

MMWAVE SDK Release Notes



Product Release 3.5

Release Date: Sept 30, 2020

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 [Limitations](#)
 - 3.5.1 [mmWave Suite/Demos Limitations](#)
 - 3.5.2 [RadarSS Limitations](#)
 - 3.5.2.1 [RadarSS firmware \(patch\) for xwr14xx, xwr16xx, xwr18xx](#)
 - 3.5.2.2 [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](#)
-



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



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.

2. Release overview

2.1. 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

2.2. 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 ²
AWR6443 ES2.0	N/A ²
AWR6843 HS ES2.0 ¹	N/A ²
AWR6843AOP ES2.0	N/A ²
AWR6843AOP HS ES2.0 ¹	N/A ²
AWR1843 ES1.0	AWR1843BOOST - AWR1843 Evaluation Module Rev C
AWR1843_HS ES 1.0 ¹	N/A ²
AWR1843AOP ES2.0	N/A ²
AWR1843AOP_HS ES2.0 ¹	N/A ²
AWR1642 ES2.0	AWR1642BOOST - AWR1642 Evaluation Module Rev B
AWR1642_HS ES 2.0 ¹	N/A ²



AWR1443 ES3.0 ³	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 ²
IWR6443 ES2.0	N/A ²
IWR6843_HS ES2.0 ¹	N/A ²
IWR6843AOP ES2.0	IWR6843AOPEVM (Rev F) + MMWAVEICBOOST (Rev B): IWR6843AOP Evaluation Module
IWR6843AOP_SIL2 ES2.0	N/A ²
IWR6843AOP_HS ES2.0¹	N/A ²
IWR1843 ES1.0	IWR1843BOOST - IWR1843 Evaluation Module Rev C
IWR1642 ES2.0	IWR1642BOOST - IWR1642 Evaluation Module Rev B
IWR1642_HS ES 2.0 ¹	N/A ²
IWR1443 ES3.0 ³	IWR1443BOOST - IWR1443 Evaluation Module Rev B
N/A	DCA1000EVM (Rev A) - mmWave Real-time data-capture adapter

¹ High Secure (HS) devices need additional MMWAVE-SECDEV package

² Device was internally validated using a device pin compatible EVM

³ 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.

2. 3. Component versions

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

Component	Version	Type	Comment	
mmwave sdk	3.5	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	1.0	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	9.3 or later	download link
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 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
DCA1000EVM CLI	1.0.0	Part of MMWAVE STUDIO package (use MMWAVE-STUDIO 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): 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

- 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
 - CANFD:
 - Added support for the second CANFD instance for AWR6xxx devices.
 - Removed support for soft reset for 60Ghz devices as it is no longer supported by the IP.
 - Added new testcases to test the two instances simultaneous transmit/receive capability.
 - SOC:
 - AWR18xx/AWR16xx: Added workaround in SOC-init() for Pulse width Synchronizer silicon issue related to DSP STC/power down
 - Added support for MCANB for AWR6xxx devices.
 - HWA: Added support for compression/decompression for 60GHz devices.
 - GPIO: updated driver to allow interrupt capabilities on only first 16 GPIO pins as per H/W capabilities
 - Updated driver and unit tests for QSPI/QSPI flash and SBL to showcase 80Mhz clocking speed for QSPI in 60GHz devices
 - mmWave Control layer
 - Updated the error code returned by mmWave_Open API when RadarSS reports boot-time calibration failure
 - If the BPM chirp is a valid chirp used for the current frame, then make sure custom calibration config MMWave_open to override the default calibration mask and adhere to the recommendation by mmwavelink RF driver.
 - mmWaveLink RF driver
 - Added support for monitoring functions for ASILB/SIL2 variants of 60GHz devices
 - Group Tracker
 - Implemented improvements at tracker layer to use the both static and dynamic point cloud points.
 - Added version history file to provide more insight into the changes performed in this component
 - mmWave data processing layers
 - AOA 2D DPU and HWA based ObjectDetection DPC: Added support for AWR1843AOP
 - Added more testcases in DPUs to cover cases where a higher order FFT is done for Range and Doppler
 - Added more testcases in DPUs for number of ADC Samples less than 64
 - Added check in CFAR DPU if Range or doppler bins are not greater than 2*(noiseLen+guardLen) for their respective CFAR config.
 - Minimum number of ADC samples supported have been changed from 64 to 2. CFAR DPU might impose some restrictions on RangeBins < 8.
- mmWave Demos enhancement
 - Visualizer:
 - Added option to playback recorded stream in the PLOTS tab.
 - Added option to CONFIGURE tab to allow user to save or restore the boot time RF calibration data
 - mmW demos:
 - Added support for saving and restoring boot time RF calibration data. New CLI command - calibdata - created to support this configuration.
 - Added section about error codes and their decoding logic in mmW demo doxygen
 - Updated RF gain target to 36dB for all profiles of 60GHz devices.
 - xwr18xx mmW demo: Added support for AWR1843 AOP and it antenna pattern
- Components/Tools
 - RadarSS: Updated the RadarSS component for all devices (see exact version above). Users should refer to the RadarSS release notes included under mmwave_sdk_<ver>/firmware/radarss folder for features and enhancements done in this component.
 - New python based scripts provided to parse output generated by mmW demo over UART and LVDS.
 - Linker command files: Grouped .bss, .neardata, and .rodata together to prevent linker warning/error related to near data relocation.

