MMWAVE SDK Release Notes

Texas Instruments

Product Release 1.2.0

Release Date: Mar 9, 2018

Release Notes Version: 1.0

CONTENTS

1 Introduction
2 Release overview
- 2.1 What is new
- 2.2 Platform and Device Support
- 2.3 Component versions
- 2.4 Tools dependency
- 2.5 Licensing

3 Release content
- 3.1 New Features
- 3.2 Migration section
- 3.3 Issues fixed
- 3.4 Limitations

4 Test reports
5 Installation instructions
6 Package Contents
- 6.1 Drivers
- 6.2 Control
- 6.3 Algorithm
- 6.4 Usecases
- 6.5 Demos
- 6.6 Misc folders
- 6.7 Scripts
- 6.8 Firmware
- 6.9 Tools
- 6.10 Docs

7 Related documentation/links

1. Introduction

The mmWave SDK enables the development of millimeter wave (mmWave) radar applications using xWR1443 and xWR1642 SOCs. 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

- LTS release for pre-production devices as mentioned in the "Platform and Device Support" section below
- mmWave suite enhancements
- Demo enhancements
- Tools changes

More details can be found in NewFeatures section.

2. 2. Platform and Device Support

The device and platforms supported with this release include:

Supported Devices	Supported EVM
AWR1443 ES2.0	AWR1443BOOST - AWR1443 Evaluation Module
AWR1642 ES1.0	AWR1642BOOST - AWR1642 Evaluation Module
IWR1443 ES2.0	IWR1443BOOST - IWR1443 Evaluation Module
IWR1642 ES1.0	IWR1642BOOST - IWR1642 Evaluation Module

xWR14xx terminology is used in sections that are common for AWR14xx and IWR14xx

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

xWR14xx ES1.0 is not supported in this release.

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		1.2.0	Source and Binary	Overall package release version
RadarSS firmware		2.0.0	Binary	
mmWaveLink Framework		1.0.0	Source and Binary	
FTDI		2.12	Binary	
Image Creator	gen_bincrc32	1.0	Windows and Linux binary	
	out2rprc for xwr14xx	3.3	Windows binary	Need mono to run this on Linux
	out2rprc for xwr16xx	2.0	Windows binary	Need mono to run this on Linux
	Crc multicore image for xwr16xx	1.0	Windows and Linux binary	
	Multicore image generator for xwr16xx	1.0	Windows and Linux binary	

2. 4. Tools dependency

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

Tool	Version	Download link
CCS	7.4 or later	download link Please note that CCS v7.1 or later is mandatory. CCSv6.x cannot be used
TI SYS/BIOS	6.53.02.00	Included in mmwave sdk installer
TI ARM compiler	16.9.6.LTS	Included in mmwave sdk installer
TI CGT compiler	8.1.3	Included in mmwave sdk installer
XDC	3.50.04.43	Included in mmwave sdk installer
C64x+ DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x MATHLIB (little-endian, elf/coff format)	3.1.2.1	Included in mmwave sdk installer
Mono JIT compiler	4.2.1	Only for Linux builds
mmwave device support packages	1.5.5 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
TI Emulators package	6.0.0576.0 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
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
  - Drivers
    - ADCBuf driver provides API to get and set CQ configuration.
    - Added Watchdog driver for DSS
    - Added ESM driver for DSS
    - Added multi Chirp and CQ support in the CBUFF Driver for xWR16xx
  - mmWaveLink
    - Updated to support the RadarSS firmware version as noted above
  - mmWave
    - Enhanced mmWave APIs to pass BPM chirp configuration to Radar front end.
    - Exposed config to control the async event generation at the Radar front end
  - Algorithms
    - Group Tracker
    - mmWaveLib
      - CFAR-OS: Ordered-Statistic CFAR algorithm
      - Histogram of unsigned 16-bit vector.
      - Right shift function for signed16-bit vector and signed 32-bit vector.
      - Complex vector element-wise multiplication.
      - Sum of absolute value of 16-bit vector elements.
      - Max power search on 32-bit complex data
- mmWave demo
  - Added a parallel processing chain in mmW demo for xWR16xx that uses BPM MIMO scheme instead of traditional TDM MIMO scheme. User can choose between either schemes using CLI configuration.
  - Showcase debug/instrumentation usecase in mmW demo by sending ADC and user data over LVDS lanes (xWR16xx only).
  - Added new command in mmW demo to enable user to correct phase error for objects in near field.
  - Added support for reading and streaming CQ1 and CQ2 data every chirp.
- Usecases
  - Added 2 new usecases under the ti/drivers/test.
    - csi_stream, applicable for IWR14xx only, allows user to stream ADC data over CSI lanes.
    - mem_capture allows user to understand the programming model needed to configure the device and generate ADC data in device's L3 memory.
- Utilities
  - Added an optional utility library that creates a header for the data to be shipped over LVDS lanes.
- Documentation
  - Added a top level landing page (docs\mmwave_sdk_module_documentation.html) to navigate individual SDK component's documentation
- Tools
  - Upgraded FTDI drivers to the latest version
  - gen_bincrc32.exe now works on Windows 32bit
- Build
  - Align enum sizes (i.e. set it equal to size of int) across R4F and C674X for all the mmWave platforms.

