

MMWAVE SDK Release Notes



Product Release 2.0.0

Release Date: Apr 26, 2018

Release Notes Version: 1.0

CONTENTS

- 1 Introduction
 - 2 Release overview
 - 2.1 What is new
 - 2.2 Platform and Device Support
 - 2.3 Component versions
 - 2.4 Tools dependency
 - 2.5 Licensing
 - 3 Release content
 - 3.1 New Features
 - 3.2 Issues fixed
 - 3.3 Limitations
 - 4 Test reports
 - 5 Installation instructions
 - 6 Package Contents
 - 6.1 Drivers
 - 6.2 Control
 - 6.3 Algorithm
 - 6.4 Usecases
 - 6.5 Demos
 - 6.6 Misc folders
 - 6.7 Scripts
 - 6.8 Firmware
 - 6.9 Tools
 - 6.10 Docs
 - 7 Related documentation/links
-

1. Introduction

The mmWave SDK enables the development of millimeter wave (mmWave) radar applications using TI mmWave sensors (see [list of supported Platform/Devices](#)). The SDK provides foundational components which will facilitate end users to focus on their applications. In addition, it provides few demo applications which will serve as a guide for integrating the SDK into end-user mmWave application.

Key mmWave SDK features:

- Building blocks
 - Full driver availability
 - Layered approach to programming analog front end
 - Catalog of mmwave algorithms optimized for C674x DSPs
- Demonstrations and examples
 - TI RTOS based
 - Out of box demo with easy configurability via TI cloud based GUI
 - Representation of "point cloud" and benchmarking data from demo via GUI
 - Profiles tuned to common end user scenarios such as Range, Range resolution, Velocity, Velocity resolution
- Documentation

mmWave SDK works along with the following external tools:

- Host tools including Pin Mux, Flashing utilities
- Code Composer Studio™ IDE for RTOS development

2. Release overview

2. 1. What is new

- Support for production devices mentioned in the "Platform and Device Support" section below
- Support for high secure devices (HS)
- Tools update
- Bug fixes

More details can be found in [NewFeatures](#) section.

2. 2. Platform and Device Support

The devices and platforms supported with this release include:

Supported Devices	Supported EVM
AWR1642 ES2.0	AWR1642BOOST - AWR1642 Evaluation Module
AWR1642_HS ES 2.0*	
IWR1642 ES2.0	IWR1642BOOST - IWR1642 Evaluation Module
IWR1642_HS ES 2.0*	

* xWR16xx HS devices need additional MMWAVE-SECDEV package

xWR16xx terminology is used in sections that are common for AWR16xx and IWR16xx

xWR16xx ES1.0 is not supported in this release. Prebuilt binaries in this release will not work on ES1.0 silicon.

xWR14xx is not supported in this release. No prebuilt lib/binaries/demo/radarss are provided for xWR14xx.

This release of mmWave SDK supports the foundation components for the devices mentioned in the table above . At system level, the mmWave SOC/EVM may interface with other TI ecosystem SOCs/Launchpads/EVMs and software for these other devices will not be a part of the mmWave SDK foundation components.

2. 3. Component versions



Components inside mmwave_sdk that have their own versions are shown below.

Component		Version	Type	Comment
mmwave sdk		2.0.0	Source and Binary	Overall package release version
RadarSS firmware (patch)		1.1.0	Binary	RadarSS firmware is in ROM. Only the patch is included in the mmwave sdk release
mmWaveLink Framework		1.1.0	Source and Binary	
FTDI		2.12	Binary	
Image Creator	gen_binrc32	1.0	Windows and Linux binary	
	out2rprc for xwr14xx	3.3	Windows binary	Need mono to run this on Linux
	out2rprc for xwr16xx	2.0	Windows binary	Need mono to run this on Linux
	Crc multicore image for xwr16xx	1.0	Windows and Linux binary	
	Multicore image generator for xwr16xx	1.0	Windows and Linux binary	

2. 4. Tools dependency

For building and using mmwave sdk the following tool versions are needed.

