



Nitin Sakhujia and Chethan Kumar Y.B.

摘要

本应用报告为将毫米波硬件和应用程序软件移植到 xWR68xx ES2.0 和 xWR18xx 器件提供了指导。

内容

| | |
|---|----|
| 1 引言 | 3 |
| 2 xWR1843 硬件/软件迁移 | 4 |
| 2.1 从 xWR1642 迁移到 xWR1843 | 4 |
| 3 xWR6843AoP ES2.0 迁移 | 9 |
| 3.1 硬件从 xWR6843AoP ES1.0 更改为 xWR6843AoP ES2.0 | 9 |
| 3.2 软件从 xWR6843AoP ES1.0 迁移到 xWR6843AoP ES2.0 | 11 |
| 4 有用资源 | 17 |
| 5 代码快照 | 18 |
| 5.1 针对 MMWave_open 的 SDK 3.3 API 变更 | 18 |
| 5.2 针对 ADCBuf_open 的 SDK 3.3 API 变更 | 18 |
| 5.3 针对 CANFD_init 的 SDK 3.3 API 变更 | 19 |
| 5.4 SDK 3.3 68xx 次级引导加载程序更新 | 19 |
| 5.5 SDK 3.3 16xx 与 68xx : 校准频率更新 | 20 |
| 5.6 SDK 3.3 16xx 与 68xx : SoC 定义更新 | 20 |
| 5.7 SDK 3.3 16xx 与 18xx : SoC 定义更新 | 21 |
| 5.8 SDK 3.4 xWR68xx 校准频率更新 | 21 |
| 5.9 SDK 3.4 物体检测 HWA DPC 范围 FFT 缩放 | 22 |
| 5.10 SDK 3.4 物体检测范围 HWA DPC 雷达立方体格式 | 22 |
| 5.11 xWR6843AoP ES1.0 天线几何结构 | 23 |
| 5.12 xWR6843AoP ES2.0 天线几何结构 | 23 |
| 5.13 xWR6843AoP ES2.0 天线几何结构代码更新 | 24 |
| 5.14 毫米波演示中的天线几何结构用法 | 24 |
| 5.15 xWR6843AoP ES2.0 RX 通道相位补偿 | 24 |
| 6 参考文献 | 26 |
| 7 修订历史记录 | 26 |

插图清单

| | |
|---|----|
| 图 2-1. xWR1642 器件标识 | 4 |
| 图 2-2. xWR1843 器件标识 | 4 |
| 图 2-3. xWR1642 天线图像 | 5 |
| 图 2-4. xWR1843 天线图像 | 5 |
| 图 3-1. xWR6843AoP ES1.0 和 ES2.0 之间器件的器件标识差异 | 9 |
| 图 3-2. xWR6843AoP ES1.0 天线几何结构和产生的 MIMO 虚拟天线阵列 | 14 |
| 图 3-3. xWR6843AoP ES2.0 天线几何结构和产生的 MIMO 虚拟天线阵列 | 14 |
| 图 3-4. AoA2dProc HTML 文档 | 15 |
| 图 3-5. RX 通道相位补偿 : CompRangeBiasAndRxChanPhase CLI 命令 | 16 |
| 图 5-1. 针对 MMWave_open 的 SDK 3.3 API 变更 | 18 |
| 图 5-2. 针对 ADCBuf_open 的 SDK 3.3 API 变更 | 18 |
| 图 5-3. 针对 CANFD_init 的 SDK 3.3 API 变更 | 19 |
| 图 5-4. SDK 3.3 68xx 次级引导加载程序更新 | 19 |
| 图 5-5. SDK 3.3 16xx 与 68xx : 校准频率更新 | 20 |

| | |
|---|----|
| 图 5-6. SDK 3.3 16xx 与 68xx : SoC 定义更新..... | 20 |
| 图 5-7. SDK 3.3 16xx 与 18xx : SoC 定义更新..... | 21 |
| 图 5-8. SDK 3.4 xWR68xx 校准频率更新..... | 21 |
| 图 5-9. SDK 3.4 物体检测 DPC FFT 范围缩放配置..... | 22 |
| 图 5-10. SDK 3.4 物体检测范围 HWA DPC FFT 雷达立方体格式..... | 22 |
| 图 5-11. xWR6843AoP ES1.0 天线几何结构..... | 23 |
| 图 5-12. xWR6843AoP ES2.0 天线几何结构..... | 23 |
| 图 5-13. SDK 3.2.0.6 与 SDK 3.4 : xWR6843AoP ES2.0 的天线几何结构更新..... | 24 |
| 图 5-14. 毫米波演示中的天线几何结构用法..... | 24 |
| 图 5-15. SDK 3.2.0.6 与 SDK 3.4 : RX 通道相位补偿..... | 25 |

