MMWAVE SDK Release Notes

Product Release 3.2

Release Date: May 10, 2019

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
AWR1843 ES1.0	AWR1843BOOST - AWR1843 Evaluation Module RevB
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 RevB
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		3.2	Source and Binary	Overall package release version
RadarSS firmware (patch) for xwr14xx, xwr16xx, xwr18xx		1.2.0.3	Binary	RadarSS firmware is in ROM. Only the patch is included in the mmwave sdk release
RadarSS firmware for xwr68xx		6.0.7.0	Binary	
mmWaveLink Framework		1.2.1	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	8.3.3	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	2.0.0	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
    - Drivers
        - HWA driver: Changes related to handling of silicon errata
        - I2C driver: added new test to scan all I2C addresses and report the valid address that is active on the board it is being tested on
    - mmWaveLib Enhancements
        - new accumulator function with shift
    - mmWave data processing layers
        - New datapath modules for Doppler, CFARCA and AoA using DSP for processing
        - AoA HWA DPU: added max Velocity algorithm
        - Added unit tests for AoA, Doppler, CFAR DPUs
        - DPC: New DPCs added:
            - one DPC with all low level mmwave processing in DSP and other DPC with Range processing using HWA and rest of the low level mmwave processing using DSP
- mmWave Demos enhancement
    - 68xx HWA only demo: This has been moved to xwr64xx directory and can be invoked from Visualizer by selecting xwr64xx as the platform - the actual device to use for this demo is still IWR6843;
        - Added max velocity and lvds streaming feature
        - CQ feature is added but currently disabled due to limitation in the RadarSS firmware
        - CFAR cfg command now accepts the threshold in dB instead of linear value
    - 68xx HWA+DSP enhanced demo: This is located in 68xx directory and can be invoked from Visualizer by selecting 68xx as the platform;
        - Added lvds streaming feature
        - CQ feature is added but currently disabled due to limitation in the RadarSS firmware
        - BPM feature is added but currently disabled due to limitation in the RadarSS firmware
        - Added max velocity CLI command but is currently disabled as this feature is not yet supported in AoA DSP DPU
        - CFAR cfg command now accepts the threshold in dB instead of linear value
    - 18xx HWA only demo:
        - Added max velocity, lvds streaming and CQ features.
        - CFAR cfg command now accepts the threshold in dB instead of linear value
    - 16xx DSP only demo: Migrated the old demo to the new architecture and has support for all features except for max velocity (it is currently disabled as this feature is not yet supported in AoA DSP DPU) and near field correction
    - mmWave Visualizer enhancement
        - Added a console clear button to clear up console messages
        - Added a button to export tuned profile after using the real-time tuning or advanced commands
        - Added a maximize/restore button in every plot's pop-up toolbar to enlarge any given plot
- mmWave Utilities
    - Added utility module to generate Gaussian distributed random complex number with specified variance
- Components/Tools
    - Updated RadarSS firmware for IWR6843 as noted above
    - Updated CGT toolchain to 8.3.3 as noted above

3. 2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.2.0 release from 3.1.1 release. See release notes archive in the SDK release package for migrating from other older releases.

