Copyright (C) 2019, Texas Instruments Incorporated
http://www.ti.com

MMWAVE SDK Release Notes

Product Release 3.4

Release Date: Mar 30, 2020

Release Notes Version: 1.0

CONTENTS

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 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	N/A**
AWR1843 ES1.0	AWR1843BOOST - AWR1843 Evaluation Module Rev C
AWR1843_HS ES 1.0 *	N/A**
AWR1642 ES2.0	AWR1642BOOST - AWR1642 Evaluation Module Rev B
AWR1642_HS ES 2.0 *	N/A**
AWR1443 ES3.0	AWR1443BOOST - AWR1443 Evaluation Module Rev B
IWR6843 ES2.0	IWR6843ISK (Rev B)+MMWAVEICBOOST (Rev B): IWR6843 Evaluation Module
IWR6443 ES2.0	N/A**
IWR6843_HS ES2.0	N/A**
IWR6843AOP ES2.0	IWR6843AOPEVM (Rev E) + MMWAVEICBOOST (Rev B): IWR6843AOP Evaluation Module
IWR1843 ES1.0	IWR1843BOOST - IWR1843 Evaluation Module Rev C
IWR1642 ES2.0	IWR1642BOOST - IWR1642 Evaluation Module Rev B
IWR1642_HS ES 2.0*	N/A**
IWR1443 ES3.0	IWR1443BOOST - IWR1443 Evaluation Module Rev B
N/A	DCA1000EVM (Rev A) - mmWave Real-time data-capture adapter

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

** Device was internally validated using a device pin compatible EVM

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		3.4	Source and Binary	Overall package release version
RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx		1.2.5.2	Binary	RadarSS firmware is in ROM. Only the patch is included in the mmwave sdk release
RadarSS firmware for xwr68xx		6.2.1.5	Binary	
mmWaveLink Framework		1.2.5	Source and Binary	
FTDI		2.12	Binary	
Image Creator	gen_bincrc32	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 IWR6443 ES2.0, AWR6843 ES2.0, IWR6843AOP ES2.0, IWR6843 HS ES2.0 devices.
  - AWR68xx is added as a distinct target in the build environment.
  - Added distinct xwr64xx ccsdebug binary for IWR6443 device that contains only MSS executable.
  - xwr64xx: SBL and SOC driver avoid DSS unhalt procedures when xwr64xx device is detected.
- mmWave Suite enhancement
  - Drivers
    - HWA driver:
      - Changes related to enabling new AoA 2D algorithm
      - Allow concurrent access to common set registers while HWA is active for xwr68xx ES2
    - EDMA driver: Updated the checkers for the various data integrity and device hung related EDMA IP erratas. These checkers are disabled by default for performance reasons but can be enabled easily via global variables exposed by the EDMA driver.
    - SOC driver:
      - Added new API for MSS to detect if the device has a DSP core
      - Added new API for MSS to workaround MSS DMA MPU silicon errata. SOC_init() leverages this new API to reset DMA state on boot-up.
  - mmWave Control layer
    - Enabled all boot time calibration and runtime calibration for xwr6xxx devices
    - Added capability to restore phase shifter calibration data for xwr18xx and xwr6xxx devices
  - mmWave data processing layers
    - New datapath module for AoA that uses 2D FFT method while leveraging HWA
    - Range HWA based DPU and Objectdetection DPC now allows users to set the scaling values for butterfly stages and converting from internal 24-bit to 16-bit output.
    - Removed additional scaling of ADC input in objectdetection HWA based DPC for xwr68xx device after integrating DFP 6.2.x that allows runtime calibrations and calibrated Rx signal level.
  - New "board" directory is added to hold EVM related configuration such as antenna geometry.
  - New helper utilities added to mathutils for AoA 2D algorithm
- mmWave Demos enhancement
  - Visualizer: Added support for displaying device temperature report
  - Visualizer: Added support for configuring baud rate of DATA_port higher than default value of 921600
  - mmW demos:
    - Support 2 new optional commands configDataPort and queryDemoStatus to allow configuring baud rate of DATA_port higher than default value of 921600
    - Added support for device temperature report every frame/subframe as a new TLV (MMWDEMO_OUTPUT_MSG_TEMPERATURE_STATS)
  - 68xx mmW demo:
    - Added support for BPM-MIMO
    - Added support for CQ/CP streaming over LVDS
  - 68xx HWA only (xwr64xx) mmW demo:
    - This demo has been ported for IWR6843AOP and is available as a separate binary in the same demo directory (in addition to the binary for IWR6xxx ISK).
    - For the AOP variant of the demo, it leverages the new AoA 2D algorithm and antenna geometry definition.
    - Added support for CQ/CP streaming over LVDS
- Components/Tools
  - RadarSS: Updated the RadarSS component for IWR68xx (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.