表格清单

| | |
|--|----|
| 表 1-1. 迁移参考..... | 3 |
| 表 2-1. 器件特性比较表..... | 5 |
| 表 2-2. xWR1642 至 xWR1843 软件迁移..... | 6 |
| 表 3-1. xWR6843AoP ES1.0 至 xWR6843AoP ES2.0 硬件变更..... | 9 |
| 表 3-2. xWR6843AoP ES2.0 软件 - 平台更新..... | 11 |

商标

所有商标均为其各自所有者的财产。

1 引言

此处提供的信息适用于以下任何情况：

- 当前在 xWR6843 ES1.0 上部署了硬件/软件，并希望将其迁移到 xWR6843 ES2.0
- 当前在 xWR1642 上部署了硬件/软件，并希望将其迁移到 xWR6843 ES2.0
- 当前在 xWR1642 上部署了硬件/软件，并希望将其迁移到 xWR1843
- 当前在 xWR6843AOP ES1.0 上部署了硬件/软件，并希望将其迁移到 xWR6843AOP ES2.0

本文档中提供的信息包括：

- 基本器件和新目标器件的比较，以及有关这些差异如何影响现有硬件和软件描述。
- 新目标器件所需的 SDK 版本，以及应用程序构建基础架构 (makefile 和/或 CCS 工程、链接器命令文件等) 所需的更新
- 应用程序源代码中所需的更新，例如 API 更新、新结构参数等。
- 提供了示例源代码比较快照以便于参考。

有关特定于您当前和目标器件的信息，请参阅以下各节。

表 1-1. 迁移参考

| 当前器件 | 目标器件 | 章节 |
|------------------|------------------|---|
| xWR6843 ES1. | xWR6843 ES 2.0 | 从 xWR6843 ES1.0 迁移到 xWR6843 ES2.0 : 节 3.2 |
| xWR1642 | xWR6843 ES2.0 | 从 xWR1642 迁移到 xWR6843 ES2.0 : 节 2.1 |
| xWR1642 | xWR1843 | 从 xWR1642 迁移到 xWR1843 : 节 2 |
| xWR6843AoP ES1.0 | xWR6843AoP Es2.0 | 从 xWR6843AoP ES1.0 迁移到 xWR6843AoP ES2.0 : 节 3 |

2 xWR1843 硬件/软件迁移

本节提供了将硬件和软件从 xWR1642 移植到 xWR1843 器件的迁移指南。此处提供的信息旨在介绍在撰写本文时迁移到特定 MMWAVE-SDK 版本的主要变更。有关更多信息，请参阅 [MMWAVE-SDK 版本说明](#) 中的 [迁移](#) 部分。

2.1 从 xWR1642 迁移到 xWR1843

2.1.1 器件比较

[表 2-1](#) 列出了从硬件和软件迁移角度需要考虑的 xWR1642 和 xWR1843 器件的主要特性。有关更多信息，请参阅 [节 6](#) 中的器件特定的数据表和 [工业毫米波雷达系列技术参考手册](#)。

[图 2-1](#) 和 [图 2-2](#) 显示了器件标识上的器件符号从 xWR1642 更改为 xWR1843。

左侧器件标识显示 xWR1642 器件，右侧器件标识显示 xWR1843 器件。有关器件标识的更多详细信息，请参阅特定于器件的勘误表。

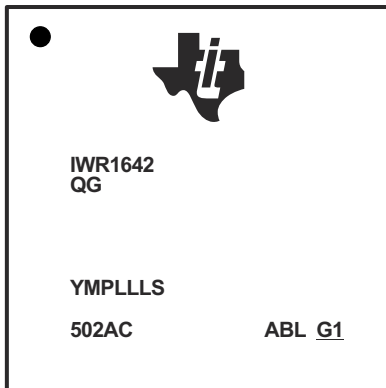


图 2-1. xWR1642 器件标识

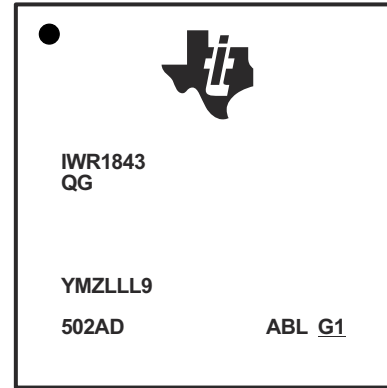


图 2-2. xWR1843 器件标识

- [IWR1642 器件勘误表](#)
- [AWR1642 器件勘误表](#)
- [IWR1843 器件勘误表](#)
- [AWR1843 器件勘误表](#)