Summary	Component/s	Subcomponent	Behavior of impact

Directory move of xwr68xx HWA only demo	Demos	xwr68xx	This demo has been moved to 64xx directory and can be invoked from Visualizer by selecting 64xx as the platform - the actual device to use for this demo is still IWR6843; In addition to the features supported in previous releases, new CLI command to support the following have been added. - max velocity - lvds streaming feature. - CQ feature (this CLI is added but should be always set to disabled due to limitation in the RadarSS firmware) - CFAR cfg command now accepts the threshold in dB instead of linear value to simplify user's migration path due to change in Q format of detection matrix across demos User can use ti/demo/<platform>/mmw/profiles /mmwDemo_<platform>_update_config.pl to migrate CFG files from previous release to the current one. It will require manual editing to migrate the CFAR CFG's threshold change.
Feature addition to xwr18xx HWA only demo	Demos	xwr18xx	In addition to the features supported in previous releases, new CLI command to support the following have been added. - max velocity - lvds streaming feature. - CQ feature - CFAR cfg command now accepts the threshold in dB instead of linear value to simplify user's migration path due to change in Q format of detection matrix across demos User can use ti/demo/<platform>/mmw/profiles/mmwDemo_<platform>_update_config.pl to migrate CFG files from previous release to the current one. It will require manual editing to migrate the CFAR CFG's threshold change.
IWR68xx DFP updated	RadarSS firmware	-	Updated to latest radarss firmware for IWR6843 device
TI CGT updated to 8.3.3	Tools	CGT	setenv scripts are updated. Also updated CFLAGS for C674x from -ms0 to -mf3in mmwave_sdk.mak If the application doesn't use the default flags defined in mmwave_sdk.mak, it needs to update the CFLAGS manually
GTRACK: updated gating related structures	alg	gtrack	Updated struct for gateLimits and GTRACK_gatingParams. User should refer to the gtrack internal documentation at mmwave_sdk_<ver>\packages\ti\alg\gtrack\docs
sys_types.h: added alignment on complex structure types	Common	system defines	Added alignment for complex structures for DSP to fulfill the 4 byte alignment needed by optimized DSP routines/intrinsics. No action needed in the application except that there might be slight increase in heap usage due to alignment
AoA HWA DPU config API updated to request more EDMA/HWA resources	Datapath	AoA HWA DPU	AoA HWA DPU: Needs more system resources (EDMA, HWA) for the ping-pong scheme to enable faster execution time. See objdethwa DPC and xwr64xx/xwr18xx mmw_res.h resources file for integration details
AoA HWA DPU: added max Velocity algorithm	Datapath	AoA HWA DPU	To support this new feature, added max velocity related params to config and control APIs
AoA HWA DPU: process function returns errCode as Int32 data type and not uint32_t (which was incorrect)	Datapath	AoA HWA DPU	This is a result of bug fix where AoA HWA DPU was incorrectly reporting back errorCode (signed value) as uint32_t. Now it correctly reports back using Int32 data type.
CFAR DPU: Needs the CFAR threshold to be specified in Q8 format	Datapath	CFAR HWA DPU	CFAR CFG dynamic IOCTL expects the threshold to match the detection matrix which is now moved to use Q8 format
Doppler DPU: Produces detection matrix in Q8 format. Range Profile TLV will also be in Q8 format	Datapath	Doppler DPU	For simpler integration across HWA and DSP based DPUs/demos, the detection matrix and Range profile TLV will now be provided in Q8 format irrespective of whether Doppler DPU uses HWA or DSP for processing. The gain due to Virtual antennas would be consistent and equal to: (N / N') where N is numVirtualAntennas and N' = 2^ceil(log2(N)). For HWA based demos: This change means migration from Q9 format to Q8 format For DSP based demos: This change means the detection matrix will have the updated gain which was N in old demos and now its changed to (N/N') as explained above.
DPEDMA: updated the name of the library from libdpedma_dsp to libdpedma_base	Datapath	DPEDMA	Updated the library name to be more consistent with the functionality.

