SAF85XX

RN00004

Release Notes for the SAF85xx RFE SW

Rev. 0.8.19 — 29 September 2023 Release Notes

Document information

Info	Content
Keywords	Radar, SAF85xx, RFE FW, RFE SW
Abstract	Release notes describing release contents, changes, limitations and known issues.

Revision history

Rev	Date	Description
0.8.19	29 September 2023	Release Notes for SAF85xx RFE SW EAR 0.8.19 D230929
0.8.18	23 August 2023	Release Notes for SAF85xx RFE SW EAR 0.8.18 D230823
0.8.17	21 July 2023	Release Notes for SAF85xx RFE SW CD 0.8.17 D230721
0.8.16	30 June 2023	Release Notes for SAF85xx RFE SW CD 0.8.16 D230630
0.8.15	28 April 2023	Release Notes for SAF85xx RFE SW CD 0.8.15 D230428
0.8.14	17 March 2023	Release Notes for SAF85xx RFE SW EAR 0.8.14 D230317
0.8.13	27 January 2023	Release Notes for SAF85xx RFE SW CD 0.8.13 D230127
0.8.12_HF01	17 February 2023	Release Notes for SAF85xx RFE SW HF01 0.8.12 D230217
0.8.12	16 December 2022	Release Notes for SAF85xx RFE SW CD 0.8.12 D221216
0.8.11	30 September 2022	Release Notes for SAF85xx RFE SW CD 0.8.11 D220930
0.8.10	26 August 2022	Release Notes for SAF85xx RFE SW CD 0.8.10 D220826
0.8.9 HF1	27 July 2022	Release Notes for SAF85xx RFE SW_HF01_0.8.9_D220727
0.8.9	15 July 2022	Release Notes for SAF85xx RFE SW CD 0.8.9 D220715
0.8.8	01 July 2022	Release Notes for SAF85xx RFE SW EAR 0.8.8 D220701
0.8.7	10 June 2022	Release Notes for SAF85xx RFE SW CD 0.8.7 D220610
0.8.6	24 May 2022	Release Notes for SAF85xx RFE SW CD 0.8.6 D220524
0.8.5	09 May 2022	Release Notes for SAF85xx RFE SW CD 0.8.5 D220509
0.8.4	26 April 2022	Release Notes for SAF85xx RFE SW CD 0.8.4 D220426
0.8.3	08 April 2022	Release Notes for SAF85xx RFE SW CD 0.8.3 D220408
0.8.2	15 March 2022	Release Notes for SAF85xx RFE SW CD 0.8.2 D220315
0.8.1	18 February 2022	Release Notes for SAF85xx RFE SW CD 0.8.1 D2202
0.8.0	28 January 2022	Release Notes for SAF85xx RFE SW Code Drop 0.8.0
0.4.2	03-December-2021	SAF85xx RFE SW Documentation Update 0.4.2
0.4.1	05-November-2021	SAF85xx RFE SW Hot Fix 0.4.1
0.4	15 October 2021	Initial for SAF85xx RFE SW Code Drop 0.4.0
0.2	30 August 2021	Draft for SAF85xx RFE SW Code Drop 0.3.0
0.1	28 July 2021	Draft for SAF85xx RFE SW Code Drop 0.2.0

1. Introduction

The SAF85xx is an NXP RFCMOS One-chip radar IC integrating mixed signal radar transceiver and radar processing SoC.

SAF85xx RFE SW contains:

- Definition of the RFE Abstract API
- RFE demo control application running on RFE control core
- RFE driver running on RFE control core
- RFE FW running on RFE-M7 core

Note: The RFE control core is ARM Cortex-A53 (APP-A53) or ARM Cortex-M7 (APP-M7) as supported by this release.

This document describes the release contents, changes, limitations and known issues.

The *SAF85xx RFE SW Reference Manual* is included in the release, it describes the usage (*User Manual*), the RFE Abstract API, and RFE Driver code and design.

Details on the control of RFE HW using the RFE Abstract API and the RFE parameters can be found in *RM00266 SAF85xx RFE Reference Manual*.

Refer to *SAF85xx_RFE_SW_Release_Test_Report.pdf* for details on what is functional\tested in this release.

2. Release Contents

This release (code drop) is targeting SAF85xx ES1.1 hardware. This release contains the items as per Table 1 .

Table 1: Release Contents

Item	Version
Documentation	
• SAF85xx_RFE_SW_ReleaseNotes.pdf	0.8.19
• SAF85xx_RFE_SW_Reference_Manual o User Manual o API reference o RFE Driver code reference o RFE Driver design reference	0.8.19
• SAF85xx_RFE_SW_Release_Test_Report.pdf	0.8.19
RFE firmware (RFE-M7)	
• DATA and CODE	0.8.19
RFE software (APP-A53/APP-M7)	
• RFE Abstract API	0.8.19
• RFE Driver o RFE Abstract API o RFE configuration o RFE dynamic table o RFE command stub o RFE command client	0.8.19
• AUTOSAR RFE Complex Device Driver o RFE CDD API o EB tresos Plugin	0.8.19
• RFE Basic Example App o S32DS project o Example RFE Configuration o T32 boot scripts for RFE FW and APP-A53 /APP-M7 Application • RFE AUTOSAR Basic Example App o S32DS project o Example RFE Configuration T32 boot scripts for RFE FW and APP-A53 /APP-M7 Application	0.8.19
Tools	
• RFE Configuration Tool o User input example configuration o Generated example configuration and schedule o SAF85xx RFE Config Generator User Manual	0.8.19

3. SAF85xx RFE SW Release

The release package contains the items depicted in Figure 1.

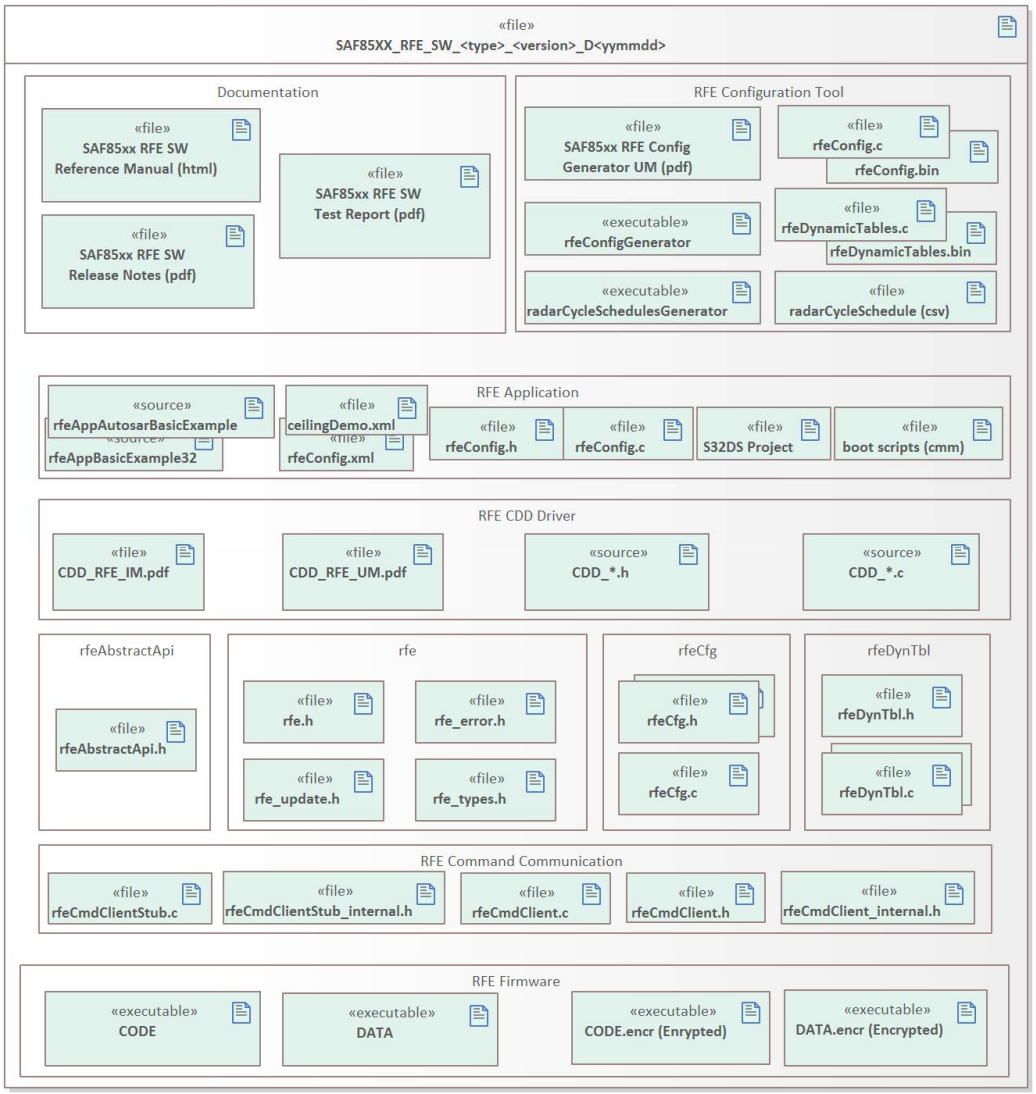


Figure 1 Release Content

3.1.1 Release Structure

The release is contained in the following folder structure.