表 2-1. 器件特性比较表

| 否 | 器件特性差异 | xWR1642 | xWR1843 | 硬件和软件影响 |
|---|---------------|---------------------|--------------------------|---|
| 1 | 发送通道数量 | 2 | 3 ⁽¹⁾ | 需要设计第 3 根变送器天线。更新 chirpCfg 中的 TX 位图 |
| 2 | 最大采样率 | 6.25MHz 复频率 | 12.5MHz 复频率 | xWR1843 上提供了更高的 IF 带宽和采样率 |
| 3 | 最大 I/F (中频) | 5MHz | 10MHz | |
| 4 | 片上存储器 | 1.5MB | 2.0MB | 如果需要, 软件可以利用额外的存储器。 |
| 5 | 雷达加速器 | 不适用 | 用于 FFT、滤波和 CFAR 处理的硬件加速器 | xWR1843 可在硬件加速器或 DSP 上灵活地处理数据 |
| 6 | Tx 波束形成 | 不支持 | 支持 | xWR1843 具有支持可操纵光束的移相器。注意: 天线需要设计为支持 TX 波束形成操作 |
| 7 | MMWAVE-SDK 支持 | SDK 2.1 (LTS) 及更高版本 | SDK 3.3.0 及更高版本 | 常规软件移植要求对 xWR1843 进行编译。有关更多信息, 请参阅节 2.1.4。 |

(1) 仅在 1V LDO 旁路和 PA LDO 禁用模式下支持三个 Tx 同时操作。在这种模式下, 需要在 VOUT PA 引脚上提供 1V 电源。

2.1.2 硬件迁移说明

2.1.2.1 添加天线

从 xWR1642 到 xWR1843, 都需要引入第三根天线。有关更多信息, 请参阅提供天线详细信息的设计文件包。详细的视场和辐射可在下面列出的用户指南中找到。

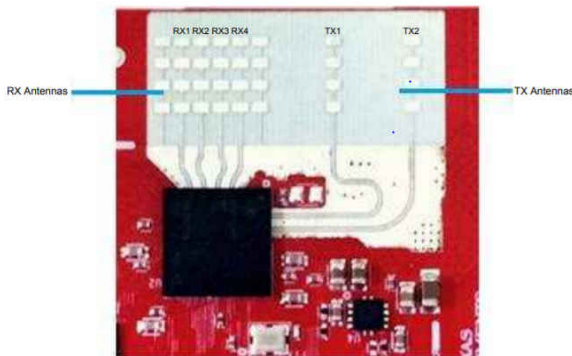


图 2-3. xWR1642 天线图像

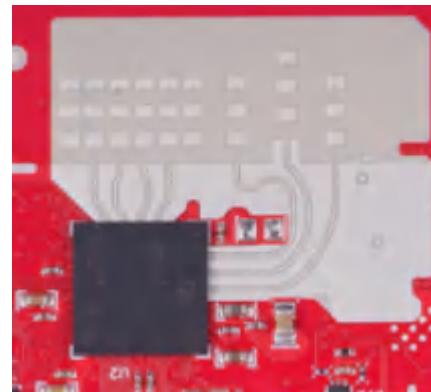


图 2-4. xWR1843 天线图像

- [xWR1642BOOST 布局和设计文件](#)
- [xWR1642 EVM \(xWR1642BOOST\) 单芯片毫米波传感解决方案用户指南](#)
- [xWR1843BOOST 硬件文件](#)
- [xWR1843 评估模块 \(xWR1843BOOST\) 单芯片毫米波传感解决方案用户指南](#)

2.1.3 硬件设计检查表

xWR1642 的硬件设计 (原理图、布局、启动/唤醒) 检查表可在 <http://www.ti.com/cn/lit/zip/swrr151> 上获得, 而对于 xWR1843, 硬件设计 (原理图、布局、启动/唤醒) 检查表, 可在 <http://www.ti.com/cn/lit/zip/spracl2> 上获得。