RangeProc DSP DPU: DPU_RangeProcDSP_process now accepts DPU_RangeProcDSP_OutParams instead of DPU_RangeProcDSP_OutParams (bug fix)	Datapath	Range DSP DPU	Minor bug fix with no change in functionality
CBUFF driver: Default CBUFF library allows single chirp mode with ADC and CP_ADC_CQ format	Drivers	CBUFF	Default CBUFF library allows single chirp mode with ADC and CP_ADC_CQ format. Users can customize the supported formats table in the platform specific section of the driver to add other modes as needed. Multi-chirp mode and CP_ADC_CQ format has a chance of causing underflow in hardware which can mislead the receiver and hence users are advised to evaluate their config before enabling this in the driver
EDMA: return error if the H/W instance doesn't allow completion /error interrupts and user specifies callback function during instance initialization	Drivers	EDMA	This was added as extra checks in the driver and calling apolication should make sure to avoid such situations since callback will never be called in this scenario where interrupts are not hooked up in the H/W.
HWA: deprecated HWA individual paramset completion interrupt due to silicon errata	Drivers	HWA	HWA silicon has an errata which prevents concurrent access of command static registers while HWA is active. This paramset interrupt happens while HWA is active and needs a read of common static register while servicing -> hence this API is disabled for the current HWA silicon revision. Workaround: User could have dummy EDMA transfers at the end of desired paramset and use the EDMA paramset interrupt to achieve their desired functionality
HWA: changed the read address for hot signature from HWA common set registers to a copy in local core memory due to silicon errata	Drivers	HWA	HWA silicon has an errata which prevents concurrent access of command static registers while HWA is active. This hot signature read from common static register is required while HWA is active -> hence the driver now keeps a local copy of this signature and returns the local memory address when user calls HWA_getDMAconfig(). No change is required at the application level when using the driver APIs to get the EDMA programming parameters for Hot signature feature.
CLI: CLI_open now accepts a user defined string to override the default Platform that gets displayed in the version command	Utils	CLI	This is added to allow xwr64xx demo to run on a IWR68xx superset device using Visualizer.
SBL: Increased the wait time from 20secs to 90 secs to allow user more time to stop the autoboot process	Utils	SBL	This is added to allow enough time to users to stop the autoboot process on PC's where UART detection could take longer time
SBL: User can use only the Enter and Space key to stop the autoboot process	Utils	SBL	The list of keys to stop autoboot has been restricted to certain keys since certain PC's could have drivers that send some handshake data over COM /UART before user gains access to the COM port. To prevent false stopping of autoboot process, SBL now listens to allow certain keys. User can update the SBL code to use different set of keys as desired on their setup.

3. 3. Issues fixed

Issue Type	Key	Summary
Bug	MMWSDK-1860	SBL doesn't disable the interrupts before switching to application
Bug	MMWSDK-1854	Visualizer does not accept manually created profile config with un-necessary spaces in the commands
Bug	MMWSDK-1851	CCS/IWR6843: Incorrect extern C in two header files of MMWAVE_SDK 3.1.1.2
Bug	MMWSDK-1848	DPM missing semaphore delete during deinit
Bug	MMWSDK-1833	AoA doxygen figure correction
Bug	MMWSDK-1831	Missing doppler window generation before doppler config in the execute chain of HWA obj detection DPC

Bug	MMWSDK-1825	HWA driver: disallow individual paramset interrupt due to silicon errata
Bug	MMWSDK-1798	IWR6843 variable alignment issue in MMWAVE SDK
Bug	MMWSDK-1764	I2C driver: inaccurate settings for clock and issues in handling errors in ISR
Bug	MMWSDK-1760	Wrong limits set to the bittiming values of MCAN
Bug	MMWSDK-1706	Azimuth heatmap calculation produces extra EDMA event
Bug	MMWSDK-1703	Workaround for HWA DMA based trigger mechanism
Bug	MMWSDK-1696	objdetHwa DPC cleanup
Bug	MMWSDK-1692	cmplx16ImRe_t and cmplx16ReIm_t types must be properly aligned when used by intrinsics
Bug	MMWSDK-1678	SDK 3.1 package: Missing 3.0 RN in the relnotes_archive folder
Bug	MMWSDK-1654	AOA DPU process should return Int32 instead of Uint32
Bug	MMWSDK-1556	Visualizer: Error message is persistent after GUI changes focus
Bug	MMWSDK-1078	mmw demo CQ data corruption when streaming and CQ mon enabled

