Application Note

Migrating to xWR68xx and xWR18xx Millimeter Wave Sensors

Nitin Sakhuja and Chethan Kumar Y.B.

ABSTRACT

This application report provides guidance for porting mm-wave hardware and application software to the xWR68xx ES2.0 and the xWR18xx devices.

Table of Contents

1 Introduction ... 3
2 xWR1843 Hardware/Software Migration ... 4
2.1 Migrating From xWR1642 to xWR1843 ... 4
3 xWR6843AoP ES2.0 Migration ... 9
3.1 Hardware Changes From xWR6843AoP ES1.0 to xWR6843AoP ES2.0 ... 9
3.2 Software Migration From xWR6843AoP ES1.0 to xWR6843AoP ES2.0 ... 11
4 Helpful Resources ... 17
5 Code Snapshots ... 18
5.1 SDK 3.3 API Change for MMWave_open ... 18
5.2 SDK 3.3 API Change for ADCBuf_open ... 18
5.3 SDK 3.3 API Change for CANFD_init ... 19
5.4 SDK 3.3 68xx Secondary Bootloader Update ... 19
5.5 SDK 3.3 16xx vs 68xx: Calibration Frequency Update ... 20
5.6 SDK 3.3 16xx vs 68xx: SoC Definition Updates ... 20
5.7 SDK 3.3 16xx vs 18xx: SoC Definition Updates ... 21
5.8 SDK 3.4 xWR68xx Calibration Frequency Update ... 21
5.9 SDK 3.4 Object Detect HWA DPC Range FFT Scaling ... 22
5.10 SDK 3.4 Object Detect Range HWA DPC Radar Cube Format ... 22
5.11 xWR6843AoP ES1.0 Antenna Geometry ... 23
5.12 xWR6843AoP ES2.0 Antenna Geometry ... 23
5.13 xWR6843AoP ES2.0 Antenna Geometry Code Update ... 23
5.14 Antenna Geometry Structure Usage in mmw demo ... 24
5.15 xWR6843AoP ES2.0 RX Channel Phase Compensation ... 24
6 References ... 26
7 Revision History ... 26

List of Figures

Figure 2-1. xWR1642 Device Marking ... 4
Figure 2-2. xWR1843 Device Marking ... 4
Figure 2-3. xWR1642 Antenna Image ... 5
Figure 2-4. xWR1843 Antenna Image ... 5
Figure 3-1. Silicon Device Marking Difference Between xWR6843AoP ES1.0 and ES2.0 ... 9
Figure 3-2. xWR6843AoP ES1.0 Antenna Geometry and Resulting MIMO Virtual Antenna Array ... 14
Figure 3-3. xWR6843AoP ES2.0 Antenna Geometry and Resulting MIMO Virtual Antenna Array ... 14
Figure 3-4. AoA2dProc HTML Documentation ... 15
Figure 3-5. RX Channel Phase Compensation: CompRangeBiasAndRxChanPhase CLI Command ... 16
Figure 5-1. SDK 3.3 API Change for MMWave_open ... 18
Figure 5-2. SDK 3.3 API Change for ADCBuf_open ... 18
Figure 5-3. SDK 3.3 API Change for CANFD_init ... 19
Figure 5-4. SDK 3.3 68xx Secondary Bootloader Update ... 19
Figure 5-5. SDK 3.3 16xx vs 68xx: Calibration Frequency Update ... 20

Figure 5-6. SDK 3.3 16xx vs 68xx: SoC Definition Updates....20
Figure 5-7. SDK 3.3 16xx vs 18xx: SoC Definition Updates....21
Figure 5-8. SDK 3.4 xWR68xx Calibration Frequency Update....21
Figure 5-9. SDK 3.4 Object Detection DPC FFT Range Scaling Configuration....22
Figure 5-10. SDK 3.4 Object Detect Range HWA DPC FFT Radar Cube Format....22
Figure 5-11. xWR6843AoP ES1.0 Antenna Geometry....23
Figure 5-12. xWR6843AoP ES2.0 Antenna Geometry....23
Figure 5-13. SDK 3.2.0.6 Vs SDK 3.4: Antenna Geometry Update for xWR6843AoP ES2.0....24
Figure 5-14. Antenna Geometry Structure Usage in mmw demo....24
Figure 5-15. SDK 3.2.0.6 Vs SDK 3.4: RX Channel Phase Compensation....25

List of Tables

Table 1-1. Migration Reference....3
Table 2-1. Device Feature Comparison Table....5
Table 2-2. xWR1642 to xWR1843 Software Migration....6
Table 3-1. xWR6843AoP ES1.0 to xWR6843AoP ES2.0 Hardware Changes....9
Table 3-2. xWR6843AoP ES2.0 Software - Platform Updates....11

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The information presented here is applicable to any of the following scenarios:

- Have hardware/software currently deployed on xWR6843 ES1.0 and want to migrate it to xWR6843 ES2.0
- Have hardware/software currently deployed on xWR1642 and want to migrate it to xWR6843 ES2.0
- Have hardware/software currently deployed on xWR1642 and want to migrate it to xWR1843
- Have hardware/software currently deployed on xWR6843AOP ES1.0 and want to migrate to xWR6843AOP ES2.0

The information presented in this document covers:

- Comparison of the base and the new target device along-with a description of how those differences impact existing hardware and software.
- SDK version required for the new target device and updates needed in application build infrastructure (makefiles and/or CCS projects, linker command files, and so forth)
- Updates needed in application source code, for example, API updates, new structure parameters, and so forth.
- Example source code comparison snapshots are provided for easy reference.

For information specific to your current and target device, see the following sections.

Table 1-1. Migration Reference

Current Device	Target Device	Section
xWR6843 ES1.	xWR6843 ES 2.0	Migrating from xWR6843 ES1.0 to xWR6843 ES2.0 : Section 3.2
xWR1642	xWR6843 ES2.0	Migrating from xWR1642 to xWR6843 ES2.0 : Section 2.1
xWR1642	xWR1843	Migrating from xWR1642 to xWR1843 : Section 2
xWR6843AoP ES1.0	xWR6843AoP Es2.0	Migrating from xWR6843AoP ES1.0 to xWR6843AoP ES2.0 :Section 3

2 xWR1843 Hardware/Software Migration

This section provides migration guidance to port Hardware and software from the xWR1642 to the xWR1843 device. The information provided here is meant to cover the major changes for migrating to a particular MMWAVE-SDK release at the time of writing. For more information, see the *Migration* section in the MMWAVE-SDK Release Notes.

2.1 Migrating From xWR1642 to xWR1843

2.1.1 Device Comparison

Table 2-1 lists the key features of the xWR1642 and the xWR1843 devices that need to be considered from Hardware and software migration perspective. For more information, see the device-specific data sheets and the *Industrial mmWave Radar Family Technical Reference Manual* in Section 6.

Figure 2-1 and Figure 2-2 show the device symbolization change from xWR1642 to xWR1843 on device part marking.

The left side device marking shows the xWR1642 silicon and the right side device marking shows the xWR1843 silicon. For more details on the device marking, see the device-specific Errata.

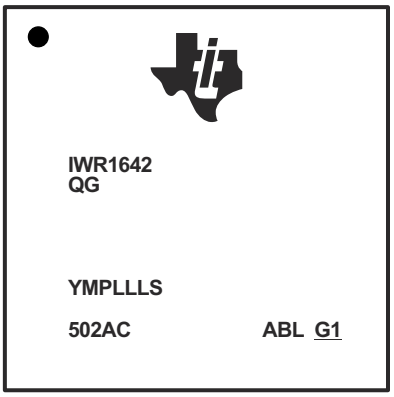

Figure 2-1. xWR1642 Device Marking

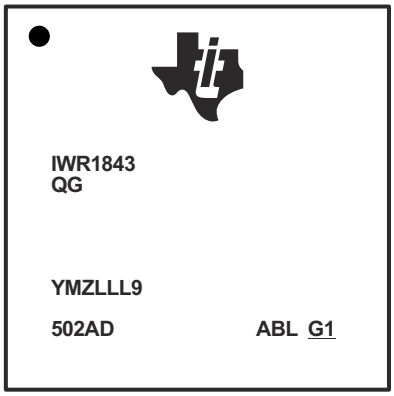

Figure 2-2. xWR1843 Device Marking

- *IWR1642 Device Errata*
- *AWR1642 Device Errata*
- *IWR1843 Device Errata*
- *AWR1843 Device Errata*

Table 2-1. Device Feature Comparison Table

No	Device Feature Differences	xWR1642	xWR1843	Hardware and Software Impact
1	Number of Transmit Channels	2	3 (1)	3rd Transmitter Antenna need to be designed. Update TX bitmap in chirpCfg
2	Maximum Sampling Rate	6.25 MHz complex	12.5 MHz complex	Higher IF bandwidth and Sampling rates are available on xWR1843
3	Max I/F (Intermediate Frequency)	5 MHz	10 MHz	
4	On-chip memory	1.5MB	2.0MB	Software can leverage the additional memory if needed.
5	Radar Accelerator	Not Applicable	Hardware accelerator for FFT, filtering, and CFAR processing	xWR1843 has flexibility of data processing on Hardware accelerator or DSP
6	Tx beam forming	No support	Supported	xWR1843 has phase shifters which supports the steerable beams. Note: Antennas need to be designed to support TX beam forming operation
7	MMWAVE-SDK support	SDK 2.1 (LTS) and above	SDK 3.3.0 and above	General software porting required compiling for xWR1843. For more information, see the Section 2.1.4.

(1) Three Tx Simultaneous operation is supported only with 1-V LDO bypass and PA LDO disable mode. In this mode, the 1-V supply needs to be fed on the VOUT PA pin.

2.1.2 Hardware Migration Notes

2.1.2.1 Antenna Addition

From xWR1642 to xWR1843, the third antenna needs to be introduced. For more information, see the design file package that provides the antenna details. Detailed field of view and radiations can be found in the user's guides listed below.