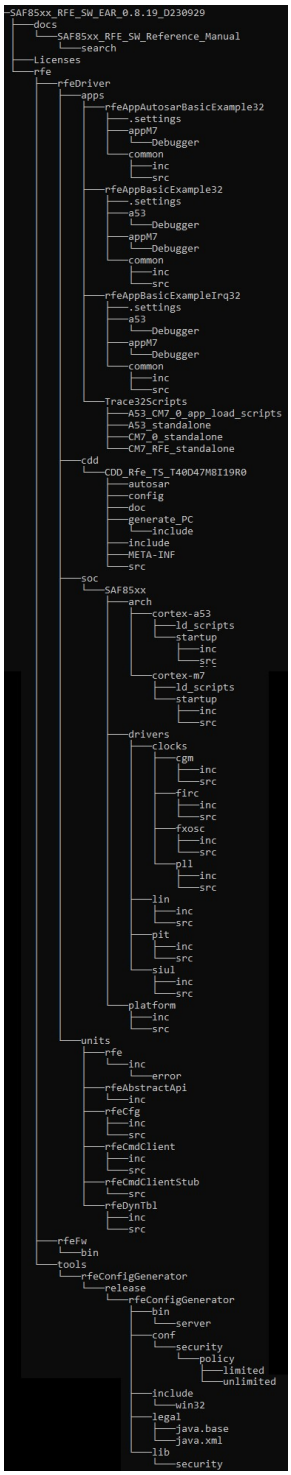


Figure 2 Release Folder Structure

3.1.2 Usage Instructions

This release replaces all previous releases.

Please unzip the content of the release to the folder structure as per Figure 2.

Please refer to *SAF85xx_RFE_SW_Reference_Manual*, chapter *RFE App Basic Example*, on how to use this SW package for integration with APP-A53/APP-M7 Application.

To access RFE SW Reference manual please run:
SAF85XX_RFE_SW_xxxx_xxxx\docs\SAF85xx_RFE_SW_Reference_Manual.html

Notes:

For the product code, NXP strongly recommends using RFE Configuration Tool to generate RFE configurations (that are going to be used in the production software) rather than using the *rfeCfg.h* utility functions to create\modify the configuration BLOB in the code Utility functions are provided for development purposes. For run-time updates of RFE parameters *rfe_updateParam\rfe_updatePush()* functions are to be used in the product code.

Also, the same recommendation should be followed for dynamic parameters, the RFE Configuration Tool should be used to create dynamic tables unless run-time updates of dynamic table are required (*rfeDynTbl.c*).

The SAF85xx RFE SW release does not contain the following items to build and run RFE Basic Example Application:

- S32 Design Studio to compile APP-A53/APP-M7 code;
- Trace32 SW (Lauterbach ICD box and license required) for booting RFE FW and RFE Application via JTAG.
- HSE Firmware, APP-A53/APP-M7 application image and RFE FW image can be booted also using (customer) boot application.

4. Changes

4.1 Change Log

4.1.1 SAF85xx RFE SW EAR 0.8.19 D230929

RFE Documentation

* Updated to reflect the 0.8.19 release content.

* RFE API profiling updated in Software Reference Manual.

RFE FW

* Heart-beat signal updated- Following are the instances when Heart beat signal will be sent from RFE M7

  After RFE Sync command is received

  At the start of radar cycle.

  After R1-faults recovery is successful..

* Rx Noise floor improvements for gain of 37db with IF frequencies less than 5MHz

  RX conversion gain control LUT/Algo updated to optimize VGA1 and VGA2 regarding Noise Figure.

* Issue related RFE Software entering FuSa fault state post functional reset fixed

  Registers used by FIT (Fault Injection Test) were not properly initialized by RFE software in case if it is re-initialized via POR_B. This was because these uninitialized registers are data-path registers that are not connected POR_B but connected to power supply-based reset. Now, the RFE software initializes these registers that are connected to power supply based reset before FIT execution, hence resolving the issue related to re-initialization of RFE software.

* LOIF LDOs must be all enabled in all mode to avoid SOA issue

  Level shifters were not properly biased when LDOs are disabled

  RFE software will keep the driver, LNA and PA LDOs ON always - with this change the DC current of all active LDOs is less than 14mA

* FuSa

  Issues related to SM19, SM20, SM21, SM22, SM64 and SM86 fixed

  FIT2 for SM55 added

  For detailed FuSa fault pass list please refer to Section 5

* Issue related to TxEnable/TxTransmissionEnable fixed

txEnable	txTransmissionEnable	Output chirp transmission
0	0	NO chirp
0	1	NO chirp
1	0	NO chirp
1	1	Chirp

Table 2: TxEnable and TxTransmission Enable

Tx was always enabled if the PR_Cal_Enable was enabled. The PR_cal_enable signal should have been enabled only during PR calibration and should have been disabled after once the calibration done. Also in the SW, txEnable input should be taken from the xml in sequenceConfig, and not from the transmissionEnable data (which is in the chirpConfig). Table 2 shows the behavior in 0.8.19. Also it is now observed that current drops by roughly 250-300mA when using txEnable = 0 per Tx Channel

RFE API

* Added rfe_testParam_copybackConfig_e to rfe_testParam_t to enable the readback of RFE config BLOB from RFE M7 TCM to the memory address which indicated in the parameter. The size of the memory to be allocated should be 1024 bytes. Also this features is for development purposes only

* rfe_error.h moved from
<Folder>\SAF85xx_RFE_SW_EAR_0.8.19_D230929\rfe\rfeDriver\units\rfe\inc to

<Folder>\SAF85xx_RFE_SW_EAR_0.8.19_D230929\rfe\rfeDriver\units\rfe\inc**error**

* Following new error codes added in rfe_error_t enumerated datatypes present in rfe_error.h

rfe_error_api_configParamCrossCheck_conflictingIoAssignment

rfe_error_api_configParamCrossCheck_notSupportedByRfeRole_e

rfe_error_api_configParamCrossCheck_conflictingCsi2TxModes_e

In enumerated data type rfe_testParam_t, new entry rfe_testParam_copybackConfig_e added at the bottom

Minor updates made in rfe_update.h and other files to resolve MISRA warnings

In rfe_paramUpdate_t enumerated datatype present in rfe_types.h a new value rfe_paramUpdate_general_chirpActiveOut_e has been added to support the Chirp Active GPIO

rfe_testParam_enableClockRetuning_e Deprecated

rfe_testParam_enableClockRetuning_e was provided to enable/disable retuning of the clock PLL to prevent it to go out-of-lock, with default being false. From this release on, the retuning is enabled by Default, no input is required from the control application . This test parameter rfe_testParam_enableClockRetuning_e has been deprecated, the clk pll retuning cannot be controlled using this. In future release this enum will be renamed as rfe_testParam_Reserved. To ensure backward compatability this enum is not yet removed

RFE Configuration

* Added new parameter in the General Section for Chirp Active signal that configures IO pins (rfe io 3 and rfe io 5) as indicated in Figure 3. This is an optional parameter and default is zero (none). If not used then this can be skipped in the XML

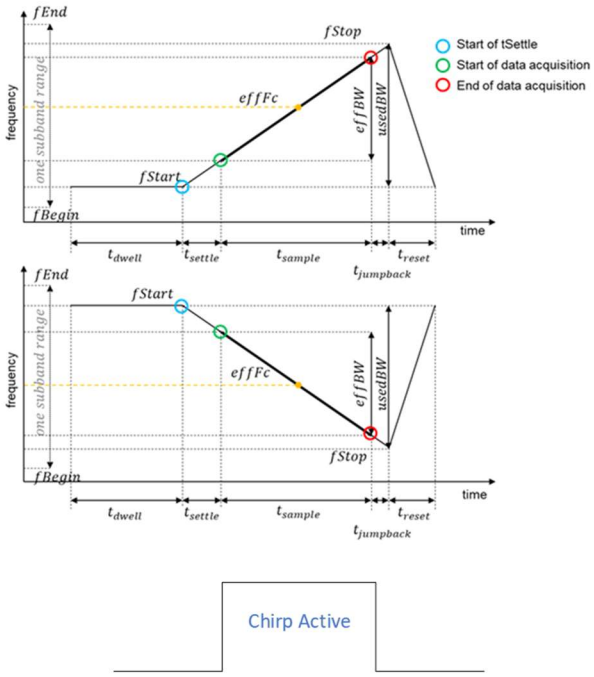


Figure 3 Chirp Active Signal

RFE Applications

* Adapted the Applications to be compatible with the updated RFE API and Configuration.

4.1.2 SAF85xx RFE SW EAR 0.8.18 D230823

RFE Documentation

* Updated to reflect the 0.8.18 release content.

* RFE API profiling updated in Software Reference Manual.

RFE FW

* With ES2 samples, the RX gain has improved by approximately 2dB. Gain compensation programmed in software has been removed from ES1.1 to ES2.

* Tx Calibration Changes

  Profile independent calibration is performed at the center frequency of all the profiles (previously done at the max frequency of radar cycle)

Profile dependent PA calibration is performed at the center frequency of profile (previously done at the max frequency) as shown in Figure 4

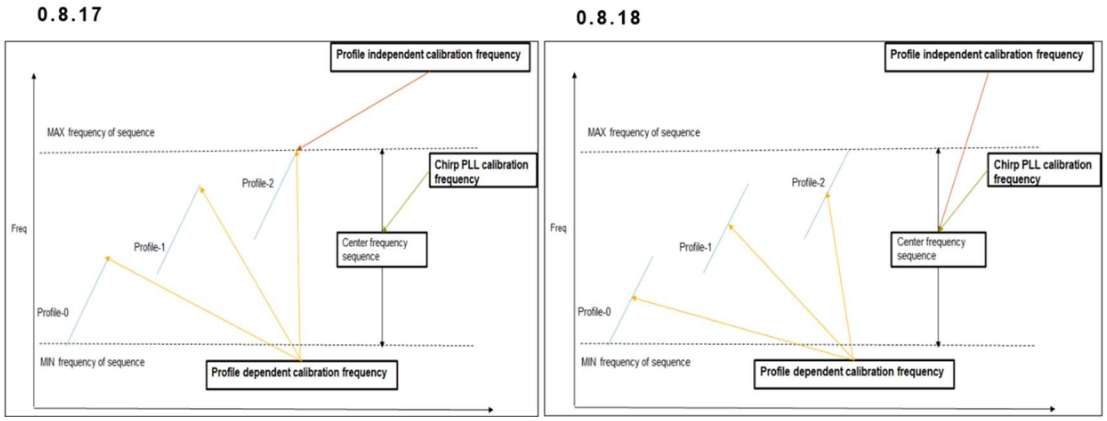


Figure 4: Figure indicating change in calibration centre frequency

Phase difference is resolved by applying constant value for Buf2B2C for all TX channels

* FuSa and BIST

  At the end of BIST slot rfeFw checks if any single bit error is detected and does a memory refresh to recover from the single bit error in DTCM

  When enabled, Heart beat signal will be sent at start of every radar cycle to A53/APP-M7 as an interrupt

  For detailed FuSa fault pass list please refer to Section 5

RFE API

* Added rfe_testParam_assertErrorNSignal_e, rfe_testParam_deAssertErrorNSignal_e and rfe_testParam_assertHeartBeatSignal_e to the enumerated data type rfe_testParam_t, which can be used in rfe_testSetParam function

RFE Configuration

* No change from 0.8.17 release

RFE Applications

* RFE CDD updated to enable AUTOSAR 21.11 integration
* Adapted the Applications to be compatible with the updated RFE API and Configuration.
* Common folder structure implemented across example apps.