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	UNIFLASH-1195	mmwave device IWR6843: Unable to flash 2 MetaImages via the command line package	This issue exists only when using the command line package from Uniflash and not from the GUI. When 2 metaImages are provided to be flashed onto IWR6843 device, only the first metaImage gets flashed properly.
Bug	MMWSDK-1872	EDMA transfer can stall while running DPC test code on AWR18xx	When the HWA based DPC test code is configured to repeat 10 identical frames, the EDMA transfer can stall at random while repeating identical frame executions. It appears that the problem is related to EDMA read issue while crossing the 4K boundary. Some of the EDMA source and destination addresses in L2 memory happened to be just before the 4K boundary, and the EDMA transfer from HWA to L2 after several frame repetitions became stalled during the transfer from HWA to L2 memory. After forcing the addresses of input/ouput buffers in L2 memory just above 4K boundary, the problem appeared to go away. These EDMA transfers are configured as A-sync transfers with bcount = 1.
Bug	MMWSDK-1871	If the chirp idle times between different bursts /subframes are different in advanced subframe profile for xwr64xx and xwr68xx devices, it can lead to RF CPU fault	The RadarSS in DFP 6.1 cannot handle the case where certain profile config has <10us chirp idle time and others have >10us chirp idle times and all these profile configs are active in the advanced frame configuration. Workaround (choose any one): 1. Increase the idle time of the profiles to ensure they're always >= 10us. (OR) 2. Issue a Dynamic Power Save Configuration API with all fields disabled before frame trigger. (rlRfDynamicPowerSave)
Bug	MMWSDK-1870	LVDS streaming of S/W data can get stuck sometimes depending on timing and/or buffer placements	This issue is under debug and rootcause/workaround is unknown. When such situation happens, underlying CBUFF IP doesn't provide the completion event for that session and in mmW demo, this can lead to debug assert as the previous frame would be marked incomplete and demo will prevent the next frame to proceed.

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) 1.2.0.3 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. This section captures additional known issues present in this release of RadarSS which are not captured in the release notes.

Issue type	Key	Summary
Bug	AUTORADAR-1951	ESM self-test monitoring in periodic runtime digital monitoring doesn't work as expected. Workaround: The “ESM MONITORING EN” field in “AWR MONITOR RF DIG PERIODIC CONF SB” should be left disabled.
Bug	AUTORADAR-1943	Issues seen with digital monitoring in “AWR MONITOR RF DIG LATENTFAULT CONF SB” API in long tracking tests. Workaround: Disable Latent fault digital monitoring API
Bug	AUTORADAR-1941 AUTORADAR-1950	Race condition seen rarely with register read-back checks with digital monitoring enabled which would cause RadarSS to throw fatal error
Bug	AUTORADAR-1971	There is a possibility of race condition in getting available time for calibration and monitoring chirps in inter burst /frame time, which would cause firmware fatal error

Bug	AUTORADAR-1998 MMWAVE_SOC-58	There is a possibility of Rampgen parity self-test monitoring generating G2 rampgen parity error in long overnight runs Workaround: Rampgen parity self test should be disabled
Bug	AUTORADAR-2006	Sequencer extension ECC and access continuous monitoring and reporting is not enabled as part of digital monitoring
Bug	AUTORADAR-2009	Fault injection can not be tested for VCO and IF stage monitoring. Workaround: Disable fault injection for VCO and IF stage monitoring
Bug	AUTORADAR-2013	There is a possibility of race condition in accessing sequencer extension RAM when tx gain phase monitoring is enabled.
Bug	AUTORADAR-1992 AUTORADAR-2028	Tx internal analog monitor failures are seen with higher temperatures
Bug	AUTORADAR-1994 AUTORADAR-2057 AUTORADAR-2046	Rx internal analog monitor failures are seen with higher temperatures
Bug	AUTORADAR-2061	There is a possibility of FRC lock step test to fail in latent fault API
Bug	AUTORADAR-2055	BSS periodic monitor failures seen at neg20C-neg30C Tj
Bug	AUTORADAR-2074	The gain calibration might fail at -40deg C
Bug	AUTORADAR-2087	calibration store and restore API can cause APLL control voltage monitor to fail.
Bug	AUTORADAR-2090	Calibration store and restore API returns INVALID DATA occasionally.
Bug	AUTORADAR-2089	TX gain phase monitor reports Phase mismatch failure even though the reported phases are within the threshold for some phase values
Bug	AUTORADAR-2093	Rx gain phase monitor reporting mode 1 and 2 fails if RF gain target is set to 34dB and 26dB.
Bug	AUTORADAR-2096	PA loop back option in advance frame config API loopback burst causes RF CPU fault.

