1.2.0	Review. Testing on hardware.

Module	Features	Introduced in release	Testing approach
	Support for TEF810X ES3 radar transceiver (non-SPICE), synchronized with version 5.2.2 of its low level driver	1.2.1	Review. Smoke testing on hardware.
SPT Example application	Demonstrates the integration of SPT Driver and kernels library. Baseband radar samples acquisition emulated by the CPU reading data files from host PC through the T32 debugger.	0.1 (0.4.0)	Review. Testing on hardware.
	Integrate 2D and 3D Peak Search SPT kernels	0.4.1	Review. Testing on hardware.
	Integrate new SPT Driver and Kernel features	0.4.2	Review. Testing on hardware.
“1RF 4Antennas demo” application	Sample app demonstrating how to integrate all RadarSDK modules on the RFBeam Microwave MR3003_RD-NXP device	0.4.2	Review. Testing on hardware.
	Sample app add support for BeamForming and DOA and CFAR threshold computation. ETH Communication now is made on Z4 using a new feature called proxy intercom and radar processing is done on Z7 For more details consult Sample Apps UserGuide .	0.9.0	Review. Testing on hardware.
	TD-MIMO sample application was integrated in this demo. Also, configuration file was added for easier usage. For more details consult Sample Apps UserGuide .	1.0.0	Review. Testing on hardware.
	Support for S32R372 and TEF810x	1.1.0	Review. Smoke testing on hardware
	Added RTP streaming support (experimental feature)	1.2.0	Review Smoke testing on hardware
	Monitoring TEF810X errors	1.2.1	Review Smoke testing on hardware

Module	Features	Introduced in release	Testing approach
	Basic platform setup code needed for sample applications.	0.4.1	Review. Testing on hardware.
	Add Interrupt priority and Core ID to the interrupt handler registration function.	0.4.2	Review. Testing on hardware.
	Improved platform setup Easy to use linker files Multicore easy to use and robust mechanism: proxy intercom For more details consult Application support tools	0.9.0	Review. Testing on hardware.
	The folder structure was modified to assure a better and clear view over source files.	1.0.0	Review. Testing on hardware.
	Improved TFTP client robustness and latency. Added ARP support. Added experimental RTP streaming support.	1.1.0	Review. Smoke testing on hardware.
	Added experimental UART driver	1.1.0	Review. Smoke testing on hardware.

2.4 Known Issues and Limitations

ID	Affected Modules	Description
RSDK-1232	SPT Driver	Software workaround not implemented for hardware issue S32R274_2N58R_Errata_Rev1 e10459: "Spurious CS_AHB_ERR can be generated in the SPT_DMA_ERR_STATUS register"
RSDK-1425	RFE_Abstract	(pertains to TEF810X only) In certain error cases, calling reinitialization function for frontend leaves code in a never-ending loop. Do not reinit if error received on first attempt, without hard-resetting frontend unit.
RSDK-1458	Sample apps	Section .bss_shared is not zeroed during startup code. Variables placed in this section have unpredictable content. Workaround: have application memset these variables to zero.
RSDK-1520	Sample apps	flash_rchw section size too large in linker file
RSDK-1571	RFE_Abstract	(pertains to TEF810X only) Polling on FreqReadValue.bits_st.DoneFlag in hal_es3_cafc_SubBandSelectBinarySearch() does not feature timeout. Code can get stuck running while statement.

RSDK-1658	RFE_Abstract	(pertains to TEF810X only) glue_Toc() does not measure time accurately and it introduces 10usec delay
RSDK-1673	RFE_Abstract	(pertains to TEF810X only) Missing timeout for while statement on hal_es3_ISM_FIT. If DynamicPowerControlEnable.bits_st.ChirpDyPowerControl == code will get stuck running on while statement
RSDK-1674	RFE_Abstract	(pertains to TEF810X only) Software architecture and detailed design artifacts are not in synch with current code state
RSDK-1675	RFE_Abstract	(pertains to TEF810X only) There exist functions with complexity (CCM) higher than 30
RSDK-1729	Sample apps	(pertains to TEF810X only) 1RF_4Antennas_demo monitors radar frontend status each frame, with no corrective action being taken. Check frontend user manual for description of the problem and possible actions.
	RFE_Abstract	(pertains to TEF810X only) LVDS and CIF interfaces of low level driver are not maintained (only CSI2 is). No MISRA analysis has been performed on these.
	RFE_Abstract	(pertains to TEF810X only) OPALKELLY source code branch is not maintained/MISRA analyzed. Thus, deci parameter of chip_ISM_Init() is hardcoded to invalid value of 0, as it is not used on S32R274 during Init_Act_16(), but only on OPALKELLY branch
	RFE_Abstract	(pertains to TEF810X only) RFBIST code is not invoked