2.1.4 软件迁移说明

表 2-2 列出了将现有 xWR1642 应用程序代码移植到 xWR1843 所需的变更。

备注

本节提供的迁移说明仅适用于迁移到 MMWAVE-SDK 3.3。

在将现有 xWR1843 应用程序迁移到 MMWAVE-SDK 3.3 以外的 SDK 版本时，应遵循相应 SDK 版本说明中提供的增量迁移说明。

表 2-2. xWR1642 至 xWR1843 软件迁移

| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|--|-------------------------|---|
| 1 | xWR1843 需要 MMWAVE-SDK 3.2.1 或更高版本 注意： 建议使用 SDK 3.3.0 或更高版本以包含最新的 API 更新。 | Makefile 或 CCS 工程 | 应用程序代码必须使用 MMWAVE-SDK 3.3.0 或更高版本重新编译，才能在 xWR1843 上运行 Makefile： 如果您使用 SDK makefile，则无需更改，因为这会在 SDK 3.3 环境设置脚本中自动处理： C:\ti\mmwave_sdk_03_03_xx_xx\packages\scripts\windows\setenv.bat 或 CCS Projectspec： 如果应用程序是使用 CCS projectspec 编译的，则需要更新 DSS projectspec 和 MSS projectspec 中的 products 属性，如下所示。 <property name="products" value="com.ti.rtsc.SYSBIOS:6.73.01.01;com.ti.MMWAVE_SDK:3.3.0.03;"/> 示例： 有关 xWR1843 的参考 CCS 工程，请参阅“18xx - 毫米波 SDK 演示”，位于： 毫米波工业工具箱 。 |
| 2 | 更改器件类型 | Makefile 或 CCS 工程 | Makefile： 对于基于 SDK makefile 的构建，请在 setenv.bat 中设置 MMWAVE_SDK_DEVICE=iwr18xx/awr18xx。 C:\ti\mmwave_sdk_03_03_xx_xx\packages\scripts\windows\setenv.bat 或 CCS Projectspec： 如果使用 CCS projectspec 编译应用程序，请在 DSS projectspec 和 MSS projectspec 中将定义 SOC_XWR16XX 更改为 SOC_XWR18XX。 示例： 有关 xWR1843 的参考 CCS 工程，请参阅“18xx - 毫米波 SDK 演示”，位于： 毫米波工业工具箱 。 |
| 3 | 更新 RadarSS 固件文件路径 | Makefile 或 CCS 工程 (mss) | 需要在 metaimage 生成步骤中使用 xWR18xx_radarss_rprc.bin。 Makefile： 如果您使用 SDK makefile，则无需更改，因为这会在基于 MMWAVE_SDK_DEVICE 变量的 SDK 3.3 环境设置脚本中自动处理。 或 CCS Projectspec： 如果使用 CCS projectspec 编译应用程序，请在 metaimage 生成步骤（编译后处理步骤）中将 xwr16xx_radarss_rprc.bin 替换为 xWR18xx_radarss_rprc.bin 示例： 有关 xWR1843 的参考 CCS 工程，请参阅“18xx - 毫米波 SDK 演示”，位于： 毫米波工业工具箱 。 |
| 4 | 使用 xWR18xx 平台链接器命令文件 | Makefile 或 CCS 工程 | Makefile： 如果您使用 SDK makefile，则无需更改，因为这会在基于 MMWAVE_SDK_DEVICE 变量的 SDK 3.3 环境设置脚本中自动处理。 或 CCS Projectspec： 如果使用 CCS projectspec 编译应用程序，请将 r4f_linker.cmd 和 c674x_linker.cmd 的包含路径分别更新为： COM_TI_MMWAVE_SDK_INSTALL_DIR/packages/ti/platform/xwr18xx/r4f_linker.cmd 和 COM_TI_MMWAVE_SDK_INSTALL_DIR/packages/ti/platform/xwr18xx/c674x_linker.cmd。 示例： 有关 xWR1843 的参考 CCS 工程，请参阅“18xx - 毫米波 SDK 演示”，位于： 毫米波工业工具箱 。 |

表 2-2. xWR1642 至 xWR1843 软件迁移 (continued)

| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|---|------------------------|--|
| 5 | 包括 xWR18xx 驱动程序和 CLI 库 | Makefile 或 CCS 工程 | <p>Makefile : 如果您使用 SDK makefile, 则无需更改, 因为这会在基于 MMWAVE_SDK_DEVICE 变量的 SDK 3.3 环境设置脚本中自动处理。</p> <p>或</p> <p>CCS Projectspec : 如果使用 CCS projectspec 编译应用程序, 请更新链接器包含路径以选择 *_xwr18xx.aer4f 和 *_xwr18xx.xe674 库版本, 例如: -llibsoc_xwr18xx.ae674、-llibsoc_xwr18xx.xe674、-llibcli_xwr18xx.aer4f</p> |
| 6 | 更新传感器前端配置参数 | CLI 配置文件 (.cfg) 和/或源代码 | <p>更新 chirpCfg CLI 命令和/或 API 中的 TX 通道位图以解决第 3 个 TX 的问题。</p> <p>示例 : 有关更多信息, 请参阅 C:\ti\mmwave_sdk_03_03_xx_xx\packages\demo\xwr18xx\mmw\profiles 中的示例配置文件。</p> |
| 7 | 将 16xx SOC 定义替换为 18xx 等效项。 | MSS/DSS 源代码 | <p>将源代码中的 SOC_XWR16XX_* 定义/宏替换为相应 SOC_XWR18XX_* 定义。</p> <p>例如 :</p> <p>将 SOC_XWR16XX_MSS_ADCBUF_BASE_ADDRESS 替换为 SOC_XWR18XX_MSS_ADCBUF_BASE_ADDRESS , 类似地, 在 Pinmux 配置代码中: 将 SOC_XWR16XX_PINN5_PADBE 替换为 SOC_XWR18XX_PINN5_PADBE, 等等。</p> <p>下图显示了 SDK 16xx 和 18xx 毫米波演示之间的参考代码差异</p> <p>文件 : mmwave_sdk_03_03_xx_xx\packages\demo\xwr18xx\mmw\mss\mss_main.c 代码快照 : 请参阅 节 5.7。</p> |
| 8 | 针对 MMWave_open 的 API 更新 SDK 3.3 要求将新参数传递到 MMWave_open | MSS/DSS 启动代码 | <p>MMWave_open : 应用程序必须在调用 MMWave_open 之前设置 calibMonTimeUnit 参数的值, 如下所示。下图显示了 SDK 68xx 毫米波演示中的参考代码更新 (同样适用于 18xx 毫米波演示)</p> <p>文件 : mmwave_sdk_03_03_xx_xx\packages\demo\xwr68xx\mmw\mss\mss_main.c 代码快照 : 请参阅 节 5.1。</p> |
| 9 | 针对 ADCBuf_open 的 API 更新 SDK 3.3 要求将新参数传递到 ADCBuf_open | MSS/DSS 启动代码 | <p>ADCBUF_open : 应用程序必须在调用 ADCBUF_open 之前, 在 ADCBufparams 结构中设置 socHandle 的值, 如下所示。下图显示了 SDK 68xx 毫米波演示中的参考代码更新 (同样适用于 18xx 毫米波演示)。</p> <p>文件 : mmwave_sdk_03_03_xx_xx\packages\demo\utils\mmwdemo_adconfig.c 代码快照 : 请参阅 节 5.2。</p> |
| 10 | 针对 CANFD_init 的 API 更新 SDK 3.3 要求将新参数传递到 CANFD_init | 驱动程序 | <p>CANFD_init : 使用 CANFD 驱动程序的应用程序必须将实例 ID 传递给 CANFD_init API, 如下所示。目前仅支持值 0。下图显示了 SDK CANFD 驱动程序测试中的参考代码更新 (同样适用于 18xx)。</p> <p>文件 : mmwave_sdk_03_03_xx_xx\packages\drivers\canfd\test\xwr618xx\main.c 代码快照 : 请参阅 节 5.3。</p> |

表 2-2. xWR1642 至 xWR1843 软件迁移 (continued)

| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|------------------|--------|---|
| 11 | 有关 CLI 配置文件的一般说明 | 传感器配置 | 对于重复使用毫米波演示框架的应用程序，请确保配置命令 (profileCfg、chirpCfg、frameCfg 等) 遵循毫米波演示目录中提供的示例配置文件中提供的格式： C:\ti\mmwave_sdk_03_03_xx_xx\packages\ti\demo\wr18xx\mmw\profiles。 有关更多信息，请参阅 毫米波 SDK 用户指南 的 配置文件格式 部分。请参考 节 6 。 |

3 xWR6843AoP ES2.0 迁移

本节提供将硬件和软件从 xWR6843AoP ES1.0 移植到 xWR6843AoP Es2.0 器件的迁移指南。此处提供的信息旨在介绍在撰写本文时迁移到特定 MMWAVE-SDK 版本的主要变更。有关更多信息，请参阅 [MMWAVE-SDK 版本说明](#) 中的 [迁移](#) 部分。

3.1 硬件从 xWR6843AoP ES1.0 更改为 xWR6843AoP ES2.0

本节介绍了与将 xWR6843AoP ES1.0 硬件迁移到 xWR6843AoP ES2.0 相关的变更。[图 3-1](#) 显示了器件标识上的器件符号从 ES1.0 更改为 ES2.0。

左侧器件标识显示 ES1.0 器件，右侧器件标识显示 ES2.0 器件。有关器件标识的更多详细信息，请参阅 [xWR6843 器件勘误表](#)，[器件版本 1.0](#) 和 [2.0](#)。

