

MMWAVE SDK Release Notes



Product Release 3.1.1

Release Date: Jan 18, 2019

Release Notes Version: 1.1

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 [Known Issues](#)
 - 3.5 [Limitations](#)
 - 4 [Test reports](#)
 - 5 [Installation instructions](#)
 - 6 [Package Contents](#)
 - 6.1 [Drivers](#)
 - 6.2 [Control](#)
 - 6.3 [Datapath](#)
 - 6.4 [Algorithm](#)
 - 6.5 [Usecases](#)
 - 6.6 [Demos](#)
 - 6.7 [Misc folders](#)
 - 6.8 [Scripts](#)
 - 6.9 [Firmware](#)
 - 6.10 [Tools](#)
 - 6.11 [Docs](#)
 - 7 [Related documentation/links](#)
-
-



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 | | 3.1.1 | 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 | | 6.0.6.0 | 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 | 8.1.3 | Included in mmwave sdk installer |
| XDC | 3.50.08.24 | Included in mmwave sdk installer |
| C64x+ DSPLIB | 3.4.0.0 | Included in mmwave sdk installer |
| C674x DSPLIB | 3.4.0.0 | Included in mmwave sdk installer |
| C674x MATHLIB (little-endian, elf /coff format) | 3.1.2.1 | Included in mmwave sdk installer |
| Mono JIT compiler | 4.2.1 | Only for Linux builds |
| mmWave Radar Device support package | 1.6.1 or later | Upgrade to the latest using CCS update process (see SDK user guide for more details) |
| TI Emulators package | 7.0.188.0 or later | Upgrade to the latest using CCS update process (see SDK user guide for more details) |
| MMWAVE-SECDEV | 2.0.0 | Needed for xWR16xx high secure (HS) devices only Can be requested from link |

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

The following tools are needed at runtime

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

2. 5. Licensing

Please refer to the [mmwave_sdk_software_manifest.html](#), which outlines the licensing status for mmwave_sdk package.

3. Release content

3. 1. New Features

- Updated RadarSS firmware and corresponding mmWaveLink driver update for IWR6843

3. 2. Migration section

This section describes the changes that are relevant for users migrating to the mmWave SDK 3.1.1 release from 3.1.0 release. See release notes archive in the SDK release package for migrating from other older releases.

| Summary | Component /s | Subcomponent | Behavior of impact |
|--------------------|--------------|--------------|---|
| Updated components | - | - | Updated to latest radarss firmware for IWR6843 device |

3. 3. Issues fixed

Since only RadarSS for IWR6843 device was updated in this release, see [mmwave_sdk_<ver>/firmware/radarss/mmwave_xwr6xxx_radarss_release_notes.pdf](#) for issues fixed in that component.

3. 4. Known Issues

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

| Issue Type | Key | Summary | Comments |
|------------|-----------------------------|---|---|
| Bug | MMWSDK-1654 | AOA DPU process should return Int32 instead of Uint32 | The process() function uses return value to communicate any errors encountered during AoA processing. These errors are encoded in negative space and hence the return value should be int32 instead of uint32. Currently the object detection DPC (the one that invokes AoA DPU) uses check of "!=0" to flag any errors returned by AoA DPU and hence this bug is not disastrous. Any other DPC or application should use this as a workaround as well. |
| Bug | MMWSDK-1542 | AoA DPU: RX phase calibration does not work when measurement is done with less than the possible max antenna size (#tx < 3, #rx < 4 in case of IWR6843) | Documented procedure in past releases always mentioned that all the available antennas on the device be turned on for measurement - so this is not creating any deviation from that. This is listed as known issue so that user are aware of the limitation. |
| Bug | MMWSDK-1363 | Range processing hwa DPU crashes when number of RX antenna is 4, and range fft size is 1024 | For 1 TX 4 RX and numRangeBins = 1024, the BdstIndex for EDMA copy will go beyond its limit of 32768. The calculation is follows: BytesPerChirp = numRangeBins * numRxAnt * sizeof(cmplx16ImRe_t) = 16KB. For 1 TX antenna, due to ping/pong scheme, the jump will be 2 * BytesPerChirp = 32KB. The same case is solved by manually setting destination address in rangeProc DSP based implementation. For rangeProcHWA, the manually setting of destination address is not doable. |
| Bug | MMWSDK-1497 | Intermittent failure in "monitoring results" for mmwavelink unit test for awr16xx | This issue is seen in noisy lab environment only. One out of many reports for noise figure has failure status. Observed noise figure from that report are logged at the end of the test run and can be used for debugging further, in case this is seen in other scenarios. |
| Bug | MMWSDK-1078 | Limitation in processing chain + LVDS instrumentation use case | See limitations section below |
| Task | MMWSDK-533 | GUI of mmw demo running slow from Firefox browser | <u>Workaround:</u> Please switch to Chrome browser. |
| Story | MMWSDK-319 | CAN driver: DMA mode is not supported | |
| Story | MMWSDK-252 | UART driver has not tested for Data Length 5 and 6 | |

