

## MMWAVE SDK Release Notes



**Product Release 3.6 LTS**  
**Release Date: June 13, 2022**  
**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 Migration section
    - 3.3 Issues fixed
    - 3.4 Known Issues
      - 3.4.1 mmWave Suite/Demos Known Issues
      - 3.4.2 RadarSS Known Issues
        - 3.4.2.1 RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx
        - 3.4.2.2 RadarSS firmware for xwr68xx
    - 3.5 Future Patch Release
    - 3.6 Limitations
      - 3.6.1 mmWave Suite/Demos Limitations
      - 3.6.2 RadarSS Limitations
        - 3.6.2.1 RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx
        - 3.6.2.2 RadarSS firmware for xwr68xx
  - 4 Test reports
  - 5 Installation instructions
    - 5.1 Installation in GUI mode
    - 5.2 Installation in unattended command line mode
    - 5.3 Post Installation
  - 6 Package Contents
    - 6.1 Drivers
    - 6.2 Control
    - 6.3 Datapath
    - 6.4 Algorithm
    - 6.5 Usecases
    - 6.6 Demos
    - 6.7 Misc folders
    - 6.8 Scripts
    - 6.9 Firmware
    - 6.10 Tools
    - 6.11 Docs
  - 7 Related documentation/links
- 



## 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



**NOTICE:** This software product is used to configure TI's mmWave devices, including RF emissions parameters for such devices. Note that many countries or regions impose regulations governing RF emissions. Users are responsible for understanding local RF emission regulations and operating the product within those regulations.

## Release overview

### What is new

- Support for devices mentioned in the "Platform and Device Support" section below
- New features can be found in [New Features](#) section.
- Bug fixes
- Tools update

### Platform and Device Support

The devices and platforms supported with this release include:

Supported Devices	Supported EVM
AWR6843 ES2.0	AWR6843ISK (Rev B)+MMWAVEICBOOST (Rev B): AWR6843 Evaluation Module
AWR6843 QM ES2.0	N/A <sup>2</sup>
AWR6443 ES2.0	N/A <sup>2</sup>
AWR6843 HS ES2.0 <sup>1</sup>	N/A <sup>2</sup>
AWR6843AOP ES2.0	N/A <sup>2</sup>
AWR6843AOP HS ES2.0 <sup>1</sup>	N/A <sup>2</sup>
AWR1843 ES1.0	AWR1843BOOST - AWR1843 Evaluation Module Rev C
AWR1843_HS ES 1.0 <sup>1</sup>	N/A <sup>2</sup>
AWR1843AOP ES2.0	N/A <sup>2</sup>
AWR1843AOP_HS ES2.0 <sup>1</sup>	N/A <sup>2</sup>
AWR1642 ES2.0	AWR1642BOOST - AWR1642 Evaluation Module Rev B
AWR1642_HS ES 2.0 <sup>1</sup>	N/A <sup>2</sup>
AWR1443 ES3.0 <sup>3</sup>	AWR1443BOOST - AWR1443 Evaluation Module Rev B
IWR6843 ES2.0	IWR6843ISK (Rev B)+MMWAVEICBOOST (Rev B): IWR6843 Evaluation Module
IWR6843 SIL-2 ES2.0	N/A <sup>2</sup>



IWR6443 ES2.0	N/A <sup>2</sup>
IWR6843_HS ES2.0 <sup>1</sup>	N/A <sup>2</sup>
IWR6843AOP ES2.0	IWR6843AOPEVM (Rev F) + MMWAVEICBOOST (Rev B): IWR6843AOP Evaluation Module
IWR6843AOP_SIL2 ES2.0	N/A <sup>2</sup>
IWR6843AOP_HS ES2.0 <sup>1</sup>	N/A <sup>2</sup>
IWR1843 ES1.0	IWR1843BOOST - IWR1843 Evaluation Module Rev C
IWR1642 ES2.0	IWR1642BOOST - IWR1642 Evaluation Module Rev B
IWR1642_HS ES 2.0 <sup>1</sup>	N/A <sup>2</sup>
IWR1443 ES3.0 <sup>3</sup>	IWR1443BOOST - IWR1443 Evaluation Module Rev B
N/A	DCA1000EVM (Rev A) - mmWave Real-time data-capture adapter

<sup>1</sup> High Secure (HS) devices need additional MMWAVE-SECDEV package

<sup>2</sup> Device was internally validated using a device pin compatible EVM

<sup>3</sup> mmW (OOB) demo is not supported for this device in this release



xWR terminology is used in sections that are common for AWR and IWR devices

**Silicon versions other than the ones in the table above are not supported**



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.