3. 2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 1.2.0 release from 1.1.0 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component/s	Subcomponent	Behavior of Impact
Update TI dependent tools (compiler, BIOS, XDC, etc) in mmWave SDK package to their LTS versions	Build		SDK dependencies are updated as follows as per their LTS status: BIOS: 6.53.02.00 R4F CGT: 16.9.6 LTS XDC: 3.50.04.43 • SDK bundled package will contain these updated tools and hence user need not worry about installation. • The auto-generated environment file will be populated with these values for users • No performance impact was seen

Align enum sizes across R4F and C674X for all the mmWave platforms	Build		This change involves using the explicit value of 'int' for enum_type in makefiles and BIOS configuro option. With 16.x ARM compiler upgrade in last SDK releases, the enum size definition is not the same for R4F and C674X. If one uses the “packed” config, then size of enum will be variable and dependent on its values which needs special attention by user when understanding the underlying storage format. To facilitate interoperability, SDK makefiles uses explicit value of ‘int’. This involves compiling all R4 code (including BIOS generated files, BIOS pre-generated libraries, etc) with enum types forced to be int32. This is achieved by passing "--enum_type=int" option to compiler and configuro flags in mmwave_sdk.mak and to BIOS via application CFG file. All user demos should take care of adding this option in their application CFG file to avoid linker warnings and runtime errors. These changes were done for xWR16xx only. xWR14xx does not have to worry about interfacing with DSP and if enum is forced to be int32 it will increase memory consumption on already low memory device. After this change on xWR16xx, an increase of about 1KB in program memory on R4 was noticed.
Capture demo has been deprecated	Demos	Capture	Capture demo has been deprecated from the SDK and has been replaced by multiple targeted products: • mmWave Studio that allows user to capture raw ADC samples from mmWave Sensors and thereby enable performance characterization and algorithm development for mmWave technology • mmW Demo (xWR16xx) within mmWave SDK showcases the instrumentation usecase where raw ADC samples can be collected from the device while mmWave sensor continues to process those samples within the device • CSI stream (IWR14xx) application within mmWave SDK that showcases how to stream ADC data over CSI-2 interface • "mem_capture" application within mmWave SDK that helps a user get started with our mmWave sensors and the SDK by showcasing how to configure the front end and EDMA data from ADCbuf to L3. See doxygen of these new SDK modules for more internal implementation details.
Renamed Mailbox Driver Handle definition to avoid redefinition of SysBIOS structure	Drivers	Mailbox	Driver handle for Mailbox is renamed from Mailbox_Handle to Mbox_Handle since original definition conflicts with the Sysbios Mailbox module. This primarily applies to existing 16xx based demos that will now need to use this new handle type.
Update XWR16xx MSS ESM event mapping	Drivers	sys_common, ESM	ESM mapping defines have been updated to match the hardware documentation
Fix for crash seen during 1D processing when chirps with smaller chirp duration are enabled	Front end config, Control, Demos	RadarSS, mmWaveLink, mmWave, mmW demo	For the usecase of small chirps, CPU loading is high and any other interrupt in the system can delay 1D FFT chirp processing. When testing small chirps, 2 sources of such interrupts were identified: RadarSS Frame start async message and BIOS timer. With DFP 0.9, a feature was introduced to allow application to skip the Frame start notification since the chirp interrupt was a good enough indication. mmW demo will disable the frame start async notification from RadarSS so that it doesnt interfere with chirp processing. In BIOS prior to 6.52, the hooks to disable timer interrupt had undesirable side-effects which made disabling timer interrupt not practical. This was fixed in BIOS 6.52 (packaged as part of SDK 1.1 already). However after the frame start fix, disabling timer interrupt was not needed.
Correction for near by objects due to violation of the "far field" assumption in angle estimation	xWR16xx Demos	Processing Chain, Config File	SDK 1.1 OOB processing chain assumes that the object of interests are located in the far field so that the rays between the object and the multiple TX/RX antennas are parallel. However for very close by objects this assumption (of parallel lines) is not valid and can induce a significant phase error when processed using regular FFT techniques. User can use the new command 'nearFieldCfg' to enable the near field correction algorithm. This feature is new and existing user that do not wish to use this mode will only need to do the necessary CLI changes to disable the feature. The perl script mmwDemo_xwr16xx_update_config.pl can be used to automatically update the existing 1.1 based SDK profiles.

