MMWAVE SDK Release Notes

Product Release 1.0.0

Release Date: May 2, 2017

Release Notes Version: 1.0

CONTENTS

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.

2. Release overview

2. 1. 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. 2. Component versions

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

Component	Version	Type	Comment
mmwave sdk	1.0.0	Source and Binary	Overall package release version
RadarSS firmware	1.7.0.4	Binary	
mmWaveLink Framework	0.7.1.1	Source and Binary	
FTDI	2.10	Binary	
Image Creator	1.0.0.0	Windows and Linux binary	

2. 3. Tools dependency

For building and using mmwave sdk the following external tool versions are needed. These tools are not included in the mmwave sdk release and need to be downloaded and installed separately. Instructions for installing the tools that don't have a download link are given in the mmwave_sdk_user_guide.

Tool	Version	Download link
CCS	7.1 or later	download link Please note that CCS v7.1 or later is mandatory. CCSv6.x cannot be used
TI SYS/BIOS	6.50.1.12 or later	download link
TI ARM compiler	16.9.1.LTS or later	Included in CCS v7.1

TI CGT compiler	8.1.3 or later	Included in CCS v7.1
XDC	3.50.00.10 or later	download link Use the installer that has no mention of JRE which by default has JRE (this installer will be bigger than the one without JRE). Also it is recommended that this be installed separately even if CCS has this version installed since the CCS version is without JRE.
C64x+ DSPLIB	3.4.0.0	Please download the installer for **C64x+**. download link
C674x MATHLIB (little-endian, elf/coff format)	3.1.2.1	Please download the installer for **C674x**. download link
Perl	5.20.2 or later	download link
CRC.pm	0.21	download link This is an add-on to the Perl installation. This file needs to be copied to <PERL installation folder>\perl\lib\Digest folder.
Mono JIT compiler	3.2.8	Only for Linux builds
mmwave device support packages	1.5.3 or later	Upgrade to the latest using CCS update process
TI Emulators package	6.0.0576.0 or later	Upgrade to the latest using CCS update process
Pinmux tool (optional)	Latest	download link Used to generate pinmux configuration for custom board
Doxygen (optional)	1.8.6	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	4.1.2.1329 or later	link Uniflash Standalone Flash Tool download page
mmWave Demo Visualizer	Latest	link TI Gallery APP for configuring mmWave sensors and visualizing the point cloud objects generated by the mmWave SDK demo

2. 4. Licensing

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

3. Release content

3. 1. 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
- Tools
  - Host tools including Pin Mux, Flashing utilities
  - Code Composer Studio™ IDE for RTOS development
- Documentation

3. 2. Features list by component

3. 2. 1. Drivers

Drivers/Hardware IP	Platform supported	Driver Functionality Implemented in mmWave SDK
ADCBUF	xWR14xx R4F xWR16xx R4F xWR16xx DSP	All features of IP are implemented in the driver except for CQ
CAN	xWR14xx R4F xWR16xx R4F	Following features of IP are implemented in the driver: Configure Rx and Tx I/O Control registers Configure DCAN mode of operation Configure DCAN controller, interrupts, ECC, parity Set bit time parameters Configure Rx and Tx message objects Receive and Transmit a CAN message Retrieve Tx message object transmission status and Rx message object reception status Check the validity of the received message
CANFD	AWR16xx R4F	Following features of IP are implemented in the driver: • Reset MCAN driver • Initialize MCAN clock stop controls, auto wakeup, MCAN mode - classic versus FD mode, transceiver delay conpensation • Configure MCAN controller and global filters • Configure MCAN mode of operation • Set bit time parameters • Configure message filters, Rx/Tx FIFOs • Add and cancel Tx requests • Transmits a CAN message • Receive a CAN message • Check the validity of the received message • Configure interrupt multiplexer to service message objects • Retrieve interrupt line status, interrupt pending status, parity error status, bit error status, ECC diagnostics status, ECC error status and MCAN error status • Clear interrupt pending status, ECC diagnostics error status, ECC error status and MCAN error status • Configure MCAN parity function, self test mode, ECC Diagnostic mode

