MMWAVE DFP 00.07.00 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. It also includes RF evaluation tool (RadarStudio) to help user 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, IWR1642 SOCs/EVMs. The basic components of DFP are included in mmWave SDK.

> Note: This is a pre-production release and has undergone limited validation under nominal conditions for nominal device process

2. Release Overview

2.1 Platform and Device Support

The device and platforms supported with this release include:

Supported Devices	Supported EVMs
AWR1243 ES2.0	AWR1243BOOST : AWR1243 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 these other devices will not be a part of the DFP

2.2 Release contents and component versions

Component	Version	Type
RadarSS Firmware	1.7.0.4	Binary
MSS Firmware (AWR1243 only)	1.7.0.0	Binary
mmWaveLink Framework	0.7.1.1	Source and Library
FTDI Driver	2.10	Binary
RadarStudio	1.7.4.0	Executable
mmWaveLink example	0.7.1.0	Source and Binary
mmwlstudio	0.7.1.2	Library
Docs	Release Notes DFP user guide RadarStudio user guide Interface Control Document mmWaveLink Programmer's guide	PDF PDF PDF PDF Doxygen HTML

2.3 Directory Structure

Directory Name	Content
docs	AWR1xx_Radar_Interface_Control.pdf mmwave_dfp_ReleaseNotes.pdf mmwave_dfp_user_guide.pdf
firmware	RadarSS and Master SS firmware binary files
ti	mmWaveLink framework and example source code
rf_eval	RF evaluation firmware FTDI Driver RadarStudio GUI TSW1400 firmware files RadarStudio User's Guide.pdf

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

2.4.4 Radar Studio

The Radar Studio GUI supports RF/System evaluation of TI mmWave radar devices. Radar Studio is designed to communicate with TI mmWave Radar devices for characterization and performance evaluation. The Radar device is configured and controlled from the Radar Studio by sending commands to mmWave radar device over SPI. ADC data is captured using TSW1400 EVM Board and the post processing is performed through Matlab and the results can be viewed in the GUI.

RadarStudio GUI utilizes mmWaveLink framework to communicate with the device through FTDI FT4232H device.

Key features of the Radar Studio Software are

- Board Control (SOP Change, Reset Control)
- RS232 Connection to device
- Firmware Download Capability
- Execute mmWaveLink APIs
- Flash Programming
- Interaction with TSW1400 EVM
- Post-Processing

2.5 Tools and dependencies

mmWave DFP contains the pre-built binaries for all the software components. Below tools are required to run RadarStudio and to build mmWaveLink example application

Tools	Version	Download Link
HSDC Pro Software	4.2 or later	download link
Matlab Runtime Engine	8.5.1 only	download link

Microsoft Visual Studio 2010	10.0.30319.1	download link
FTDI Driver	2.10	Included in the package

2.6 Licensing

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

3. Release Contents

3.1 New Features

- Support for Autonomous run time calibrations which are scheduled on a user defined periodicity
- mmWaveLink example application

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 00.06.02.00)

Type	Key	Description
Bug	AUTORADAR-908	Defer the RADAR SS MB ACK to a point that the packet is processed or sent over SPI Avoid update to DFTREG2 in PBIST setup
Feature	AUTORADAR-907	Upgrading the compiler to 16.9.1 compiler, adding the build option to allow for SPI operation even if Flash is connected for the RF evaluation of 14xx.
Feature	AUTORADAR-881	Added error code response according to ICD Add support to get config using AR_DEV_CONF_GET_MSG

3.2.3 mmWaveLink framework

Type	Key	Description
Feature	AUTORADAR-803	Implement mmWaveLink example application
Feature	MMWSDK-450	Add Calibration Configuration and Report APIs and asynchronous events
Feature	MMWSDK-322	Add Phase Shift (PS), Power Amplifier (PA), Intermediate frequency(IF) loopback APIs

Feature	MMWSDK-457	Remove/deprecate APIs (rlSetCalibFuncDisableConfig, rlGetChannelConfig, rlGetAdcOutConfig and rlGetLowPowerModeConfig)

3.2.4 Radar Studio

- Allows control of calibrations and viewing of the status reports
- Allows setting of frequency limits over which the device operates

3.3 Known Issues

- Calibration report should have minimum periodicity of 100 msec.
- On AWR1243 and XWR1443 devices, the NError is signaled due to an ESM group2 error (from VIM) being reported in the startup sequence. This will be resolved in the production version of the device.
- rlGetAdvFrameConfig API only returns frame sequence configuration and doesn't return data path configuration.
- rlRfGetTemperatureReport API doesn't return error in xWR14xx and xWR16xx
- Only 1 profile can be configured using profile configuration API. 'cnt' should be 1 in rlSetProfileConfig API.

4. Notes

Serial Data FLASH Supported:

AWR1443/xWR1443
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 ES1.0 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.