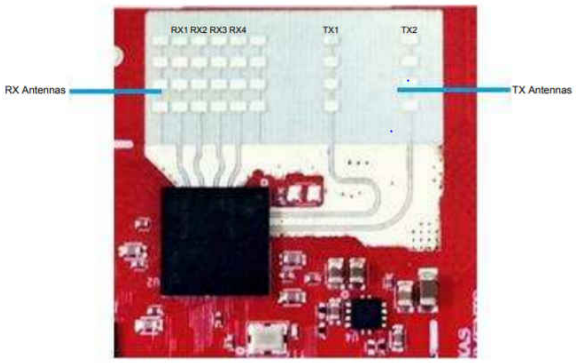

Figure 2-3. xWR1642 Antenna Image

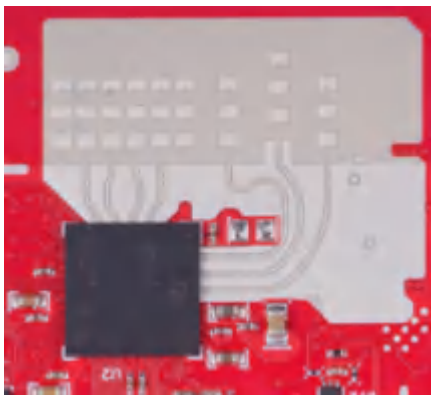

Figure 2-4. xWR1843 Antenna Image

- xWR1642BOOST Layout and Design Files
- *xWR1642 EVM (xWR1642BOOST) Single-Chip mmWave Sensing Solution User's Guide*
- xWR1843BOOST Hardware Files
- *xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution User's Guide*

2.1.3 Hardware Design Checklist

xWR1642 has the hardware design (schematic, Layout, bring-up/wakeup) checklist is available at http://www.ti.com/lit/zip/swrr151 and for the xWR1843 hardware design (schematic, Layout, Bring up/wakeup) checklist is available at http://www.ti.com/lit/zip/spracl2.

2.1.4 Software Migration Notes

Table 2-2 lists the changes required to port existing xWR1642 application code to xWR1843.

Note

The scope of the migration notes provided in this section is limited to migrating to MMWAVE-SDK 3.3.

When migrating existing xWR1843 applications to SDK releases beyond MMWAVE-SDK 3.3, you should follow the incremental migration instructions provided in the corresponding SDK release notes.

Table 2-2. xWR1642 to xWR1843 Software Migration

No	Summary	Components Impacted	Required Changes
1	MMWAVE-SDK 3.2.1 or above required for xWR1843 **NOTE:** It is recommended to use SDK 3.3.0 or above to include the latest API updates.	Makefile **OR** CCS projects	Application code must be re-compiled with MMWAVE-SDK 3.3.0 or above to run on xWR1843 **Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.3 environment setup script: C:\ti\mmwave_sdk_03_03_xx_xx\packages\scripts\windows\setenv.bat **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, you need to update the products property in DSS and MSS projectspecs as shown below. <property name="products" value="com.ti.rtsc.SYSBIOS:6.73.01.01;com.ti.MMWAVE_SDK:3.3.0.03;"/> **Example:** For reference CCS projects for xWR1843, see the 18xx – mmWave SDK Demo available in: MMWAVE Industrial Toolbox.
2	Change device type	Makefile **OR** CCS projects	**Makefile:** For SDK makefile based build, set MMWAVE_SDK_DEVICE=iwr18xx/awr18xx in setenv.bat. C:\ti\mmwave_sdk_03_03_xx_xx\packages\scripts\windows\setenv.bat **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, change the define SOC_XWR16XX to SOC_XWR18XX in DSS and MSS projectspecs. **Example:** For reference CCS projects for xWR1843, see the 18xx – mmWave SDK Demo available in: MMWAVE Industrial Toolbox.
3	Update RadarSS firmware file path	Makefile **OR** CCS projects (mss)	Need to use xWR18xx_radarss_rprc.bin in the metaimage generation step. **Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.3 environment setup script based on the MMWAVE_SDK_DEVICE variable. **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, replace xwr16xx_radarss_rprc.bin with xWR18xx_radarss_rprc.bin in the metaimage generation steps (postbuild steps) **Example:** For reference CCS projects for xWR1843, see the 18xx – mmWave SDK Demo available in: MMWAVE Industrial Toolbox.

Table 2-2. xWR1642 to xWR1843 Software Migration (continued)

No	Summary	Components Impacted	Required Changes
4	Use xWR18xx platform linker command file	Makefile **OR** CCS projects	**Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.3 environment setup script based on the MMWAVE_SDK_DEVICE variable. **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, update the include paths for r4f_linker.cmd and c674x_linker.cmd to: COM_TI_MMWAVE_SDK_INSTALL_DIR/packages/ti/platform/xwr18xx/r4f_linker.cmd and COM_TI_MMWAVE_SDK_INSTALL_DIR/packages/ti/platform/xwr18xx/c674x_linker.cmd, respectively. **Example:** For reference CCS projects for xWR1843, see the 18xx – mmWave SDK Demo available in: MMWAVE Industrial Toolbox.
5	Include xWR18xx driver and CLI libs	Makefile **OR** CCS projects	**Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.3 environment setup script based on the MMWAVE_SDK_DEVICE variable. **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, update the linker include paths to select the *_xwr18xx.aer4f and *_xwr18xx.xe674 lib versions, for example: -llibsoc_xwr18xx.ae674, -llibsoc_xwr18xx.xe674, -llibcli_xwr18xx.aer4f
6	Update sensor front-end configuration parameters	CLI config file (.cfg) and/or source code	Update TX channel bitmap in chirpCfg CLI command and/or API to account for the 3rd TX. **Example:** For more information, see the sample config files in C:\ti\mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr18xx\mmw\profiles.
7	Replace 16xx SOC definitions with 18xx equivalents.	MSS/DSS source code	Replace **SOC_XWR16XX_*** definitions/macros in source code with corresponding **SOC_XWR18XX_*** definitions. **For instance:** Replace SOC_XWR16XX_MSS_ADCBUF_BASE_ADDRESS with SOC_XWR18XX_MSS_ADCBUF_BASE_ADDRESS, Similarly, in Pinmux configuration code: Replace SOC_XWR16XX_PINN5_PADBE with SOC_XWR18XX_PINN5_PADBE and so forth. The image below shows reference code difference between the SDK 16xx and 18xx mmw demos **File:**mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr18xx\mmw\mss\mss_main.c **Code Snapshot: see** Section 5.7.
8	API update for MMWave_open SDK 3.3 requires new parameter to be passed to MMWave_open	MSS/DSS start-up code	**MMWave_open:** Application must set the value of calibMonTimeUnit parameter before calling MMWave_open as shown below. The image below shows reference code updates in the SDK 68xx mmw demo (same applies to 18xx mmw demo) **File:**mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c **Code Snapshot: see** Section 5.1.

Table 2-2. xWR1642 to xWR1843 Software Migration (continued)

No	Summary	Components Impacted	Required Changes
9	API update for ADCBuf_open SDK 3.3 requires new parameter to be passed to ADCBuf_open	MSS/DSS start-up code	**ADCBUF_open:** Application must set the value of socHandle in the ADCBufparams structure before calling ADCBUF_open as shown below. The image below shows reference code updates in the SDK 68xx mmw demo (same applies to 18xx mmw demo). **File:**mmwave_sdk_03_03_xx_xx\packages\ti\demo\utils\mmwdemo_adcconfig.c **Code Snapshot: see** Section 5.2.
10	API update for CANFD_init SDK 3.3 requires new parameter to be passed to CANFD_init	Drivers	**CANDF_init:** Applications using CANFD driver must pass instance ID to the CANFD_init API as shown below. Only a value of 0 is supported at this time. The image below shows reference code updates in the SDK CANFD driver test (same for 18xx). **File:**mmwave_sdk_03_03_xx_xx\packages\ti\drivers\canfd\test\xwr618xx\main.c **Code Snapshot: see** Section 5.3.
11	General note on CLI configuration file	Sensor Configuration	For applications that re-use the mmWave demo framework, ensure that the configuration commands (profileCfg, chirpCfg, frameCfg, and so forth) follow the format provided in the sample configuration files provided in the mmw demo directory: C:\ti\mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr18xx\mmw\profiles. For more information, see the *Configuration File Format* section of the *mmwwave SDK User's Guide*. See Section 6.

3 xWR6843AoP ES2.0 Migration

This section provides migration guidance to port Hardware and software from the xWR6843AoP ES1.0 to the xWR6843AoP Es2.0 device. The information provided here is meant to cover the major changes for migrating to a particular MMWAVE-SDK release at the time of writing. For more information, see the *Migration* section in the MMWAVE-SDK Release Notes.

3.1 Hardware Changes From xWR6843AoP ES1.0 to xWR6843AoP ES2.0

The changes described in this section are relevant when migrating xWR6843AoP ES1.0 hardware to xWR6843AoP ES2.0. Figure 3-1 shows the device symbolization change from ES1.0 to ES2.0 on device part marking.

Left side device marking shows ES1.0 silicon and right side device marking shows ES2.0 silicon. For more details on the device marking, see the *xWR6843 Device Errata, Silicon Revisions 1. and 2.0*.