Tool	Version	Download link
CCS	7.4 or later	download link
TI SYS/BIOS	6.53.02.00	Included in mmwave sdk installer
TI ARM compiler	16.9.6.LTS	Included in mmwave sdk installer
TI CGT compiler	8.1.3	Included in mmwave sdk installer
XDC	3.50.04.43	Included in mmwave sdk installer
C64x+ DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x MATHLIB (little-endian, elf/coff format)	3.1.2.1	Included in mmwave sdk installer
Mono JIT compiler	4.2.1	Only for Linux builds
mmWave Radar Device support package	1.5.9 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
TI Emulators package	7.0.188.0 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
MMWAVE-SECDEV	2.0.0	Needed for xWR16xx high secure (HS) devices only Can be requested from link
Pinmux tool (optional)	Latest	Used to generate pinmux configuration for custom board https://dev.ti.com/pinmux (Cloud version)
Doxygen (optional)	1.8.11	Only needed if regenerating doxygen docs
Graphviz (optional)	2.36.0 (20140111.2315)	Only needed if regenerating doxygen docs

The following tools are needed at runtime

Runtime tool	Version	Link
Uniflash	Latest	Uniflash tool is used for flashing xWR1xxx devices Cloud version (Recommended): https://dev.ti.com/uniflash Offline version: http://www.ti.com/tool/uniflash
mmWave Demo Visualizer	Latest	TI Gallery APP for configuring mmWave sensors and visualizing the point cloud objects generated by the mmWave SDK demo https://dev.ti.com/mmWaveDemoVisualizer

2. 5. Licensing

Please refer to the `mmwave_sdk_software_manifest.html`, which outlines the licensing status for `mmwave_sdk` package.

3. Release content

3.1. New Features

- mmWave suite enhancement
 - Support for xWR16xx ES2.0 production device.
 - Increased L3 memory
 - Support for RadarSS patch binary
 - Drivers
 - Crypto driver for xWR16xx HS devices
 - mmWaveLink updated to support the RadarSS firmware version as noted above

This section describes the changes that are relevant for users migrating to the mmWave SDK 2.0.0 release from 1.2.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component/s	SubComponent	Behavior of Impact
xWR16xx ES2.0: Update DSS L3 memory allocations	Build, Drivers	Linker command file, Demo makefiles, SOC driver	There is no need to reserve 1 bank of L3 memory (128KB) for RadarSS in xWR16xx production devices and can be claimed back for application usage. The linker command file and the SDK metalmage (ccsdebug and mmW demo) have been modified to reflect this increased memory usage. Application code should use the right metalmage syntax to gain access to this increased memory space. When using the SDK provided default linker command file, they can use this space for code/memory as per their requirement.
Reserve memory for CQ0 in CHIRP_INFO buffer	Demos	mmW demo	CQ0 is reserved for RadarSS usage and application should program the CQ1 memory offset to start after CQ0 end to avoid memory corruption. mmW demo has been updated to reflect this correct programming
Update profiles to use low power ADC mode	Demos, mmWave Demo Visualizer	mmW demo	Low power mode has to be enabled for xWR16xx else RadarSS firmware will throw an appropriate error. This change is reflected in the packaged mmW demo profiles and the auto-generated profiles by the mmW demo Visualizer. This is mandatory and existing user of xWR16xx should take care of this in their custom application code and/or custom profile configuration script for mmW demo. When using the perl script mmwDemo_xwr16xx_update_config.pl to automatically update the existing 1.2 based SDK profiles, it will throw an error if the "lowPower" command is not passed with correct setting.
DFP 1.1 integration: Boot time IQMM calibration should disabled in mmWave	Control	mmWave API	This is handled internally by the mmWave control layer and no change is required from the application.
DFP 1.1 integration: De-featuring of non-verbose mode of RX gain and phase monitor and TX gain and phase monitor	Control	mmWaveLink API	Report mode 1 and 2 are no longer supported for RX gain phase monitor and TX gain phase monitors
rIRfCalibDataStore and rIRfCalibDataRestore shouldnt expose the chunk level detail to application	Control	mmWaveLink API	rIRfCalibDataRestore and rIRfCalibDataStore API has been updated to explicitly request array of structure rICalDataStore_t from the application code
Structure rIRfInitomplete_t name changed to rIRfInitComplete_t (was a typo in the structure name)	Control	mmWaveLink API	Compilation will fail if application code refers to rIRfInitomplete_t structure. Change rIRfInitomplete_t to rIRfInitComplete_t wherever it is used in application code and recompile
xWR16xx ES2.0: ROM BL leaves WDT in reset state and it needs to be pulled out of reset in application space	Drivers	Watchdog	This is handled internally by the Watchdog driver and no other change is needed by the application
remove UART from DSS on xWR16 mmw demo	Demos	mmW demo	UART driver was initialized in DSP to allow mmW demo to ship detected objects over the DSS UART instance. This mode has been deprecated to allow more code space on DSP for signal processing algorithm.