## Component versions

Components inside mmwave\_sdk that have their own versions are shown below.

Component	Version	Type	Comment	
mmwave sdk	3.6	Source and Binary	Overall package release version	
RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx	1.2.6.11	Binary	RadarSS firmware is in ROM. Only the patch is included in the mmwave sdk release	
RadarSS firmware for xwr68xx	6.3.2.x	Binary		
mmWaveLink Framework	1.2.6	Source and Binary		
FTDI	2.12	Binary		
Image Creator	gen_binrc32	1.0	Windows and Linux binary	
	out2rprc	2.0	Windows binary	Need mono to run this on Linux
	Crc multicore image	1.0	Windows and Linux binary	
	Multicore image generator	1.0	Windows and Linux binary	
	create_ConfigRPRC	1.1	Windows and Linux binary	MSS Logger is disabled by disabled through the binary in the latest version

## Tools dependency

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

Tool	Version	Download link
CCS	9.3 or later	<a href="#">download link</a>



TI SYS/BIOS	6.73.01.01	Included in mmwave sdk installer
TI ARM compiler	16.9.6.LTS	Included in mmwave sdk installer
TI CGT compiler	8.3.3	Included in mmwave sdk installer
XDC	3.50.08.24	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	1.6.1 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	2.0.1 or later	Needed for high secure (HS) devices only Can be requested from <a href="#">link</a>
Pinmux tool (optional)	Latest	Used to generate pinmux configuration for custom board <a href="https://dev.ti.com/pinmux">https://dev.ti.com/pinmux</a> (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
DCA1000EVM CLI	1.0.0	Part of MMWAVE STUDIO package (use <a href="#">MMWAVE-STUDIO</a> version 2.1.0 or later)

The following tools are needed at runtime

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

## Licensing

Please refer to the [mmwave\\_sdk\\_software\\_manifest.html](#), which outlines the licensing status for mmwave\_sdk package.



## Release content

### New Features

- Added support for AWR1843AOP ES2.0, AWR6843 QM ES2.0, AWR6843AOP HS ES2.0, AWR1843AOP HS ES2.0, IWR6843AOP HS ES2.0, IWR6843AOP\_SIL2 ES2.0, AWR6443 ES2.0, AWR6843AOP ES2.0, AWR6843 HS ES2.0, IWR6843 SIL-2 ES2.0 devices.
- mmWave Suite enhancement
  - Drivers
    - I2C:
      - Added support for the sub address mechanism in I2C slave mode.
    - EDMA: ISR latency reduction using transfer completion code context passed to the transfer complete ISR.
  - mmWave data processing layers
    - Antenna spacing based correction factor for xyz estimation improvement.
    - New beta version DPU/DPC to support compression/decompression: Objdetcmphwa DPC invoking rangeproccmp HWA, dopplerproccmp HWA and aoaproccmp HWA DPUs.
- mmWave Demos enhancement
  - mmW demos:
    - Added support for antenna board variants (ISK, ODS, AOP) and corresponding antenna spacing factor.
    - Added low power APIs library as libsleep utils library and updated the xWR68xx demo to take CLI command inputs.
    - Enabled RTRIM programming feature for all Gen1 devices by providing a switchable macro. xWR68xx uses a DFP API to do the same and other Gen1 devices uses rtrim utils library to set and get RTRIM value.
    - Custom beta demo - xwr64xx\_compression, to demonstrate compression/decompression of radar data cube, based on the existing xwr64xx demo.

### Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.6.0\_LTS release from 3.5.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component /s	Subcomponent	Behavior of impact
New mandatory CLI command 'compressCfg' added to mmW demo CLI of xwr18xx & xwr64xx demos, to configure compression feature	Demos	mmW	See SDK user guide on details about this new command.  User can access sample CFG files with compression command, at ti/demo/<platform>/mmw/profiles/.
BOARD_DEFINE added to mmwave_sdk_<device_type>.mak file at mmwave_sdk_<ver>\packages\ti\common, to support antenna variant (ISK, ODS, AOP) selection	Common	mmW	See mmwave_sdk_<device_type>.mak files at mmwave_sdk_<ver>\packages\ti\common. The antenna spacing factor for xwr68xx variants (ISK, ODS, AOP) is selected in antenna_geometry.c file, based on this definition. To retain the default xyz estimation calculations, set this defines to NO_ANTENNA_CORRECTION, which is the default definition.
New library libsleep added into utils to support low power operations on xwr68xx device	utils	libsleep	See SDK user guide on details about the new library.  User has the option to enable/ disable the usage of this library.
New optional CLI commands 'idlePowerDown' and 'idlePowerCycle' to xwr68xx demo to support low power mode software techniques	Demo	mmW	See SDK user guide on the details about the new commands.  User can access sample CFG files with power APIs at ti/demo/platform/mmw/profiles.
SYS_COMMON_XWR68XX_LOW_POWER_MODE_EN added to mmwave_sdk_<ver>\packages\ti\common\sys_common_xwr68xx_mss.h file to support enable/disable of low power APIs	Common	mmW	See SDK user guide on details about the library.  User has the option to enable/ disable the usage of this library by defining/ undefining this macro
New library rtrim added into utils to support RTRIM programming on all Gen1 devices	utils	rtrim	See SDK user guide on details about the new library.  User has the option to enable/ disable the usage of this library.
SYS_COMMON_RTRIM_MODIFY_EN added to mmwave_sdk_<ver>\packages\ti\common\sys_common.h file to support enable/disable of RTRIM programming	Common	mmW	See SDK user guide on details about the library.  User has the option to enable/ disable the usage of this library by defining/ undefining this macro

### Issues fixed



This section captures the issues that were fixed in this release for mmWave Suite/Demos. For RadarSS related issues that are fixed as part of this release can be found in RadarSS release notes included under `mmwave_sdk_<ver>/firmware/radarss` folder.

Issue Type	Key	Summary
Bug	MMWSDK-2503	CRC_open does not initialize errCode
Bug	MMWSDK-2334	Missing AOP Pinmux definition in MMWAVE-SDK 3.x
Bug	MMWSDK-2555	EDMA Transfer Completion Interrupt Latency
Bug	MMWSDK-2642	Range/Noise profile deviation with height estimation correction changes, when clutter removal is enabled
Bug	MMWSDK-2554	AWR1843AOP demo elevation incorrect sign
Bug	MMWSDK-2550	Height computation error for overhead scenarios
Bug	MMWSDK-2548	Conflict with TopRCM register space in the device while BSS is booting up
Bug	MMWSDK-2445	Allocation of cfarDopplerDetOutBitMaskSize in DPC can cause chain to crash
Bug	MMWSDK-2470	QSPIFlash Doxygen incorrect for QSPIFlash_singleRead
Bug	MMWSDK-2571	SDK UG supported EVMs to use web names
Bug	MMWSDK-2601	GPIO interrupt mask causes assert issue for pins with no interrupt provision
Bug	MMWSDK-2582	Crash at AoAProchWA_peakGrouping in aoa2dprochwa DPU
Bug	MMWSDK-2640	Additional delay of 3.2mS with change in image size

