MMWAVE SDK Release Notes

Product Release 3.0.0

Release Date: Oct 29, 2018

Release Notes Version: 1.2

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
AWR1843 ES1.0	AWR1843BOOST - AWR1843 Evaluation Module
AWR1642 ES2.0	AWR1642BOOST - AWR1642 Evaluation Module RevB
AWR1642_HS ES 2.0 *	
AWR1443 ES3.0	AWR1443BOOST - AWR1443 Evaluation Module RevB/RevA
IWR6843 ES1.0	IWR6843ISK+MMWAVEICBOOST - IWR6843 Evaluation Module
IWR1843 ES1.0	IWR1843BOOST - IWR1843 Evaluation Module
IWR1642 ES2.0	IWR1642BOOST - IWR1642 Evaluation Module RevB
IWR1642_HS ES 2.0*	
IWR1443 ES3.0	IWR1443BOOST - IWR1443 Evaluation Module RevB/RevA

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

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		3.0.0	Source and Binary	Overall package release version
RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx		1.2.0	Binary	RadarSS firware is in ROM. Only the patch is included in the mmwave sdk release
RadarSS firmware for xwr68xx		6.0.5	Binary	
mmWaveLink Framework		1.2.0	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	7.4 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	8.1.3	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	2.0.0	Needed for xWR16xx high secure (HS) devices only Can be requested from link

Pinmux tool (optional)	Latest	Used to generate pinmux configuration for custom board https://dev.ti.com/pinmux (Cloud version)
Doxygen (optional)	1.8.11	Only needed if regenerating doxygen docs
Graphviz (optional)	2.36.0 (20140111.2315)	Only needed if regenerating doxygen docs

The following tools are needed at runtime

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

2. 5. Licensing

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

3. Release content

3. 1. New Features

- mmWave Suite enhancement
    - Support for IWR68xx and xWR18xx mmWave devices.
    - Drivers
        - mmWaveLink updated to support the RadarSS firmware version as noted above
        - EDMA CC1 can now be instantiated on MSS (except on xwr14xx where only one CC is available) in polling mode
        - CANFD: added feature for range filtering on RX message IDs
        - HWA: new helper APIs to assist in EDMA-HWA integration and provide memory bank information based on device
        - OSAL: new module to abstract the cycle profiler capability that might be needed in drivers/modules that are OS independent
        - SOC: allow user to bypass MPU and DSS unhalt configuration during SOC initialization
    - mmWaveLib enhancements
        - Multiple peak search for Azimuth estimation on 32-bit float vector
        - Second peak search for Azimuth estimation on 32-bit float vector
        - Power computation combined with max power search for 32-bit complex vector
        - Peak pruning for CFAR post-processing
        - Twiddle table generation for 32x32 and 16x16 FFTs
        - FFT window coefficients generation utility
        - Matrix transpose for 32-bit matrix
        - DC signature removal on 32-bit float complex vector
        - Additional vector multiplication functions
        - 16-bit to 32-bit Vector accumulation
        - 16-bit Vector subtraction
        - Right shift 32-bit vector to 16-bit vector
        - DFT sine/cosine table generation for DFT single bin calculation
        - Variation of the windowing functions with I/Q swap since most of the fixed point FFT functions in DSPLib only support one format of complex types.
    - Group tracker
        - Support for 3D point cloud data that has range, azimuth and elevation information along with doppler.
        - Updated test vectors for unit testing to be based on usecases
    - mmWave data processing layers
        - Defined new scalable architecture for the SDK to allow quick programming of the device and easy adaptation to different deployment scenarios.
        - Introduced new modules for the data processing layer:
            - Data processing units (DPUs)
            - Data processing chain (DPCs)
            - Datapath interface (DPIF)
            - Datapath manager (DPM)
            - EDMA helper utilities tuned for mmWave processing (DPEDMA)
- mmWave Demos enhancement
    - Added demo for xwr68xx using the newly defined data processing layers
    - Supports 2D (azimuth only) and 3D (azimuth + elevation) profiles using HWA as the processing engine
    - New capability for Field of view filtering in the CLI/mmWave Demo Visualizer
    - Added floating point based Point Cloud format for mmWave detections post CFAR/AoA processing
- Utilities