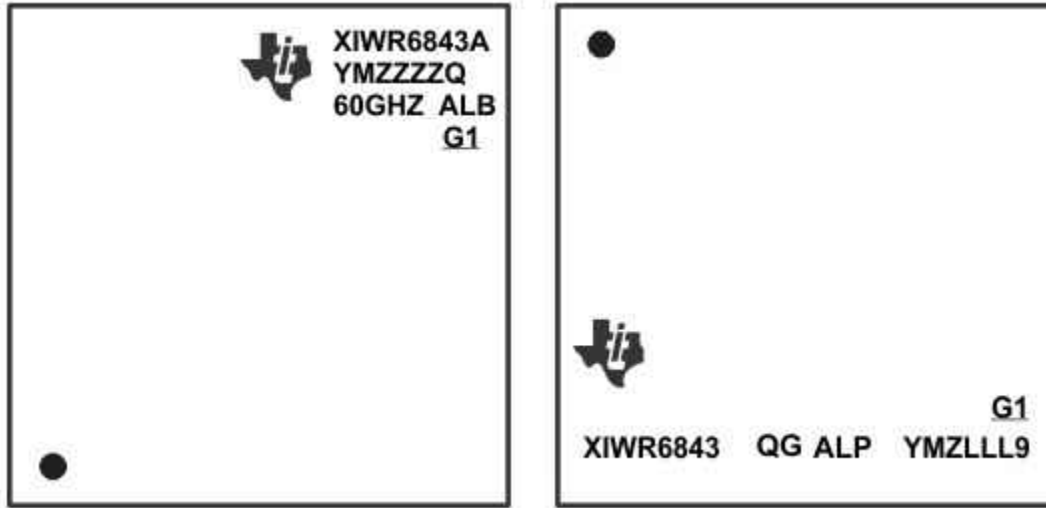


图 3-1. xWR6843AoP ES1.0 和 ES2.0 之间器件的器件标识差异

表 3-1. xWR6843AoP ES1.0 至 xWR6843AoP ES2.0 硬件变更

| 否 | 总结 | xWR6843AoP ES1.0 | xWR6843AoP ES2.0 |
|---|---|-----------------------|-------------------------------------|
| 1 | QSPI 接口速度得到提高。这可以加快引导加载速度，请注意， 毫米波传感器支持的闪存型号 中列出了支持的闪存。 | 最大 40MHz | 最大 80MHz |
| 2 | 引导加载程序得到增强。这样就可以实现更快的引导及器件间稳定性 | 用于执行 APLL 校准的引导加载程序代码 | 闭环 APLL 校准将由 BSS 完成 |
| 3 | 引入了 TX 波束扫描 | 不支持 | 支持 |
| 4 | 存储器压缩（根据雷达数据立方体的压缩比，更大的存储器可用于代码和剩余数据） | 不支持 | 支持 |
| 5 | 支持校准（这可以提高器件在整个温度范围内的性能和稳定性） | 无需校准 | 支持校准 |
| 6 | 上电时的时钟门控和基于用例的 IP 时钟门控，这应该可以提高节能效果 | 无时钟门控 | 未使用的外设上有时钟门控。器件低级驱动程序根据所使用的外设取消时钟门控 |
| 7 | 射频改进 - RX NF（范围和精度得到改进） | 基线 | 已改进（有关确切的数字，请参考数据表） |
| 8 | 射频改进 - CLK PN（精度得到提高） | 基线 | 已改进（有关确切的数字，请参考数据表） |
| 9 | 封装变更 | 基线 | 封装已改进（有关详细封装信息，请参考数据表） |

表 3-1. xWR6843AoP ES1.0 至 xWR6843AoP ES2.0 硬件变更 (continued)

| 否 | 总结 | xWR6843AoP ES1.0 | xWR6843AoP ES2.0 |
|----|-----------|------------------|---|
| 10 | 天线虚拟阵列的变化 | 基线 | 封装布线的改进导致天线元件发生变化，因此 ES1 和 ES2.0 之间的虚拟天线阵列发生了变化。请参阅 表 3-2 |

3.2 软件从 xWR6843AoP ES1.0 迁移到 xWR6843AoP ES2.0

本节介绍了与将基于 SDK 3.2.0.6 的 xWR6843AoP ES1.0 软件迁移到 xWR6843AoP ES2.0 和 SDK 3.4 相关的变更。

除了增加封装上的天线和不同的天线方向图外，xWR6843AoP ES2.0 还重复使用相同的器件。因此，软件从 xWR6843AoP ES1.0 迁移到 xWR6843AoP ES2.0 按顺序大致包括以下步骤：