3.2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.5.0 release from 3.4.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component /s	Subcomponent	Behavior of impact
Added presence detection to group tracker algorithm	Alg	Gtrack	See mmwave_sdk_<ver>/packages/ti/alg/gtrack /version_history.txt for details on changes and their impact to application
mmWave Open API will return MMWAVE_ECALFAIL instead of MMWAVE_ERFINIT on boot time calibration failure.	Control	mmWave API	If application takes different actions based on error code returned by mmWave Open API, then they should update the action for MMWAVE_ECALFAIL accordingly.

mmwavelink now supports monitoring APIs for safety devices in 60GHz family	Control	mmwavelink	<p>No changes to existing API but user can issue the following APIs to enable the corresponding features in safety devices for 60GHz family:</p> <ul style="list-style-type: none"> • rIRfDigMonEnableConfig • rIRfDigMonPeriodicConfig • rIRfAnaMonConfig • rIRfTempMonConfig • rIRfRxGainPhMonConfig • rIRfRxIfStageMonConfig • rIRfTxPowrMonConfig • rIRfTxBallbreakMonConfig • rIRfSynthFreqMonConfig • rIRfExtAnaSignalsMonConfig • rIRfTxIntAnaSignalsMonConfig • rIRfRxIntAnaSignalsMonConfig • rIRfPmClkLoIntAnaSignalsMonConfig • rIRfGpadcIntAnaSignalsMonConfig • rIRfPllContrlVoltMonConfig • rIRfDualClkCompMonConfig • rIRfRxIfSatMonConfig • rIRfRxSigImgmMonConfig • rIRfRxMixerInPwrConfig • rIRfAnaFaultInjConfig
TX Ball Break monitoring configuration API has a new frequency field.	Control	mmwavelink	See doxygen of rIRfTxBallbreakMonConfig for more details
New option to disable TX PS DAC monitoring as part of the TX internal analog signals monitoring configuration API.	Control	mmwavelink	See doxygen of rIRfTxIntAnaSignalsMonConfig for more details
New restrictions have been added on the allowed range for start frequency and frequency slope values.	Control	mmwavelink	<p>See doxygen of following APIs for more details:</p> <ul style="list-style-type: none"> • rISetProfileConfig • rISetChirpConfig • rISetLoopBckBurstCfg • rISetDynChirpCfg • rISetContModeConfig • rIRfTxBallbreakMonConfig
New mandatory CLI command 'calibdata' added to the mmW demo CLI to configure the save/restore feature of boot time RF calibrations	Demos	mmW	<p>See SDK user guide on details about this new command.</p> <p>User can use <code>ti/demo/<platform>/mmw/profiles/mmwDemo_<platform>_update_config.pl</code> to migrate CFG files from previous release to the current one.</p>
GPIO driver is updated to allow interrupt capabilities on only first 16 GPIO pins as per H/W capabilities	Drivers	GPIO	GPIO hardware only supports interrupt capability on GPIO pins 0-15 - driver will now return an error if any pins outside that range are requested for interrupt capability.
HWA driver API is updated to support compression /decompression feature in 60GHz devices	Drivers	HWA	Application need to config these new fields only if compression/decompression is desired. This applies to 60Ghz devices only.
SOC_getDeviceRFFreqScaleFactor() will return compile time RF frequency scale factor if the device part number is not a known part number	Drivers	SOC	No change in application is required for this.