Add support for CQ in ADCBUF, CBUFF and mmW demo	Drivers, Demo	ADCBuf, mmW demo	New APIs are added in ADCBuf to set and get CQ addresses. mmW demo enhanced to configure the front end to generate CQ1 and CQ2 data. This data is EDMA'ed into L2 memory every chirp interrupt. No processing is done on this data at the moment. Note CQ is not supported for Advanced frame configuration. Refer to ADCbuf, mmWavelink and mmW demo doxygen for more details.
Limit maximum number of chirps per chirp available event to 8	Demos	mmW demo	In SDK 1.1, mmwdemo finds the maximum number of full chirps that can fit in the ADCBUF memory and allows these many number of chirps to be processed per chirp available event. However, the maximum number of CPs (and CQs) per chirp event are limited to 8 as per hardware. Since CP/CQ is tightly coupled with ADCBuf, the maximum number of chirps processed per event will be limited to 8. Refer to "Configuration file format" section in SDK user guide for more details.
Allow user to stream raw ADC data and/or user buffer over high speed interface LVDS while processing samples within the mmW demo	Demos	mmW demo	New feature added that allows user to stream out raw ADC data over the LVDS high speed interface while simultaneously processing it within the device. Additionally user can see prototype of how to send user data (detected objects) over the same interface after the interframe processing. Note: This feature is available only for xWR16xx device. There is a known issue that prevents streaming of CQ over LVDS while processing chain is active. New command is added to the input configuration to enable and configure the various sessions. This feature is new and existing user that do not wish to use this mode will only need to do the necessary CLI changes to disable the feature. The perl script mmwDemo_xwr16xx_update_config.pl can be used to automatically update the existing 1.1 based SDK profiles. See demo doxygen for more details.
Cbuff driver enhancements for CQ, headers and multiple chirps/interrupt	Drivers	CBUFF	• 'size' field used to specify userBuffer is now uint16_t to match the hardware limits. • New parameter added to H/W Session create parameters to signify number of chirps available per interrupt so that CBUFF can stream out the entire ADCBuf/CP/CQ buffer every interrupt. This should match the 'threshold' parameter passed to adcBuf driver. This parameter should be 1 for xWR14xx device • Updated CQ h/w session parameters to accept CQ sizes from the application. CBUFF driver internally uses ADCBuf driver handle to query the enabled state of respective CQn and its address. Application needs to configure CQ using ADCBuf driver before creating CBUFF h/w session. • Added documentation under the CBUFF driver (CBUFF_transfers.pptx) to showcase the flow of data over LVDS under various modes
Add HSI Header Module to facilitate LVDS data parsing at the receiver end	Utils		This is a new helper module for streaming data over LVDS when appending headers. Application has a choice to use this module as is or create their own header. It adds fields that helps the receiving application to make sense of the internal contents of the data buffer. This is located under ti/utils/hsiheader. Refer to doxygen docs of this module for more details.