3.2. Issues fixed



The following issues from previous releases were fixed in this release

Issue Type	Key	Summary
Bug	MMWSDK-1156	CANFD: Transmit complete interrupt can generate multiple callbacks
Bug	MMWSDK-1162	Docs: profileCfg.freqSlopConst description and units are unclear
Bug	MMWSDK-1133	mmw demo: Reserve memory for CQ0 in CHIRP_INFO buffer
Bug	MMWSDK-1151	Update profiles to use low power ADC mode
Bug	MMWSDK-1153	DC calibration (antenna coupling signature removal) mismatch between CLI documentation and implementation
Bug	MMWSDK-1182	chirp startFreqVar parsing not correct in mmw demo
Bug	DBGTRC-3696	XDS110 UART data duplication error observed at 921600 baud rates (the fix is in "TI Emulators package" version mentioned in the tools section)

Known Issues

The following issues are known at the time of this release.

Issue Type	Key	Summary	Comments
Bug	MMWSDK-1078	Limitation in mmW demo + LVDS instrumentation use case	See limitations section below
Task	MMWSDK-533	GUI of mmw demo running slow from Firefox browser	<u>Workaround:</u> Please switch to Chrome browser.
Story	MMWSDK-319	CAN driver: DMA mode is not supported	
Story	MMWSDK-252	UART driver has not tested for Data Length 5 and 6	
Bug	MMWSDK-1157	Rare failure seen in UART loopback driver unit test - HW limitation	

3.3. Limitations

Some of these limitations are captured in the "known issues" list shown in previous section.

1	CAN driver: <ul style="list-style-type: none"> DMA and FIFO mode are not supported
2	CANFD driver: <ul style="list-style-type: none"> DMA and Timestamping are not supported
3	CBUFF/CSI2/LVDS: <ul style="list-style-type: none"> Driver does not support the following functionality: <ul style="list-style-type: none"> Multiple packets 3 channels CSI2: ADC streaming has only been tested under 1 configuration in csi_stream usecase
4	CRC driver: "Auto" mode is not implemented.
5	DMA driver: MPU and Parity Feature not implemented.
6	EDMA driver: Privilege feature not implemented.
7	HWA driver: Any modes/algorithm outside the scope of mmWave demo are not tested (however they are implemented in the driver).
8	I2C driver: Verified only loopback mode on mmWave device TI EVM (however all features are implemented in the driver).



9	QSPI/QSPI Flash driver: <ul style="list-style-type: none">▪ dual-Read/Quad read in configuration mode is not supported▪ setting write protections bits is not supported
10	SPI (MIBSPI) Limitations: <ul style="list-style-type: none">• For xWR14xx, MIBSPI is only supported on SPIA, hence driver only supports SPIA. SPIB is not supported in xWR14xx. In xWR16xx, both instances are MIBSPI and are supported within the driver.• When MIBSPI mode is used in 4-pin slave mode, for every CHARLEN (8 bits or 16 bits), CS signal(from Master) has to be toggled and 2 VBUSP cycles need to be inserted. This needs to be taken care on SPI master device.
11	DMA based transactions are not supported for CRC and Mailbox driver.
12	mmW demo: See demo's doxygen page for more details.
13	mmW demo + LVDS instrumentation: <ul style="list-style-type: none">▪ This feature is not available for xWR14xx due to ADC Buffer being unavailable for streaming while datapath processing is active.▪ For xWR16xx, CQ cannot be streamed out reliably when datapath processing is also enabled. The data corruption for CQ data over LVDS lanes is seen more pronounced when multiple chirps/chirp event is enabled

