MMWAVE DFP 01.02.00 Release Notes

1 Introduction

TI mmWave Device Firmware Package (DFP) enables the development of millimeter wave (mmWave) radar applications using AWR1243 SOC/EVM. It includes necessary components which will facilitate end users to integrate AWR1243 SOC with their choice of processor. The user is expected to use mmWaveStudio to measure RF and key system performance on TI mmWave Radar devices.

In addition, DFP provides mmWaveLink framework and example application which will serve as a guide for integrating the AWR1243 with external processor.

Note: mmWave Software Development Kit (SDK) enables the development of radar applications on AWR1443, AWR1642, IWR1443 and IWR1642 SOCs/EVMs. The basic components of DFP are included in mmWave SDK.

2 Release Overview

2.1 Platform and Device Support

The device and platforms supported with this release include:

Supported Devices	Release Status	Supported EVMs
AWR1243 ES3.0	RTM Release (DFP 1.2 contents)	AWR1243BOOST : AWR1243 ES3.0 Booster pack + MMWAVE-DEVPACK

Note: DFP supports the foundation components for the device mentioned in the table above. At system level, the mmWave SOC/EVM may interface with other SOCs/EVMs and software for other devices will not be a part of the DFP

2.2 Release contents and component versions

Component	Version	Type
RadarSS Firmware	ROM: 2.0.0.1, Patch: 1.2.0.0	Binary
MSS Firmware (AWR1243 ES3.0 only)	ROM: 1.10.0.20, Patch: 1.2.0.1	Binary
mmWaveLink Framework	1.2.0.0	Source and Library
Docs	Release Notes DFP user guide Interface Control Document mmWaveLink Programmer's guide	PDF PDF PDF Doxygen HTML

2.3 Directory Structure

Directory Name	Content
Docs	AWR1xxx_Radar_Interface_Control.pdf mmwave_dfp_release_notes.pdf mmwave_dfp_user_guide.pdf
firmware	RadarSS and Master SS firmware binary files in RPRC format metaImage binary (xwr12xx_metaImage.bin) for xWR1243 ES3.0 for boot over Flash metaImage header file (xwr12xx_metaImage.h) for xWR1243 ES3.0 for boot over SPI
ti	mmWaveLink framework and example source code
rf_eval	RF evaluation firmware

2.4 Component Descriptions

2.4.1 RadarSS Firmware

Refer to Radar SS firmware release notes in firmware\radarss folder.

2.4.2 Master SS Firmware

The main software components of the MSS firmware are

- System services – provides infrastructure services (error handling, mmWaveLink Host communication protocol manager) used by the functional firmware.
- Functional firmware – Is responsible for the external host API communication, RADAR SS API handshake, data path control.

2.4.3 mmWaveLink framework

Radar SS is a closed subsystem whose internal blocks are configurable using messages coming over mailbox.
TI mmWaveLink framework acts as driver for Radar SS and exposes services of Radar SS. It includes APIs to configure HW blocks of Radar SS and provides communication protocol for message transfer between external processor and AWR1243

- Link between application and Radar SS
- Platform and OS independent which means it can be ported into any processor which provides communication interface such as SPI and basic OS routines. The mmWaveLink framework can also run in single threaded environment

2.5 Licensing

Please refer to the mmwave_dfp_manifest.html, which outlines the licensing information for mmWave DFP package.

3 Release Contents

3.1.1 New Features (Compared to DFP 1.0 and DFP 1.1)

- Software trigger of sub-frames API
- TX phase shifter calibration and monitoring
- API option to allow disabling of PA LDO (used in simultaneous 3 TX use case)
- Support for daisy chain cascade mode
- CQ Monitoring enabled for all 4 profiles
- New Die ID read API
- Tx Phase shifter calibration LUT override GET and SET API
- Updated dynamic chirp configuration API with new Chirp row select option
- 20GHz Sync monitoring (for AWR1243 cascade mode)
- Enabled the RX peak detectors and mixer LO power monitoring
- Enabled boot time IQMM calibration