3. 2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.4.0 release from 3.3.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component/s	Subcomponent	Behavior of impact
Name of RadarSS binary for xwr6xxx devices is now switched to xwr6xxx_radarss_rprc.bin instead of iwr6xxx_radarss_rprc.bin.	firmware		setenv.bat and setenv.sh already reflect this change in the value of XWR68XX_RADARSS_IMAGE_BIN
Created a new structure MMWave_CalibrationData that encapsulates rlCalibrationData_t and rlPhShiftCalibrationData_t to allow restore of multiple device calibration related parameters	Control	mmWave	All applications using mmWave for restoring device calibrations will need to port their application code according to the new API parameters. If application want not using the calibration restore feature can continue to pass NULL for that parameter.
mmWave now enables all valid calibrations during init time and runtime for xwr6xxx devices	Control	mmWave	Application should pass valid values for freqLimitLow and freqLimitHigh in mmWave_Open API and can now enable periodic calibrations in mmWave_Start API

New TLV with device temperature report is now shipped out of mmW demos when the stats parameter is enabled in guiMonitor command	Demos	mmW	Users can choose to ignore or consume this additional output over UART.
Range Azimuth HeatMap TLV stream from mmW demo is replaced by Range Azimuth/Elevation TLV stream for IWR6843AOP mmW demo	Demo Datapath	xwr64xx Demo (for IWR6843AOP only)	IWR6843AOP antenna pattern has multiple rows/columns for azimuth and elevation symbols for a given range/doppler index and based on debugging needs, any of the rows/columns could be used for azimuth and elevation heatmap calculations. Hence AoA 2D DPU provides all the virtual antenna symbols for the zero Doppler radar cube matrix and xwr684xx Demo (for IWR6843AOP only) ships all these symbols across all range bins as rangeAzimuthElevation heatmap. The same bit <rangeAzimuthHeatMap> in "guiMonitor" is repurposed to enable /disable rangeAzimuthElevation heatmap. Note that this change only applies to xwr64xx demo when compiled for IWR6843 AOP. When it is compiled for standard IWR6843 ISK antenna pattern, the original AoA DPU is used for processing and the heatmap is just range-azimuth heatmap (with no elevation related symbols) as in previous releases. See xwr64xx demo doxygen for more details.
Minor updates to existing AoA DPU interface to unify with the new AoA 2D DPU	Datapath	DPU	Some new fields are added to the original AoA HWA DPU to align the interface with the new AoA 2D DPU but these fields are unused and reserved for now. There should be no impact to existing applications.
Objectdetection DPC now accepts antenna geometry to enable wider configurations of Tx/Rx antennas besides the standard antenna pattern shown on BOOST and ISK mmWave EVMs	Datapath	DPC	This field is mandatory only for HWA based Objectdetection DPC when compiled to use the new AoA 2D algorithm. For DSP based DPC and for HWA based DPC that uses standard AoA DPU, this field is unused.
Range HWA based DPU and Objectdetection HWA based DPCs now allow users to set the scaling values for butterfly stages and converting from internal 24-bit to 16-bit output	Datapath	RangeHWADPU ObjDetHWA ObjDetRangeHWA	Application now need to pass range FFT scaling parameters as part of DPC/DPU configuration. See Object detection DPC doxygen for more details and mmW demos for reference implementation. With this API change and update in RadarSS binary, the artificial scaling down of Range FFT window is now removed from the HWA based DPCs and application are now in full control to config the HWA scaling for Range FFT as per their chirp design and environment where the sensor is deployed
Objectdetection Range HWA DPC now allows user to specify the radar cube format to allow flexibility in integrating various DSP based algorithms/processing chains	Datapath	ObjDetRangeHWA	Application now need to specify the radar cube format when configuring this DPC. Note: mmW demos only supports DPIF_RADARCUBE_FORMAT_1.
DPEDMA structures are updated to use int16_t for specifying B and C index to match the API definition of EDMA driver	Datapath	DPEdma	No change should be needed at the DPU/DPC or application level since users of this API would have been following the guidelines of EDMA IP when configuring their flow.
SOC_triggerWarmReset now accepts warm reset request type as an input parameter	Drivers	SOC	Application should pass appropriate values to this new parameter as per API doxygen
Several calibrations that were unsupported in DFP 6.2.0 have now been enabled in the firmware.	Control	mmwavelink	The calibration configuration in the below mentioned APIs needs to be modified to enable the new calibrations. ▪ rlRfRunTimeCalibConfig ▪ rlRfInitCalibConfig Application invoking the mmwavelink API and not using mmwave layer need to take care of this migration. When using MMWave_open() to initialize the RF and MMWave_start() to start chirping, application doesn't need to worry about this parameter as mmwave module takes care of this internally.