4. Test reports

Results of the unit tests can be found in the docs/test folder. The test folder has separate folders for all the SoC variants. System level test is run using demos.

5. Installation instructions

mmwave_sdk installer is available as a Windows Installer and a Linux installer.

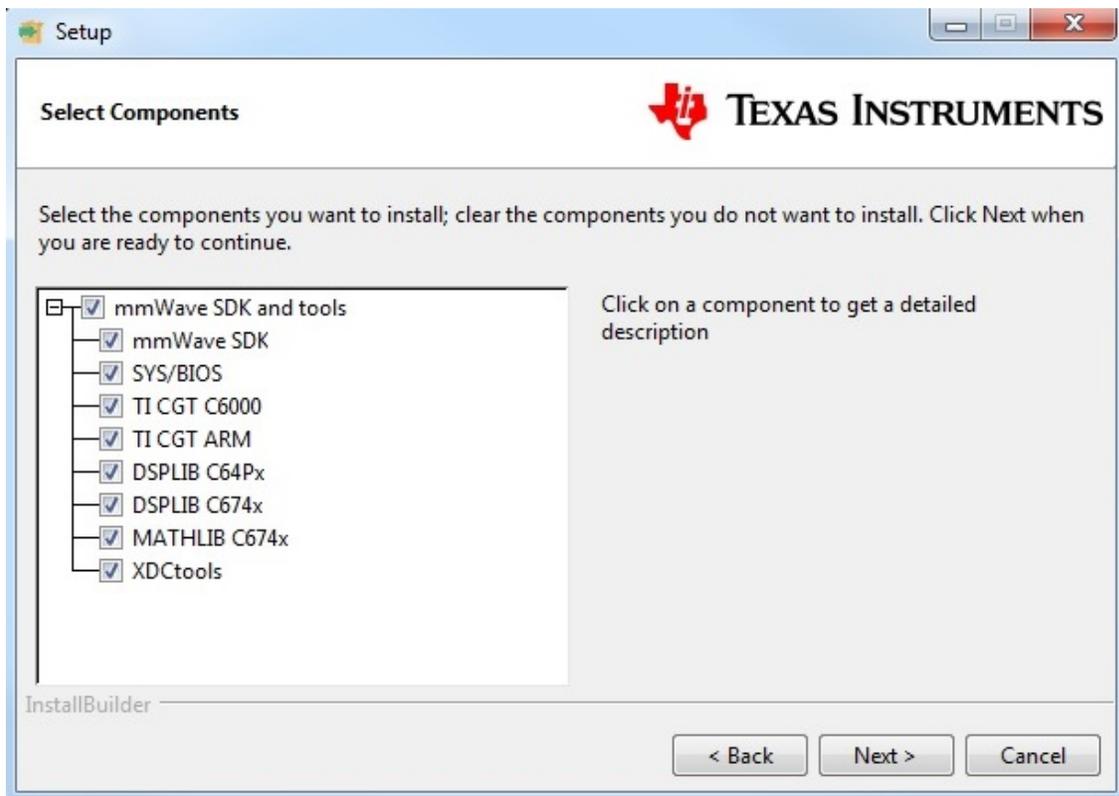
- **mmwave_sdk_<version>-Windows-x86-Install.exe: Windows installer verified on Windows 7 and Windows 10 machines**
- **mmwave_sdk_<version>-Linux-x86-Install.bin: Linux installer verified on Ubuntu 14.04 & Ubuntu 16.04 64 bit machines.**