3.2 Feature List by Components

3.2.1 Radar SS firmware

Refer to Radar SS firmware release notes in firmware\radarss folder

3.2.2 Master SS firmware (Compared to DFP 1.0)

Type	Key	Summary
Feature	AUTORADAR-1797 AUTORADAR-1669	New feature is added which reports any errors in register write followed by read back through CPU fault async event
Bug	AUTORADAR-1718	When the MSS CR4 is continuously active, some spurs were observed. MSS entering WFI seems to reduce the spurs. While it is not guaranteed that the core will be in WFI always, it is helping when the core is not active.
Feature	AUTORADAR-1668	Changes for Get version API to support build and debug version reporting
Feature	AUTORADAR-1667	Currently, the firmware reports async reports of the latent fault tests and periodic monitoring tests if reporting mode is set to 1, and reports are disabled if reporting mode is set to 0. To align with BSS implementation the definition is changed as below: reporting mode 0: send the async reports reporting mode 1: async reporting of test status is disabled
Bug	AUTORADAR-1613	On system abort, the watch dog expires on MSS R4F leading to a reset. Watch dog is serviced on abort errors for HOST to take action.
Feature	AUTORADAR-1562	APLL continuous monitoring is enabled post BSS power up.
Bug	AUTORADAR-1561	When the reporting mode is turned off for Latent Fault tests, if any of the latent fault tests fails the failure was not reported to the HOST. Firmware is fixed to send out Latent Fault failure status reports upon failures, even if the reporting mode is off.
Bug	AUTORADAR-1505	Fix for CQ transfer for 12/14 bit mode which was leading to data corruption.
Bug	AUTORADAR-1836	Fix for sending async msg after downloading firmware over SPI.
Bug	AUTORADAR-1884	The MSS could miss out internal interrupts (like chirp available, frame start) when there a SPI command from the host. This might cause failure send out the chirps data correctly over the CSI/LVDS interface. Fix to correctly service all interrupts.
Bug	AUTORADAR-1648	Fix for SPI commands failing when there is no SFLASH connected and the file download (Meta image with patches) is performed over SPI.

3.2.3 mmWaveLink framework (Compared to DFP 1.0)

Type	Key	Description
Bug	MMWL-89	Remove IFA only saturation indicator option from RX

		saturation detector configuration
Feature	MMWL-91	Enable reading of second digital temperature sensor in xWR1642
Feature	MMWL-86	Provide API support for VIO supply indication and LDO short circuit monitor
Bug	MMWL-87	De-featuring of non-verbose mode of RX gain and phase monitor and TX gain and phase monitor
Bug	MMWL-80	Report mode field missing in Rx Gain Phase monitoring configuration structure
Feature	MMWL-81	Add the program mode parameter in dynamic chirp configuration APIs
Bug	MMWL-83	RX mixer input power monitoring configuration structure is incorrect
Feature	MMWL-78	Add an option in advanced frame config to support sub-frame trigger
Feature	MMWL-72	Update the RX IF Monitor report structure to include the expected IF gain
Feature	MMWL-73	Support for sub-frame trigger API
Feature	MMWL-75	Update the single shot GPADC read API (for external signals) to enable buffer option

3.2.4 mmWaveLink framework (Compared to DFP 1.1)

Type	Key	Description
Feature	MMWL-138	Added comments for IWR6843 ES1.0 support in link & enabled SOURCE_BROWSER filed in doxygen file to get doxygen comments for all link files.
Feature	MMWL-136	Added phase shifter thresholds parameter in TX BPM monitoring configuration structure and mentioned ranges in report structure.
Feature	MMWL-135	Added RL_AR_DEVICETYPE_18XX as device type for 18xx
Task	MMWL-133	Added note points for Frame config, MSS latent fault, MSS periodic fault and Analog fault APIs according to ICD
Feature	MMWL-132	Added calibApply filed in phase shifter calibration data restore/restore structure to apply calibration for each TX
Bug	MMWL-131	Fixed comment for rlRfDevCfg_t Asynch Event direction
Task	MMWL-130	Reordered reserved fields, after adding too many new fields in configuration structures & report structures.
Feature	MMWL-127	Added new filed chirpRowSelect in rlDynChirpCfg_t to configure 48 chirps in one command.
Task	MMWL-126	Merged SDK development branch to DFP_1.2 branch for

