

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: 1.2.0.3	Binary
MSS Firmware (AWR1243 ES3.0 only)	ROM: 1.10.0.20, Patch: 1.2.0.2	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 metalmage binary (xwr12xx_metalmage.bin) for xWR1243 ES3.0 for boot over Flash metalmage header file (xwr12xx_metalmage.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.
Bug	AUTORADAR-1916	Fix for ADC data output over SPI does not work on xWR1243 ES3.0.
Bug	AUTORADAR-1648	Fix for SPI overrun issue, When BSS goes into a fault that results in an MSS IRQ, MSS does not correctly report ASYNC event to the host.

Bug	AUTORADAR-1938	Fix for LVDS calibration mode doesn't enable the data on Frame clock.
Bug	AUTORADAR-1913	Fix for periodic digital monitoring stability issue

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 rIRfDevCfg_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 rIDynChirpCfg_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 rIRfRunTimeCalibReport_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 rIRfAnaFaultInjConfig 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 rIRfInitCalConf_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 rGpAdcCfg_t structure. Application needs to accommodate this change while calling rISetGpAdcConfig API
LOW	The reserved field in the structure rIRfTempData_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 rILowPowerModeCfg_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 rIRfInitComplete_t structure which is sent by the device on RL_RF_AE_INITCALIBSTATUS_SB async event
LOW	A reserved field has been updated in rIRfLdoBypassCfg_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 rIRfSetLdoBypassConfig API
MEDIUM	A reserved field is updated in rIAdvFrameSeqCfg_t structure (part of rIAdvFrameCfg_t structure). Application needs to set this parameter if it is using sub-frame triggered mode and use the new API rISetSubFrameStart for triggering the sub-frames.
LOW	Report modes 1 and 2 are not supported in TX gain phase monitor (rIRfTxGainPhaseMismatchMonConfig) and RX gain phase monitors (rIRfRxGainPhMonConfig). Application needs to enable only report mode 0 when configuring these monitors
LOW	A reserved field is used in rIMonRxIfStageRep_t structure to indicate the expected IF gain. Application needs to take care of this when interpreting the report
MEDIUM	Few status flags in rIMonRxIntAnaSigRep_t structure have been made reserved. Application should ignore the reserved status flags
MEDIUM	A reserved field has been updated in rIFwVersionParam_t structure to indicate firmware build and debug versions. Application needs to read this parameter in rIDeviceGetMssVersion and rIDeviceGetRfVersion APIs for recording the correct firmware versions
MEDIUM	The 2 bits [bits 7:6] of rxGain parameter in rIProfileCfg_t structure have been used to indicate RF gain target to the MMIC device. Application needs to configure these bits appropriately before issuing the rISetProfileConfig API
LOW	The IF loopback configuration structure rIRfIFLoopbackCfg_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 rIDynChirpCfg_t and rIDynPerChirpPhShftCfg_t structures are modified and used as a new parameter programMode. Application needs to program this parameter appropriately when issuing rISetDynChirpCfg and rISetDynPerChirpPhShifterCfg APIs
LOW	Enabling of only IFA1 saturation monitor is no longer supported in rIRxSatMonConf_t structure. Application should not enable this option when configuring the RX IFA/ADC saturation monitor via the rIRfRxlIfSatMonConfig API
LOW	The rIlatentFault_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 rIDevContStreamingModeCfg_t is removed and made reserved as this parameter was no longer used. Application should make the reserved parameter as 0 before calling the rIDeviceSetContStreamingModeConfig API
MEDIUM	The parameter tsMode from the structure rITestSourceEnable_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 rITestSourceEnable API
HIGH	Structure parameter for rIRfCalibDataStore and rIRfCalibDataRestore API has been changed, now this structure contains all three calibration chunk data in it. So application needs to create a structure variable of rICalibrationData_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 rIChanCfg_t API to support daisy chain mode of cascading the AWR devices.
MEDIUM	Added a new parameter txCalibEnCfg in rIProfileCfg_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 rIProfileCfg API.
MEDIUM	Added a new option in rIRfLdoBypassCfg 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 rIRfInitCalConf_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 rIRfInitComplete_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, rIRfPhShiftCalDataStore and rIRfPhShiftCalDataRestore 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 rIMonTxBpmRep structure to include the TX phase shifter monitoring results.
MEDIUM	Updated rIMonPmclkloIntAnaSigRep structure to include 20 GHz sync in monitoring results.
LOW	Added new parameter ldoScEn in rIMonAnaEnables_t to enable/disable LDO short circuit monitor.
MEDIUM	Added new parameters in rITxBpmConf structure to enable phase shifter monitoring configuration.
MEDIUM	Updated rIPmClkLoIntAnaSignalsMonConf 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 rISetDynChirpCfg 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 rIRunTimeCalibConf_t.
LOW	Added new rIBssEsmFault_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.
LOW	Changes in names for "reservedxx" fields, Reserved fields in many APIs were changed from 'reserved0' to 'reserved' and so on. Since the reserved fields should not be accessed within the calling code, the impact of this change should be minimal to none
MEDIUM	Change in storage types for some existing fields, - rIRfRunTimeCalibReport_t->temperature field changed from unit16 to int16 - rIChirpCfg_t->freqSlopeVar field changed from int16 to unit16 - rITempMonConf_t->tempDiffThresh field changed from int16 to unit16 - rITxGainPhaseMismatchMonConf_t->txPhaseMismatchThresh field changed from int16 to unit16

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 (metalmage) needs to be downloaded to the SFLASH or loaded over the SPI interface.

Device	File Format	
xWR1243 ES 3.0 device	Multicore image to be downloaded over SFLASH/SPI	
	SFLASH	mmwave_dfp_01_02_00_01\firmware\xwr12xx_metalmage.bin
	SPI	mmwave_dfp_01_02_00_01\firmware\xwr12xx_metalmage.h

5.2 Serial Data FLASH Supported

AWR1243 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 variants)

For further information on supported SFLASH variants, please refer [SPRACH9](#).