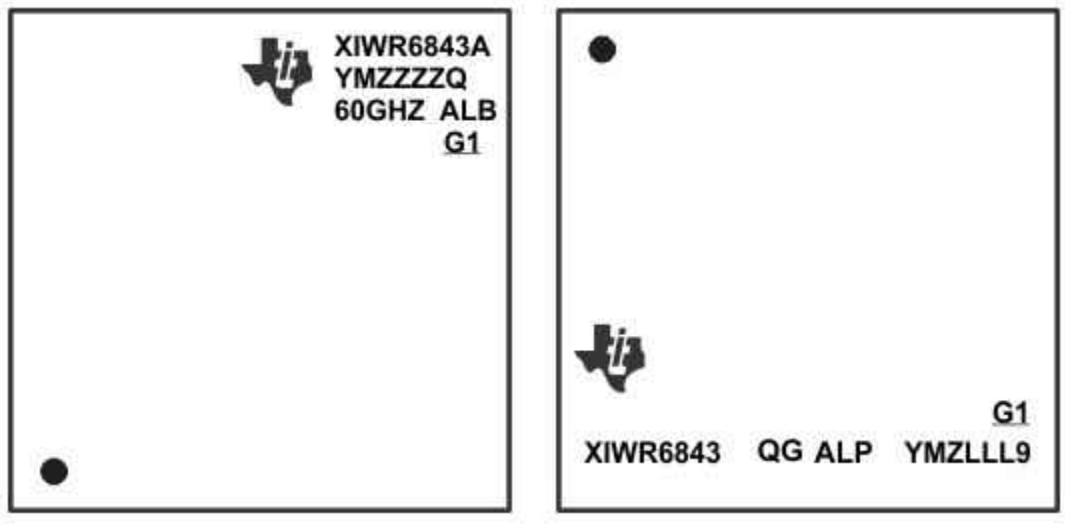

Figure 3-1. Silicon Device Marking Difference Between xWR6843AoP ES1.0 and ES2.0

Table 3-1. xWR6843AoP ES1.0 to xWR6843AoP ES2.0 Hardware Changes

No	Summary	xWR6843AoP ES1.0	xWR6843AoP ES2.0
1	QSPI interface speed has been improved. This enables faster boot loading, note that supported flashes are listed in the *Flash Variants Supported by the mmWave Sensor*.	Max 40 MHz	Max 80 MHz
2	Boot loader enhancement has been made. This allows faster boot and stability across devices	Boot loader code used to do the APLL calibration	Closed loop APLL calibration will be done by BSS
3	Tx beam scanning is introduced	No support	Supported
4	Memory compression (Depending upon the compression ratio of the RADAR data cube larger memory would be available for code and remaining data)	No support	Supported
5	Calibration is supported (This improves the performance and stability of the device across temperature)	No Calibration	Calibration supported
6	Clock gating at power-up and IP clock gating based on use-case, this should improve the power saving	No clock gating	Clock gated on unused peripherals. Device low level drivers un-gates the clock depending upon the peripheral used
7	RF Improvements –RX NF (Improved range and accuracy)	Baseline	Improved (Please refer to the data sheet for exact number)
8	RF Improvements –CLK PN (Improved accuracy)	Baseline	Improved (Please refer to the data sheet for exact number)

Table 3-1. xWR6843AoP ES1.0 to xWR6843AoP ES2.0 Hardware Changes (continued)

No	Summary	xWR6843AoP ES1.0	xWR6843AoP ES2.0
9	Package change	Baseline	Improved package (Please refer to the data sheet for detailed package information)
10	Changes in Antenna virtual Array	Baseline	Improvement in package routing caused changes in the antenna elements, hence there is change in virtual antenna array between ES1 and ES2.0. See Table 3-2

3.2 Software Migration From xWR6843AoP ES1.0 to xWR6843AoP ES2.0

The changes described in this section are relevant for migrating the xWR6843AoP ES1.0 software based on the SDK 3.2.0.6 to xWR6843AoP ES2.0 and SDK 3.4.

Besides the addition of the Antenna on Package and a different antenna pattern, xWR6843AoP ES2.0 re-uses the same silicon. Hence software migration from xWR6843AoP ES1.0 to xWR6843AoP ES2.0 broadly includes the following steps in order:

1. Initial migration of software to xWR6843ES2.0 (from MMWAVE-SDK 3.2.0.6 to MMWAVE-SDK 3.4). (Referred to below as **Platform Software Updates**)
2. Angle of Arrival Processing updates for the updated antenna pattern on xWR6843AoP ES2.0. (Referred to below as **AoA Software Updates**)

Note

MMWAVE-SDK 3.4.0 is the first baseline SDK release for xWR6843AoP ES2.0 device and the scope of migration notes provided in this section is limited to migrating to MMWAVE-SDK 3.4.0

When migrating existing xWR6843 AoP ES2.0 applications to SDK releases beyond MMWAVE SDK 3.4, you should follow the incremental migration instructions provided in the corresponding SDK release notes.

3.2.1 xWR6843AoP ES2.0 - Platform Software Updates

Table 3-2. xWR6843AoP ES2.0 Software - Platform Updates

No	Summary	Components Impacted	Required Changes
1	MMWAVE-SDK 3.4.0 or above required for xWR6843AoP ES2.0	Makefile **OR** CCS projects	Application code must be re-compiled with MMWAVE-SDK 3.4.0 or above to run on xWR6843AoP ES2.0 as prior SDK versions are not compatible with ES2.0. Conversely, SDK 3.4.0 is not compatible with xWR6843AoP ES1.0 devices. **Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.4 environment setup script: C:\ti\mmwave_sdk_03_04_xx_xx\packages\scripts\windows\setenv.bat **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, you need to update the products property in DSS and MSS projectspecs as shown below. <property name="products" value="com.ti.rtsc.SYSBIOS:6.73.01.01;com.ti.MMWAVE_SDK:3.4.0.03;"/> **Example:** For reference CCS projects for xWR6843AoP ES2.0, see the 68xx AoP – mmWave SDK Demo available in MMWAVE Industrial Toolbox.
2	Change the value of SHMEM_ALLOC parameter in MetaImage (flashable) binary generation step.	Makefile **OR** CCS projects (mss).	The value of SHMEM_ALLOC parameter should be set to 0x00000006 for ES2.0 (it was 0x02000006 for ES1.0 device). **Makefile: No change required if you are using SDK makefiles** build, as this is automatically handled in the SDK 3.4 device specific makefiles. **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, update the postBuildStep in MSS projectspec to replace the value 0x02000006 with 0x00000006. **Example:** For reference CCS projects for xWR6843AoP ES2.0 , see the 68xx AoP – mmWave SDK Demo available in MMWAVE Industrial Toolbox.

Table 3-2. xWR6843AoP ES2.0 Software - Platform Updates (continued)

No	Summary	Components Impacted	Required Changes
3	Update RadarSS firmware file name	Makefile **OR** CCS projects (mss)	The RadarSS binary for xwr6xxx devices is now called xwr6xxx_radarss_rprc.bin instead of iwr6xxx_radarss_rprc.bin. **Makefile: No change required if you are using SDK makefiles**, as this is automatically handled in the SDK 3.4 environment setup script based on the MMWAVE_SDK_DEVICE variable. **OR** **CCS Projectspec:** If the application is compiled using CCS projectspecs, replace iwr6xxx_radarss_rprc.bin with xwr6xxx_radarss_rprc.bin in the metaimage generation steps (postbuild steps) **Example:** For reference CCS projects for xWR6843AoP ES2.0 , see the 68xx AoP – mmWave SDK Demo available in MMWAVE Industrial Toolbox.
4	API update for MMWave_open SDK 3.3 and above requires a new parameter to be passed to MMWave_open	MSS/DSS start-up code	**MMWave_open**: Application must set the value of calibMonTimeUnit parameter before calling MMWave_open as shown below. The image below shows reference code updates in the SDK 68xx mmw demo **File:** mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c **Code Snapshot: see** Section 5.1
5	API update for ADCBuf_open SDK 3.3 and above requires a new parameter to be passed to ADCBuf_open	MSS/DSS start-up code	**ADCBUF_open**: Application must set the value of socHandle in the ADCBufparams structure before calling ADCBUF_open as shown below. The image below shows reference code updates in the SDK 68xx mmw demo. **File:** mmwave_sdk_03_04_xx_xx\packages\ti\demo\utils\mmwdemo_adcconfig.c **Code Snapshot: see** Section 5.2
6	API update for CANFD_init SDK 3.3 and above requires new parameter to be passed to CANFD_init	Drivers	**CANDF_init**: Applications using CANFD driver must pass instance ID to the CANFD_init API as shown below. Only a value of 0 is supported at this time. The image below shows reference code updates in the SDK CANFD driver test. **File:** mmwave_sdk_03_04_xx_xx\packages\ti\drivers\canfd\test\xwr68xx\main.c **Code Snapshot: see** Section 5.3
7	SDK 3.3 and above removes support for Bus error interrupt from the DMA driver for xWR6843 ES2 as that interrupt is not hooked up to the device.	Drivers	Application would get an error code back from the xwr68xx driver if DMA_enable Interrupt API is called for DMA_IntType_BER. You can either remove the call to the above API or ignore the error; however you should review the DMA usage to make sure there is no invalid memory access via MSS DMA engine.
8	General note on CLI configuration file	Sensor Configuration	For applications that re-use the mmWave demo/CLI framework, ensure that the configuration commands (for example, profileCfg, chirpCfg, frameCfg, and so forth) follow the format provided in sample configuration files provided in the mmw demo directory: C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr64xx\mmw\profiles. for more details, see the *Configuration File Format* section in the mmwwave SDK User's Guide. Section 6

Table 3-2. xWR6843AoP ES2.0 Software - Platform Updates (continued)

No	Summary	Components Impacted	Required Changes
9	BSS clock un-gate required in Secondary bootloader	Secondary Bootloader	**Note: This update is not related to the main application**. It is needed only if you are using a custom secondary bootloader in your system. The Secondary Bootloader must ungate BSS clock using SOC gate/ungate API before downloading image to RadarSS/BSS memory as shown below. The image below shows reference code updates in the SDK secondary bootloader example. **File:** C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\utils\sbl\platform\sbl_xwr68xx.c **Code Snapshot: see** Section 5.4
10	SDK 3.4 mmWave layer enables all valid init time and runtime calibrations for xwr6xxx devices	MSS/DSS start-up code	Application should pass valid values for **freqLimitLow** and **freqLimitHigh** in **mmWave_Open** API and can now enable periodic calibrations in **mmWave_Start** API The image below shows reference code updates in the SDK 68xx mmw demo. **File:** mmwave_sdk_03_04_00_03\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c **Code Snapshot: see** Section 5.8
11	Object detection DPC accepts antenna geometry to enable wider configurations of Tx/Rx antennas	DPCconfiguration	This field is mandatory only for HWA-based Object detection DPC when compiled to use the new AoA 2D algorithm (in the xwr64xx AoP mmw demo). For DSP-based DPC and for HWA-based DPC that uses standard AoA DPU, this field is unused. The image below shows the reference code in the SDK 64xx mmw demo. File: mmwave_sdk_03_04_00_03\packages\ti\demo\xwr64xx\mmw\main.c **Code Snapshot: see** Section 5.14
12	Object Detection HWA DPC now accepts Range FFT Scaling Parameters	DPC configuration	Range HWA-based DPU and Object detection HWA-based DPCs now allow you to set the scaling values for butterfly stages and converting from internal 24-bit to 16- bit output The image below shows the reference code in the SDK 64xx mmw demo. File: mmwave_sdk_03_04_00_03\packages\ti\demo\xwr64xx\mmw\main.c **Code Snapshot: see** Section 5.9
13	Objectdetection Range HWA DPC now allows user to specify the radar cube format	DPC Configuration	ObjDetRangeHWA DPC allows user to specify the radar cube format to allow flexibility in integrating various DSP based algorithms/ processing chains Note: mmW demos support only DPIF_RADARCUBE_FORMAT_ The image below shows the reference code in the SDK 68xx mmw demo. File: mmwave_sdk_03_04_00_03\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c **Code Snapshot: see** Section 5.10
14	Updates related to saving/restoring device calibration parameters (Phase shift calibration parameters)	For more details on this and other calibration related updates, see the MMWAVE-SDK 3.4.0 release notes in the Migration Notes.	