3. 3. 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-2293	ESM_deregisterNotifier function doesn't remove notifier
Bug	MMWSDK-2289	GPIO: Only first 16 GPIO pins can be configure for interrupt
Bug	MMWSDK-2250	Error in formula in AoA Doxygen for doppler FFT output size condition
Bug	MMWSDK-2239	The calibrationStatus is recorded but masked (The mask is missing the RXIQMM cal bit)
Bug	MMWSDK-2155	Missing Header/Prototype for DPIFUtils_convertPointCloudCartesionToSpherical
Bug	MMWSDK-2154	In math utility, on a DSP, log2 functions are incorrect for some input values.
Bug	MMWSDK-2116	EDMA driver does not clear EERH register at init time



3. 4. Known Issues

3. 4. 1. 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-1497	Intermittent failure in "monitoring results" for mmwavelink unit test for awr16xx	This issue is seen in noisy lab environment only. One out of many reports for noise figure has failure status. Observed noise figure from that report are logged at the end of the test run and can be used for debugging further, in case this is seen in other scenarios.
Bug	MMWSDK-1363	Range processing hwa DPU crashes when number of RX antenna is 4, and range fft size is 1024	For 1 TX 4 RX and numRangeBins = 1024, the BdstIndex for EDMA copy will go beyond its limit of 32768. The calculation is follows: BytesPerChirp = numRangeBins * numRxAnt * sizeof(cmplx16ImRe_t) = 16KB. For 1 TX antenna, due to ping/pong scheme, the jump will be 2 * BytesPerChirp = 32KB. The same case is solved by manually setting destination address in rangeProc DSP based implementation. For rangeProcHWA, the manually setting of destination address is not doable.
Bug	MMWSDK-1157	Rare failure seen in UART loopback driver unit test - HW limitation	
Bug	MMWSDK-1078	Limitation in processing chain + LVDS instrumentation use case	See limitations section below
Task	MMWSDK-533	GUI of mmw demo running slow from Firefox browser	Workaround: 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	

3. 4. 2. RadarSS Known Issues

3. 4. 2. 1. 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.

3. 4. 2. 2. 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.

3. 5. Limitations

3. 5. 1. 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"> DMA and FIFO mode are not supported
2	CANFD driver: <ul style="list-style-type: none"> DMA and Timestamping are not supported AWR6843: 2nd CANFD instance is 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 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">▪ 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	Processing chain + 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, 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.

3. 5. 2. RadarSS Limitations

3. 5. 2. 1. 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.

3. 5. 2. 2. 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.

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 16.04 and Ubuntu 18.04 64 bit machines.