* Example apps and RFE Driver are compatible with GHS Compiler version GHS MULTI 7.1.6d COMPILER 2021.1.4.

4.1.3 SAF85xx RFE SW CD 0.8.17 D230721

RFE Documentation

* Updated to reflect the 0.8.17 release content.

* RFE API profiling detailed and updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

* 38 GHz Improvement

LOI buffer activation is modified be done close to the start of the chirp sequence as possible.

* FuSa and BIST

FuSa and BIST enabled. For detailed FuSa fault list please refer to Section 5

RFE API

* Added rfe_state_waitingForCalibration_e item to rfe_state_t (at the end) to implement Cascading use cases:

* New enumerated data type rfe_role_t introduced with rfe_role_standalone_e, rfe_role_leader_e and rfe_role_follower_e

* In rfe_state_t enumerated present in rfe_types.h the enumerated data rfe_state_testContinuousWaveTransmission_e has been renamed to rfe_state_continuousWaveTransmission_e

* In enumerated data type present in rfe_types.h added new value rfe_chirpProfileIndex_bist_e at the end

* In rfe_types.h, RFE_DATA_OUT_DEST_WDMA added to enable data output over WDMA for cascading use case

* rfe_testContinuousWaveTransmissionStart() moved from rfe_error.h to rfe.h and renamed to rfe_continuousWaveTransmissionStart()

* rfe_testContinuousWaveTransmissionStop() moved from rfe_error.h to rfe.h and renamed to rfe_continuousWaveTransmissionStop()

* In rfe_type.h rfe_gpio_t has been renamed as rfe_io_t and here the default value is changed from rfe_gpio_none_e (value 8) to rfe_io_none_e (value 0) as shown in Figure 5. Please note only this change is not backward compatible.

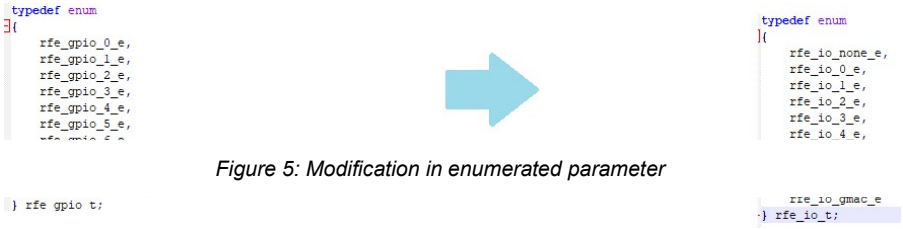


Figure 5: Modification in enumerated parameter

RFE Configuration

* Added configuration for CSI2 Pin Swap especially for LiP

* RFE Config parameter modifications in general section

Parameter Name	New Name	New Type Name	New Range	New Default in XML
radarCycleStartIoOut	radarCycleStartIoOutradarCycleStartIoOut	rfe_io_t	rfeIo0-rfeIo7/none	none
chirpSequenceActiveSignalGpio	chirpSequenceActiveIoOut	rfe_io_t	rfeIo0-rfeIo7/none	none
pdcDecimationFilter	pdcDecimationFilter	rfe_pdcDecimationFilter_t	narrow/steepNarrow/real	steepNarrow
dataOutDest	dataOutDest	rfe_dataOutDest_t	Add following bit to bit mask: RFE_DATA_OUT_DEST_WDMA	RFE_DATA_OUT_DEST_PACKET_PROCESSOR

Important note: Except for name change (GPIO -> IO), this implementation is backwards compatible

Parameter Name	Type	Size	Range	Defautl Value
radarCycleStartIoIn	rfe_io_t	1 byte	none/rfeIo0-rfeIo7/gmac	None
cascadingTriggerIo	rfe_io_t	1 byte	none/rfeIo0-rfeIo7	None
loifOutputPower	int8_t	1 byte	-3 to 6 dBm (1 dB resolution)	3dbm
wdmaOutputBufferAddress	sysMemAddress	4 bytes	0,0x33E8000-0x341FFFFF	0x33E80000
wdmaOutputBufferSizePerRx	uint32_t	4 bytes	0-4194304	65536
csi2TxDataRate	rfe_csi2TxMode_t	1 byte	adcDataRate/320/640/1280/2560 Mbps	adcDataRate

Important note: This implementation is backwards compatible

RFE Applications

* Adapted the Applications to be compatible with the updated RFE API and Configuration.

* Restructured the rfeDrivers folder to avoid duplication and bring in more commonality across example apps.

Refer to RFE SW Reference Manual for the details.

4.1.4 SAF85xx RFE SW CD 0.8.16 D230630

RFE Documentation

* Updated to reflect the 0.8.16 release content.

* RFE API profiling detailed and updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

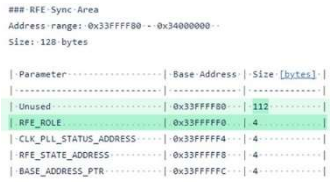
* ES2 Changes

  Current firmware adapted according to the ES2 changes. This release is no longer backward compatible with ES1.1 hardware.

* FuSa and BIST

  In the current release FuSa and BIST are disabled and not supported. All FuSa faults need to be masked in the configuration and also the BIST interval should be set to none.

* A new variable added to the reserved space of RFE Sync area. Variable required for cascading. RFE Sync Area have to be zero-initialized including non-used partition..



```
### RFE Sync Area
Address range: 0x33FFFF80 - 0x34000000
Size: 128 bytes

| Parameter                | Base Address | Size [bytes] |
|--------------------------|--------------|--------------|
| Unused                   | 0x33FFFF80   | 112          |
| RFE_ROLE                 | 0x33FFFFF0   | 4            |
| CLK_PLL_STATUS_ADDRESS   | 0x33FFFFF4   | 4            |
| RFE_STATE_ADDRESS        | 0x33FFFFF8   | 4            |
| BASE_ADDRESS_PTR         | 0x33FFFFFC   | 4            |
```

RFE API

* New API *rfe_configureInterrupt()* added to allow RFE Driver to be used with RFE Firmware interrupt support.

* With ES2, LLDO 0.9 v safety monitor from RFE digital reports fault(s) directly to RTS FCCU instead of RFE FCCU. As a consequence, two FuSa faults, fuSaFault_R2_sm41_ov_ldo_dig and fuSaFault_R2_sm41_uv_ldo_dig are removed from FuSa fault list. FuSa fault count has been reduced from 14 to 12 in rfe_error.h.

* Enumerated value added to rfe_effectiveSamplingFrequency_t enumerated data type in Chirp Profile Section.

RFE Configuration

* Enumerated value added to rfe_effectiveSamplingFrequency_t enumerated data type in Chirp Profile Section. Table 3 section sizes shown below:

Section	Amount	Size in release 0.8.15 (in bytes)	New size as per release 0.8.16 (in bytes)
metadata	1	5	8
general	1	11	44
monitorAndSafety	1	149	164
radarCycle	1	55	72
chirpSequence	8	26	36
chirpProfile	8	38	56

Table 3: Updated section sizes as per the new configuration

* RFE Config Tool adapted to support 1kB rfeCfg Structure.

* Fast-Reset Duration and Delay are moved from ChirpSequence Section to Profile Section. Fast-Reset Enable remains to be at chirp sequence level.

RFE Applications

* Adapted the Applications to be compatible with ES2 along with modifications in CMM scripts.

* Additional Example application added to demonstrate the interrupt feature.

* Restructured the rfeDrivers folder to avoid duplication and bring in more commonality across example apps.

Refer to RFE SW Reference Manual for the details.

4.1.5 SAF85xx RFE SW CD 0.8.15 D230428

RFE Documentation

* Updated to reflect the 0.8.15 release content.

* RFE API profiling detailed and updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

* FuSa Update

  Following Fit 2 Added

  SM54_CHIRPPLL_RMS VCO Detector + Amplitude Monitor

  SM62_TE_LOCKSTEP

  SM64_ATB-ADC + SW

  SM92_RFE_M7_CORE_SWT

  Safety Mechanisms validated (refer to section 5 for the list of validated FuSa Faults)

  SM98, SM204 added

* Updated RFE Init Error Codes

  Power-up and initialization of RFE is done autonomously without RFE Abstract API control. During this stage RFE is self-starting the RFE FW which will power-up and initialize all the modules. The master clock generator (CLk PLL) will be started. Upon CLK PLL lock, 0xC0DE0001 is written to RFE Sync Shared memory region (0x33fffff4).

  In case of error, 0xC0DE<internalCode> is written instead, where internalCode =

  0x10A1 => wrong 0.9[V] supply

  0x10A2 => lack of\unstable XO

  0x10A3 => wrong Clock PLL supply derrived from 0.9[V]

  0x10A4 => CLK PLL not locked

Refer to RFE SW Reference Manual for the details.