Bug	AUTORADAR-2128	false alarm seen in RX power detector status bits in RX internal signal monitor
Bug	AUTORADAR-2026	The LPF cutoff monitor reports high deviations under normal operating conditions. Such deviations need not be construed as device failure indication. Workaround: Recommended to use 128% LPF cutoff monitor error in AWR MONITOR RX IFSTAGE CONF SB API (Effectively disable the LPF cutoff monitor by using 128% error threshold)
Bug	AUTORADAR-2002	DCC monitoring occasionally shows failure for Rampgen clock. Workaround: Disable rampgen clock DCC clock monitor in AWR MONITOR DUAL CLOCK COMP CONF SB API
Bug	MMWAVE RFANA-185	TI production test is unstable for IQ mismatch calibration failure.
Bug	AUTORADAR-2067	Tx gain phase mismatch monitor for TX3 reports higher than the expected values from simulation. Workaround: Do not use TX gain phase mismatch monitor.
Bug	AUTORADAR-2065	TX power accuracy and inter-channel balance may degrade at high backoff (8dB backoff or more). Workaround: Use 0dB back-off setting for best inter-TX matching performance
Bug	MMWAVESYS-159	1. The following monitors are susceptible to corruption by interference from other radar sensors. The monitors may result in false alarms under the influence of interference. 1. RX GAIN PHASE MONITOR (Can be mitigated through Host based Solution) 2. RX NOISE FIGURE MONITOR 3. TX GAIN PHASE MISMATCH MONITOR 4. TX0 BPM MONITOR, TX1 BPM MONITOR, TX2 BPM MONITOR 5. RX MIXER INPUT POWER MONITOR 2. The following boot-time calibrations are susceptible to corruption by interference. The calibrations may result in false configuration of the RF analog sections due to corruption by interference during the calibration measurements. 1. Enable RX gain calibration 2. Enable TX Phase calibration 3. Enable RX IQMM calibration
Bug	MMWAVESYS-158	RX noise figure monitor is performed with RX RF LNA disabled (to suppress external interference's influence) and reports numbers with high variations and inconsistent with full RX noise figure. Workaround: Do not use RX noise figure monitor.
Bug	AUTORADAR-2077	The Rampgen memory ECC self-test is failed once in a long tracking stress test, which is looping Latent fault API calls infinite time with all digital monitoring tests enabled. Workaround: Disable Rampgen ECC test in AWR MONITOR RF DIG LATENTFAULT CONF SB API (Boot time rampgen ECC test is always done).

3. 4. 2. 2. RadarSS firmware (patch) 6.0.7.0 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

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) 1.2.0.3 for xwr14xx, xwr16xx, xwr18xx

This section captures unsupported features/APIs in this release of RadarSS.

API	Feature	Description
FORCE VCO SEL feature in AWR PROFILE CONF SET SB	Force VCO select	The Force VCO select in profile configuration API is not validated at system level. It is recommended not to use the same.