1. 软件到 xWR6843ES2.0 的初始迁移（从 MMWAVE-SDK 3.2.0.6 到 MMWAVE-SDK 3.4）。（以下简称为平台软件更新）
2. xWR6843AoP ES2.0 上更新的天线方向图的到达角处理更新。（以下称为 AoA 软件更新）

备注

MMWAVE-SDK 3.4.0 是 xWR6843AoP ES2.0 器件的第一个基准 SDK 版本，本节提供的迁移说明范围仅限于迁移到 MMWAVE-SDK 3.4.0

在将现有 xWR6843 AOP ES2.0 应用程序迁移到 MMWAVE SDK 3.4 以外的 SDK 版本时，应遵循相应 SDK 版本说明中提供的增量迁移说明。

3.2.1 xWR6843AoP ES2.0 - 平台软件更新

表 3-2. xWR6843AoP ES2.0 软件 - 平台更新

| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|---|--------------------------|---|
| 1 | xWR6843AoP ES2.0 需要 MMWAVE-SDK 3.4.0 或更高版本 | Makefile 或 CCS 工程 | <p>应用程序代码必须使用 MMWAVE-SDK 3.4.0 或更高版本重新编译，才能在 xWR6843AoP ES2.0 上运行，因为之前的 SDK 版本与 ES2.0 不兼容。反过来，SDK 3.4.0 与 xWR6843AoP ES1.0 器件不兼容。</p> <p>Makefile：如果您使用 SDK makefile，则无需更改，因为这会在 SDK 3.4 环境设置脚本中自动处理： <code>C:\ti\mmwave_sdk_03_04_xx_xx\packages\scripts\windows\setenv.bat</code> 或</p> <p>CCS Projectspec：如果应用程序是使用 CCS projectspec 编译的，则需要更新 DSS projectspec 和 MSS projectspec 中的 products 属性，如下所示。</p> <pre><property name="products" value="com.ti.rtsc.SYSBIOS:6.73.01.01;com.ti.MMWAVE_SDK:3.4.0.03;"/></pre> <p>示例：有关 xWR6843AoP ES2.0 的参考 CCS 工程，请参阅毫米波工业工具箱中的“68xx AoP - 毫米波 SDK 演示”。</p> |
| 2 | 在 MetalImage（可擦写）二进制文件生成步骤中更改 SHMEM_ALLOC 参数的值。 | Makefile 或 CCS 工程 (mss)。 | <p>对于 ES2.0，SHMEM_ALLOC 参数的值应该设置为 0x00000006（对于 ES1.0 器件，该值为 0x02000006）。</p> <p>Makefile：如果您使用 SDK makefile，则无需更改，因为这会在 SDK 3.4 器件特定的 makefile 中自动处理。</p> <p>或</p> <p>CCS Projectspec：如果使用 CCS projectspec 编译应用程序，请更新 MSS projectspec 中的 postBuildStep，以将值 0x02000006 替换为 0x00000006。</p> <p>示例：有关 xWR6843AoP ES2.0 的参考 CCS 工程，请参阅毫米波工业工具箱中的“68xx AoP - 毫米波 SDK 演示”。</p> |

表 3-2. xWR6843AoP ES2.0 软件 - 平台更新 (continued)