3. 3. Issues fixed

The following issues from previous releases were fixed in this release

Issue Type	Key	Summary
Bug	MMWSDK-1095	Limit maximum number of chirps per event to 8
Bug	MMWSDK-1085	peakGrouping.EndRangeIndex parameter description in MMWAVE SDK 1.1 User Guide is incomplete
Bug	MMWSDK-1077	FTDI driver for devpack need to be signed with latest windows 10 certificates
Bug	MMWSDK-1060	Fix adcbuf address for CBUFF Continuous Mode
Bug	MMWSDK-1059	Supply missing doxygen documentation for calib DC feature in OOB demo.

Bug	MMWSDK-1058	mmwDemo: Intermittent mmwave control error received when switching between profiles
Bug	MMWSDK-1019	The minimum and maximum range limits are not honoured if peak grouping is disabled.
Bug	MMWSDK-1017	profileCfg.txStartTime is unsigned (should be signed)
Bug	MMWSDK-1016	Invalid data format CBUFF_DataFmt_USER in HW Session configuration
Bug	MMWSDK-1015	Align enum sizes across R4F and C674X for all the mmWave platforms
Bug	MMWSDK-1010	mmw demo returns error during multiple GPIO button presses
Bug	MMWSDK-998	User can no longer go to the plot tab page directly after browser page load and successfully perform load cfg
Bug	MMWSDK-984	Fix issues seen during CBUFF Driver Session Create/Delete
Bug	MMWSDK-953	Incorrect adcbuf address for non-interleaved cbuff transfer mode
Bug	MMWSDK-943	Corrections to User guide (COM port typo, interleaved setting)
Bug	MMWSDK-930	gen_bincrc32 does not work on Windows 32 bit machine
Bug	MMWSDK-905	Reject Frame configuration if num of loops setting is less than 8
Bug	MMWSDK-891	Visualizer: sensor Stop button doesnt get activated after first sensorStart for xWR14xx
Bug	MMWSDK-860	Mailbox Driver Handle definition conflicts with same definition in SysBIOS
Bug	MMWSDK-840	Azimuth heat map is no longer at zero Doppler for the improved scheme (DATA_PATH_CHAIN_COMBINED_LOGMAG) implemented in SDK 1.1
Bug	MMWSDK-800	Match LVDS Initialization Sequence as per h/w documentation
Bug	MMWSDK-763	ghost objects seen with OOB demo [on about 50% of boards] for near by objects due to violation of the "far field" assumption
Bug	MMWSDK-651	Small chirp times causes crash (MIPS overflow) in 1D processing.

Known Issues

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

Issue Type	Key	Summary	Comments
Bug	MMWSDK-1078	Limitation in mmW demo + LVDS instrumentation use case	See limitations section below
Task	MMWSDK-533	GUI of mmw demo running slow from Firefox browser	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. 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	mmW demo + 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.