Depending on your development environment run the appropriate installer

- In Windows environment, double clicking the Windows installer from Windows explorer should start the installation process
- If in Linux environment,
 - Enable execute permission for the Linux installer by running "chmod +x mmwave_sdk_<version>-Linux-x86-Install.bin" command
 - Run the installer using "./mmwave_sdk_<version>-Linux-x86-Install.bin" command

Installation steps:

- Setup
- Choose Destination Location: Select the folder to install (default is c:\ti on windows and ~/ti on linux)
- Select Components: The installer includes all the tools needed for building the mmWave SDK. You should see a screen like below (except that each component will also have version information appended). The only reason to deselect a tool is if the exact tool version is already installed in the destination folder.



- Review installation decisions
- Ready to install
- Once installation starts all the selected components will be installed (if a component with the same version exists in the destination folder it will be overwritten)
- Installation complete

After the installation is complete the following folder structure is expected in the installation folder (except that each component will have appropriate version number in place of the VERSION placeholder shown below)

- ▼ ti
 - > bios_[VERSION]
 - > dsplib_c64Px_[VERSION]
 - > dsplib_c674x_[VERSION]
 - > mathlib_c674x_[VERSION]
 - > mmwave_sdk_[VERSION]
 - > ti-cgt-arm_[VERSION].LTS
 - > ti-cgt-c6000_[VERSION]
 - > xdctools_[VERSION]_core

Under the mmwave_sdk <ver> folder you should have the following directory structure.

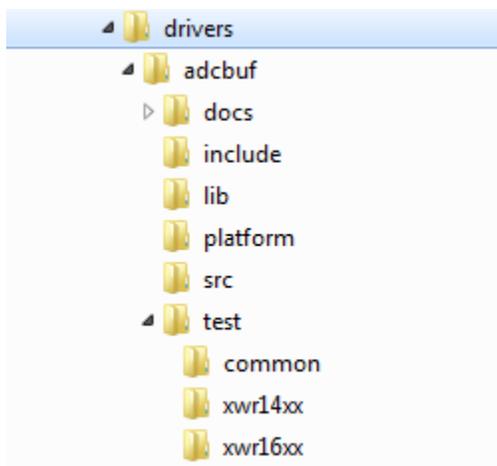
- > .metadata
- ▼ docs
 - .relnotes_archive
- > test
- ▼ firmware
 - radarss
- ▼ packages
- ▼ scripts
 - > ImageCreator
 - unix
 - windows
- ▼ ti
 - > alg
 - common
 - > control
 - > demo
 - > drivers
 - > platform
 - > utils
- ▼ tools
 - > ftdi

6. Package Contents

The mmwave sdk release package contains the following major components/folders.

6. 1. Drivers

Drivers can be found under `mmwave_sdk_<ver>/packages/ti/drivers` folder. The directory structure of all drivers is similar to the one shown below for `adcbuf` (some drivers do not have a unit test as shown in the table below)



- docs: Driver API documentation done with doxygen
- include: Include files
- lib: Prebuilt libraries
- platform: Platform files
- src: Driver Source files
- test/<platform>: Unit test src files and prebuilt unit test binary for that <platform: xwr14xx, xwr16xx>
- test/common: Unit test src files common for all platforms
- driver base folder has external header file, make files

Content of each driver is indicated in the table below.

Component	Source & prebuilt library	API Document (doxygen)	Unit test (source & prebuilt binary)
ADCBUF	X	X	X
CAN	X	X	X
CANFD	X	X	X
CBUFF/LVDS	X	X	X
CRC	X	X	X
CRYPTO ¹	X	X	X
CSI2	X	X	X
DMA	X	X	X
EDMA	X	X	X
ESM	X	X	
GPIO	X	X	X
HWA	X	X	X
I2C	X	X	X
MAILBOX	X	X	X
OSAL	X	X	
PINMUX	X	X	
QSPI	X	X	X
QSPIFLASH	X	X	X
SOC	X	X	
SPI	X	X	X
UART	X	X	X
WATCHDOG	X	X	X