CBUFF	xWR14xx R4F xWR16xx R4F xWR16xx DSP	Following features of IP are implemented in the driver: Supports Platform defined HSI: LVDS or CSI (IWR14xx only). Initialize and setup the CBUFF Driver Configure the Linked List and EDMA Channels to support the data transfer Supports Interleaved and Non-Interleaved data mode Supports the data formats: ADC, ADC_CP, CP_ADC, CP_ADC_CQ, ADC_USER, CP_ADC_CQ_USER, USER Supports CRC
CBUFF (LVDS)	xWR14xx R4F xWR16xx R4F xWR16xx DSP	Following features of IP are implemented in the driver: LVDS driver supports the chirp and continuous mode of data transmission. Supports only 2 and 4 lane configuration in Format0
CRC	xWR14xx R4F xWR16xx R4F xWR16xx DSP	All features of IP are implemented in the driver including: CRC-16 CRC-32 CRC-64
CSI-2	IWR14xx R4F	Following features of IP are implemented in the driver: Initialization and Setup of the Protocol Engine Initialization and configuration of the DSI PHY DSI Phy Parameters can be customized by the application
DMA	xWR14xx R4F xWR16xx R4F	Following features of IP are implemented in the driver: software and hardware triggered transfer frame based transfer block based transfer Addressing mode (Constant, Indexed, Post Increment) FTC, BTC, LFS, HBC interrupts channel chaining auto-initiation mode interrupt based and polling based channel completion APIs
EDMA	xWR14xx R4F xWR16xx R4F xWR16xx DSP	All features of IP are implemented in the driver except "privilege" feature
ESM	xWR14xx R4F xWR16xx R4F	Default ESM FIQ Interrupt handler
GPIO	xWR14xx R4F xWR16xx R4F	All features of IP are implemented in the driver
HWA	xWR14xx R4F	All features of IP are implemented in the driver
I2C	xWR14xx R4F xWR16xx R4F	All features of IP are implemented in the driver

MAILBOX	xWR14xx R4F xWR16xx R4F xWR16xx DSP	All features of IP are implemented in the driver.
OSAL	xWR14xx R4F xWR16xx R4F xWR16xx DSP	Provides an abstraction layer for some of the common OS services: Semaphore Mutex Debug Interrupts Clock Memory
PINMUX	xWR14xx R4F xWR16xx R4F	All Pinmux fields can be set and all device pad definitions are available
QSPI	xWR14xx R4F xWR16xx R4F	All features of IP are implemented in the driver.
QSPIFLASH	xWR14xx R4F xWR16xx R4F	All features of IP are implemented in the driver.
RTI	xWR14xx R4F xWR16xx R4F	Part of TI RTOS offering
SOC	xWR14xx R4F xWR16xx R4F xWR16xx DSP	Provides abstracted APIs for system-level initialization. See User guide for more details.
SPI (MIBSPI)	xWR14xx R4F xWR16xx R4F	All features of IP are implemented in the driver including: 4-wire Slave and master mode 3-wire Slave and Master mode both polling mode and DMA mode for read/write char length 8-bit and 16-bit.
VIM	xWR14xx R4F xWR16xx R4F	Part of TI RTOS offering
UART	xWR14xx R4F xWR16xx R4F xWR16xx DSP	All features of IP are implemented in the driver including: Standard Baud Rates: 9600, 14400, 19200 till 921600 Data Bits: 7 and 8 Bits Parity: None, Odd and Even Stop Bits: 1 and 2 bits Blocking and Polling API for reading and writing to the UART instance

3. 2. 2. Control

3. 2. 2. 1. mmwavelink framework

Radar SS is a closed subsystem whose internal blocks are configurable using messages coming over mailbox.
TI mmWaveLink framework acts as driver for Radar SS and exposes services of Radar SS. It includes APIs to configure HW blocks of Radar SS and provides communication protocol for message transfer between MSS/DSS and RADAR SS.

- Link between application and Radar SS
- Handles communication errors, Notifies exceptions
- Platform and OS independent
- Can work in single threaded (non OS) environment

3. 2. 2. 2. mmwave high level api

- Abstraction over mmWaveLink
- Handles mmWaveLink integration onto the mmWave SDK for both xWR14xx and xWR16xx
- Simple set of APIs for init, config and start of the mmWave device

See mmWave SDK user guide for more details.

3. 2. 3. Algorithms

3. 2. 3. 1. mmwavelib

- collection of algorithms that provide basic functionality needed for FMCW radar-cube processing.
  - Windowing (16-bit complex input, 16 bit complex output, 16-bit windowing real array)
  - Windowing (16-bit complex input, 32 bit complex output, 32-bit windowing real array)
  - log2 of absolute value of 32-bit complex numbers
  - vector arithmetic (accumulation)
  - CFAR-CA, CFAR-CASO, CFAR-CAGO (logarithmic input samples)
  - 16-point FFT of input vectors of length 8 (other FFT routines are provided as part of DSPLib)
  - single DFT value for the input sequences at one specific index
- Optimized and available for xWR16xx C674x DSP only

3. 2. 4. Demo

- capture demo
- mmw demo

See mmWave SDK user guide for more details on these demos.