3.2.2 xWR6843AoP ES2.0 - AoA Software Updates

Figure 3-2 and Figure 3-3 compare the antenna geometries of xWR6843AOP ES1.0 and xWR6843AOP ES2.0.

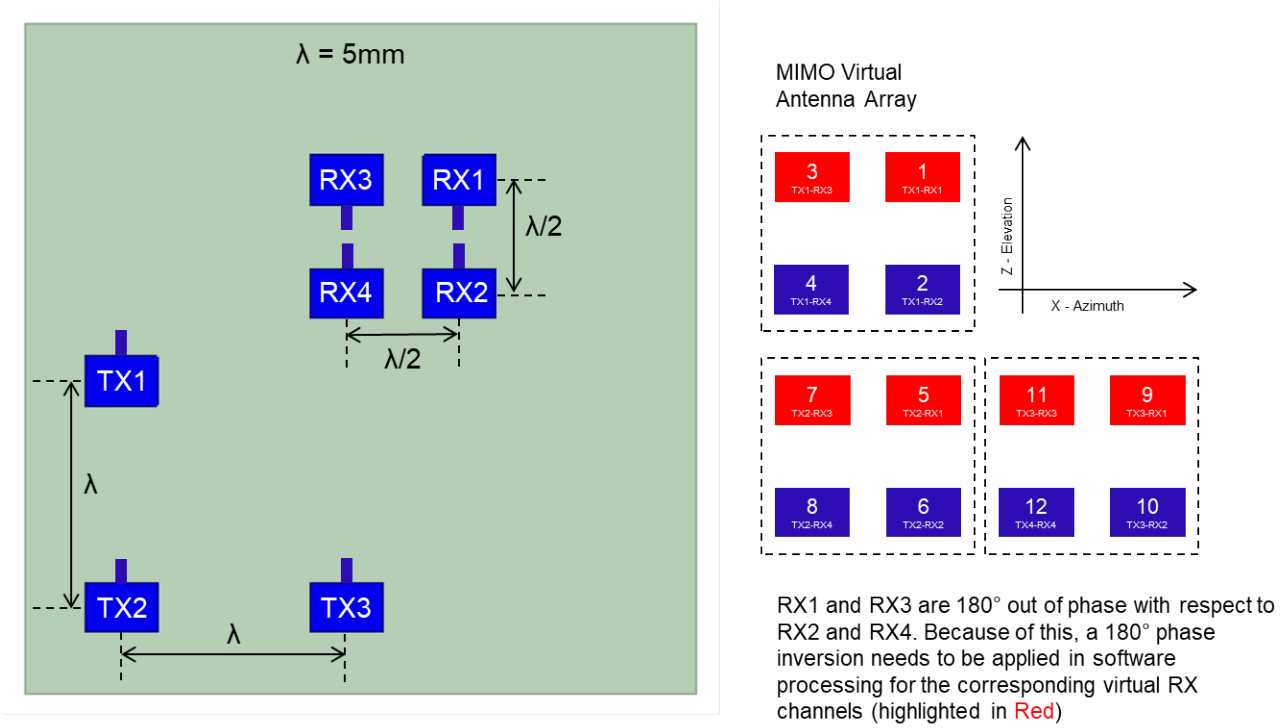

Figure 3-2. xWR6843AoP ES1.0 Antenna Geometry and Resulting MIMO Virtual Antenna Array

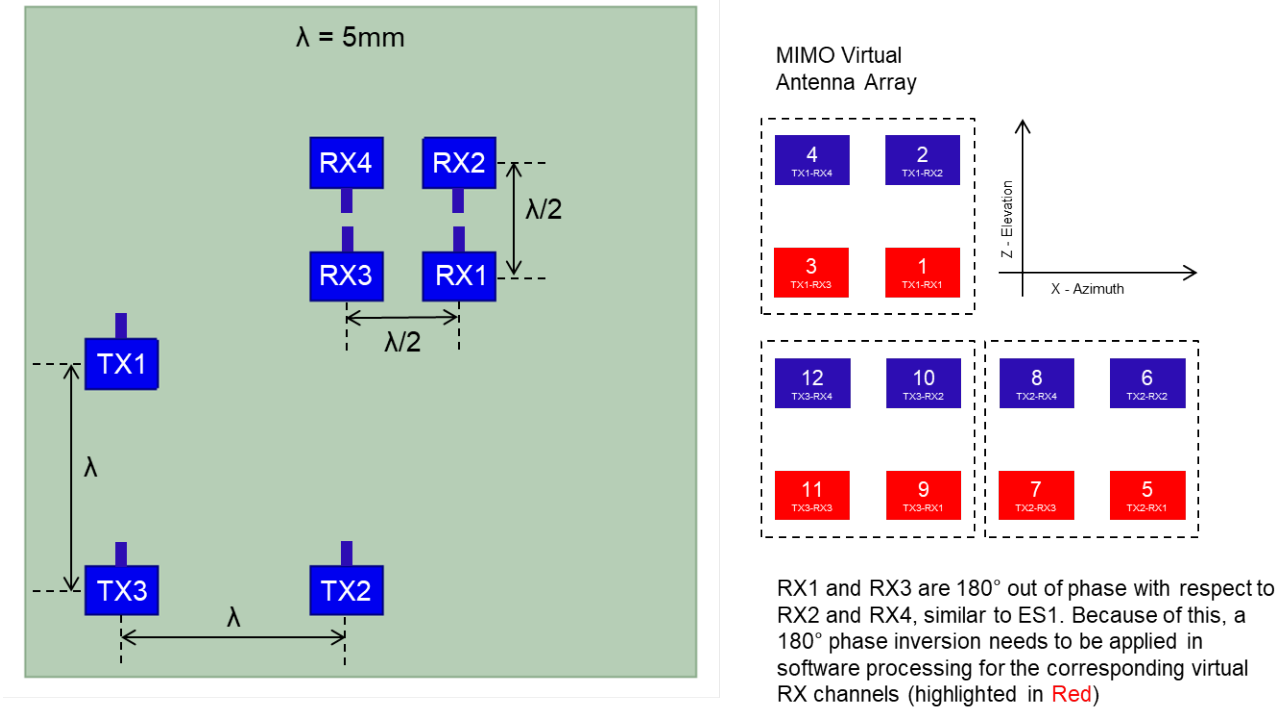

Figure 3-3. xWR6843AoP ES2.0 Antenna Geometry and Resulting MIMO Virtual Antenna Array

The key antenna updates in xWR6843AOP ES2, as shown above are:

- **RX Antennas:** RX1 and RX2 are swapped on xWR6843AOP ES2. Similarly RX3 and RX4 are swapped
- **TX Antennas:** TX2 and TX3 are swapped on xWR6843AOP ES2.
- **Line Feed:** The RX line feeds on xWR6843AOP ES2 are same as on ES1 i.e. RX1 and RX2 are fed from opposite ends, which results in a 180° phase difference between RX1 and RX2. Similarly, RX3 and RX4 are

out of phase by 180°. To compensate for the opposite line feeds, a 180° phase inversion needs to be applied in software processing for the corresponding virtual channels as shown in Figure 3-3.

MMWAVE-SDK 3.2.0.6 and MMWAVE-SDK 3.4 include the AoA2dProc DPU which performs Angle of Arrival processing for the xWR6843 AoP antenna array using the Hardware Accelerator. The AoA2dProc DPU (Datapath Processing Unit) is used in the xWR64xx AoP mmw demo for angle of arrival processing.

To understand the AoA updates needed for xWR6843AOP ES2, it is recommended to understand the antenna geometry concept defined in AoA2dProc DPU.

1. Navigate to C:\ti\mmwave_sdk_03_04_xx_xx\docs and and open the file mmwave_sdk_module_documentation.html in a browser.
2. Click on the AoA using 2D FFT method link as highlighted in the picture below:

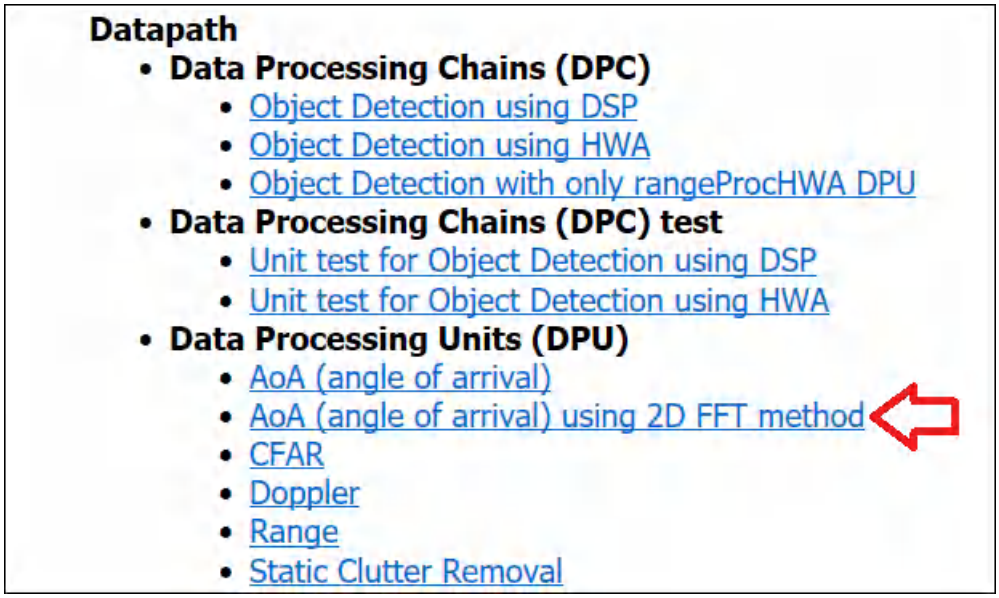

Figure 3-4. AoA2dProc HTML Documentation

3. Scroll down to the section named Antenna Geometry Definition, which explains how the generic antenna geometry structure is defined and used by the HWA AoA2dProc DPU code. The antenna geometry for a specific antenna (for example, xWR6843AoP ES2.0) is defined in the corresponding C structure in **mmwave_sdk_03_04_xx_xx\packages\ti\board\antenna_geometry.c**.