5. 1. Installation in GUI mode

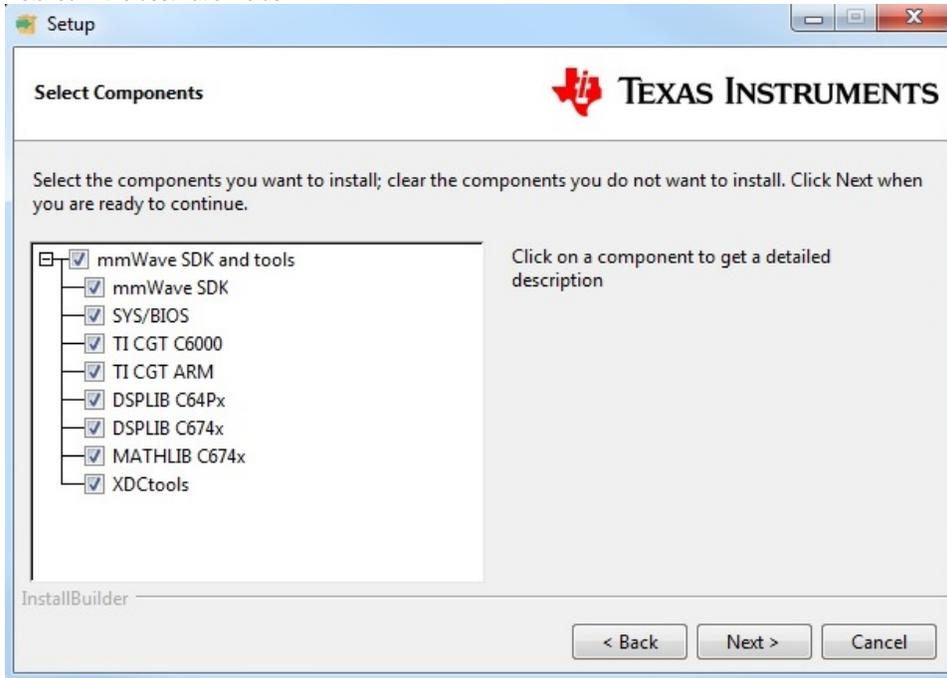
Depending on your development environment run the appropriate installer



- 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

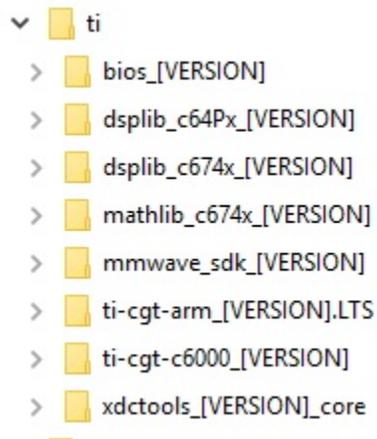
5. 2. Installation in unattended command line mode

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

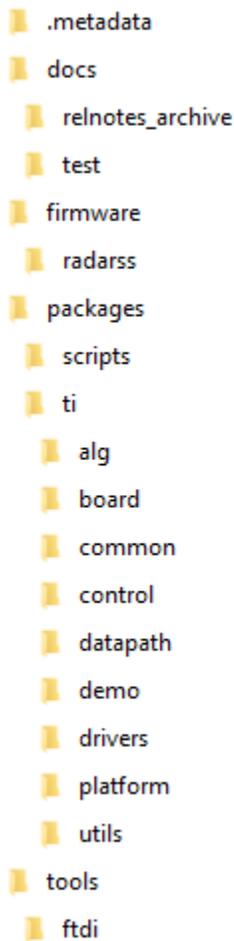
- In Windows environment
 - Run the installer using "mmwave_sdk_<version>-Windows-x86-Install.exe --prefix <installation folder> --mode unattended" command. This will install all the components in the installer.
 - Please note that even though the command may finish immediately it takes sometime for 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"

5. 3. 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)



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

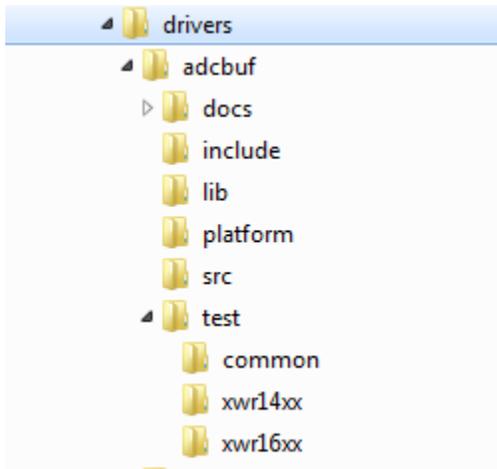


6. Package Contents

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 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 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 high secure (HS) devices

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

6. 3. Datapath

Datpath 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 ¹	X	X	X

¹ No pre-built library for Object Detection DPC

6. 4. 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

6. 5. Usecases

Usecases 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
mem_capture	X	X	X

6. 6. 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 ¹	X	X	X

¹ Demo is supported on all devices except for xwr14xx in this release

6. 7. 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

6. 8. Scripts

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

6. 9. 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. 10. 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

6. 11. 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

7. 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)