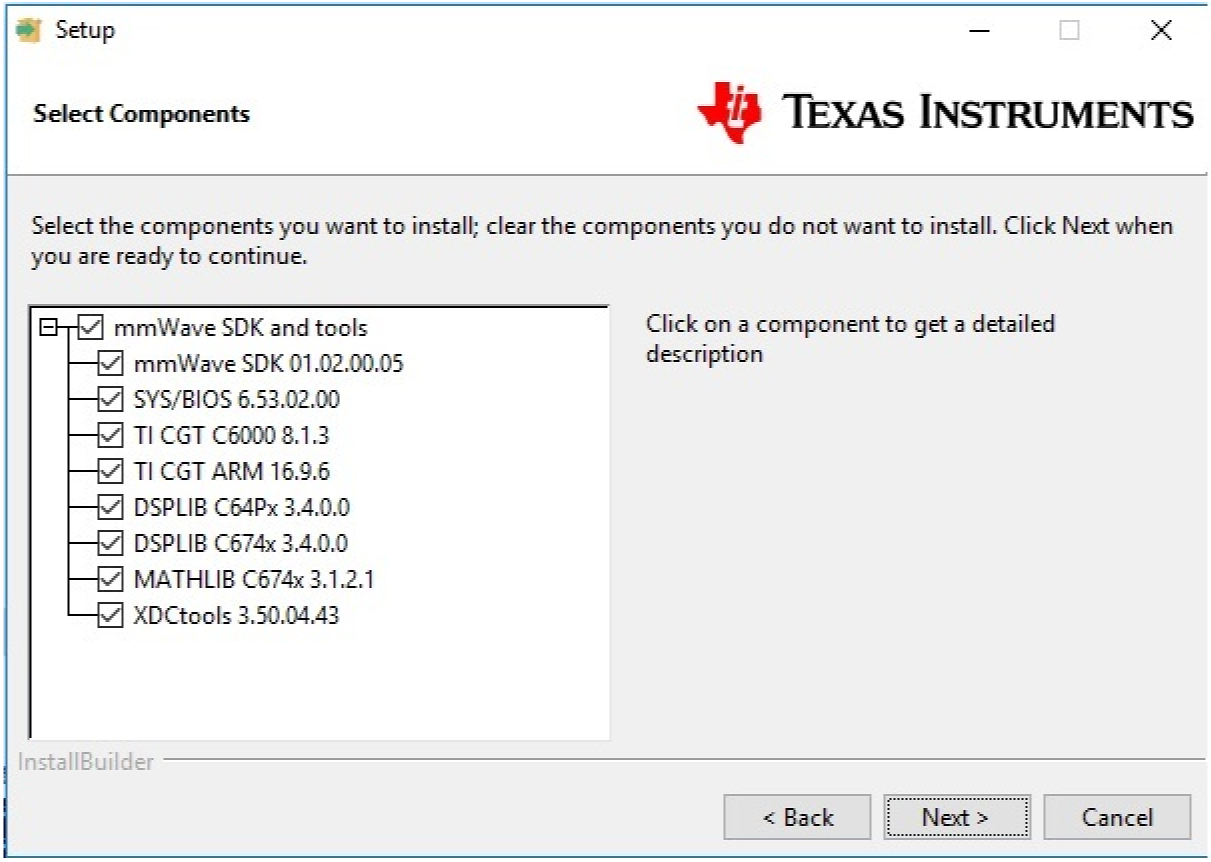
- **mmwave_sdk_<version>-Windows-x86-Install.exe: Windows installer verified on Windows 7 and Windows 10 machines**
- **mmwave_sdk_<version>-Linux-x86-Install.bin: Linux installer verified on Ubuntu 14.04 & Ubuntu 16.04 64 bit machines.**

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)
- Select Components: The installer includes all the tools needed for building the mmWave SDK. You should see a screen like below (the exact version in the installation may be different from the one shown below). The only reason to deselect a tool is if the tool is already installed in the destination folder.
-



Select Components

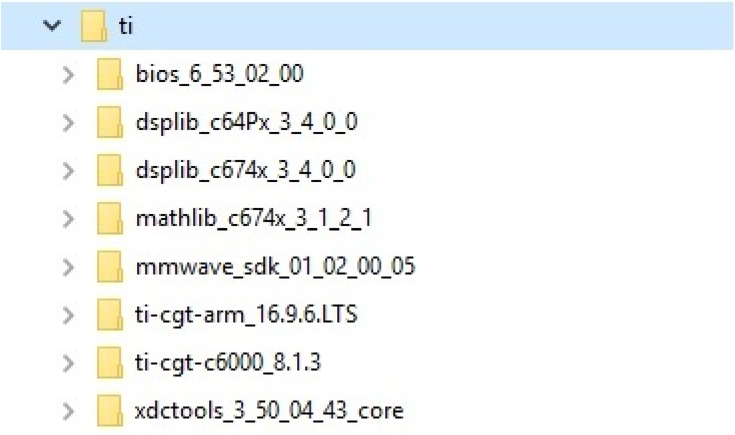
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ mmWave SDK and tools
  - ☑ mmWave SDK 01.02.00.05
  - ☑ SYS/BIOS 6.53.02.00
  - ☑ TI CGT C6000 8.1.3
  - ☑ TI CGT ARM 16.9.6
  - ☑ DSPLIB C64Px 3.4.0.0
  - ☑ DSPLIB C674x 3.4.0.0
  - ☑ MATHLIB C674x 3.1.2.1
  - ☑ XDCtools 3.50.04.43