    - Added reference implementation of secondary bootloader
    - Added math utilities to perform some common operations such as log2, rounding, saturation based on the core they need to run on (R4F, C674x)

3. 2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.0.0 release from 2.1.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Componen t/s	Subcompone nt	Behavior of impact

Updated components	-	-	▪ RadarSS firmware now includes a separate firmware for xwr68xx device ▪ BIOS and XDC tools have been updated to the latest to bring in support for xwr68xx device ▪ mmWave Device support package should be updated to the version listed above to be able to create xwr68xx based ccxml file in CCS
Gtrack enhancements for handling 3D point cloud	alg	gtrack	Gtrack module is enhanced to handle 3D point cloud in addition to existing 2D point cloud. To balance the memory and mips requirement, this module provides switching between the 2D and 3D mode via compile time options. As a result, there are 2 pre-built libraries that are provided one for each option. Existing applications would need to update their makefile to pick up the right library. Additionally, they will need to define the flag GTRACK_2D or GTRACK_3D before including gtrack.h This support involves more changes at the API level and user should refer to the gtrack internal documentation at mmwave_sdk_<ver>\packages\ti\alg\gtrack\docs
Macro cleanup in sys_common.h	Common		Part of this cleanup is initiated by the new scalable architecture where the boundary between and application and processing chain have been defined more crisply. Additionally, non-platform specific defines have been moved to sys_defs.h and sys_types.h to allow modules to be platform independent when using such macros. sys_common.h still includes these new files and hence no change in the application is needed to continue to use these defines. To summarize the changes: • New file sys_types.h added to define the existing complex data types (no change at application level) • New file sys_defs.h added to define the existing CSL and MAX/MIN defines (no change at application level) • sys_defs.h also defines memory related macros. They existed as MMWDEMO_MEMORY_ALLOC prefixed defines and now they are renamed to be more generic: SYS_MEMORY_ALLOC_ • New module mathutils added under ti/utils to provide some common math utilities. round and saturate related defines from the sys_common.h are now provided by mmwave_sdk_<ver>/packages/ti/utils/mathutils/mathutils.h
Several changes to the demo directory based on the new scalable architecture	Demos	mmW Demo	• ti/demo/io_interface has been removed. The config structures are now provided by the DPUs in the new architecture. For applications that are still based off old architecture should copy the mmw_xxx.h from older SDK releases to their private application directories and customize it as per their needs • ti/demo/utils/mmw_monitor.c|h has been temporarily removed. This file implemented the CQ based monitoring and current 68xx device does not support this feature. This will be updated as per new architecture and added back once the mmW demos/device in SDK support CQ monitoring. For applications that are still based off old architecture should copy the mmw_monitor files from older SDK releases to their private application directories and customize it as per their needs • ti/demo/utils/rx_ch_bias_measure.c|h has been removed. This functionality has been migrated to the AoA DPU in the new architecture. For applications that are still based off old architecture should copy the rx_ch_bias_measure files from older SDK releases to their private application directories and customize it as per their needs
xwr14xx and xwr16xx demos have been removed	Demos	mmW Demo	This SDK release introduces the new scalable architecture and showcases that framework using xwr68xx mmW demo. To promote the architecture in a cleaner way, demos (and the test vector based dsp_edma and hwa_edma) written on top of old architecture are not distributed with this release. Having said that, complete mmWave Suite components are provided for xwr16xx and xwr14xx (similar to past releases) and users with their custom application for xwr16xx and xwr14xx will be able to still migrate to this SDK release by following the migration section