RFE API

* No Change

RFE Configuration

* No Change.

RFE Applications

* No change except version updates.

4.1.6 SAF85xx RFE SW EAR 0.8.14 D230317

RFE Documentation

* Updated to reflect the 0.8.14 release content.

* RFE API profiling detailed and updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

* FuSa Update

4 FIT2 Tests added

2x SW Register CRC (SM68, SM69),

1x HW Register CRC (SM94) and

1x FTTI WDT (SM63)

R1 auto recovery mode added - One config element "recovery mode" is introduced to rfe config to enable\disable automatic recovery when radar cycle is executed in autonomous mode

Because safety monitor sm42 was not a real safety mechanism, it has been removed. As a consequence, two safety faults have been removed from the faults list.

MCGEN/ADPLL registers integrity check - During RFE Init MCGEN/ADPLL SW configured registers are read back to check the integrity. Error in register integrity will be reports as initialization error (internal error can be read out)

Safety Mechanisms validated (refer to section 5 for the list of validated FuSa Faults)

Migrated to Arm Compiler for Embedded FuSa 6.16.2 LTS.

* di/dt

GLDO2 undershoot during PLL calibration and recalibration solved

* Fix for sporadic spurs on 5MHz and 10MHz 1st FFT (range) bins (fix same as in 0.8.12 HF01)

* Memory synchronization barriers (DSB) for race condition avoidance.

* BIST Power Management added (as per 0.8.13 debug)

* In RFE AUTOSAR CDD Driver, EB tresos plug added for importing in EB tresos Studio (S32 Design Studio configurator is not supported for RFE CDD)

* LiP support

* CMD Client \ Server optimized for speed

* PR Calibration issue resolved

Functional reset executed during every PR Calibration

Functional reset in Power ON executed only if Re-Calibration scheduled

Refer to RFE SW Reference Manual for the details.

RFE API

* In `rfe_error.h`, `RFE_FUSA_R1_FAULT_COUNT_MAX` reduced from 79 to 77

* In `rfe_error.h` in the enumeration `rfe_fuSaFault_t`, faults `rfe_fuSaFault_R1_sm42_dc_det_low_mcgen_e` and `rfe_fuSaFault_R1_sm42_dc_det_high_mcgen_e` have been removed

* In `rfe_types.h` in the enumeration `rfe_bistInterval_t`, `rfe_bistInterval_none_e` added indicating not to perform BIST in the radar cycle.

* In `rfe_types.h` under enumeration `rfe_paramUpdate_t`, new enumeration `rfe_paramUpdate_monitorAndSafety_autoErrorRecoveryMode_e` added for R1 auto recovery mode.

RFE Configuration

* FuSa updates
  - Two faults removed
  - Byte count remains the same but the masks are adjusted to reflect the removal of SM42
  - R1 Error recovery mode `"errorRecovery mode"` added.

Refer to RFE SW Reference Manual on detailed speciation of RFE BLOB.

RFE Applications

* Restructured the rfeDrivers folder to avoid duplication and bring in more commonality across example apps.

* In RFE AUTOSAR CDD Driver, EB tresos plug in added so that the CDD can be imported in EB tresos Studio

4.1.7 SAF85xx RFE SW CD 0.8.13 D230127

RFE Documentation

* Updated to reflect the 0.8.13 release content.

* RFE API profiling detailed and updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

* Calibration

  ATB-ADC SE (Single Ended) Trimming

  Chirp PLL Loop Filter Bandwidth Calibration (Chirp PLL is also validated for 4GHz Now), refer to RFE SW reference Manual for details on how to set center frequency and select VCO in reference to this calibration.

  Tx Pout Real Time Monitor (RTM) Calibration

* FuSa Update

  During RFE Initialization (FIT1), HW Fault Injection Tests are executed

  Safety Mechanisms validated (refer to section 5 for the list of validated FuSa Faults)

  R1 Recovery introduced

* BIST

  RX BIST added (Refer to the RFE BIST section of RFE SW Reference Manual for the details)

  CRC BIST added (Refer to the RFE BIST section of RFE SW Reference Manual for the details)

* RFE Initialization errors reported via RFE Sync area, the errors can be read via *rfe_testGetInternalError* function.
* Datapath bit width autoconfiguration

  RFE Configure general::pdcBitwidth parameter

  - Datapath can be set to 12 \ 14 \ 16 bit independently from Calibrations and BIST

  - Calibration and BIST will use 16bit regardless the data path bit width setting in RFE Configure
* RFE Software Secure Boot

  Release package contains encrypted binaries of RFE FW which can be used for secure boot using HSE Firmware (refer to HSE documentation)
* Large Dynamic Tables
* RFE Driver: ported to on App-M7 Bare Metal
* RFE CDD Driver: ported to App-M7 Bare Metal
* Use of separate VCs per profiles enabled

Refer to RFE SW Reference Manual for the details.

RFE API

* In the `void rfe_getFuSaFaults(uint8_t *pFuSaR1R2FaultList, RFE_ERROR_FUNCTION_PARAMETER)` message in the pay load has changed due to changes in FuSa fault flags. Refer to RFE SW Reference Manual (CMD_RSP_Format.pdf) for more details

* In the function `void rfe_getFuSaFaultStatistics(uint16_t *pR1FaultPromotedToR2, rfe_radarCycleCount_t *pRadarCycleCount, uint8_t *pFuSaR1FaultCountList, RFE_ERROR_FUNCTION_PARAMETER);` message in the pay load has changed due to changes in FuSa fault flags. Refer to RFE SW Reference Manual (CMD_RSP_Format.pdf) for more details

RFE Configuration

* FuSa updates

  - FuSa fault flag list extended to reflect additional safety mechanism, see `t` in `rfe_error.h`

  - RFE configuration BLOB fusaFaultMask `unit8_t[RFE_FUSA_R1_R2_MASK_UNMASK_FAULT_BYTE_COUNT ]` set to 12 bytes
* Calibration and BIST updates

- RFE Configuration BLOB format changed select the transmitters for which BIST will be performed. `txSelectForTxBist` changed from `unit32_t to rfe_txSelect_t`

- RFE Configuration BLOB, TX peak power detector threshold (power in the steps of 0.1 [dBm]) `txPpdThreshold` changed to `int16_t[RFE_CHIRP_PROFILES_MAX]`, with min of -90 and max of 150

Refer to RFE SW Reference Manual on detailed specification of RFE BLOB.

RFE Applications

* SoC drivers are added in /rfe/rfeDriver/ folder.

* rfeAppAutosarBasicExample32 added for demonstrating use of AUTOSAR complex device driver (CDD) RFE Driver on bare metal APP-M7

4.1.8 SAF85xx RFE SW 0.8.12 HF01 D230217

RFE Documentation

* Updated to reflect the 0.8.12_HF release content.

RFE FW

* Specific fix for sporadic spurs on 5MHz and 10MHz 1st FFT (range) bins on to top of 0.8.12 release

4.1.9 SAF85xx RFE SW CD 0.8.12 D221216

RFE Documentation

* Updated to reflect the 0.8.12 release content.

* RFE API profiling updated (profiling of API from RFE Driver code as measured from APP-A53).

RFE FW

* Supports only SAF85xx ES1.1 A1-E5-T2MF

* ADPLL retune timing violation solved

* di\dt

- di\dt for the end of the first chirp sequence with re-calibration enabled has been improved.

- GLDO2 voltage undershoot occurring during param update has been solved

- GLDO2 voltage drop to 0.9V occurring between chirp sequences has been solved

- GLDO2 voltage undershoot occurring between chirp sequences with different TX Pout settings has been solved

* Calibration strategy Changed back to 0.8.10. All profiles present in configuration file are calibrated during configuration and visible in spectrum, only used profiles calibrated at rec-calibration.

* Chirp PLL loop bandwidth updated for 4GHz VCO

* RX IF Calibration based on feedback loop is introduced, calibration sets the RX IF to meet targets (LPF, HFP, Rx Gain) for process corners.

Note:

RX IF Calibration procedure requires always 16 bits data path, that is `pdcBitwidth = rfe_pdcBitwidth_16bit_e`.

RX IF Calibration is called by `rfe_configure()` *only, see RFE Abstract API. Note that, RX IF Calibration is not called during Recalibration.*

In case the radar application is using 14 bit or 12 bit samples during radar data acquisition the following procedure must always be followed:

- the PDC bit width (data path) must be set to 16 bit in the RFE configuration file for the first RX IF Calibration.

- `pdcBitwidth` *must be set to 12 bit or 14 bit directly after calling* `rfe_configure()`. *Use* `rfe_updateParam()` *to update* `pdcBitwidth`.

- `pdcBitwidth` *must be set to 16 bit just before calling* `rfe_configure()`. *Use* `rfe_updateParam()` *to update* `pdcBitwidth`.

* TX Pout accuracy improved across temperatures to meet +\-0.5dB.

* During power down mode, supply of Chirp PLL, PDC and ADC hardware is switched off reducing power dissipation

- extra power state introduced for PDC\ADC power management

see RFE Power Management Control chapter of RFE SW Reference Manual for the details.

* Dynamic power management (DPM) implemented: RFE FW powers down hardware autonomously, depending on idle time windows between chirp sequences, time windows are function of user configuration for radar timing.

* FuSa updates:

- faults validated as per pass list

- RFE ERROR_N is de-asserted after RFE initialization, after `rfe_sync()`.

Note: To clear RFE ERROR_N Main FCCU should be cleared after RFE FW is up, or Main FCCU RFE ERROR_N input (NCF[118]) should be programmed to hardware recoverable fault so it is automatically cleared upon RFE ERROR_N desertion, to start further monitoring.

* BIST updates

The following RFE BIST tests are present

- ChirpPLL LLDO monitoring

- TX12\23\34 phase difference monitoring

- TX1\2\3\4 phase step monitoring

Refer to RFE SW Reference Manual for the details.