There are no changes to the API parameters for mmwavelink module in this release. However, many APIs that were not supported in the previous release are now enabled in the firmware.	Control	mmwavelink	User can issue the following APIs to enable the corresponding features. • rlRfSetCalMonFreqLimitConfig • rlRfTxFreqPwrLimitConfig • rlRfCalibDataRestore • rlRfPhShiftCalibDataRestore • rlRfCalibDataStore • rlRfPhShiftCalibDataStore • rlSetSubFrameStart • rlSetBpmCommonConfig • rlSetBpmChirpConfig • pfVcoSelect feature in rlSetProfileConfig • loopBackCfg feature in rlSetAdvFrameConfig • rlSetDynChirpCfg • rlSetDynChirpEn • rlSetDynPerChirpPhShifterCfg • rlRxGainTempLutSet • rlTxGainTempLutSet • rlRxGainTempLutGet • rlTxGainTempLutGet • rlRfSetPhaseShiftConfig • rlRfSetProgFiltCoeffRam • rlRfSetProgFiltConfig
numOfCascadeDev parameter in the calibration and monitoring time unit configuration API rlRfSetCalMonTimeUnitConfig needs to be set to the default value of '1' for non-cascaded systems (single chip).	Control	mmwavelink	Application invoking the mmwavelink API and not using mmwave layer need to take care of this migration. When using MMWave_open() to initialize the RF, application doesnt need to worry about this parameter as mmwave module takes care of this internally.
Updates to the RF gain target field in "Profile configuration set" API and the "Continuous streaming config set" API for xwr6xxx devices	Control	mmwavelink	The RF gain target field in "Profile configuration set" API and the "Continuous streaming config set" API has been updated for IWR6843 ES2 devices From • 30dB (0b00) • 34dB (0b01) • Reserved (0b10) • 26dB (0b11) To • 30dB (0b00) • 33dB (0b01) • 36dB (0b10) • Reserved (0b11) mmwavelink APIs impacted: rlSetProfileConfig rlSetContModeConfig rlSetLoopBckBurstCfg Application should take care of this updated mapping when invoking MMWave_addProfile() API as well Application should take care of this updated mapping when specifying the <rxGain> parameter of profileCfg CLI command for mmW demos.
Update to rlCalibrationData structure format for xwr6xxx devices	Control	mmwavelink	Due to changes in some of the calibration algorithms in xWR6483, the calibration results data structure is different from the 1st gen devices like 12xx/14xx/16xx/18xx. See mmwavelink doxygen of rlRfCalibDataStore and rlRfCalibDataRestore APIs and their parameters for more details on the updated structure.
Update to calling sequence for rlRfSetMiscConfig	Control	mmwavelink	Issue rlRfSetMiscConfig with the per-chirp phase shifter configuration field enabled first in the sequence if rlRfSetPhaseShiftConfig and rlSetDynPerChirpPhShifterCfg are issued down in the sequence.