		18xx link test.
Bug	MMWL-125	Changed temperature filed in rlRfRunTimeCalibReport_t from unsigned to signed type.
Feature	MMWL-124	Added a new API to get the die ID of the device
Feature	MMWL-123	Added phase shifter calibration store & restore APIs
Bug	MMWL-121	Fixed mapping of pgaGain index numbers to the actual gain in rfPsLoopBackCnfg
Feature	MMWL-117	Added documentation for number of transmitters which can be turned on in profile configuration
Feature	MMWL-116	Updated code to conform to C90 standard
Feature	MMWL-112	TX BPM monitoring configuration API is enhanced to support phase shifter monitoring
Feature	MMWL-111	Added support for storing and restoring phase shifter calibration data
Feature	MMWL-107	Added an API option to disable PA LDO input (for 3 TX use case)
Feature	MMWL-106	Added 20 GHz sync in monitoring configuration option
Feature	MMWL-105	Added an API for advanced cascade configuration
Bug	MMWL-102	Added examples in MMWL porting section documentation
Bug	MMWL-99	Corrected the description of chirp config's startFreqVar, freqSlopeVar and profile config's freqSlopeConst
Bug	MMWL-98	Updated the comments for rlRfAnaFaultInjConfig API that this API will be supported in 1243 ES3.0 and 1642 ES2.0 onwards
Bug	MMWL-71	Corrected MSS latent fault structure report mode comments
Bug	MMWL-66	Updated comments for few variables in the structure rlRfInitCalConf_t which were not matching with ICD definition

3.3 Known Issues

Refer to Radar SS firmware release notes in firmware\radarss folder.

4 Migration Guide

This section explains the steps to migrate from DFP 1.0 release to this package

Impact	Change list
MEDIUM	Few reserved fields are removed and new parameters are added in rlGpAdcCfg_t structure. Application needs to accommodate this change while

	calling rlSetGpAdcConfig API
LOW	The reserved field in the structure rlRfTempData_t has been updated to indicate the value of second digital temperature sensor. This value will be updated only in xWR1642 devices.
MEDIUM	Only low power ADC mode is supported in 5 MHz device variants (for e.g. xWR1642). Application needs to accommodate this change before calling rlLowPowerModeCfg_t API
MEDIUM	RX IQ mismatch calibration is disabled by default in Radar SS firmware. Application needs to ignore the RX IQMM calibration status flag in the rlRfInitComplete_t structure which is sent by the device on RL_RF_AE_INITCALIBSTATUS_SB async event
LOW	A reserved field has been updated in rlRfLdoBypassCfg_t structure to indicate IO supply to the MMIC device. This field is used by the MMIC device for IO supply monitoring. Application needs to set this parameter to the correct value before calling the rlRfSetLdoBypassConfig API
MEDIUM	A reserved field is updated in rlAdvFrameSeqCfg_t structure (part of rlAdvFrameCfg_t structure). Application needs to set this parameter if it is using sub-frame triggered mode and use the new API rlSetSubFrameStart for triggering the sub-frames.
LOW	Report modes 1 and 2 and not supported in TX gain phase monitor (rlRfTxGainPhaseMismatchMonConfig) and RX gain phase monitors (rlRfRxGainPhMonConfig). Application needs to enable only report mode 0 when configuring these monitors
LOW	A reserved field is used in rlMonRxIfStageRep_t structure to indicate the expected IF gain. Application needs to take care of this when interpreting the report
MEDIUM	Few status flags in rlMonRxIntAnaSigRep_t structure have been made reserved. Application should ignore the reserved status flags
MEDIUM	A reserved field has been updated in rlFwVersionParam_t structure to indicate firmware build and debug versions. Application needs to read this parameter in rlDeviceGetMssVersion and rlDeviceGetRfVersion APIs for recording the correct firmware versions
MEDIUM	The 2 bits [bits 7:6] of rxGain parameter in rlProfileCfg_t structure have been used to indicate RF gain target to the MMIC device. Application needs to configure these bits appropriately before issuing the rlSetProfileConfig API
LOW	The IF loopback configuration structure rlRfIFLoopbackCfg_t has updated comments for IF loopback frequency. None of the structure parameters are modified or updated. Application needs to configure the parameters correctly for the correct IF loopback frequency.
LOW	A reserved parameter in rlDynChirpCfg_t and rlDynPerChirpPhShftCfg_t structures are modified and used as a new parameter programMode. Application needs to program this parameter appropriately when issuing rlSetDynChirpCfg and rlSetDynPerChirpPhShifterCfg APIs
LOW	Enabling of only IFA1 saturation monitor is no longer supported in rlRxSatMonConf_t structure. Application should not enable this option when configuring the RX IFA/ADC saturation monitor via the rlRfRxIfSatMonConfig