The image below shows the antenna geometry structure update for xWR6843AoP ES2.0 as compared to xWR6843AoP ES1.0 in MMWAVE-SDK 3.2.0.6.
Code Snapshot: see Section 5.13

The antenna geometry structure is passed to the Object Detection DPC during initialization in **mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr64xx\mmw\main.c**
Code Snapshot: see Section 5.14

RX Channel Phase Compensation: To compensate for the opposite line feeds as shown in Section 5.12, a 180° phase inversion is applied to the corresponding RX channels (including virtual channels) using the compRangeBiasAndRxChanPhase CLI command available in the mmw demo.

Figure 3-5, from the MMWAVE-SDK user's guide, explains the structure of this command.

compRangeBiasAndRxChanPhase	Command for datapath to compensate for bias in the range estimation and receive channel gain and phase imperfections. Refer to the procedure mentioned here The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running. This is a mandatory command.		
		<rangeBias> Compensation for range estimation bias in meters	supported
		<Re(0,0)> <Im(0,0)> <Re(0,1)> <Im(0,1)> ... <Re(0,R-1)> <Im(0,R-1)> <Re(1,0)> <Im(1,0)> ... <Re(T-1,R-1)> <Im(T-1,R-1)> Set of Complex value representing compensation for virtual Rx channel phase bias in Q15 format. Pairs of I and Q should be provided for all Tx and Rx antennas in the device	For xwr1843, xwr6843 and xwr6443 demos: 12 pairs of values should be provided here since the device has 4 Rx and 3 Tx (total of 12 virtual antennas). Note the sign reversal required for phase compensation coefficients in xwr6443 demo running on IWR6843AOP device. For xwr1642 demo: 8 pairs of values should be provided here since the device has 4 Rx and 2 Tx (total of 8 virtual antennas)

Figure 3-5. RX Channel Phase Compensation: CompRangeBiasAndRxChanPhase CLI Command

To understand the CompRangeBiasAndRxChanPhase values configured in the example AoP profile configuration provided in MMWAVE-SDK, see Section 5.15.

4 Helpful Resources

The following resources provide example source code, makefile and CCS projects for the xWR6843 ES2.0 and the xWR1843 devices.

Resource Name	File-System Path / Web URL	Content Reference
MMWAVE-SDK 3.3 mmw demo	68xx - C:\ti\mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr68xx\mmw 18xx - C:\ti\mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr18xx\mmw	Source code, Makefiles, Configuration files (.cfg)
MMWAVE-SDK 3.4 mmw demo	68xx - C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr68xx\mmw 64/68xxAoP C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr64xx\mmw	
MMWAVE Industrial Toolbox	MMWAVE Industrial Toolbox 68xx ISK – mmWave SDK Demo – DSP Version 64/68xx AoP - mmWave SDK Demo 68xx AoP 18xx – mmWave SDK Demo And various other demos included in Industrial Toolbox	Reference CCS Projectspecs for mmWave SDK mmw demos and other application specific demos.

5 Code Snapshots

This section provides code snapshots for the migrations notes presented in the previous sections.