| | | |
|-----|-----------------------------|---|
| Bug | MMWSDK-1157 | Rare failure seen in UART loopback driver unit test - HW limitation |
|-----|-----------------------------|---|

3.5. Limitations

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

| | |
|----|---|
| 1 | CAN driver: <ul style="list-style-type: none"> DMA and FIFO mode are not supported |
| 2 | CANFD driver: <ul style="list-style-type: none"> DMA and Timestamping are not supported |
| 3 | CBUFF/CSI2/LVDS: <ul style="list-style-type: none"> Driver does not support the following functionality: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> dual-Read/Quad read in configuration mode is not supported setting write protections bits is not supported |
| 10 | SPI (MIBSPI) Limitations: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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.

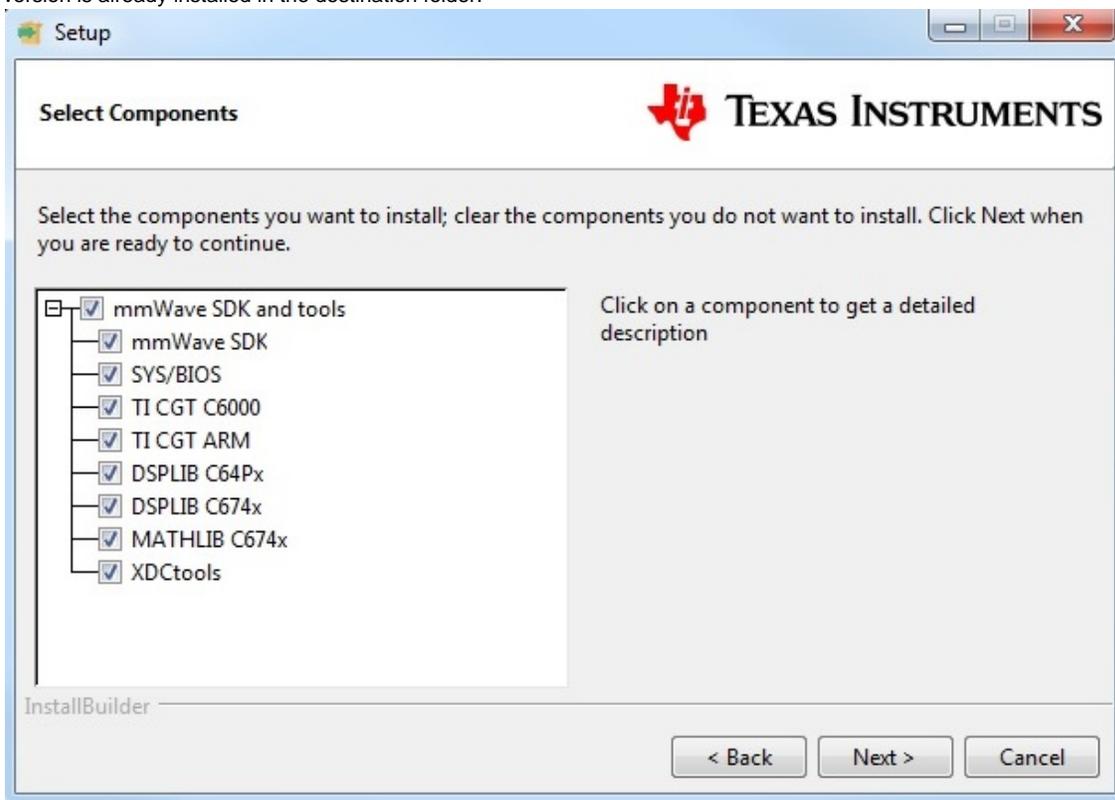
- **mmwave_sdk_<version>-Windows-x86-Install.exe:** Windows installer verified on Windows 7 and Windows 10 machines
- **mmwave_sdk_<version>-Linux-x86-Install.bin:** Linux installer verified on Ubuntu 14.04 & Ubuntu 16.04 64 bit machines.