	API
LOW	The rllatentFault_t structure comments are updated to indicate the correct latent fault test status from Master SS. Application should take care of this and ensure reserved bits are not monitored.
MEDIUM	The parameter dataTransSize from the structure rlDevContStreamingModeCfg_t is removed and made reserved as this parameter was no longer used. Application should make the reserved parameter as 0 before calling the rlDeviceSetContStreamingModeConfig API
MEDIUM	The parameter tsMode from the structure rlTestSourceEnable_t has been removed and made reserved as this parameter was no longer used. Application should make the reserved parameter as 0 before calling the rlTestSourceEnable API
HIGH	Structure parameter for rlRfCalibDataStore and rlRfCalibDataRestore API has been changed, now this structure contains all three calibration chunk data in it. So application needs to create a structure variable of rlCalibrationData_t and pass it to these APIs to store/restore the calibration data.

This section explains the steps to migrate from DFP 1.1 release to this package

Impact	**Change list**
MEDIUM	Added new parameters in rlChanCfg_t API to support daisy chain mode of cascading the AWR devices.
MEDIUM	Added a new parameter txCalibEnCfg in rlProfileCfg_t API. This parameter indicates to the firmware if more than one transmitter is to be turned on during the TX power calibration. If the same number of transmitters are turned on during TX power calibration as in functional use-case, then it will improve the output power accuracy by about 1 dB. Application needs to set this parameter to the correct value before calling the rlProfileCfg API.
MEDIUM	Added a new option in rlRfLdoBypassCfg API which will disable the PA LDO in the LDO bypass mode. In this mode, the VIN_13RF2 should be shorted to VOUT_PA on board. This is needed specifically in simultaneous 3 TX use cases to improve the package reliability.
LOW	Defined a new bit for enabling/disabling TX phase shifter calibration in rlRfInitCalConf_t API. This calibration can be enabled/disabled on devices which have support for TX phase shifter.
LOW	Defined a new bit for TX phase shifter calibration status in rlRfInitComplete_t async event. The status bit will be set only on successful completion of TX phase shifter calibration on devices which support TX phase shifter. On other devices this status bit should be ignored.
MEDIUM	Added 2 new APIs, rlRfPhShiftCalDataStore and rlRfPhShiftCalDataRestore for storing and restoring phase shifter calibration values. These APIs are applicable only on those devices which have support for TX phase shifter.
MEDIUM	Updated rlMonTxBpmRep structure to include the TX phase shifter monitoring results.
MEDIUM	Updated rlMonPmclkloIntAnaSigRep structure to include 20 GHz sync in

	monitoring results.
LOW	Added new parameter ldoScEn in rlMonAnaEnables_t to enable/disable LDO short circuit monitor.
MEDIUM	Added new parameters in rlTxBpmConf structure to enable phase shifter monitoring configuration.
MEDIUM	Updated rlPmClkLoIntAnaSignalsMonConf structure to enable 20 GHz sync monitoring. This is applicable only on devices which support cascading.
LOW	Added a new bit for reading device Die ID status.
MEDIUM	Added new option in rlSetDynChirpCfg API which will configure 48 chirps dynamically in one call. This option speedup the dynamic chirp configuration.
MEDIUM	Added PD calibration in runtime calibration structure rlRunTimeCalibConf_t.
LOW	Added new rlBssEsmFault_t structure for BSS ESM fault report.
LOW	Enabled SOURCE_BROWSER filed in doxyfile to generate doxygen files for all mmwavelink files
MEDIUM	The host shall wait for MSS_BOOTERRORSTATUS_AE_MSG Async-event, while downloading the firmware over SPI, before sending any functional APIs to radar device.