CLI commands for IWR68xx mmW demo	Demos	mmW Demo	CLI commands for IWR68xx demo follow the same structure as the preceding devices - xwr14xx, xwr16xx - except for the following high level changes. For more details, refer to the mmWave SDK user guide: • Antenna mask pattern in chirpCfg similar to xwr14xx • 60Ghz band for specifying the start frequency in profileCfg • low Power mode can be enabled/disabled based on sampling rate • ADCBuf supports only non-interleaved mode and chirpThreashold=1 • Updated cfarcfg to take peakGrouping as an additional parameter. Removed the duplicate peakGroupingCfg command • Added new FOV related commands
TLV output for IWR68xx mmW demo	Demos	mmW Demo	TLV output for IWR68xx demo follow the same structure as the preceding devices - xwr14xx, xwr16xx - except for the following high level changes. For more details, refer to the mmwave_sdk_<ver>/packages/ti/demo/xwr68xx /mmw doxygen documentation.: ▪ Updated TLV for detected objects: ▪ The structure for each detected object is now changed to conform to DPIF_PointCloudCartesian. This means float values in meters for x,y,z and float value in m/s for velocity. ▪ Since the values are in float, there is no dataObject descriptor preceding the array of detected objects now. ▪ New TLV for providing side info for each detected points ▪ SNR and NoiseValue are now provided for each detected point in a separate TLV and follows the structure DPIF_PointCloudSideInfo.
Enable EDMA#1 on MSS to help the usecase for CBUFF on MSS for instrumentation usecase	Drivers	EDMA	To enable CBUFF/LVDS instrumentation to be controlled via MSS while processing chain is controlled via DSS, EDMA CC#1 needs to be available as an instance on MSS when using EDMA driver. EDMA#1 control registers are memory mapped in MSS space but the completion and error interrupts are not hooked up to MSS core. Hence EDMA driver is enhanced to instantiate CC#1 in polling mode (i.e. no interrupt functionality) on MSS. This is an internal driver change but the instance info returned during open call is now enhanced to feed back the polling/interrupt capabilities of the instance being opened. Application can look at the returned EDMA instance info to decide on completion and error handling mechanism No change is required for existing applications.
Intermittent failure in QSPIFlash mmap read /write API test	Drivers	QSPI	With new Macronix flash, intermittment failures are seen in QSPIFlash mmap read/write unit test. This was due to QSPI flash memory being marked as not strongly ordered in the MPU settings. Such setting needs the memory to be flushed using dsb instruction (MEM_BARRIER function in sys_defs.h) before reading from a recently written flash location. Any application that needs to write to a flash location and then read it back immediately should call MEM_BARRIER before calling the read APIs.
Expose SOC MPU functionality to user so that they can override the defaults in SOC_init.	Drivers	SOC	SOC_init API is enhanced to accept a flag to bypass the default MPU settings. This allows application to own the MPU settings and configure it as per its needs. Existing application don't need a change if they are already memsetting the SOC_Cfg structure to 0 before using it for SOC_init API call.
Allow user to bypass the unhalting of DSS during SOC initialization	Drivers	SOC	SOC_init API is enhanced to accept a flag to bypass the unhalting of DSS. This allows application to keep DSS in its original state (unhalted or spinning in a while loop). One would use this functionality in the context of secondary bootloader or when running a mmwave application that doesn't involve the DSP. Existing application don't need a change if they are already memsetting the SOC_Cfg structure to 0 before using it for SOC_init API call.
SOC handle is now needed to be passed to the CLI module	Utils	CLI	CLI module now uses SOC APIs to determine the mmWave device during run time and thereby provide custom handling of certain CLI commands based on device type Existing application using CLI module should now populate the SOC handle in the CLI config structure before calling CLI_init. Failure to do so will result in init API failing and returning an error.

3. 3. Issues fixed

The following issues from previous releases were fixed in this release

Issue Type	Key	Summary

Bug	MMWSDK-1547	Intermittent Macronix flash write test failure seen due to incorrect programming sequence
Bug	MMWSDK-1528	Intermittent failure in QSPIFlash mmap read /write API test due to missing MEM_BARRIER in test code
Bug	MMWSDK-1224	Fix and clean up X-Y scatter plot
Bug	MMWSDK-1215	CANFD-Range Filter Not working

Known Issues

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

Issue Type	Key	Summary	Comments

Bug	MMWSDK-1571	xwr68xx demo: Profile with 1RX and 1TX crashes after a frame fails to complete execution	Due to a bug in objdethwa DPC for xwr68xx, the HWA engine can enter bad state and stalls the processing chain. To fix this:
			