RFE API

* The `rfe_getBistZeroHourReferenceData()` function parameter types changed to reflect signed format for zero hour references. Several parameters used in CMD message changed from unsigned into signed types, refer to RFE SW Reference Manual for details. This update has no effect on the CMD message structure.

The RFE FW performs the TX phase difference, RX phase difference and RX gain difference measurements only if the corresponding fault bits are unmasked in the config BLOB.

* `uint32_t rfe_testGetInternalError()` function added.

RFE Configuration

* FuSa updates

  - FuSa fault list extended to reflect individual BIST faults, see `t` in `rfe_error.h`

  - Number of PPD thresholds is equal to number of profiles (8). In earlier releases, the number of PPD thresholds is equal to number of Tx (4).

* BIST updates

  - RFE Configuration BLOB format changed to reflect signed format for `zeroHourReferences`

Refer to RFE SW Reference Manual on detailed speciation of RFE BLOB.

4.1.10 SAF85xx RFE SW CD 0.8.11 D220930

RFE FW

* ChirpPLL VCO (AAFC) Calibration longer lock time, causing RF stop at cold, is solved.

4.1.11 SAF85xx RFE SW CD 0.8.10 D220826

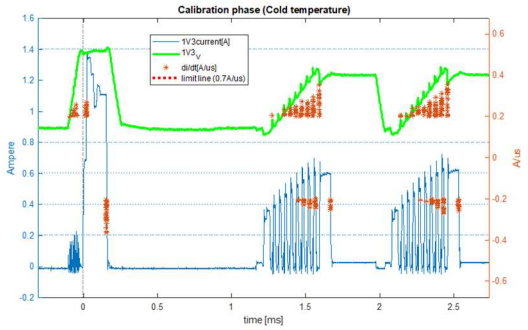
RFE Documentation

* included update of Functional Safety and BIST API.

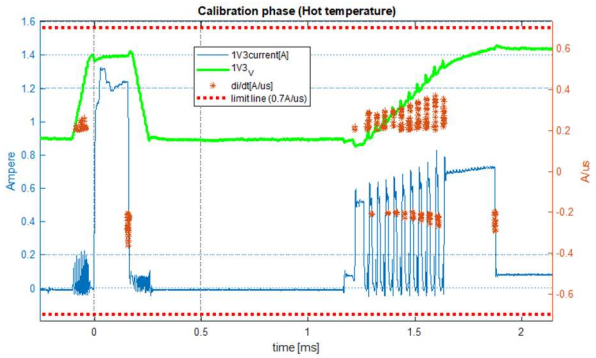
* added the new state “start offset” in the RADAR cycle state diagram.

RFE FW

* di/dt at cold temperature during ‘configuration calibration’.



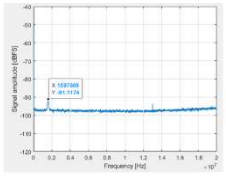
* di/dt at hot temperature during ‘configuration calibration’.



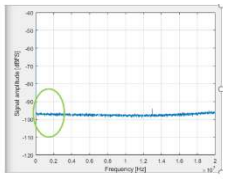
* GLDO undershoots are generally not signaled as di\dt on NXP board\test set-ups, so it is not possible to have automated full-proof di\dt release tests. Instead error prone manual inspection of waveforms has to be done which in nature cannot be exhaustive as limited wave-forms can only be inspected

* Tx power accuracy improved compared to 0.8.9HF.

* ADC dither updated to remove spurious



Dither setting 24, -95dBFS input signal at 13MHz



Dither setting 31, -95dBFS input signal at 13MHz

* Occupied Bandwidth (OBW): visible only for in-band per defined frequencies;

* Calibration phase in a RADAR cycle excludes chirp profiles not used in current RADAR cycle actually. Calibration phase in a RADAR cycle includes transmitters having Tx transmission enabled in one or more chirp profiles in current RADAR cycle.

RFE API

* The function

  - `uint8_t rfe_getFuSaErrors( rfe_fusaError_t *pFusaErrorList, uint8_t listLength, uint32_t *pFwInternalErrorCode);`

  is changed to

  `void rfe_getFuSaFaults( uint8_t *pFuSaR1R2FaultList )`

* New Functional Safety and BIST API functions.:

  - `void rfe_getFuSaFaultStatistics( uint16_t *pR1FaultPromotedToR2, rfe_radarCycleCount_t *pRadarCycleCount, uint8_t *pFuSaR1FaultCountList );`

  - `void rfe_getBistZeroHourReferenceData( rfe_txBistZeroHourRefData_t *pTxReferenceData, rfe_rxBistZeroHourRefData_t *pRxReferenceData );`

* Many error types are updated in `rfe_error.h`.

* FuSa faults masking\unmasking removed from `rfe_testSetParam()`.

* FuSa faults masking\unmasking added to RFE Configuration data structure.

* The function `rfe_updateParam()` extended with functions for the newly added RFE parameters

RFE Configuration

* FuSa related changes

  - `PdcClipping` renamed to `AdcClipping`

* Parameters added for FuSa \ BIST, `rfeCfg_param_t` adapted:

  - `thresholdValueToPromoteR1Faults`
  - `zeroHourReferenceForTxPhaseDiff`
  - `txPhaseDiffThresholdTolerance`
  - `txPhaseStepThresholdTolerance`
  - `txPowerLevelForBist`
  - `txFrequencyForBist`
  - `rxFrequencyForBist`
  - `zeroHourReferenceForRxPhaseDiff`
  - `rxPhaseDiffThresholdTolerance`
  - `zeroHourReferenceForRxGainDiff`
  - `rxGainDiffThresholdTolerance`
  - `injectTestToneBeforeLna`
  - `txPpdThreshold`
  - `fuSaFaultMask`

* API upgraded with “Start time offset for first chirp sequence in RADAR cycle”; Parameter added for chirp sequence 0 start time offset
  - `chirpSequenceStartTimeOffset[0]`
* Utility functions for access to the above parameters added

RFE Configuration Tool

* Updated for Functional Safety and BIST parameters in RFE Config.
* Updated for “Start time offset for first chirp sequence in RADAR cycle”.

RFE Driver

* Adapted to support Functional Safety and BIST changes.

RFE Applications

* rfeAppBasicExample32 adapted for Functional Safety and BIST API changes

4.1.12 SAF85xx_RFE_SW_HF01_0.8.9_D220727

RFE FW

* Large di\dt jumps caused by PA Enable during Calibration are resolved.
* Notch Filter setting in combination with dynamic table required time out increase.

4.1.13 SAF85xx RFE SW CD 0.8.9 D220715

RFE Documentation

* Updated to reflect 0.8.9 changes: DC Filter RFE configuration settings.
* DC Filter settings explained: coefficient calculation for a required corner frequency

RFE FW

* DC Filter settings added per chirp sequence instead of static\global setting
* TX PR Calibration\ReCalibration enabled
* Maximum time for Calibrations is set in RFE FW. When maximum time is elapsed and calibration did not finish in time, an error is reported via RFE SW error.
* di\dt reduced

- in general within specification of 0.7 [A]/[us], as measured on NXP WG4.1 boards
- in rare cases di\dt is above specification at 0.8-0.9 [A]/[us] maximum

* Internal SPI access set at 80MHz (as validated) to compensate for di\dt related calibration time increase.

RFE Configuration Tool

* Updated for DC Filter setting per chirp sequence

* RFE Configuration tool Reference Manual updated accordingly

RFE Applications

* APP-A53 SoC drivers are moved out from /rfe/rfeDriver/rfeAppBasicExample32/ to a separate /rfe/rfeDriver/a53drivers/ folder.

4.1.14 SAF85xx RFE SW EAR 0.8.8 D220701

RFE Documentation

* Updated to reflect 0.8.8 changes: Dynamic Tables (frequency drift), Frequency Drift documented.

RFE FW

* Static Frequency drift supported, using RFE Configuration (parameter was present already)

* Dynamic Frequency drift supported, using Dynamic Table

* All Calibrations and Recalibrations supported

- except for TX PR Calibration and TX PR Re-Calibration

* Partial updates of Dynamic Tables supported during param update

Note: Radar cycle timing adapted in RFE FW FSM: chirp sequences offset (start) time increased, use RFE config tool to adapt radar application

RFE Configuration Tool

* Updated for Dynamic Tables frequency drift support: `chirpFrequencyDrift` parameter added

* Updated for Dynamic Tables removing settle time support `settleTimeTicks` parameter removed

* RFE Configuration tool Reference Manual updated accordingly

RFE Driver

* RFE Driver implemented as AUTOSAR CDD driver, as a layer on top of existing bare metal RFE Driver, supports AUTOSAR version 4.4

* RFE Driver supports both AUTOSAR and Bare Metal

4.1.15 SAF85xx RFE SW CD 0.8.7 D220610

RFE Documentation

* Updated to reflect 0.8.7 changes: API (Tx power), Radar Cycle FSM (power down mode)

RFE FW

* di\dt issue solved

  - limitation exists when RFE configuration uses different power levels per chirp sequences of the same radar cycle, to overcome it:

    i. use RFE configurations with single power level for all chirp sequences in a given radar cycle and

      a. at run-time reconfigure RFE using `rfe_configure()` for a new radar cycle which uses different power

      b. at design-time use different application images per power level used

* far-away objects attenuation issue present in previous releases solved

* FFT noise improved at full temperature range

* Power down mode replacing standby power mode during sensor idle introduced

  - radar cycle timing modified, use RFE Configuration Tool to adapt application timing accordingly.

* Tx Power read out introduced, Tx power range and power step adjusted

* Tx1 and Tx4 Calibration improved

* RFE FW tested on ES1.1 (A1MF)

RFE API

* Tx Power read out for each transmitter and each profile introduced in `rfe_monitorRead()`

  - `rfe_monitorValues_t` extended with

  ```
  int16_t txPower[RFE_CHIRP_PROFILES_MAX][RFE_TX_COUNT];
  ```

RFE Abstract API change is backwards compatible with new functionality extended.