¹ CRYPTO is only supported on xWR16xx high secure (HS) devices

No pre-built libraries or binaries for xwr14xx in this release

6. 2. Control

Control modules can be found under mmwave_sdk_<ver>/packages/ti/control folder. Content of each of the control module is shown below

Component	Source & Prebuilt Library	API Document (doxygen)	Unittest (source & prebuilt binary)
-----------	---------------------------	------------------------	-------------------------------------

mmwavelink framework	X	X	X
mmwave high level api	X	X	X

No pre-built libraries or binaries for xwr14xx in this release

6.3. Algorithm

Algorithms can be found under mmwave_sdk_<ver>/packages/ti/alg folder. Currently algorithms applicable for mmwave functionality are provided under this folder:

Component	Source & Prebuilt Library	API Document (doxygen)	Unittest (source & prebuilt binary)
gtrack	X	X	X
mmwavelib	X	X	X

No pre-built libraries or binaries for xwr14xx in this release

6.4. Usecases

Useases can be found under mmwave_sdk_<ver>/packages/ti/drivers/test folder.

Component	Source	API Document (doxygen)	Unittest (source & prebuilt binary)
csi_stream (IWR14xx only)	X	X	X
dsp_edma (xWR16xx only)	X	X	X
hwa_edma (xWR14xx only)	X	X	X
mem_capture	X	X	X

No pre-built libraries or binaries for xwr14xx in this release

6.5. Demos

Demos can be found under mmwave_sdk_<ver>/packages/ti/demo/<platform>. The following demos are included in the mmwave sdk package. Details on running demos can be found in the mmwave_sdk_user_guide.

Component	Source & Prebuilt Binary	Demo document (doxygen)	Demo GUI
mmw	X	X	X

No mmw demo for xwr14xx in this release

6.6. Misc folders

Following folders are also part of mmwave_sdk_<ver>/packages/ti folder.

- common: Common header files needed across all components
- platform: platform specific files
- utility: Contains
 - ccs debug utility which is the MSS/DSSbinary that needs to be flashed when connecting/developing using CCS (details can be found in mmwave_sdk_user_guide)
 - cli which is the cli helper utility used by the demos
 - cycleprofiler which is the helper utility used for profiling the various components inside the SDK
 - hsiheader which is a helper utility that creates a header for the data to be shipped over LVDS lanes.

- testlogger which is the helper utility for driver unit tests

6.7. Scripts

Build scripts can be found in `mmwave_sdk_<ver>/packages/scripts` folder. Build instructions can be found in `mmwave_sdk_user_guide`.

6.8. Firmware

RadarSS firmware for all supported devices is included under `mmwave_sdk_<ver>/firmware/radarss` folder. Procedure to flash the radarss is covered in the `mmwave_sdk_user_guide`.

6.9. Tools

The following tools are included in the release in binary form. These can be found under `mmwave_sdk_<ver>/tools` folder.

- **Ftdi:** These Windows PC drivers are needed when interfacing to the board via MMWAVE-DEVPACK

6.10. Docs

`mmwave_sdk_<ver>/docs` folder contains important documents related to the release such as

- `mmwave_sdk_software_manifest.html`: Software Manifest
- `mmwave_sdk_release_notes.pdf`: Release Notes (this document)
- `mmwave_sdk_user_guide.pdf`: User guide
- `mmwave_sdk_module_documentation.html`: Links to individual module's documentation

`mmwave_sdk_<ver>/docs/relnotes_archive` contains release notes from previous releases

`mmwave_sdk_<ver>/docs/test` folder contains test results for each SoC. Each SoC folder in turn may contain multiple test group folders (such as `module_test`, `alglib_test`) which have the following files

- `Report.html`: Detailed Test report with links to logs
- `*.log`: Test logs for unit tests

7. Related documentation/links

Other than the documents included in the `mmwave_sdk` package the following documents/links are important references.

- SoC links:
 - [AWR1642](#)
 - [IWR1642](#)
- EVM links:
 - [AWR1642BOOST](#)
 - [IWR1642BOOST](#)
 - [MMWAVE-DEVPACK](#)