Click on a component to get a detailed description

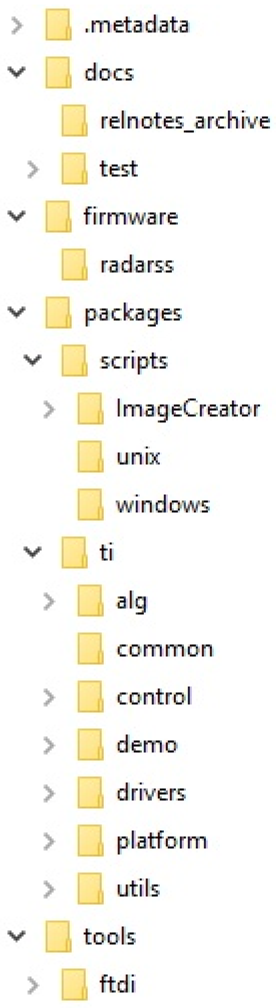
- Review installation decisions
- Ready to install
- Once installation starts all the selected components will be installed (if a component with the same version exists in the destination folder it will be overwritten)
- Installation complete

After the installation is complete the following folder structure is expected in the installation folder (please note that the version numbers may not be same as the one illustrated below)



```
ti
  bios_6_53_02_00
  dsplib_c64Px_3_4_0_0
  dsplib_c674x_3_4_0_0
  mathlib_c674x_3_1_2_1
  mmwave_sdk_01_02_00_05
  ti-cgt-arm_16.9.6.LTS
  ti-cgt-c6000_8.1.3
  xdctools_3_50_04_43_core
```

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



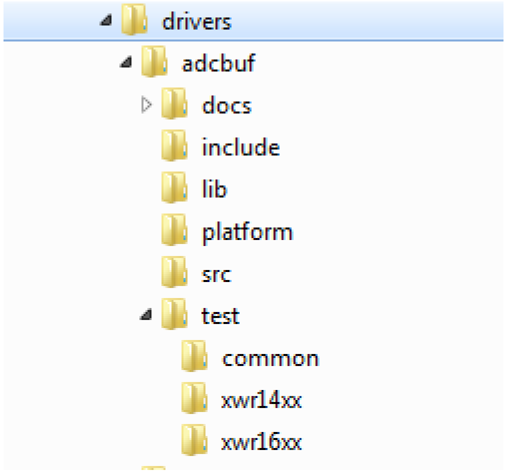
```
> .metadata
v docs
    relnotes_archive
  > test
v firmware
    radarss
v packages
  v scripts
    > ImageCreator
      unix
      windows
  v ti
    > alg
      common
    > control
    > demo
    > drivers
    > platform
    > utils
v tools
  > ftdi
```

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 that <platform: xwr14xx, xwr16xx>
- test/common: Unit test src files common for all platforms
- driver base folder has external header file, make files

Content of each driver is indicated in the table below.

Component	Source & prebuilt library	API Document (doxygen)	Unit test (source & prebuilt binary)
ADCBUF	X	X	X
CAN	X	X	X
CANFD	X	X	X
CBUFF/LVDS	X	X	X
CRC	X	X	X
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

6. 2. Control

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

Component	Source & Prebuilt Library	API Document (doxygen)	Unittest (source & prebuilt binary)
mmwavelink framework	X	X	X
mmwave high level api	X	X	X

6. 3. Algorithm

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

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

6. 4. 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
dsp_edma (xWR16xx only)	X	X	X
hwa_edma (xWR14xx only)	X	X	X
mem_capture	X	X	X

6. 5. Demos

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

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

6. 6. Misc folders

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

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

6. 7. Scripts

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

6. 8. Firmware

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

6. 9. Tools

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

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

6. 10. Docs

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

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

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

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

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

7. Related documentation/links

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

- SoC links:
  - AWR1443
  - AWR1642
  - IWR1443
  - IWR1642
- EVM links:
  - AWR1443BOOST
  - AWR1642BOOST
  - IWR1443BOOST
  - IWR1642BOOST
  - MMWAVE-DEVPACK