LOOPBACK CFG feature in AWR ADVANCED FRAME CONF SB	Loopback feature in advance frame config API	The Loopback option in advance frame config API is not validated at system level. It is recommended not to use the same.
LDO SC MONITORING EN field in AWR MONITOR ANALOG ENABLES CONF SB	Short Circuit monitor	The Short circuit protection feature is not supported. Do not enable short circuit monitor LDO SC MONITORING EN field in AWR MONITOR ANALOG ENABLES CONF SB.
RAMPGEN 100M monitor feature in AWR MONITOR DUAL CLOCK COMP CONF SB	Rampgen 100MHz clock monitor	The rampgen 100MHz clock monitor is not supported. It is recommended not to use the same.
PCR self-test feature in AWR MONITOR RF DIG LATENTFAULT CONF SB	PCR self-test	The PCR self-test is not supported in latent fault configuration API. It is recommended not to use the same.
RAMPGEN ECC self-test feature in AWR MONITOR RF DIG LATENTFAULT CONF SB	Rampgen ECC self-test	The Rampgen ECC self-test is not supported in latent fault configuration API. It is recommended not to use the same.
LPF CUTOFF FREQ ERROR THRESH feature in AWR MONITOR RX IFSTAGE CONF SB	LPF cutoff freq monitor	The LPF cutoff frequency monitor is not supported in IF stage monitor. It is recommended not to use the same.
AWR DYN CHIRP CONF SET SB AWR DYN PERCHIRP PHASESHIFTER CONF SET SB AWR DYN CHIRP ENABLE SB	Dynamic chirp configurations	Dynamic chirp configuration APIs are not validated at system level. It is recommended not to use the same.
AWR RX GAIN TEMPLUT SET SB AWR TX GAIN TEMPLUT SET SB AWR RX GAIN TEMPLUT GET SB AWR TX GAIN TEMPLUT GET SB	Rx and Tx gain calibration override	The Rx and Tx gain calibration override APIs are not validated at system level. It is recommended not to use the same.
AWR PERCHIRPPHASESHIFT CONF SB	Per-chirp phase shifter	This API is not validated at system level. It is recommended not to use the same.
AWR PROG FILT COEFF RAM SET SB AWR PROG FILT CONF SET SB	Programmable filter (xWR1642/xWR1843 Only)	These APIs are not validated at system level. It is recommended not to use the same.
AWR INTER RX GAIN PHASE CONTROL SB	Inter-RX gain phase configuration	This API is not validated at system level. It is recommended not to use the same.
AWR LOOPBACK BURST CONF SET SB	Loopback burst configuration	This API is not validated at system level. It is recommended not to use the same.
AWR INTERCHIRP BLOCKCONTROLS SB	Inter-chirp power saving configurations	This API is not validated at system level. It is recommended not to use the same.
AWR RF DFE STATISTICS REPORT GET SB	DFE statistics report	This API is not validated at system level. It is recommended not to use the same.

AWR MONITOR RX SATURATION DETECTOR CONF SB AWR MONITOR SIG IMG MONITOR CONF SB	Saturation and Image band detectors	These APIs are not validated at system level. It is recommended not to use the same.
AWR RF GPADC CFG SET SB AWR MONITOR EXTERNAL ANALOG SIGNALS CONF SB	External signal monitoring using GPADC (xWR1642 /xWR1843 Only)	This API is not validated at system level. It is recommended not to use the same.
AWR MONITOR RX NOISE FIGURE CONF SB	RX noise figure monitor	RX noise figure monitor is performed with RX RF LNA disabled (to suppress external interference's influence) and reports numbers with high variations and inconsistent with full RX noise figure. It is recommended not to use the same.
AWR MONITOR TX GAIN PHASE MISMATCH CONF SB	TX gain phase monitor	Tx gain phase mismatch monitor is susceptible to corruption by interference from other radar sensors. The monitors may result in false alarms under the influence of interference. It is recommended not to use the same.
AWR MONITOR TX0 BPM CONF SB AWR MONITOR TX1 BPM CONF SB AWR MONITOR TX2 BPM CONF SB	TX BPM monitor	The TX BPM monitor is susceptible to corruption by interference from other radar sensors. The monitors may result in false alarms under the influence of interference. It is recommended not to use the same.
AWR MONITOR RX MIXER IN POWER REPORT AE SB	Rx mixer input power monitor	The RX mixer input power monitor is susceptible to corruption by interference from other radar sensors. The monitors may result in false alarms under the influence of interference, it is recommended to use only for debug.