## Known Issues

### mmWave Suite/Demos Known Issues

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

Issue Type	Key	Summary	Comments
Bug	MMWSDK-1542	AoA DPU: RX phase calibration does not work when measurement is done with less than the possible max antenna size (#tx < 3, #rx < 4 in case of IWR6843)	Documented procedure in past releases always mentioned that all the available antennas on the device be turned on for measurement - so this is not creating any deviation from that. This is listed as known issue so that user are aware of the limitation.
Bug	MMWSDK-2444	Range bias correction erroneous	The Range bias offset correction feature in the mmw demo application corrects the range for only the target object on which the correction was measured. This correction ideally should be applied to all range bins since its an offset. This is listed as known issue so that user are aware of the limitation.
Bug	MMWSDK-2643	Detection performance degradation observed with compression demo	The custom beta demo xwr64xx_compression has a degraded performance and is partly attributed to the lossy compression. This is a known issue and will be addressed in subsequent updates
Feature	MMWSDK-2577	SBL for AWR1443 Device	SBL needs update on the memory allocation from L3 bank to TCMA and TCMB RAM. This is a known limitation for the SBL addition for xwr14xx and this would be handled in the next release.
Bug	MMWSDK-2615	Bug in UART driver Baud rate calculation	The existing uartsci baud rate calculation doesn't match with the TRM. Fixing the UART baud rate calculation according to TRM is causing UART transmission errors which is being looked into. This is a known issue and will be addressed in the next update.

### RadarSS Known Issues

#### RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx

Users should refer to the RadarSS release notes included under `mmwave_sdk_<ver>/firmware/radarss` folder for known issues in this release of RadarSS firmware.

#### RadarSS firmware for xwr68xx

Users should refer to the RadarSS release notes included under `mmwave_sdk_<ver>/firmware/radarss` folder for known issues in this release of RadarSS firmware.

### Future Patch Release



Please refer to Software/mmWave Sensors section of tirex for any further patch releases on 03.06.00.00\_LTS version  
<https://dev.ti.com/tirex/explore/node?node=AHJY4qNCowO17wH-P2ICKQ>

## Limitations

### mmWave Suite/Demos 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"> <li>DMA and FIFO mode are not supported</li> </ul>
2	CANFD driver: <ul style="list-style-type: none"> <li>DMA and Timestamping are not supported</li> <li>AWR6843: 2nd CANFD instance is not supported</li> </ul>
3	CBUFF/CSI2/LVDS: <ul style="list-style-type: none"> <li>Driver does not support the following functionality: <ul style="list-style-type: none"> <li>Multiple packets</li> <li>3 channels</li> </ul> </li> <li>CSI2: ADC streaming has only been tested under 1 configuration in csi_stream usecase</li> </ul>
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 loopback mode on all mmWave device TI EVM (however all features are implemented in the driver) and master mode using address scanning on all devices. Note that default xWR1642 BOOST EVM does not have a direct connection to I2C devices on the board from the xwr1642 device and this I2C scan test in driver will fail until board modifications are done.
9	QSPI/QSPI Flash driver: <ul style="list-style-type: none"> <li>dual-Read/Quad read in configuration mode is not supported</li> <li>setting write protections bits is not supported</li> </ul>
10	SPI (MIBSPI) Limitations: <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>
11	DMA based transactions are not supported for CRC and Mailbox driver.
12	mmW demo: See demo's doxygen page for more details.
13	Processing chain + LVDS instrumentation: <ul style="list-style-type: none"> <li>This feature is not available for xWR14xx due to ADC Buffer being unavailable for streaming while datapath processing is active.</li> <li>For xWR16xx, xWR18xx, xWR68xx, 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. Note that, for this reason, default mmW demo does not allow LVDS streaming and multiple chirps/chirp event to be enabled in the same configuration.</li> </ul>
14	xwr64xx_compression demo: <ul style="list-style-type: none"> <li>This is a custom, beta demo version based on the existing xwr64xx demo</li> <li>Compression is a HWA specific feature and hence, runs only on demos that use objdethwa DPC (rangeprochwa, dopplerprochwa DPUs), on devices with upgraded HWA engine supporting this feature</li> <li>Compression, if enabled, is bound to increase processing time notably (due to decompression being performed twice - before Doppler FFTs in dopplerprocdcmp and aoaprocdcmp DPUs)</li> <li>Supported only on interleaved ADC data format</li> <li>Features not supported/not verified using this chain: calibDcRangeSig, clutterRemoval, calibration parameter generation and range azimuth heatmap generation</li> </ul>

- |    |   |
|----|---|
| 15 | Doxygen Module Documentation <ul style="list-style-type: none"><li>• Search bar doesn't return any results in the doxygen API guide</li></ul> |
|----|---|

## RadarSS Limitations

### RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx

Users should refer to the RadarSS release notes included under `mmwave_sdk_<ver>/firmware/radarss` folder for "Unsupported Features and APIs" in this release of RadarSS firmware.