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,
 - 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

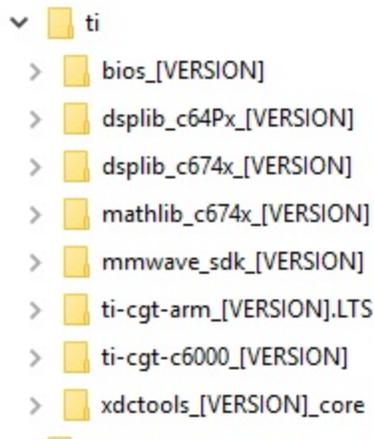
Installation steps:

- 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.

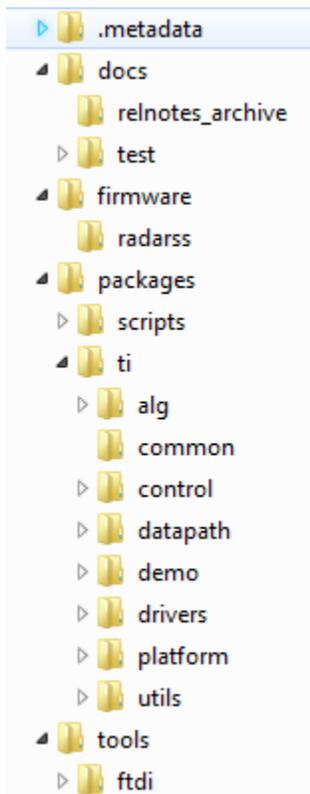


- Review installation decisions
- Ready to install
- Once installation starts all the selected components will be installed (if a component with the same version exists in the destination folder it will be overwritten)
- Installation complete

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.

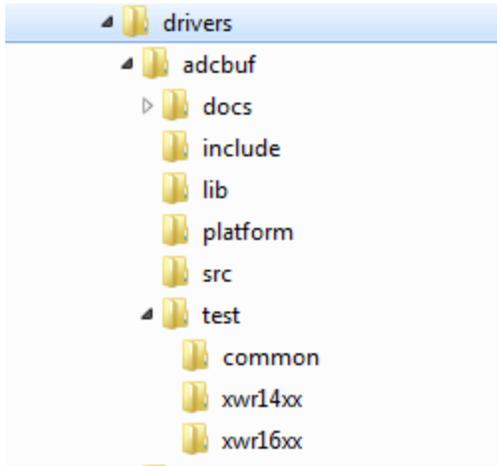


6. Package Contents

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

Datapath 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 | |
| AoA DPU | X | X | |
| Datapath EDMA | X | X | |
| Object Detection DPC | X | X | X |

6. 4. Algorithm

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

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

6. 5. Usecases

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

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

6. 6. Demos

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

| Component | Source & Prebuilt Binary | Demo document (doxygen) | Demo GUI |
|------------------|-----------------------------|----------------------------|----------|
| mmw ¹ | X | X | X |

¹ Demo is supported for only IWR68xx and xWR18xx devices 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)