5.1 SDK 3.3 API Change for MMWave_open

MMWAVE-SDK 3.2.1 vs MMWAVE-SDK 3.3.0

File: mmwave_sdk_03_03_00_0x\packages\ti\demo\xwr68xx\mmw\mss_main.c

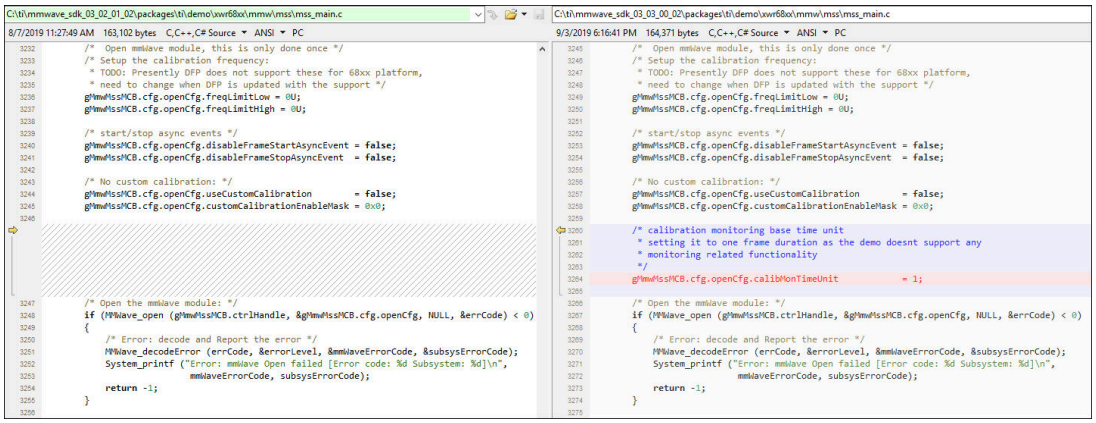

Figure 5-1. SDK 3.3 API Change for MMWave_open

5.2 SDK 3.3 API Change for ADCBuf_open

MMWAVE-SDK 3.2.1 vs MMWAVE-SDK 3.3.0

File: mmwave_sdk_03_03_00_0x\packages\ti\demo\xwr68xx\mmw\mss_main.c

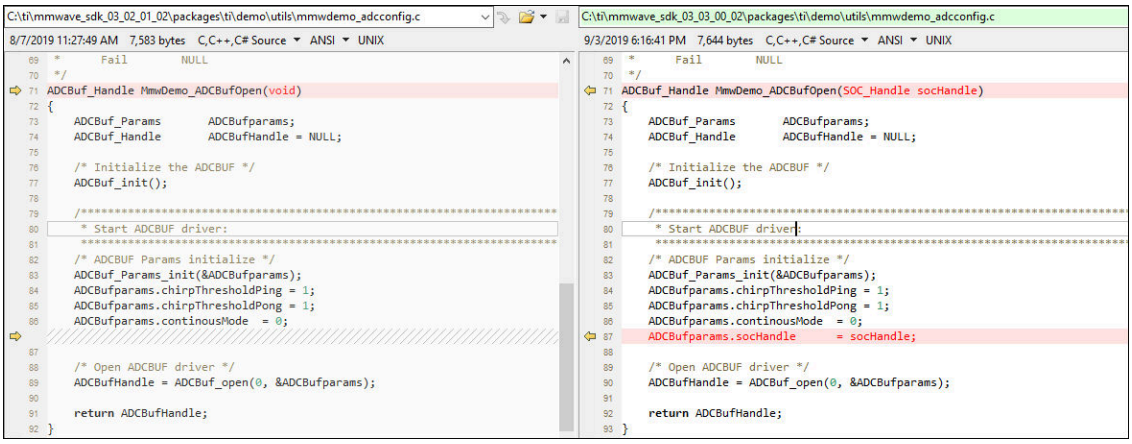

Figure 5-2. SDK 3.3 API Change for ADCBuf_open

5.3 SDK 3.3 API Change for CANFD_init

MMWAVE-SDK 3.2.1 vs MMWAVE-SDK 3.3.0

File: mmwave_sdk_03_03_00_0x\ packages\ti\drivers\canfd\test\xwr68xx\main.c

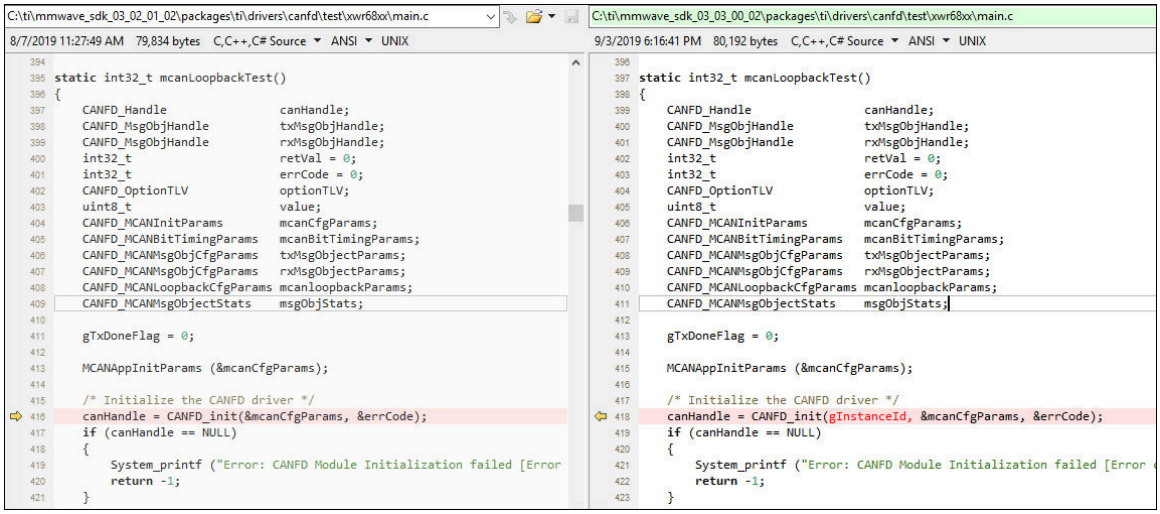

Figure 5-3. SDK 3.3 API Change for CANFD_init

5.4 SDK 3.3 68xx Secondary Bootloader Update

MMWAVE-SDK 3.2.1 vs MMWAVE-SDK 3.3.0

File: mmwave_sdk_03_03_00_02\packages\ti\utils\sbl\platform\sbl_xwr68xx.c

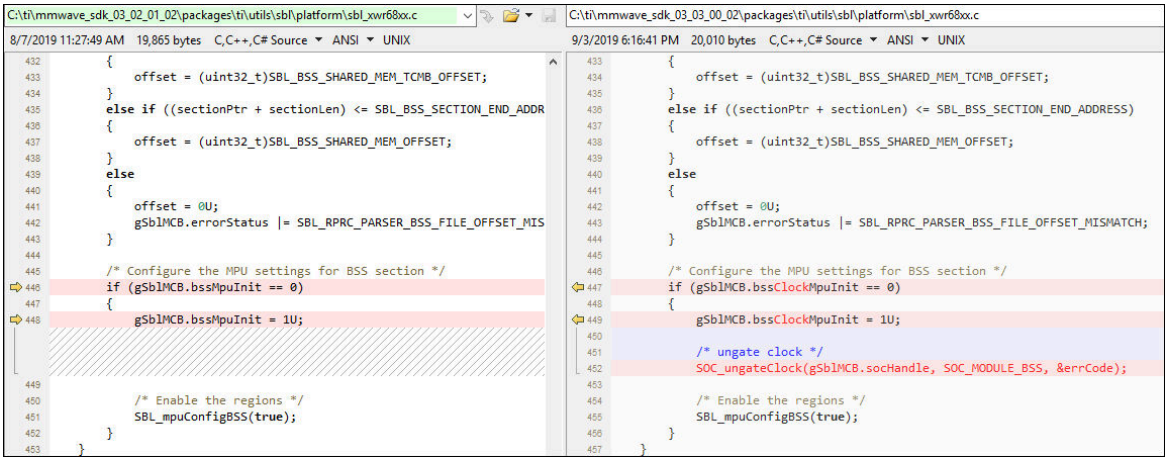

Figure 5-4. SDK 3.3 68xx Secondary Bootloader Update

5.5 SDK 3.3 16xx vs 68xx: Calibration Frequency Update

MMWAVE-SDK 3.3.0 mmw demo (16xx vs 68xx)

File: mmwave_sdk_03_03_00_0x\packages\ti\demo\xwr68xx\mmw\mss_main.c

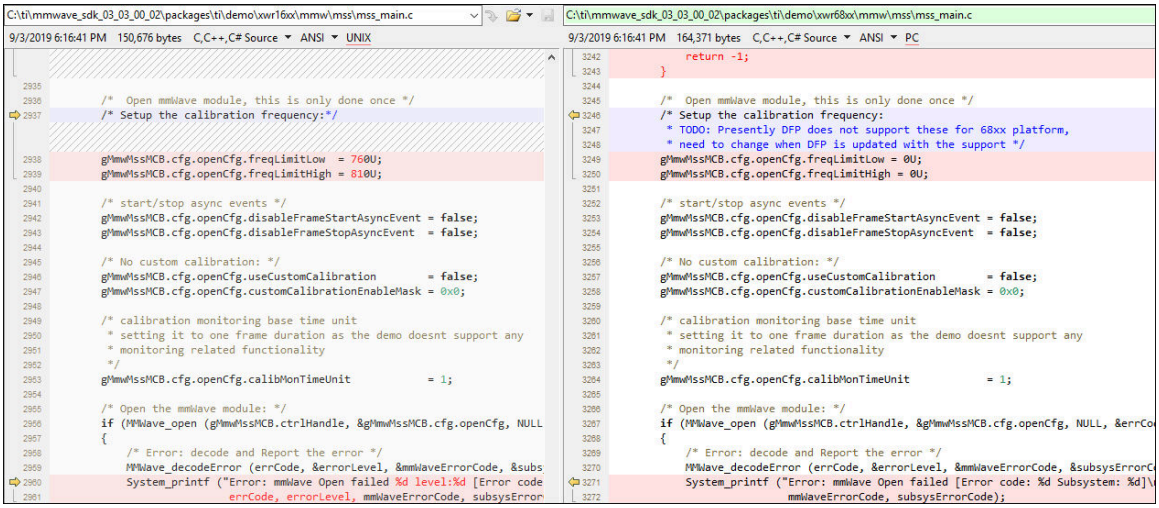

Figure 5-5. SDK 3.3 16xx vs 68xx: Calibration Frequency Update

5.6 SDK 3.3 16xx vs 68xx: SoC Definition Updates

MMWAVE-SDK 3.3.0 mmw demo (16xx vs 68xx)

File: mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr68xx\mmw\mss_main.c

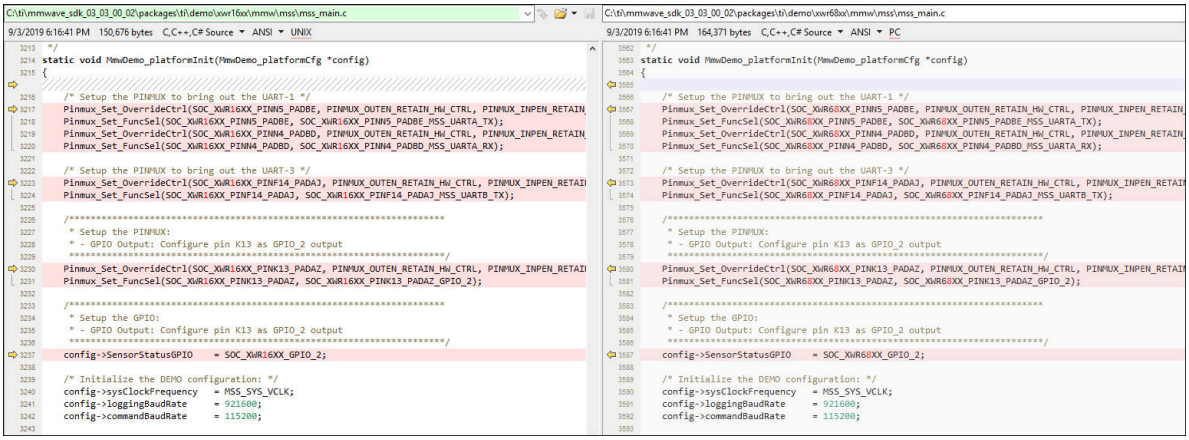

Figure 5-6. SDK 3.3 16xx vs 68xx: SoC Definition Updates

5.7 SDK 3.3 16xx vs 18xx: SoC Definition Updates

MMWAVE-SDK 3.3.0 mmw demo (16xx vs 18xx)