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-2086	Pinmux: Pinmux_Set_Pull() doesn't set the pull values in the pad register as expected
Bug	MMWSDK-2085	mmwavelib unit tests: relative directory of testdata should be updated after the move of binary file location

Bug	MMWSDK-2083	xwr6843 mmw demo: DPC_start fails after several stop/restart with no reconfig sequence in between
Bug	MMWSDK-2073	window coefficients generation in Range HWA DPU test bench
Bug	MMWSDK-2072	Source and destination B and C indices should be int16_t in struct DPEDMA_syncACfg and DPEDMA_syncABCfg
Bug	MMWSDK-2071	mmw demo doesn't report error if subframe contains chirps from 2 profiles
Bug	MMWSDK-2019	Wrong directory name in test makefile hwa_dssTest.mak in CFAR HWA DPU
Bug	MMWSDK-2018	Address static analysis issues in test code
Bug	MMWSDK-2016	mmwave doesnt return proper error code when calibmontimeunit is set to 0 in mmwave_open API
Bug	MMWSDK-2013	Correct noise variance in DPC unit test signal generation
Bug	MMWSDK-2012	AWR1642: User cannot load an image when built with L3 memory diverted to extend TCMB
Bug	MMWSDK-1989	SOC_triggerWarmReset Function name and comment is misleading
Bug	MMWSDK-1872	EDMA transfer can stall while running DPC test code on AWR18xx
Bug	MMWSDK-1602	Linker warnings seen in gtrack unit test
Bug	MMWSDK-1032	Uninitialized MCPI test result variable falsely masks failed testcase

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: ByptesPerChirp = numRangeBins * numRxAnt * sizeof(cmplx16ImRe_t) = 16KB. For 1 TX antenna, due to ping/pong scheme, the jump will be 2 * ByptesPerChirp = 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: • DMA and FIFO mode are not supported
2	CANFD driver: • DMA and Timestamping are not supported • AWR6843: 2nd CANFD instance is not supported

3	CBUFF/CSI2/LVDS:
	• Driver does not support the following functionality:
	  ◦ 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: ▪ dual-Read/Quad read in configuration mode is not supported ▪ setting write protections bits is not supported
10	SPI (MIBSPI) Limitations: • 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: ▪ 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