5 Notes

5.1 File Formats

On xWR1243/xWR1443 ES3.0 devices onwards, the file format of the image downloaded/loaded, to the SFLASH/over the SPI, has been updated and unified with the entire range of mmWave sensors (similar to the xWR1642 range of sensors). Hence, a multicore image (metaImage) needs to be downloaded to the SFLASH or loaded over the SPI interface. The tools to be used to generate the final multicore image are available in the mmWave SDK (similar to the ones used to generate images for all variants of xWR1642 devices).

Device	**File Format**	
xWR1243 ES 3.0 device	Multicore image to be downloaded over SFLASH/SPI	
	SFLASH	mmwave_dfp_01_02_00_00\firmware\xwr12xx_metaImage.bin
	SPI	mmwave_dfp_01_02_00_00\firmware\xwr12xx_metaImage.h
xWR1443 ES 3.0 device	Multicore image needs to be generated using the toolset for the xWR1642 devices. The individual components that make the final multicore (metaImage) image are: RadarSS Firmware Patch: mmwave_dfp_01_02_00_00\firmware\radarss\ xwr1xxx_radarss_rprc.bin + MSS Application image in out2rprc format Please follow the SDK user guide for generation of multicore image from	

	individual RPRC images for xWR1642 device variants.

5.2 Serial Data FLASH Supported

AWR1243/xWR1443 ES1.0 and ES2.0 devices

The AWR1243/xWR1443 ES1.0 and ES2.0 devices work only with Spansion and Macronix devices. In particular, the Flash variants that have been tested to work with the ROM bootloader are

Spansion S25FL256S
Spansion S25FL132K0XNFB010
Macronix MX25L3233F
Macronix MX25R1635FZNIH0 (wide voltage part variant)

xWR1642 devices (ES1.0 and ES2.0) and xWR1243/xWR1443 ES3.0 devices

There are the following factors that will determine if the xWR16xx ROM bootloader will be able to interface and work with the SFLASH on xWR16xx devices:

Pre-requisite

SFLASH supports the SFDP command and responds with JEDEC compliant information regarding the capabilities and command set of the flash.

ROM assisted download to the FLASH (Device Management mode - SOP5)

- The ROM assisted download should work with all flash variants that allow for “Memory mapped mode” and “Page program command (0x2)” with 1 dummy byte and 24bit addressing.

- In addition to writing to the Flash, the ROM bootloader also support setting the “Quad Enable” bit for Spansion and Macronix variants (certain specific part variants only).

In case any of the above steps fail, the device supports “Boot mode – UART” which can be used to download an application onto the MSS RAM and execute, which can be used to program the SFLASH.

ROM based load from FLASH (Functional mode – SOP4)

The ROM bootloader performs the read from the FLASH based on the highest capability mode (Quad/Dual/Single) as published by the SFLASH in response to the SFDP command. The commands used also are as published by the SFDP response. So if the Quad read is supported, the expectation is that the Quad Enable (QE) bit is already set in the FLASH. The ROM bootloader would use the Quad mode to perform the read.

Recommendation

The flash vendors have an orderable part variant with the Quad Enable (QE) bit set. It is recommended to use these variants to work with TI mmWave SOCs.

Known Issues (xWR1642 ES1.0 devices)

The ROM bootloader in xWR1642 pre-production devices is not compatible with SFLASH variants that supports extended addressing mode. In particular, the “Number of Address length” field of the SFDP command response being non-zero is not supported. The total SFLASH addressable region in xWR1642 devices is 8 MBytes. So “Number of Address length” = 0 (corresponding to 3 bytes address length) will satisfy the addressable range. However, the compatibility issue is with variants that allow for “3 or 4 bytes address length”.

This incompatibility will be addressed in our production version of the xWR1642 silicon.