RFE Configuration

* in `chirpProfile` section Tx power range adjusted

`txPower | uint8_t | 0 | 255`

changed to

`txPower | int16_t | -90 | 150`

RFE Configuration Tool

* Updated for Tx power range adjustment

RFE Driver

* RFE stub
  - txPower introduced in `rfe_monitorRead()`

see that this change does not change the RFE Command-Response message format, however the content can carry Tx Power read out request (command) and valued (response).

RFE Applications

* rfe configuration updated to new format for Tx power range adjustment
* Aurora PLL setting adjusted to avoid unlock at temperature testing

4.1.16 SAF85xx RFE SW CD 0.8.6 D220524

RFE FW

* No object detected by some chirp sequences in multiple chirp sequence solved. RX IF calibration is performed also for profiles with highest frequency, this problem caused a profile with the highest frequency to work incorrectly.
* Documenting start time offset for first chirp sequence, first chirp sequence offset is calculated by RFE Configuration timing tool, see below.
* Documenting `rfe_dataOutConfig_t`, `rfe_virtualChannel_t` RFE configuration parameters and their relation to packet processor.

RFE Configuration Tool

* Radar cycle timing calculation added and integrated with RFE Configuration Tool
* RFE Configuration Tool Reference Manual updated adding description of timing calculation tool

4.1.17 SAF85xx RFE SW CD 0.8.5 D220509

RFE FW

* Observed ChirpPLL Bandwidth is reduced in case of multi-profile multi-chirp sequences use cases (ChirpPLL overshoot of T_{reset} is reduced).

* Center Frequency is set properly for the first chirp of second and all consecutive chirp sequences in radar cycle

* CLK PLL re-tune stable, can be used for temperature testing

  - phase shift caused by re-tune as reported in 0.8.4 solved

* RFE SW tested on E5 samples

  - booting via JTAG, secure booting will be supported at later date

4.1.18 SAF85xx RFE SW CD 0.8.4 D220426

RFE FW

* Tx Calibrations (update)

  - di\dt during TX Pout Calibration is reduced via current limiter to 2.7 [A]

  - Tx Power is degraded when Tx Re-calibration is enabled, NXP recommends it to disable Tx Re-calibration.

  Note: Tx Re-Calibration does not exceed the re-calibration budget.

* CLK PLL Re-Calibration [by default disabled] added to avoid temperature dependent CLK PLL unlock situation.

  - performed every radar cycle during Power On state of every chirp sequence in radar cycle

    i. CLK PLL re-calibration does not change internal radar cycle timing as it uses time dedicated to Power-On phase

  - performed every 50[ms] in command idle state when radar cycle is inactive.

    i. Note that an RFE API command can be delayed by 1[ms] in command idle when it is issued during ADPLL Re-calibration action; this is not the case when radar cycle is running

    ii. 1[ms] time is fixed time outside of radar cycle when RFE FW is busy.

* RX IF programming enabled

  - RX IF (HPF, LPF, Gain) can be programmed via API with full range for these parameters, refer to RFE reference Manual for the ranges supported

RFE API

* CLK PLL ReCalibration control added

  - CLK PLL Re-calibration is disabled by default. It can be disabled\enabled by test parameter `rfe_testParam_enableClockRetuning_e` in function `rfe_testSetParam()`, refer to RFE SW Reference manual, example to enable CLK PLL Re-Calibration:

```
rfe_testSetParam( rfe_testParam_enableClockRetuning_e, true,
RFE_ERROR_FUNCTION_ARGUMENT )
```

This RFE Abstract API change is backwards compatible, no adaptations to application code are required to integrate this API

4.1.19 SAF85xx RFE SW CD 0.8.3 D220408

RFE FW

* Tx Calibrations (initial version) added
  - to Configuration Calibration phase
  - to Re-Calibration phase
  - Pout can be set via RFE API
  - Tx Calibrations include
    i. LOI Tx Calibration
    ii. Tx LOx2 Calibration
    iii. Tx Buffer2a Calibration
    iv. Tx Buffer2b2c Calibration
    v. Tx PA Calibration
* Chirp PLL (VCO) calibration added
  - to Re-Calibration phase
  - is present in Configuration phase since v0.8.0
* T0-T2 OTP trimming supported
* RFE parameter update enabled (rfe_update<name>() APIs)
* RFE FW image fixed size
  - CODE, DATA files 128KB and 32KB respectively.

RFE Applications

* Basic example application: SAF85xx SoC configuration (clocking, devices) added

4.1.20 SAF85xx RFE SW CD 0.8.2 D220315

Release structure aligned to common NXP release structure and naming.

RFE Documentation

* Radar Cycle FSM and Radar Cycle timing updated

RFE FW

* Tx Toggle disable is functional for multi-profile case
* Radar cycle timing optimized allowing use of shorter chirp offset times
* Tdwell +Tsettle limit decreased to 2 [µs], this limit applies only when dynamic tables are used
* Attenuation of faraway objects for positive slope chirps solved

RFE API

* New API error defined when unconfigured Dynamic Table is sent to RFE FW

4.1.21 SAF85xx RFE SW CD 0.8.1 D2202

Release structure aligned to common NXP release structure and naming.

RFE Documentation

* CRC description improved (CRC is calculated whole command response buffer)
* rfe_sync() diagram improved

RFE FW added

* CSI2 output enabled: 1280 and 2560 Mbps
* OTP trimming supported
  - trimming parameters read from OTP when present, otherwise defaults used (T0 trimming)
* Dynamic programming using dynamic table
  - Tx Phase Rotation (DDMA)
  - CIT
  - Multi mode (multiple profiles per chirp sequence)
* Multiple chirp sequences per radar cycle
* Single mode chirp sequences (one profile)
* Multiple mode chirp sequences: (multiple profile)
    - using static using rfe_config
    - using dynamic using dynamic table
  * Chirps
    - Fixed TX setting
      a. Tx power at 1.4 [V]
      b. not calibrated
      c. profiles 0 to 7 use the same fixed setting
    - Fixed RX settings
      a. gain at 46 [dB]

b. HPF at 800 [kHz]

c. LPF 25 [MHz]

d. not calibrated

e. profiles 0 to 7 use the same fixed setting

RFE API updated

* *rfe_getFuSaErrors()* can also be called in *radarCycleIdle* state

* 2 new API errors defined

RFE Configuration Parameters

* no change

RFE Driver

* no change

RFE Applications

* added support for printf()

* example extended to temperature monitors read outs, FuSa error masking, test params

RFE Configuration Tool

* no change

4.1.22 SAF85xx RFE SW Release CD 0.8.0

RFE FW added

* CODE and DATA binary files included

RFE API updated

* *void rfe_init()* renamed to *void rfe_sync()* to better reflect the functionality

* uint8_t *rfe_getFuSaErrors*(rfe_fusaError_t *pFusaErrorList, uint8_t listLength, uint32_t *pFwInternalErrorCode) replaces rfe_error_t rfe_getError()

* *rfe_getVersion()* extended with RFE FW and HW version

* error codes extended with FuSa errors (rfe_errors.h)

* test accessor functions added (rfe_test.h) to reflect extend test parameters *rfe_testSetParam_keepTxTransmissionEnabled()*, *rfe_testSetParam_chirpPllTestPinEnable()*, *rfe_testSetParam_maskError()*, *rfe_testSetParam_unmaskError()*

* all parameters that can be updated are supported by *rfe_updateParam()* (rfe_update.h)

* rfe types adapted to reflect adapted rfe configurations and extended function returns and extension of parameter that can be updated (rfe_types.h)

RFE Configuration

* utility\development functions adapted for changes in rfe parameters (*rfeCfg.h*).

RFE Parameters

* meta data section

  - extended with major, minor, and patch versions parameters for RFE SW

* general section

  - *dynamicPowerControlMode* parameter removed, not supported by RFE HW

  - *jumpbackTimeTicks* moved in from profile section, not available per profile

  - *chirpActiveSignalConfig* renamed to *chirpSequenceActiveSignalGpio* to better reflect the changed functionality (SW generated signal via selected GPIO)

* Chirp Sequence Config section:

  - *txEnable* moved in from Chirp Profile section, not available per profile

  - *rxEnable* moved in from Chirp Profile section, not available per profile

* Chirp Profile section:

  - *chirpPllLoopFilterBandwidth* moved in from Chirp Sequence section, controllable per profile

  - *txEnableReferenceTime* renamed to *txTransmissionReferenceTime* to better reflect the functionality

  - *txEnableTimeOffset* renamed to *txTransmissionTimeOffset* to better reflect the functionality

  - *resetTimeTicks* extended to 16bits

RFE Driver

* RFE client

  - Shared data initialization improved in *rfeCmdClient_sync()*

* RFE stub

  - Adapted to API changes listed above

RFE Applications

* RFE basic example application

  - includes S32DS project for building application with RFE FW

  - includes boot scripts for booting and executing example application with RFE FW

  - includes *rfeConfig* files demonstrating RFE operation (ceiling test)

RFE Configuration Tool

* Command line options added, see ..\rfe\tools\rfeConfigGenerator\release\readme.text
* RFE BLOB generation updated to match rfe configuration changes, Documentation updated accordingly.