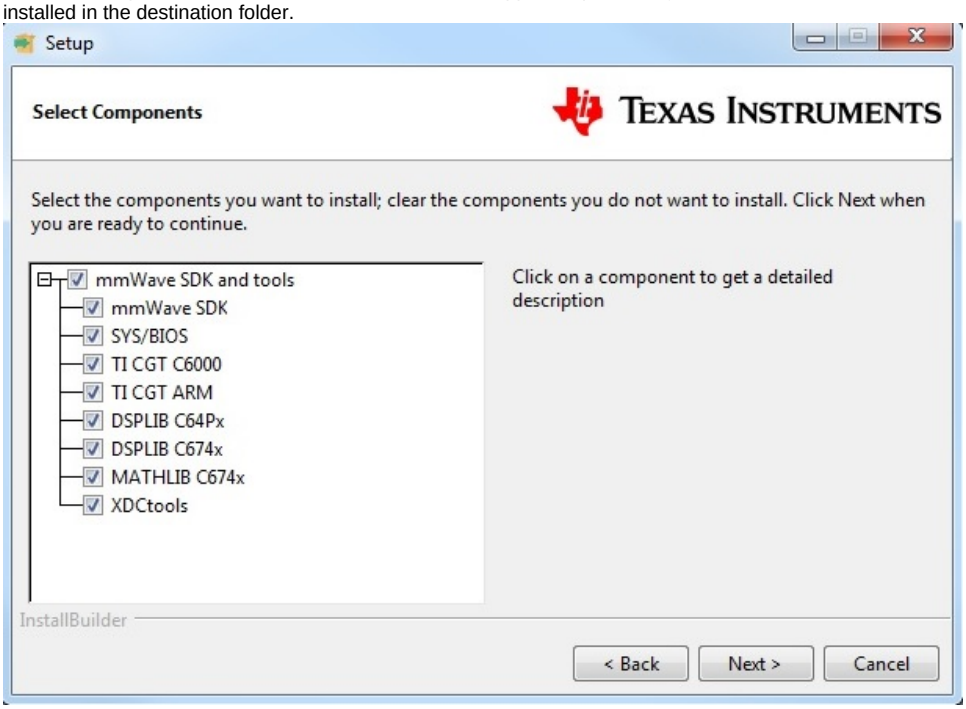
Depending on your development environment run the appropriate installer

- 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 libxtst6: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

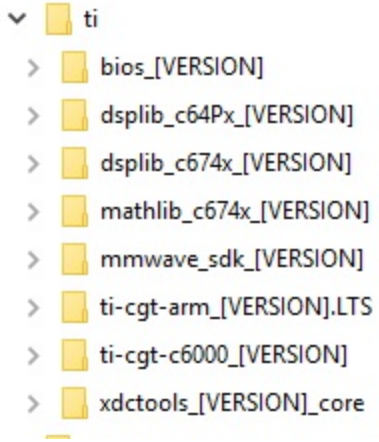
5. 2. Installation in unattended command line mode

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

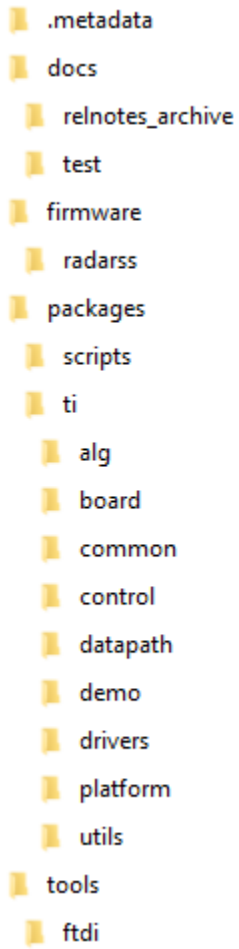
- In Windows environment
  - Run the installer using "mmwave_sdk_<version>-Windows-x86-Install.exe --prefix <installation folder> --mode unattended" command. This will install all the components in the installer.
    - Please note that even though the command may finish immediately it takes sometime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

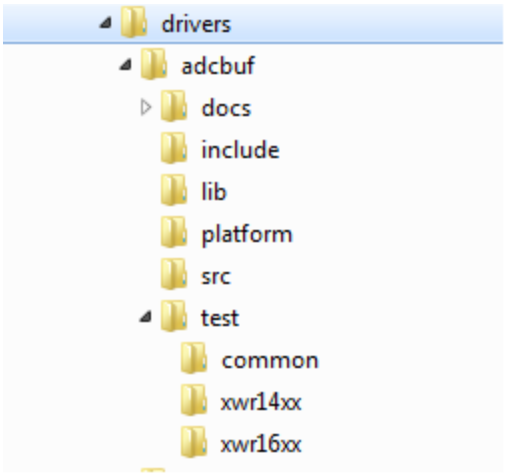


6. Package Contents

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

Datapth 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
	X	X	X

mmwavelib			

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)