File: mmwave_sdk_03_03_xx_xx\packages\ti\demo\xwr18xx\mmw\mss_main.c

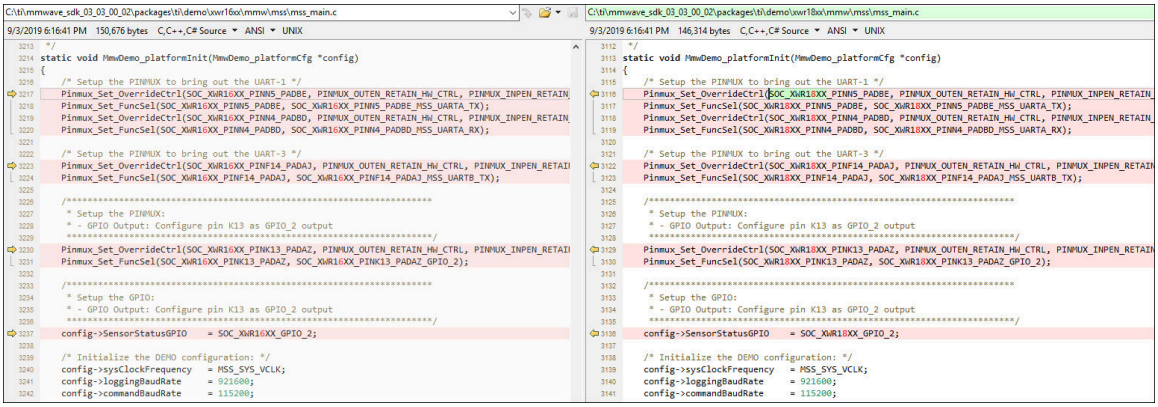

Figure 5-7. SDK 3.3 16xx vs 18xx: SoC Definition Updates

5.8 SDK 3.4 xWR68xx Calibration Frequency Update

MMWAVE-SDK 3.3 vs MMWAVE-SDK 3.4

File: mmwave_sdk_03_04_00_0x\packages\ti\demo\xwr68xx\mmw\mss_main.c

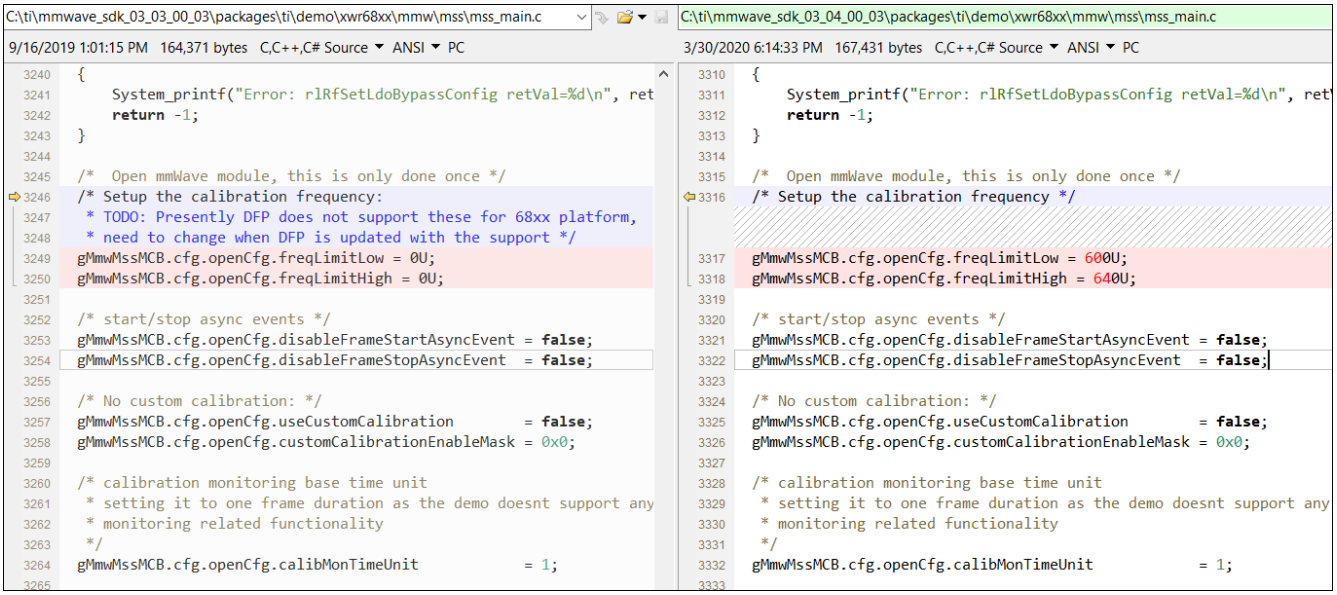

Figure 5-8. SDK 3.4 xWR68xx Calibration Frequency Update

5.9 SDK 3.4 Object Detect HWA DPC Range FFT Scaling

MMWAVE-SDK 3.3 vs MMWAVE-SDK 3.4

File: mmwave_sdk_03_04_00_0x\packages\ti\demo\xwr64xx\mmw\main.c

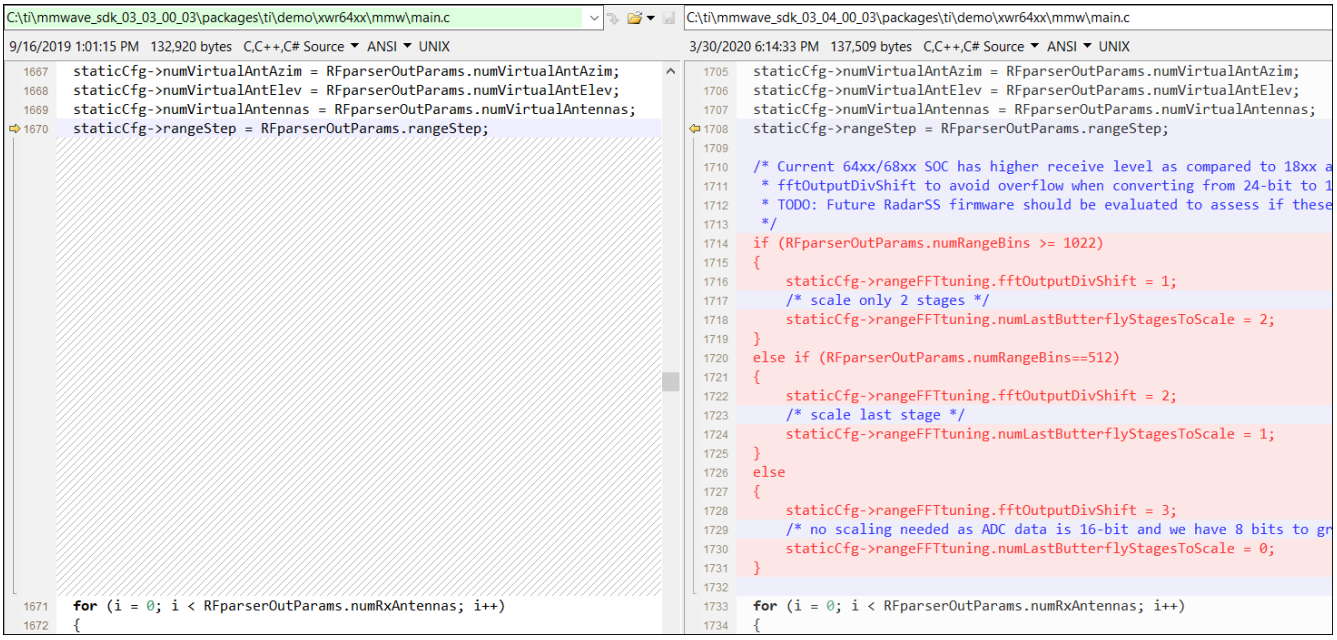

Figure 5-9. SDK 3.4 Object Detection DPC FFT Range Scaling Configuration

5.10 SDK 3.4 Object Detect Range HWA DPC Radar Cube Format

MMWAVE-SDK 3.3 vs MMWAVE-SDK 3.4

File: mmwave_sdk_03_04_00_0x\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c