### RadarSS firmware for xwr68xx

Users should refer to the RadarSS release notes included under `mmwave_sdk_<ver>/firmware/radarss` folder for limitations in this release of RadarSS firmware.

## 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.

## 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 16.04 and Ubuntu 18.04 64 bit machines.**

## Installation in GUI mode

Depending on your development environment run the appropriate installer

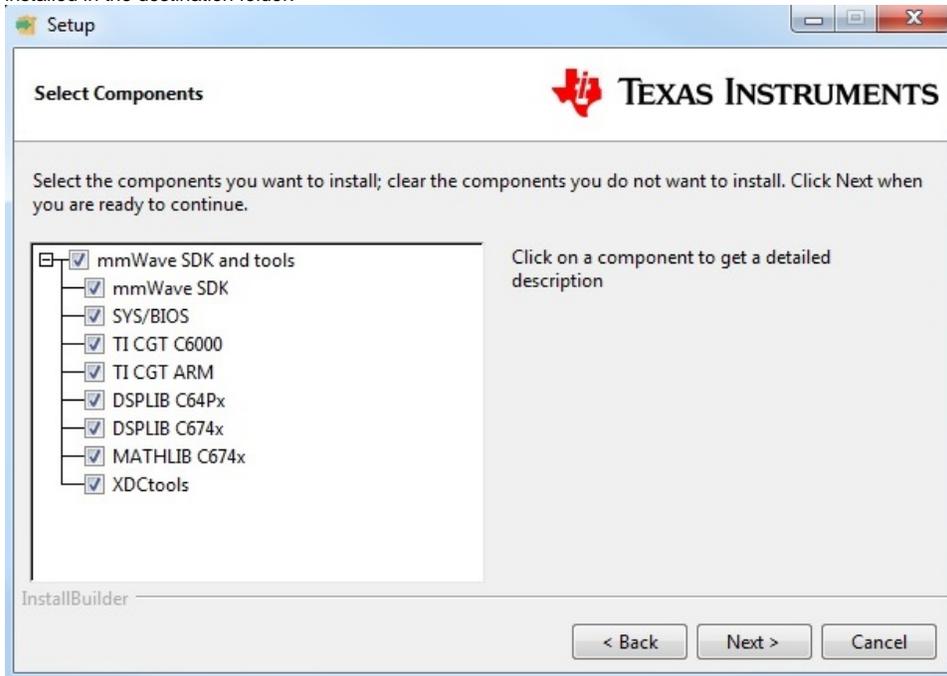
- In Windows environment, double clicking the Windows installer from Windows explorer should start the installation process
- If in Linux environment,
  - On 64-bit machines: Since `mmwave_sdk_<version>-Linux-x86-Install.bin` is a 32-bit executable, install modules that allows Linux 32bit binaries to execute: `sudo dpkg --add-architecture i386`
  - 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
  - On 64-bit machines if the GUI does not show up you may need to install additional packages: `sudo apt-get install libc6:i386 libgtk2.0-0:i386 libxst6:i386`

Installation steps:

- Setup
- Choose Destination Location: Select the folder to install (default is `c:\ti` on windows and `~/ti` on linux). **The installation folder selected should not have spaces in its full path.**