4.1.23 SAF85xx RFE SW Documentation Update 0.4.2

RFE Documentation

* RFE Driver Documentation Updated
* RFE Abstract API is merged into RFE SW User Manual

4.1.24 SAF85xx RFE SW Hot Fix 0.4.1

RFE Configuration Tool

* Configuration Tool Update
  - Tx Enable param removal to align with 0.4.0 RFE FW
  - rfeDynamicTables.bin generation corrected
* Configuration Tool Manual Update
  - Tx Enable removal
  - Description of repetitive tables generation added, chapter 5.6.1

4.1.25 SAF85xx RFE SW Code Drop 0.4.0

RFE API updated

* code documentation extended
* *rfe_configure()*
  - *dynamicTableAddress* dynamic table parameter added, no more referenced via RFE Config
* *rfe_radarCycleStop()* returns index of latest executed radar cycle
* *rfe_monitorRead()* returns radar cycle and chirp sequence for which monitors were read
* *rfe_getRadarCycleStartTime()* renamed to *rfe_getNextRadarCycleStartTime()*
  - parameter changed from *uint32_t radarCycleIndex* to *rfe_radarCycleCount_t *pRadarCycleCount*
* *void rfe_setRadarCycleStartTime()* renamed to *rfe_radarCycleCount_t rfe_setNextRadarCycleStartTime()*
  - *radarCycleIndex* parameter removed
* *rfe_testModeStart\Stop()* functions (*rfe_test.h*) split into
  - *rfe_testContinuousWaveStart\Stop()*

- *rfe_testSetParam(rfe_testParam_t testParam, uint32_t value)*

* *rfe_updatePush()* returns radar cycle count for which the update is applied.

- *scheduleApply()* and *chirpSequenceIndex()* function parameters removed, updates will be applied to the returned chirp sequence.

- implementation of param update function is extended with the following parameters:

- update of *Tdwell* time parameter enabled

- update of *Tsettle* time parameter enabled

- update of *Tjumpback* time parameter enabled

- update of *Treset* time parameter added

* *rfe_command()* function removed as obsolete

* API error codes extended (*rfe_error.h*)

* API types improved (*rfe_types.h*) reflecting the extended RFE Configuration

RFE Configuration

* Utility\development functions added for access to binary Config BLOB fields for each RFE parameter (*rfeCfg.h*).

Utility functions added for access to dynamic table fields.

RFE Parameters

* Meta data section

- dynamic table reference removed, see *rfe_configure()*

* General section

- selection of GPIO for radar cycle start I\O signal added

- *chirpPllLoopFilterBandwidth* moved to chirp sequence config section

- PDC bit width parameter added

- Meta data packet configuration added

meta data packet CSI2 virtual channel can be set

enable of meta data HW-defined fields added

SW-defined fields configuration added to meta data packet: saturation counts, and chirp sequence start RFE time stamp

* Radar Cycle section:

- BIST interval definition changed (in the future to be replaced by safety extension)

- *calibrationInterval* divided into two settings:

recalibrateProfileDependent

recalibrateProfileIndependent

to reflect SAF85xx calibration strategy

- Radar cycle timing parameters resolution increased to 25[ns]

* Chirp Sequence Config section:

- *chirpPllLoopFilterBandwidth* moved in from radar cycle section

- Fast reset control added

fastResetEnable added

fastResetDelayTicks added

fastResetDurationTicks added

- *chirpProfileRepetitionCount* parameter removed

* Chirp Profile section:

- chirp profile timing parameters resolution increased to 25[ns]

RFE Driver

* RFE client

- Shared memory addresses interface separated to a dedicated interface file

- Shared data buffer introduced with read function added: rfeCmdClient_sharedDataRead() for access to rfe state, radar cycle count and chirp sequence count.

* RFE stub

- Commands stub functions extended to cover all remote RFE Abstract API functions:

rfe_configure(), rfe_radarCycleStop(), rfe_getState(), rfe_getRadarCycleCount(), rfe_getError(), rfe_getRfeTime(), rfe_getVersion(), rfe_monitorRead(), rfe_setRadarCycleStartTime(), rfe_monitorRead(), rfe_testModeStart(), rfe_testModeStop(), rfe_updatePush(), rfe_continuousWaveTransmissionStart\Stop(), rfe_setTestParam()

- Parameter update functions implemented:

rfe_updateBegin(), rfe_updateParam (), *rfe_updateDynamicTable()*

RFE Applications

* RFE basic example application added

RFE Configuration Tool

* RFE configuration tool added

* Example user input file and generated file included

4.1.26 SAF85xx RFE SW Code Drop 0.3.0

* RFE API, Configuration and Dynamic Table functionality as per 0.2.0 release

- Integration with C++ code added (header file guard)

* RFE command communication implementation added
  - RFE Client
    Implements functions to communicate with RFE FW via shared memory command buffer
    rfeCmdClient.h, rfeCmdClient.c, rfeCmdClient_internal.h
  - RFE Stub
    Implements RFE Abstract API functions as translation to messages
    rfeCmdClientStub.c

4.1.27 SAF85xx RFE SW Code Drop 0.2.0

* Initial release

5. FuSa Pass List

Table 4 shows the list of the FuSa safety monitors that can be enabled in the current release. Please refer to Section 7 for limitations.

Table 4: FuSa Pass List

Fusa Pass - Validated List - SAF85xx_RFE_SW_EAR_0.8.19_D230929
rfe_fuSaFault_R1_sm3_bb_tx1_e
rfe_fuSaFault_R1_sm4_bb_tx2_e
rfe_fuSaFault_R1_sm5_bb_tx3_e
rfe_fuSaFault_R1_sm6_bb_tx4_e
rfe_fuSaFault_R1_sm7_paout_rtm_tx1_e
rfe_fuSaFault_R1_sm8_paout_rtm_tx2_e
rfe_fuSaFault_R1_sm9_paout_rtm_tx3_e
rfe_fuSaFault_R1_sm10_paout_rtm_tx4_e
rfe_fuSaFault_R1_sm11_bist_tx1_tx2_phaseDiff_sw_e
rfe_fuSaFault_R1_sm11_bist_tx1_phaseStep_sw_e
rfe_fuSaFault_R1_sm12_bist_tx2_tx3_phaseDiff_sw_e
rfe_fuSaFault_R1_sm12_bist_tx2_phaseStep_sw_e
rfe_fuSaFault_R1_sm13_bist_tx3_tx4_phaseDiff_sw_e
rfe_fuSaFault_R1_sm13_bist_tx3_phaseStep_sw_e
rfe_fuSaFault_R1_sm13_bist_tx4_phaseStep_sw_e
rfe_fuSaFault_R1_sm15_bb_rx1_e
rfe_fuSaFault_R1_sm16_bb_rx2_e
rfe_fuSaFault_R1_sm17_bb_rx3_e
rfe_fuSaFault_R1_sm18_bb_rx4_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx2_gainDiff_sw_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx3_gainDiff_sw_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx4_gainDiff_sw_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx2_phaseDiff_sw_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx3_phaseDiff_sw_e
rfe_fuSaFault_R1_sm23_bist_rx1_rx4_phaseDiff_sw_e
rfe_fuSaFault_R1_sm28_ov_gld1v3_gldo_e
rfe_fuSaFault_R1_sm29_ov_gld1v4_gldo_e
rfe_fuSaFault_R1_sm29_uv_gld1v4_gldo_e
rfe_fuSaFault_R1_sm31_ov_hld1v8_gldo_e
rfe_fuSaFault_R1_sm31_uv_hld1v8_gldo_e
rfe_fuSaFault_R1_sm46_xo_pllclk_det_mcgen_e
rfe_fuSaFault_R1_sm50_adpll_dco_level_max_mcgen_e
rfe_fuSaFault_R1_sm50_adpll_dco_level_min_mcgen_e

rfe_fuSaFault_R1_sm54_level_low_chirp_e
rfe_fuSaFault_R1_sm54_level_high_chirp_e
rfe_fuSaFault_R1_sm55_unlock_chirp_e
rfe_fuSaFault_R1_sm58_cg_chirp_e
rfe_fuSaFault_R1_sm59_li_chirp_e
rfe_fuSaFault_R1_sm60_sd_chirp_e
rfe_fuSaFault_R1_sm62_lockstep_te_e
rfe_fuSaFault_R2_sm63_te_swt_radar_pipeline_e
rfe_fuSaFault_R2_sm68_sm69_reg_crc_e
rfe_fuSaFault_R2_sm70_mosi_miso_crc_e
rfe_fuSaFault_R2_sm83_rfe_test_mode_active_e
rfe_fuSaFault_R2_sm87_m7_rfe_lockup_e
rfe_fuSaFault_R2_sm88_sm89_m7_tcm_sram_e
rfe_fuSaFault_R2_sm88_sm89_m7_tcm_sram_e
rfe_fuSaFault_R2_sm92_sw_wdt_e
rfe_fuSaFault_R2_sm93_xbic_integrity_check_e
rfe_fuSaFault_R1_sm94_m7_message_e2e_crc_sw_e
rfe_fuSaFault_R2_sm98_m7_spi_access_ana_e
rfe_fuSaFault_R2_sm99_spi_write_check_fail_sw_e
rfe_fuSaFault_R2_sm204_dm_rfe_m7_core_sw_redundancy_e
rfe_fuSaFault_R2_sm435_generic_sw_e
rfe_fuSaFault_R1_sm19_rf_i_low_rx1_e
rfe_fuSaFault_R1_sm19_rf_q_low_rx1_e
rfe_fuSaFault_R1_sm20_rf_i_low_rx2_e
rfe_fuSaFault_R1_sm20_rf_q_low_rx2_e
rfe_fuSaFault_R1_sm21_rf_i_low_rx3_e
rfe_fuSaFault_R1_sm21_rf_q_low_rx3_e
rfe_fuSaFault_R1_sm22_rf_i_low_rx4_e
rfe_fuSaFault_R1_sm22_rf_q_low_rx4_e
rfe_fuSaFault_R1_sm64_sup_ldo1_ov_vco_1v8_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo2_ov_vco_0v9_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo3_ov_pfdcp_1v8_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo4_ov_pfdcp_0v9_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo5_ov_pdiv_0v9_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo1_uv_vco_1v8_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo2_uv_vco_0v9_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo3_uv_pfdcp_1v8_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo4_uv_pfdcp_0v9_chirp_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo5_uv_pdiv_0v9_chirp_sw_e

rfe_fuSaFault_R1_sm64_sup_ldo1_ov_xo_core_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo2_ov_xo_out_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo3_ov_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo4_ov_dco_capbank_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo5_ov_dco_buffer_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo6_ov_div3_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo7_ov_sampler_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo8_ov_digital_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo1_uv_xo_core_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo2_uv_xo_out_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo3_uv_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo4_uv_dco_capbank_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo5_uv_dco_buffer_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo6_uv_div3_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo7_uv_sampler_mcgen_sw_e
rfe_fuSaFault_R1_sm64_sup_ldo8_uv_digital_mcgen_sw_e
rfe_fuSaFault_R1_sm86_m7_fpu_e