3 Upgrading from the Previous Release

If already using the previous RSDK release, then the following items need to be updated:

- In your build and execution environment: the tool versions described in section 1.1.
- In your custom application source code:
 - update names of the **libraries** and **API header files** for SPT Driver, Kernels and common code
 - update names of changed API functions and data structures (check headers in `<Module>/api/` and `/common/include` for changes)
 - configure the new API fields and function calls

If you have used bare TEF810X low level driver until now, following changes are required at your side:

- mind limitations marked with “*pertains to TEF810X*” mentioned in chapter Known Issues and Limitations
- LVDS and CIF interfaces of low level driver are not maintained (only CSI2 is)
- Non-const global variables have been replaced/united into a state variable named `gRfeDriverPersistentMem` of `rfeDriverPersistentMem_t->rfeTef810XPersistentMem_t` type. See `/RFE_abstract/RFE_driver/include/rfe_driver_state.h` and `rfe_driver_state.c`
- Direct use of STM0 hw timer has been replaced by a callback approach that is injected via `pDrvState->delayUsCb`. For an example, see `RsdkDelayUs()` definition in `/platform_setup/src/PPC/platform_setup_basic.c` and mind proper set of STM_0 reference clock to 60MHz.
This affects implementation of timeout mechanism applied to setting/getting various RF frontend conditions by reading/writing config registers by: introducing 10usec delay and approximating timeout values instead of having a hw timer dedicated for/directly affected by this. Later implementation will fix this limitation (see RSDK-1658 above).
- Glue layer: SPI and CSI2 drivers for the S32R2xx device are required and radarSDK does provide such drivers. As an example, one can mimic `1RF_4Antenna_demo` sample application present in the `Apps/` folder of the release package.
Proper pin configuration must be performed during initialization, radarSDK provides such a routine in `Siul2Config()` of `platform_setup/src/PPC/platform_setup_basic.c`. Mind to `#define BOARD_TEF810X_BEST3`
- Initialization of the frontend unit requires special runtime configuration for CSI2 and SPT. As an example, one can mimic `1RF_4Antenna_demo` sample application present in the `Apps/` folder of the release package.
- `HAL_ERRORCODE` is no longer used – it has been replaced by propagation of erroneous conditions via return value of functions, up to interface functions
- Value of `MAKE_CSI2_OUTPUT_UNSIGNED` `#define` has been set to 0 (using signed samples)
- `READY_INT_CFG` is currently not `#defined` for S32R274, resulting in ready int signal not being configured
- `TEF810X_CFG_SPI_CMD_TX_CRCCHK_ENABLE` and `TEF810X_CFG_SPI_CMD_RX_CRCCHK_ENABLE` `#defines` are enabled for ES3

4 Appendix A: Document History

Version	Date	Author	Description
1.0	10Apr2019	Cristian Macarascu	New in this release, known issues and upgrade from previous release chapters

5 Appendix B: Release History

Release version	Date	Description
0.1 (0.4.0)	31Mar2017	First release. Prototype
0.4.1 “SPT”	30Jun2017	“Early Access” for SPT
0.4.2 “RRU”	29Sep2017	“Early Access” for RRU SPT and mr3003 front-end
0.9.0 “RRU”	30Jan2018	Beta release
1.0.0 “RRU”	13Jun2018	RTM release
1.1.0 “S32R274 and S32R372”	30Aug2018	RTM release with EAR support for TEF810x transceiver
1.1.1 “S32R274 and S32R372”	03Sept2018	RTM release with EAR support for TEF810x transceiver
1.2.0 “S32R274 and S32R372”	29Nov2018	RTM release with RFE Abstract changes (support for TEF810x ES2 and MR3003 ES3.1 front-ends, all error related interrupts from the MR3003 front-end and CSI2 are enabled and monitored)
1.2.1 “S32R274 and S32R372”	12Apr2019	Patch release with RFE Abstract changes (support for TEF810x ES3)