			In objectdetection.c, modify the following line of code in the function DPC_ObjDet_AoAconfig from:
			
			aoaCfg.dynCfg.prepareRangeAzimuthHeatMap = &dynCfg->prepareRangeAzimuthHeatMap;
			
			to (no "&" on right hand side)
			
			aoaCfg.dynCfg.prepareRangeAzimuthHeatMap = dynCfg->prepareRangeAzimuthHeatMap;

Bug	MMWSDK-1552	xwr68xx demo: Errors in profile config or chirp config should not give exception	In xwr16xx and xwr14xx demos, such mis-configuration resulted in error on the CLI and user could re-send a new updated config after calling sensorStop and flushCfg commands. In xwr68xx mmw demo, due to lack of proper book-keeping, an assert is thrown. User can reboot the sensor and try the new config in such cases.
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-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-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-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	
Bug	MMWSDK-1157	Rare failure seen in UART loopback driver unit test - HW limitation	

3. 4. 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
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 only loopback mode on mmWave device TI EVM (however all features are implemented in the driver).
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, CQ cannot be streamed out reliably when datapath processing is also enabled. The data corruption for CQ data over LVDS lanes is seen more pronounced when multiple chirps/chirp event is enabled

4. Test reports

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

5. Installation instructions

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

- **mmwave_sdk_<version>-Windows-x86-Install.exe: Windows installer verified on Windows 7 and Windows 10 machines**

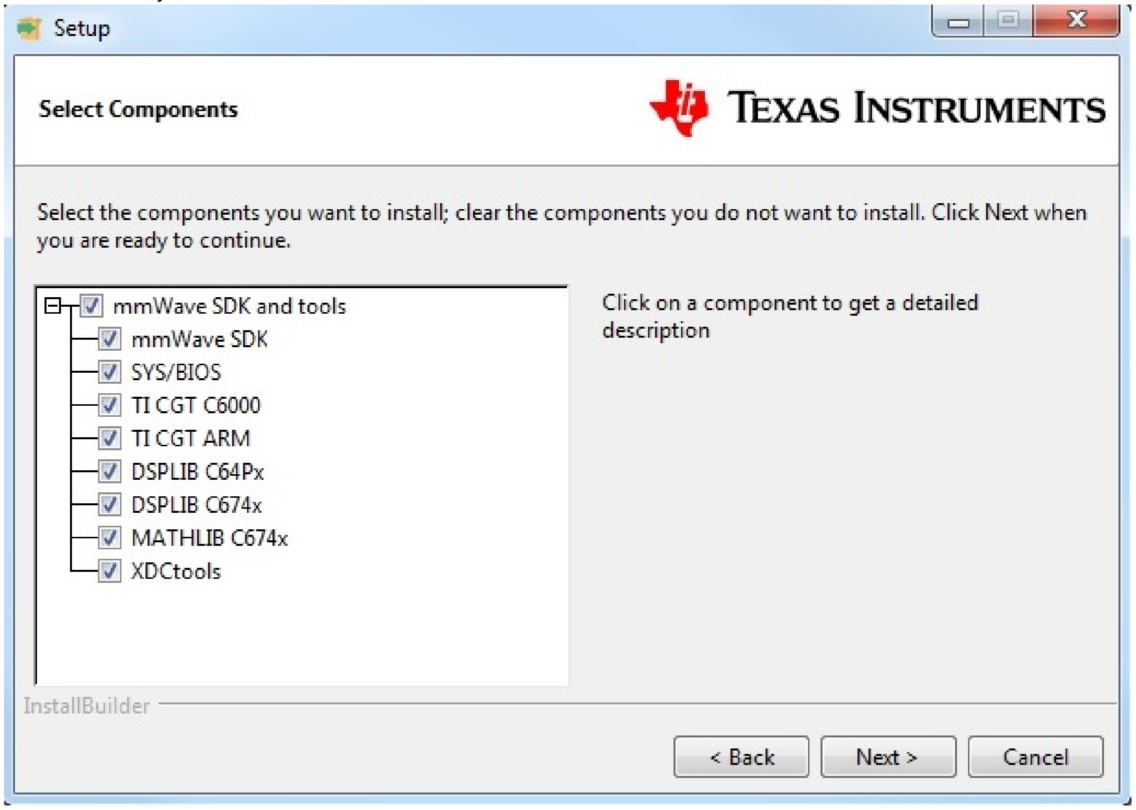
- **mmwave_sdk_<version>-Linux-x86-Install.bin: Linux installer verified on Ubuntu 14.04 & Ubuntu 16.04 64 bit machines.**

Depending on your development environment run the appropriate installer

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

Installation steps:

- Setup
- Choose Destination Location: Select the folder to install (default is c:\ti on windows and ~/ti on linux). 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.