- 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

## Installation in unattended command line mode

The installers can be run in command line mode without user intervention

- In Windows environment
  - Run the installer using "mmwave\_sdk\_<version>-Windows-x86-Install.exe --prefix <installation folder> --mode unattended" command. This will install all the components in the installer.
    - Please note that even though the command may finish immediately it takes sometime for all the folders to show up in the destination folder (double check if you have the folder structure in "Post Installation" section before proceeding)
    - For command line help including information about selective installation of components run the following command "mmwave\_sdk\_<version>-Windows-x86-Install.exe --help"
- In Linux environment:
  - On 64-bit machines: Since mmwave\_sdk\_<version>-Linux-x86-Install.bin is a 32-bit executable, install modules that allows Linux 32bit binaries to execute: "sudo dpkg --add-architecture i386"
  - 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 --prefix <installation folder> --mode unattended" command. This will install all the components in the installer.
    - For command line help including information about selective installation of components run the following command ". /mmwave\_sdk\_<version>-Linux-x86-Install.bin --help"

## Post Installation

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.

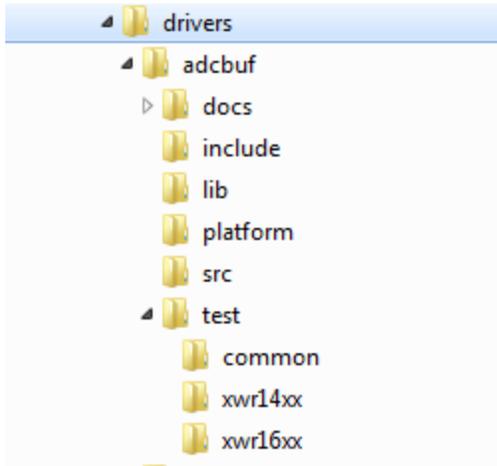
- .metadata
- docs
  - relnotes\_archive
- test
- firmware
- radarss
- packages
- scripts
- ti
  - alg
  - board
  - common
  - control
  - datapath
  - demo
  - drivers
  - platform
  - utils
- tools
- ftdi

## Package Contents

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

### 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 supported platforms
- 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 <sup>1</sup>	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

<b>SOC</b>	X	X	
<b>SPI</b>	X	X	X
<b>UART</b>	X	X	X
<b>WATCHDOG</b>	X	X	X

<sup>1</sup> CRYPTO is only supported on high secure (HS) devices

## 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)
datapath manager (dpm)	X	X	X
mmwavelink framework	X	X	X
mmwave high level api	X	X	X

## Datapath

Datapath modules can be found under mmwave\_sdk\_<ver>/packages/ti/datapath folder. Content of each of the control module is shown below

Component	Source & Prebuilt Library	API Document (doxygen)	Unittest (source & prebuilt binary)
RangeProc DPU	X	X	X
Doppler DPU	X	X	X
Static Clutter DPU	X	X	X
CFAR CA DPU	X	X	X
AoA DPU	X	X	X
AoA 2D DPU	X	X	X
Datapath EDMA	X	X	
Object Detection DPC <sup>1</sup>	X	X	X

<sup>1</sup> No pre-built library for Object Detection DPC

## 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



## Usecases

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

Component	Source	API Document (doxygen)	Unittest (source & prebuilt binary)
<code>csi_stream (IWR14xx only)</code>	X	X	X
<code>mem_capture</code>	X	X	X

## 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
<code>mmw<sup>1</sup></code>	X	X	X

<sup>1</sup> Demo is supported on all devices except for `xwr14xx` in this release

## 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.
  - `mathutil` is used to perform some common operations such as `log2`, rounding, saturation based on the core they need to run on (R4F, C674x)
  - `secondary boot loader (sbl)`
  - `testlogger` which is the helper utility for driver unit tests

## Scripts

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

## 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`.

## 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 FTDI port on MMWAVE-DEVPACK or MMWAVEICBOOST

## 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. Release notes contain migration information.



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

## Related documentation/links

Other than the documents included in the mmwave\_sdk package the following documents/links are important references.

- SoC links:
  - [Automotive mmWave Sensors](#)
  - [Industrial mmWave Sensors](#)
- Evaluation Modules (EVM) links:
  - [Automotive Evaluation modules](#) (Booster Pack, DEVPACK)
  - [Industrial Evaluation modules](#) (Booster Pack, ISK)