3. 2. 5. RadarSS firmware

RadarSS firmware is responsible for configuring RF/analog and digital front-end in real-time, as well as to periodically schedule calibration and functional safety monitoring. This enables the mmWave front-end to be self-contained and capable of adapting itself to handle temperature and ageing effects, and to enable significant ease-of-use from an external host perspective.

Features/enhancements information can be found under firmare/radarss/mmwave_radarss_release_notes.pdf

3. 3. Known Issues in MMWAVE_SDK_01_00_00 release

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

Key	Issue Type	Summary	Comments
MMWSDK-591	Bug	mmw demo azimuth heatmap does not rescale after changing range depth/width and using just the stop/start button	Workaround: Set the desired range depth/width and use the reconfiguration method (with all other config intact) instead of just stop/start.
MMWSDK-589	Bug	Capture demo with streaming and 4 RX channels enabled is not working	Workaround: Set the cbuffCfg.rxChannelStatus to active for all valid RX channels

MMWSDK-586	Bug	xWR14xx demo displays many invalid detections	The scattered points at the end of range profile across Doppler bins and azimuth angle are false detections due to sharp anti aliasing filter in front of ADC. The number of these points is more pronounced with CFAR_CASO approach, since the right side of noise averaging window would be selected more often due to sharp profile level drop. To reduce the number of these scattered points, you can either increase threshold, use CFAR_CA approach, or in peak grouping command set the upper range selection index to (0.9 x range FFT size).
MMWSDK-579	Bug	xWR16xx/xWR14xx mmw demo crash when doing multiple attempts of stop->reconfig->start	Workaround: Reboot the board and try the same configuration again.
MMWSDK-575	Bug	GUI shows Y-axis truncation of range profile for given configuration for xWR16xx	Variable level of range profile plots is because of different scaling in DSP LIB FFT functions.
MMWSDK-574	Bug	The mmWave_stop on the remote peer will not get invoked if the rlSensorStop returns non-zero value	
MMWSDK-562	Bug	xWR14xx mmw demo should abort when memory allocation fails	In function MmwDemo_dataPathCfgBuffers when memory allocation fails, there is a print that memory allocation fails but no assert is raised.
MMWSDK-553	Bug	rlSetProfileConfig API with multiple profile configures only 1 profile	Workaround: Call rlSetProfileConfig multiple times, once for every profile
MMWSDK-540	Bug	rlRfGetTemperatureReprt API returns error when executed from MSS or DSS	
MMWSDK-533	Task	GUI of mmw demo running slow from Firefox browser	Please switch to Chrome browser.
MMWSDK-528	Bug	mmwDemo should match the UART output for platform against requested output data and generate error if output data exceeds throughput	
MMWSDK-508	Bug	mmwavelink: MSS get version fails if invoked before RF-power on.	
MMWSDK-252	Story	UART driver has not tested for Data Length 5 and 6	
MMWSDK-218	Bug	Significant bias of about 8 cm seen in range measurement in mmw demo.	

3. 4. Limitations

Some of these limitations are captured in the "known issues" list shown in previous section.

1	ADCBuf driver: CQ feature is not implemented.
2	CAN driver: • DMA and FIFO mode are not supported
3	CANFD driver: • DMA and Timestamping are not supported

4	CBUFF/CSI2/LVDS:
	• Limited functionality is tested using stream demo
	  • Only ADC data format has been verified.
	  • Continuous mode support is experimental.
	• Driver does not support the following functionality:
	  • Multiple packets
	  • 3 channels

5	CRC driver: "Auto" mode is not implemented.
6	DMA driver: MPU and Parity Feature not implemented.
7	EDMA driver: Privilege feature not implemented.
8	ESM: Custom application hooks for ESM interrupts are not implemented.