Setup

Select Components

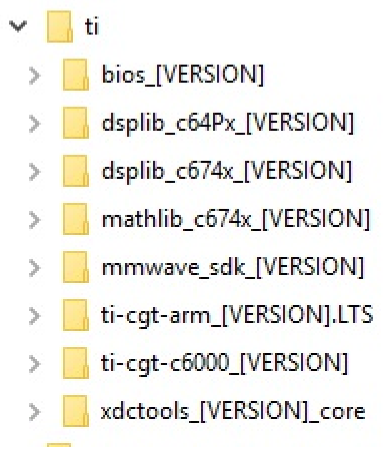
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [✓] mmWave SDK and tools
  - [✓] mmWave SDK
  - [✓] SYS/BIOS
  - [✓] TI CGT C6000
  - [✓] TI CGT ARM
  - [✓] DSPLIB C64Px
  - [✓] DSPLIB C674x
  - [✓] MATHLIB C674x
  - [✓] XDCtools

Click on a component to get a detailed description

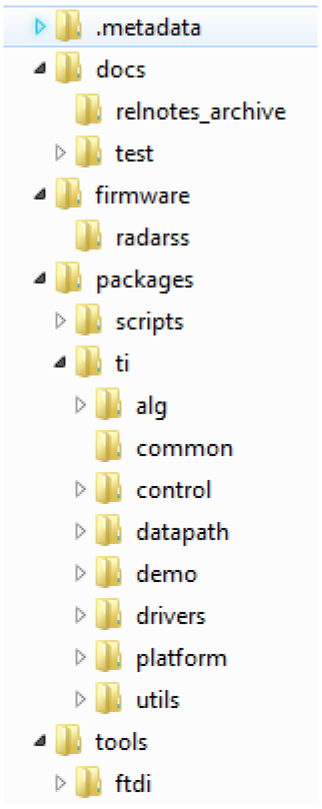
- Review installation decisions
- Ready to install
- Once installation starts all the selected components will be installed (if a component with the same version exists in the destination folder it will be overwritten)
- Installation complete

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



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

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



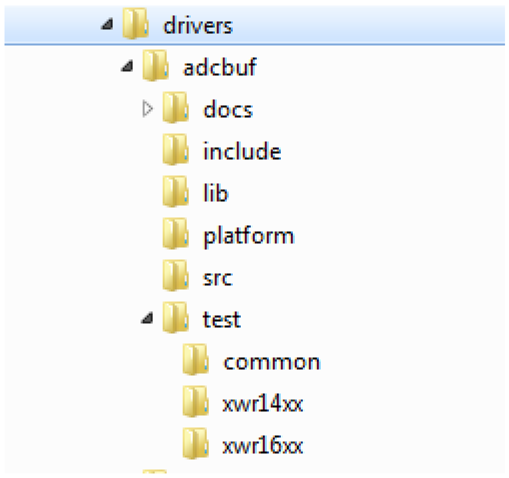
```
.metadata
docs
  relnotes_archive
  test
firmware
  radarss
packages
  scripts
  ti
    alg
    common
    control
    datapath
    demo
    drivers
    platform
    utils
tools
  ftdi
```

6. Package Contents

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

6. 1. Drivers

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



```
drivers
    adcbuf
        docs
        include
        lib
        platform
        src
        test
            common
            xwr14xx
            xwr16xx
```

- 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	
Static Clutter DPU	X	X	
CFAR CA DPU	X	X	
AoA DPU	X	X	
Datapath EDMA	X	X	
Object Detection DPC	X	X	X

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 only supported for IWR68xx device 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 MMWAVE-DEVPACK

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)