3. 5. 2. 2. RadarSS firmware (patch) 6.0.7.0 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 14.04 & Ubuntu 16.04 64 bit machines.**

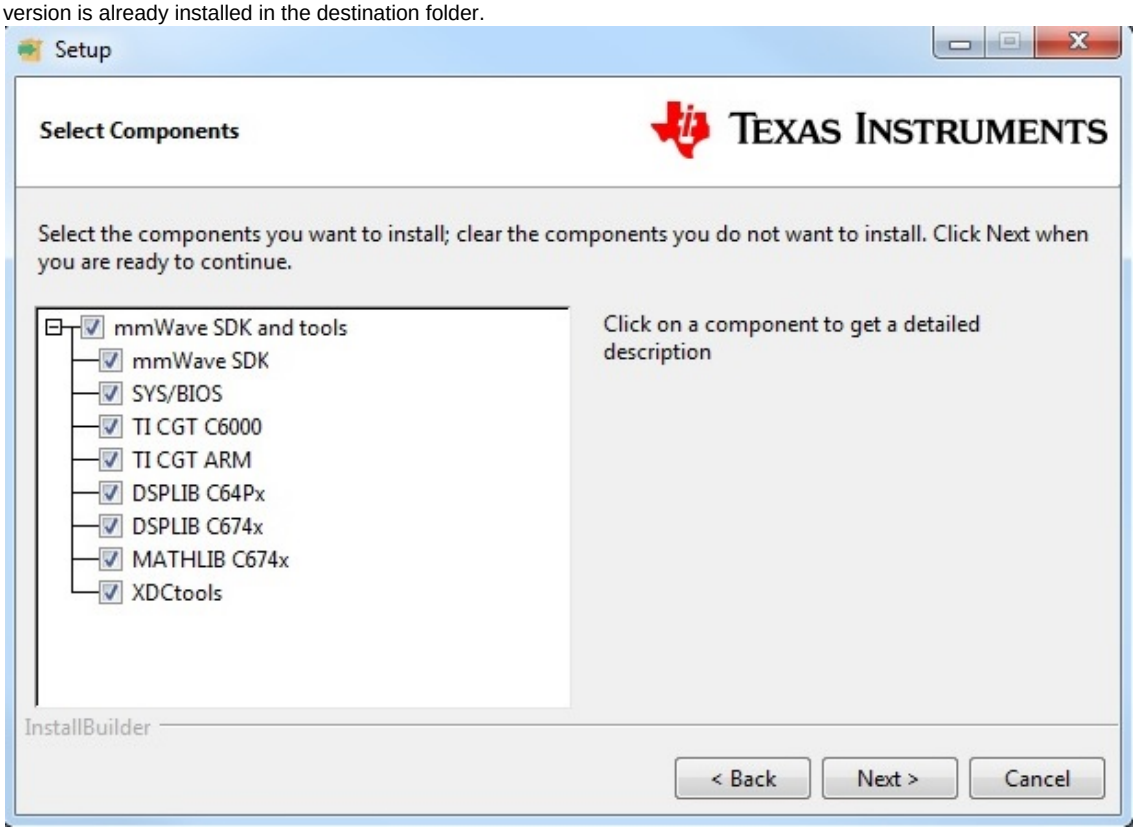
5. 1. Installation in GUI mode

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

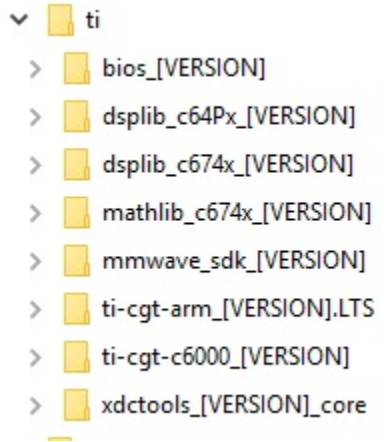
5. 2. Installation in unattended command line mode