6. Compiler versions and options

This section describes the compiler versions and options used for compiling RFE Driver

6.1.1 GCC Compiler/Linker/Assembler Options

Compiler Version : *NXP GCC 9.2 for Arm toolchain* :

- arm-none-eabi-gcc (i.e. 32-bit) for appM7
- aarch64-none-elf-gcc (i.e. 64-bit) for A53

Table 5 Compiler options

Option	Description
-Werror	Treat all enabled warnings as compilation errors*
-O3	Optimization level (Release version)
-Wall	Enables a basic set of warnings
-Wextra	Enables extra warnings not covered by -Wall
-std=c99	C Programming Language Standard used it c99
-mcpu=cortex-m7 -mcpu= cortex-a53	Compile for Cortex-M7 processor (i.e. appM7 on SAF85XX) Compile for Cortex-A53 processor
-mstrict-align	Unaligned accesses are forbidden

*An independent compliance check is done with -Werror on source code in “rfeDriver/units” folder. However, it is **NOT** currently included in the compilation flags of the RFE Driver applications as whole.

Table 6 Assembler options

Option	Description
-c	Produces an object file (called input-file.o) for each source file.

Table 7 Linker options

Option	Description
-T <path to *.ld linker file>	Option used to input the linker file

6.1.2 GHS Compiler/Linker/Assembler Options

Compiler version : Green Hills MULTI for Eclipse 2.5.5 (not yet fully functional)

Table 8 Compiler options

Option	Description
-Werror	Treat all enabled warnings as compilation errors*
-O3	Optimization level (Release version)
-Wall	Enables a basic set of warnings
-Wextra	Enables extra warnings not covered by -Wall
-std=c99	C Programming Language Standard used it c99
-mcpu=cortex-m7 -mcpu= cortex-a53	Compile for Cortex-M7 processor (i.e. appM7 on SAF85XX) Compile for Cortex-A53 processor
-mstrict-align	Unaligned accesses are forbidden

Table 9 Assembler options

Option	Description
-c	Produces an object file (called input-file.o) for each source file.

Table 10 Linker options

Option	Description
-T <path to *.ld linker file>	Option used to input the linker file

6.1.3 DIAB Compiler/Linker/Assembler Options

Compiler Version : Wind River Diab Compiler 7.0.3 (not yet integrated)

Table 11 Compiler options

Option	Description
-Werror	Treat all enabled warnings as compilation errors*
-O3	Optimization level (Release version)
-Wall	Enables a basic set of warnings
-Wextra	Enables extra warnings not covered by -Wall
-std=c99	C Programming Language Standard used it c99
-mcpu=cortex-m7 -mcpu= cortex-a53	Compile for Cortex-M7 processor (i.e. appM7 on SAF85XX) Compile for Cortex-A53 processor
-mstrict-align	Unaligned accesses are forbidden

Table 12 Assembler options

Option	Description
-c	Produces an object file (called input-file.o) for each source file.

Table 13 Linker options

Option	Description
-T <path to *.ld linker file>	Option used to input the linker file

6.2 Debugger

Refer to RFE App Basic Example section of the *RM00266 SAF85xx RFE Reference Manual*.

6.3 Examples

Refer to RFE Control Application: Reference Usage section of the *RM00266 SAF85xx RFE Reference Manual*.

7. Limitations

This release is targeting SAF85xx ES2 hardware. This release comes with limitations as per Table 14. Please also refer to the Errata sheet.

Table 14 Limitations

Item	Description
FuSa Safety Monitor SM57	SM57 that is "SM57_CHIRPPLL_Frequency_Monitor" (rfe_fuSaFault_R1_sm57_vco_freq_chirp_e) is not functional after enabling this particular SM.This fault is never triggered.
Backward compatibility	It has been observed that even though the firmware is backward compatible, recompilation of the host application along with the latest (0.8.19) is required since the *rfe_paramUpdate_general_chirpActiveOut_e* is added in between in the *rfe_paramUpdate_t* enumerated datatype present in *rfe_types.h*. Even though the RFE Config Blob content as such is backward compatible, there is a version check, hence before using the config blob from previous version, only the version needs to be updated while using with 0.8.19

8. Compatibility

Table 15 gives an overview of the compatibility of the various released software and firmware components. Combinations that are not listed are not supported and may not work.

Table 15: Software, firmware and hardware compatibility

RFE Abstract API	RFE Driver	RFE FW	RFE Configurator Tool	RFE Basic Example App	Documentation	HW Versions Supported
0.8.19	0.8.19	0.8.19	0.8.19	0.8.19	0.8.19	SAF85xx ES2 E5-T1+
0.8.18	0.8.18	0.8.18	0.8.18	0.8.18	0.8.18	SAF85xx ES2 E5-T1+
0.8.17	0.8.17	0.8.17	0.8.17	0.8.17	0.8.17	SAF85xx ES2 E5-T1
0.8.16	0.8.16	0.8.16	0.8.16	0.8.16	0.8.16	SAF85xx ES2 E5-T1
0.8.15	0.8.15	0.8.15	0.8.15	0.8.15	0.8.15	SAF85xx ES1.1 A1-E5-T2MF
0.8.14	0.8.14	0.8.14	0.8.14	0.8.14	0.8.14	SAF85xx ES1.1 A1-E5-T2MF
0.8.13	0.8.13	0.8.13	0.8.13	0.8.13	0.8.13	SAF85xx ES1.1 A1-E5-T2MF
0.8.12_HF01	0.8.12_ HF01	0.8.12 _HF01	0.8.12_HF01	0.8.12_HF01	0.8.12_HF01	SAF85xx ES1.1 A1-E5-T2MF
0.8.12	0.8.12	0.8.12	0.8.12	0.8.12	0.8.12	SAF85xx ES1.1 A1-E5-T2MF
0.8.10	0.8.10	0.8.11	0.8.10	0.8.10	0.8.11	SAF85xx ES1 E2\E5-T0-T2 SAF85xx ES1.1 A1-E5-T2MF
0.8.10	0.8.10	0.8.10	0.8.10	0.8.10	0.8.10	SAF85xx ES1 E2\E5-T0-T2 SAF85xx ES1.1 A1-E5-T2MF
0.8.9	0.8.9	0.8.9	0.8.9	0.8.9	0.8.9	SAF85xx ES1 E2\E5-T0-T2 SAF85xx ES1.1 A1-E5-T2MF
0.8.8	0.8.8	0.8.8	0.8.8	0.8.7	0.8.8	SAF85xx ES1 E2\E5-T0-T2 SAF85xx ES1.1 A1-E5-T2MF
0.8.7	0.8.7	0.8.7	0.8.7	0.8.7	0.8.7	SAF85xx ES1 E2\E5-T0-T2 SAF85xx ES1.1 A1-E5-T2MF
0.8.4	0.8.2	0.8.6	0.8.6	0.8.3	0.8.6	SAF85xx ES1 E2\E5 T0-T2
0.8.4	0.8.2	0.8.5	0.8.0	0.8.3	0.8.4	SAF85xx ES1 E2\E5 T0-T2
0.8.4	0.8.2	0.8.4	0.8.0	0.8.3	0.8.4	SAF85xx ES1 E2 T0-T2
0.8.2	0.8.2	0.8.3	0.8.0	0.8.3	0.8.3	SAF85xx ES1 E2 T0-T2
0.8.2	0.8.2	0.8.2	0.8.0	0.8.1	0.8.2	SAF85xx ES1 E2 T0
0.8.1	0.8.1	0.8.1	0.8.0	0.8.1	0.8.1	SAF85xx ES1 E2 T0
0.8.0	0.8.0	0.8.0	0.8.0	0.8.0	0.8.0	SAF85xx ES1 E2 T0
0.4.0	0.4.0	n\a	0.4.1	0.4.0	0.4.2	n\a
0.4.0	0.4.0	n\a	0.4.1	0.4.0	0.4.0	n\a
0.4.0	0.4.0	n\a	0.4.0	0.4.0	0.4.0	n\a

9. Glossary

Abbreviations	Description
RFE	mmWave mixed signal Radar Front-End of SAF85xx
SOC	System-On-Chip
APP-A53	ARM Cortex-A53, the radar processing core of SAF85xx
APP-M7	ARM Cortex-M7, the control core of SAF85xx
RFE-M7	ARM Cortex-M7, the embedded core in RFE of SAF85xx
RFE FW	Software running on RFE-M7 processor to control the RFE HW
RFE SW	RFE FW and RFE Driver
Code Drop	Preliminary sharing of code for reference purposes
RFE Abstract API	API to configure and control SAF85xx Front End
JTAG	JTAG (Joint Test Action Group) is an industry standard for verifying IC designs after manufacture.
BLOB	Binary Large Object, a data type that stores binary data.
ICD	In Circuit Debugger
CDD	Complex Device Driver
FuSa	Functional Safety - ISO 26262
MISRA	Motor Industry Software Reliability Association
ASPICE	Automotive Spice Assessment - ISO/IEC 15504
AUTOSAR	AUTomotive Open System ARchitecture
CDD	Complex Device Driver
FuSa	Functional Safety

10. Legal information

10.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

10.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

11. Contents

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 29 September 2023

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 29 September 2023