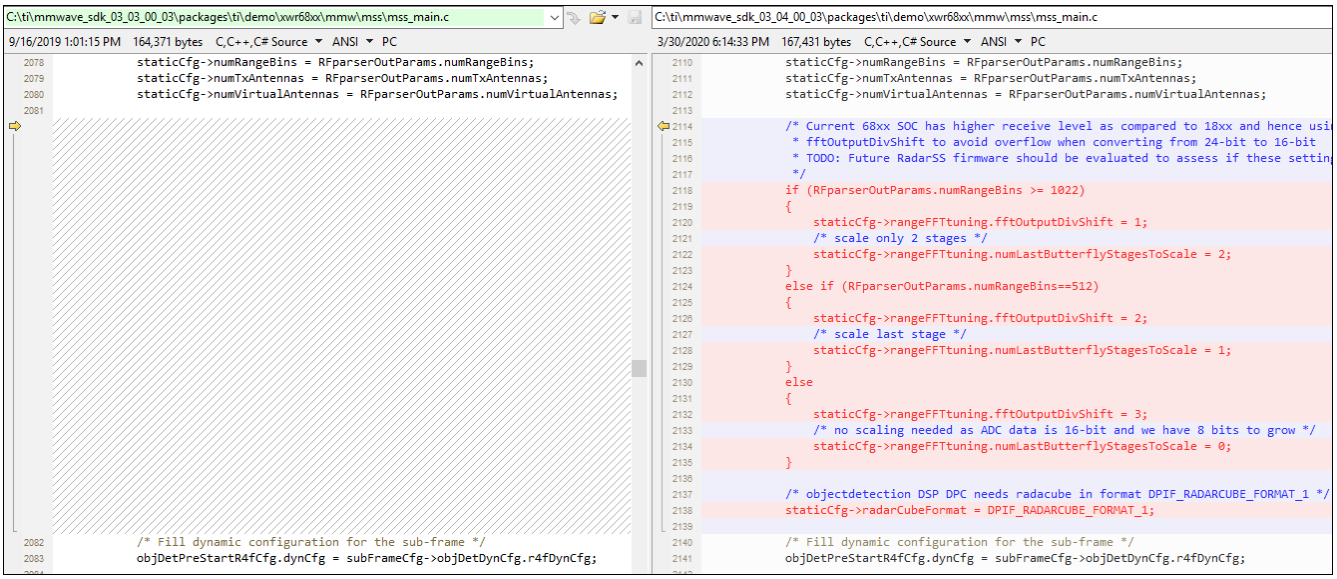

Figure 5-10. SDK 3.4 Object Detect Range HWA DPC FFT Radar Cube Format

5.11 xWR6843AoP ES1.0 Antenna Geometry

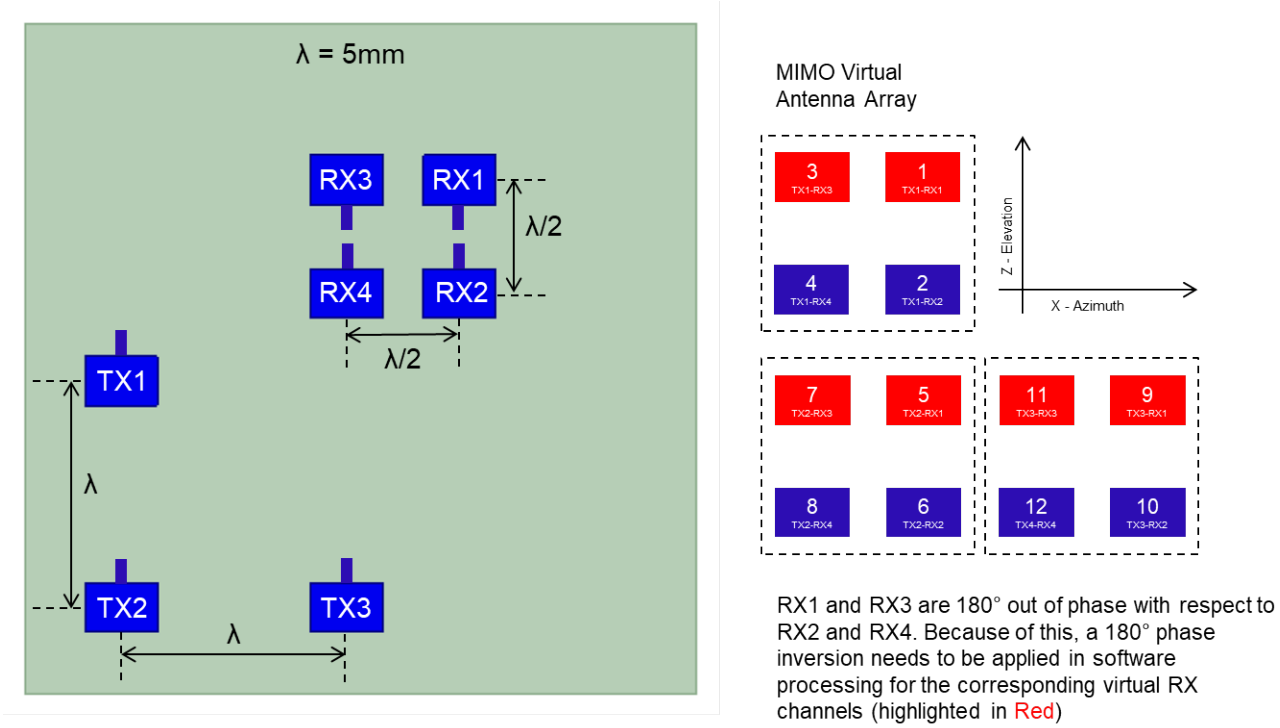

Figure 5-11. xWR6843AoP ES1.0 Antenna Geometry

5.12 xWR6843AoP ES2.0 Antenna Geometry

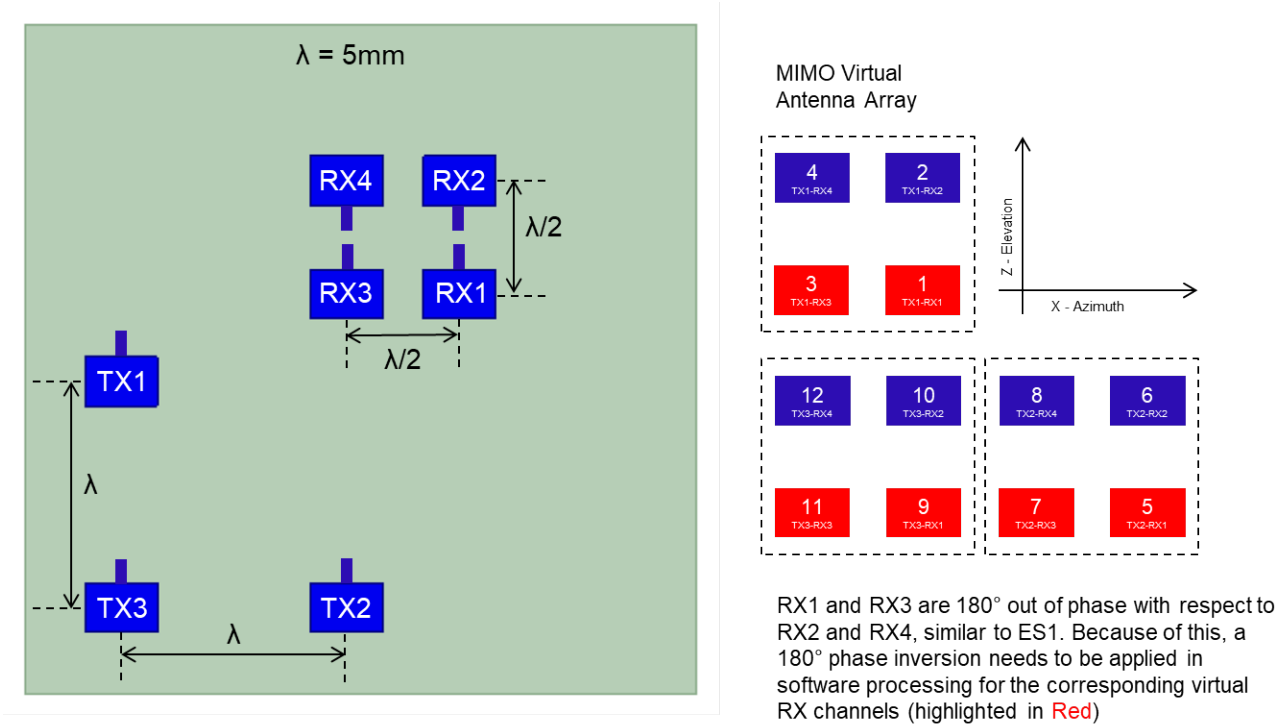

Figure 5-12. xWR6843AoP ES2.0 Antenna Geometry

5.13 xWR6843AoP ES2.0 Antenna Geometry Code Update

MMWAVE-SDK 3.2.0.6 vs MMWAVE-SDK 3.4

File: mmwave_sdk_03_04_00_0x\packages\ti\board\antenna_geometry.c

C:\ti\mmwave_sdk_03_02_00_06_AOP\packages\ti\board\antenna_geometry.c
5/31/2019 9:58:33 AM 2,822 bytes C,C++,C# Source ANSI UNIX

```
66 /**
67  * @brief Antenna geometry for IWR6843 AOP
68  *
69  */
70 ANTDEF_AntGeometry gAntDef_IWR6843AOP = {
71     .txAnt = {
72                 {0, 0},
73                 {0, 2},
74                 {2, 2}
75       },
76     .rxAnt = {
77                 {1, 0},
78                 {1, 1},
79                 {0, 0},
80                 {0, 1}

81             }
82 };
```

C:\ti\mmwave_sdk_03_04_00_03\packages\ti\board\antenna_geometry.c
3/30/2020 6:14:33 PM 2,823 bytes C,C++,C# Source ANSI UNIX

```
66 /**
67  * @brief Antenna geometry for IWR6843 AOP
68  *
69  */
70 ANTDEF_AntGeometry gAntDef_IWR6843AOP = {
71     .txAnt = {
72                 {0, 0},
73                 {2, 2},
74                 {0, 2}
75       },
76     .rxAnt = {

77                 {1, 1},
78                 {1, 0},
79                 {0, 1},
80                 {0, 0}
81             }
82 };
```

Figure 5-13. SDK 3.2.0.6 Vs SDK 3.4: Antenna Geometry Update for xWR6843AoP ES2.0

5.14 Antenna Geometry Structure Usage in mmw demo

File: mmwave_sdk_03_04_00_0x\packages\ti\demo\xwr64xx\mmw\main.c

C:\ti\mmwave_sdk_03_04_00_03\packages\ti\demo\xwr64xx\mmw\main.c
3/30/2020 6:14:33 PM 137,509 bytes C,C++,C# Source ANSI UNIX

```
1558         System_printf ("Error: Unable to get RF scale factor [Error:%d]\n", errCode);
1559         goto exit;
1560     }
1561
1562
1563     /* Copy antenna geometry definition */
1564 if defined(XWR68XX_AOP_ANTENNA_PATTERN)
1565     extern ANTDEF_AntGeometry gAntDef_IWR6843AOP;
1566     dataPathObj->objDetCommonCfg.preStartCommonCfg.antDef = gAntDef_IWR6843AOP;
1567 else
1568     extern ANTDEF_AntGeometry gAntDef_default;
1569     dataPathObj->objDetCommonCfg.preStartCommonCfg.antDef = gAntDef_default;
1570 endif
1571
1572     /* DPC pre-start common config */
1573     errCode = DPM_ioctl (dataPathObj->objDetDpmHandle,
1574                          DPC_OBJDET_IOCTL__STATIC_PRE_START_COMMON_CFG,
1575                          &dataPathObj->objDetCommonCfg.preStartCommonCfg,
1576                          sizeof (DPC_ObjectDetection_PreStartCommonCfg));
1577
1578     if (errCode < 0)
1579     {
```

Figure 5-14. Antenna Geometry Structure Usage in mmw demo

5.15 xWR6843AoP ES2.0 RX Channel Phase Compensation

MMWAVE-SDK 3.2.0.6 vs MMWAVE-SDK 3.4

File: mmwave_sdk_03_04_00_03\packages\ti\demo\xwr64xx\mmw\profiles\profile_3d_aop.cfg

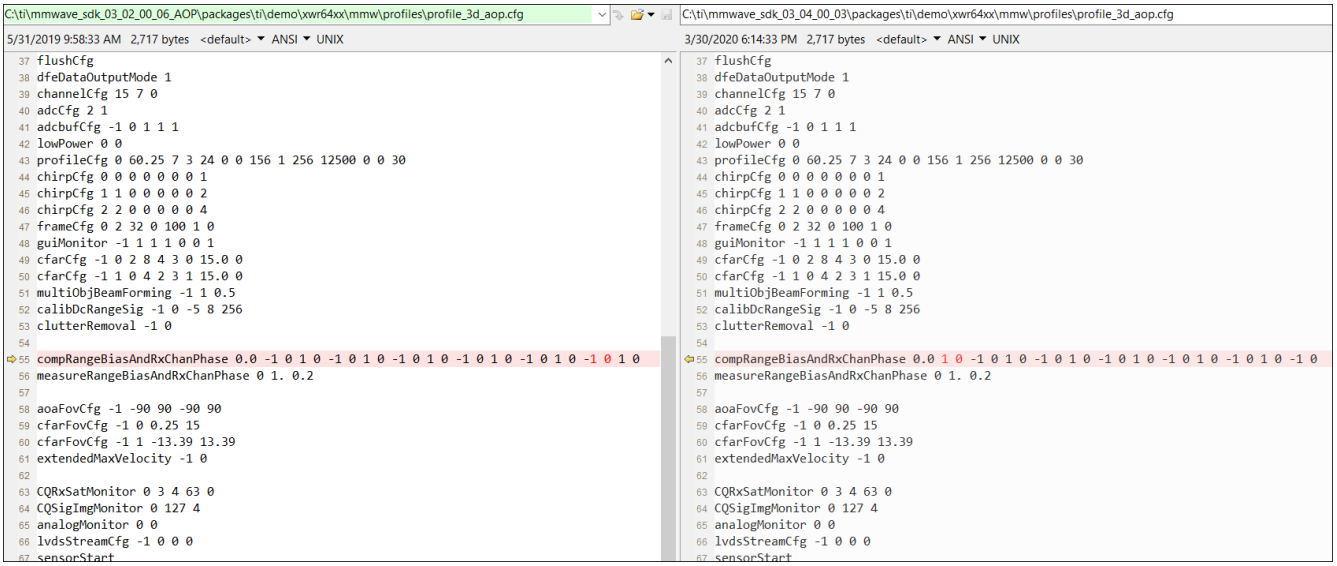

Figure 5-15. SDK 3.2.0.6 Vs SDK 3.4: RX Channel Phase Compensation

6 References

- Texas Instruments: *IWR1642 Device Errata*
- Texas Instruments: *AWR1642 Device Errata*
- Texas Instruments: *IWR1843 Device Errata*
- Texas Instruments: *AWR1843 Device Errata*
- Texas Instruments: IWR6843 Device Errata
- Texas Instruments: AWR6843 Device Errata
- xWR1642BOOST Layout and Design Files
- xWR6843AOPEVM Schematic, Assembly and Bill of Materials
- Texas Instruments: *xWR1642 EVM (xWR1642BOOST) Single-Chip mmWave Sensing Solution User's Guide*
- Texas Instruments: *MMWAVEICBOOST and Antenna Module User's Guide*
- *xWR6843 Checklist for Schematic Review, Layout Review, Bringup/Wakeup*
- xWR1843BOOST Hardware Files
- Texas Instruments: *xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution User's Guide*
- xWR6843 Product Page (Device data sheet, Silicon Errata)
- xWR6843AoP Product Page (Device data sheet, Silicon Errata)
- xWR1843 Product Page (Device data sheet, Silicon Errata)
- xWR1642 Product Page (Device data sheet, Silicon Errata)
- Texas Instruments: *IWR14xx/16xx/18xx/68xx Industrial Radar Family Technical Reference Manual*
- MMWAVE-SDK Product Page
- MMWAVE-SDK 3.3.0 download page (Release notes, User guide and SDK download)
- MMWAVE-SDK 3.4.0 download page (Release notes, User guide and SDK download)

7 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision B (May 2022) to Revision C (October 2022)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	3

Changes from Revision * (November 2019) to Revision A (May 2020)	Page
• Updates were made in Device Comparison topic	4
• Added new Section 2.1.2	5
• Updates were made in Section 2.1.2.1	5
• Added new Section 2.1.3	6
• Updates were made in Hardware Changes From xWR6843AoP ES1.0 to xWR6843AoP ES2.0 topic	9
• Update was made in Software Migration From xWR6843AoP ES1.0 to xWR6843AoP ES2.0 topic	11

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265