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

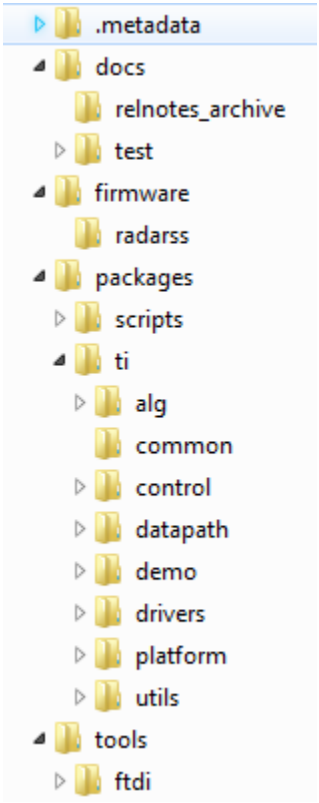
- In Windows environment
  - Run the installer using "mmwave_sdk_<version>-Windows-x86-Install.exe --prefix <installation folder> --mode unattended" command. This will install all the components in the installer.
    - Please note that even though the command may finish immediately it takes sometime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

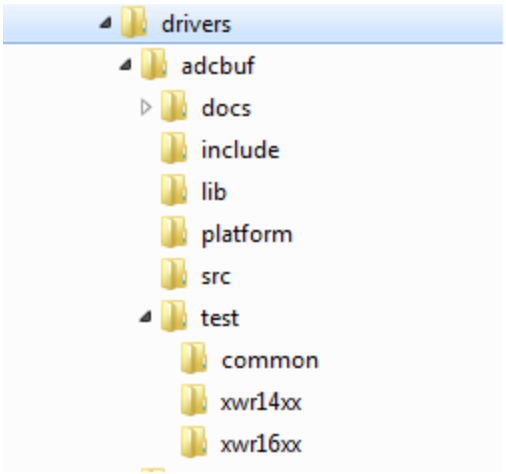


6. Package Contents

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**
Datapath EDMA	**X**	**X**	
Object Detection DPC[1]	**X**	**X**	**X**

[1] No pre-built library for Object Detection DPC

6. 4. Algorithm

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

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

6. 5. Usecases

Usecases can be found under mmwave_sdk_<ver>/packages/ti/drivers/test folder.

Component	Source	API Document (doxygen)	Unittest (source & prebuilt binary)
csi_stream (IWR14xx only)	X	X	X
mem_capture	X	X	X

6. 6. Demos

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

Component	Source & Prebuilt Binary	Demo document (doxygen)	Demo GUI
mmw[1]	X	X	X

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

6. 7. Misc folders

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

- common: Common header files needed across all components
- platform: platform specific files
- utility: Contains
  - ccs debug utility which is the MSS/DSSbinary that needs to be flashed when connecting/developing using CCS (details can be found in mmwave_sdk_user_guide)
  - cli which is the cli helper utility used by the demos
  - cycleprofiler which is the helper utility used for profiling the various components inside the SDK
  - hsiheader which is a helper utility that creates a header for the data to be shipped over LVDS lanes.
  - mathutil is used to perform some common operations such as log2, rounding, saturation based on the core they need to run on (R4F, C674x)
  - secondary boot loader (sbl)
  - testlogger which is the helper utility for driver unit tests

6. 8. Scripts

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

6. 9. Firmware

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

6. 10. Tools

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

- **Ftdi:** These Windows PC drivers are needed when interfacing to the board via FTDI port on MMWAVE-DEVPACK or MMWAVEICBOOST

6. 11. Docs

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

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

mmwave_sdk_<ver>/docs/relnotes_archive contains release notes from previous releases. Release notes contain migration information.

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

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

7. Related documentation/links

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

- SoC links:
  - Automotive mmWave Sensors
  - Industrial mmWave Sensors
- Evaluation Modules (EVM) links:
  - Automotive Evaluation modules (Booster Pack, DEVPACK)
  - Industrial Evaluation modules (Booster Pack, ISK)