9	HWA driver: Any modes/algorithm outside the scope of mmWave demo are not tested (however they are implemented in the driver).
10	I2C driver: Verified only loopback mode on mmWave device TI EVM (however all features are implemented in the driver).
11	QSPI/QSPI Flash driver: ■ dual-Read/Quad read in configuration mode is not supported ■ setting write protections bits is not supported
12	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.
13	DMA based transactions are not supported for UART, CRC and Mailbox driver.
14	capture demo: ■ Data equivalent to the L3 memory available on a given device can be collected in a given execution. ■ CSI-2: The demo has only been verified in Raw 12 and Raw14 mode with the Data format set to CBUFF_DataFmt_ADC_DATA. ■ LVDS: The demo has only been verified in the 16bit output format in DDR mode with the Data format set to CBUFF_DataFmt_ADC_DATA. ■ Testing was performed using the configuration available in the demo directory (See User Guide for more details)
15	mmW demo: See demo's doxygen page for more details.

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:

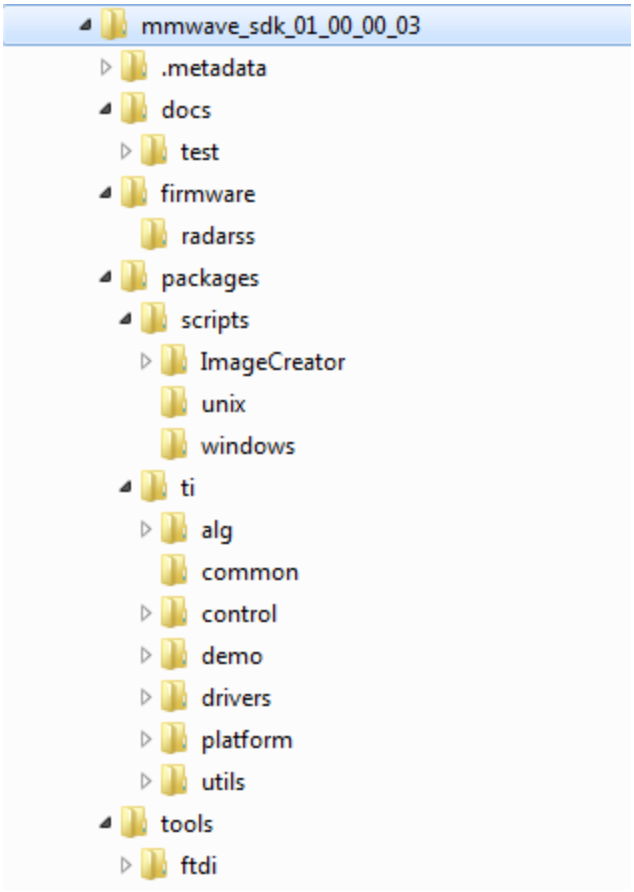
mmwave_sdk installer is available as a Windows Installer and a Linux installer.

- **mmwave_sdk_<version>-Windows-x86-Install.exe: Windows installer verified on Windows 7 PC**
- **mmwave_sdk_<version>-Linux-x86-Install.bin: Linux installer verified on an Ubuntu 14.04 64 bit machine.**

Depending on your development environment install the appropriate installer

- 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
- In Windows environment, double clicking the installer from Windows explorer should start the installation process

After the installation is complete the following folder structure is expected in the installation folder (please note that the version number following mmwave_sdk folder name will match the actual release and may not be same as the one illustrated below)

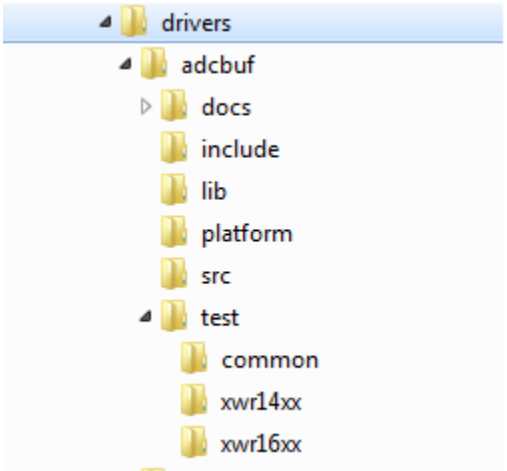


6. Package Contents

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 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

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)
mmwavelib	X	X	X

6. 4. 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
capture	X	X	
mmw	X	X	X

6. 5. 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
  - testlogger which is the helper utility for driver unit tests

6. 6. Scripts

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

6. 7. 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. 8. 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. 9. 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
- mmwave_sdk_user_guide.pdf: User guide

mmwave_sdk_<ver>/docs/test folder contains test results for each SoC. Each SoC folder in turn contains

- <soc>.pdf: Summary Test report

- Results.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