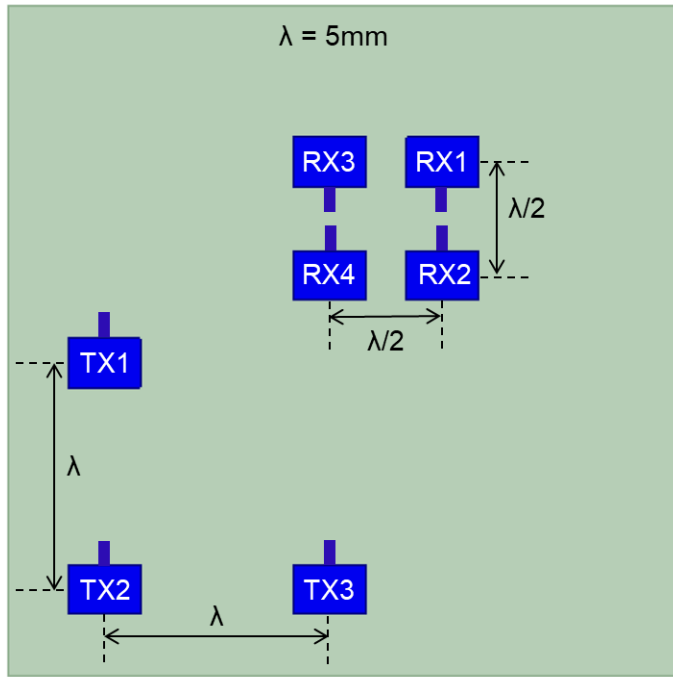
| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|---|-------------------------|--|
| 3 | 更新 RadarSS 固件文件名 | Makefile 或 CCS 工程 (mss) | xwr6xxx 器件的 RadarSS 二进制文件现在称为 xwr6xxx_radarss_rprc.bin, 而不是 iwr6xxx_radarss_rprc.bin。 Makefile : 如果您使用 SDK makefile, 则无需更改, 因为这会在基于 MMWAVE_SDK_DEVICE 变量的 SDK 3.4 环境设置脚本中自动处理。 或 CCS Projects spec : 如果使用 CCS projectspec 编译应用程序, 请在 metaimage 生成步骤 (编译后处理步骤) 中将 iwr6xxx_radarss_rprc.bin 替换为 xwr6xxx_radarss_rprc.bin 示例 : 有关 xWR6843AoP ES2.0 的参考 CCS 工程, 请参阅毫米波工业工具箱中的“68xx AoP - 毫米波 SDK 演示”。 |
| 4 | 针对 MMWave_open 的 API 更新 SDK 3.3 和更高版本要求将新参数传递到 MMWave_open | MSS/DSS 启动代码 | MMWave_open : 应用程序必须在调用 MMWave_open 之前设置 calibMonTimeUnit 参数的值, 如下所示。下图显示了 SDK 68xx 毫米波演示中的参考代码更新: 文件 : mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c 代码快照 : 请参阅 节 5.1 |
| 5 | 针对 ADCBuf_open 的 API 更新 SDK 3.3 和更高版本要求将新参数传递到 ADCBuf_open | MSS/DSS 启动代码 | ADCBUF_open : 应用程序必须在调用 ADCBUF_open 之前, 在 ADCBufparams 结构中设置 socHandle 的值, 如下所示。下图显示了 SDK 68xx 毫米波演示中的参考代码更新。 文件 : mmwave_sdk_03_04_xx_xx\packages\ti\demo\utils\mmwdemo_adconfig.c 代码快照 : 请参阅 节 5.2 |
| 6 | 针对 CANFD_init 的 API 更新 SDK 3.3 和更高版本要求将新参数传递到 CANFD_init | 驱动程序 | CANFD_init : 使用 CANFD 驱动程序的应用程序必须将实例 ID 传递给 CANFD_init API, 如下所示。目前仅支持值 0。下图显示了 SDK CANFD 驱动程序测试中的参考代码更新。 文件 : mmwave_sdk_03_04_xx_xx\packages\ti\drivers\canfd\test\xwr68xx\main.c 代码快照 : 请参阅 节 5.3 |
| 7 | SDK 3.3 及更高版本从 xWR6843 ES2 的 DMA 驱动程序中删除了对总线错误中断的支持, 因为该中断未连接到器件。 | 驱动程序 | 如果针对 DMA_IntType_BER 调用 DMA_enable 中断 API, 应用程序将从 xwr68xx 驱动程序获得错误代码。您可以删除对上述 API 的调用或忽略错误; 但是, 您应该查看 DMA 使用情况, 以确保没有通过 MSS DMA 引擎进行无效的存储器访问。 |
| 8 | 有关 CLI 配置文件的一般说明 | 传感器配置 | 对于重复使用毫米波演示/CLI 框架的应用程序, 请确保配置命令 (例如, profileCfg、chirpCfg、frameCfg 等) 遵循毫米波演示目录中提供的示例配置文件中提供的格式: C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr64xx\mmw\profiles. 有关更多详细信息, 请参阅毫米波 SDK 用户指南中的配置文件格式部分。节 6 |

表 3-2. xWR6843AoP ES2.0 软件 - 平台更新 (continued)

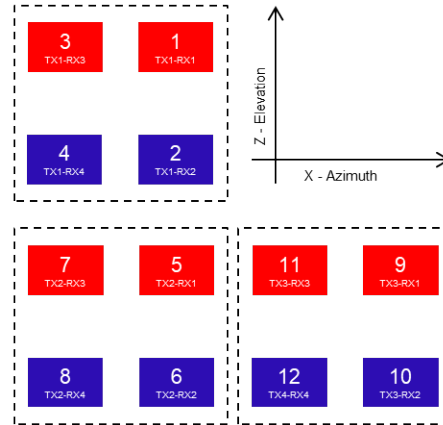
| 编号 | 总结 | 受影响的部分 | 必需的变更 |
|----|--|--|---|
| 9 | 次级引导加载程序中需要取消 BSS 时钟门控 | 次级引导加载程序 | <p>注意：此更新与主应用程序无关。 仅当您在系统中使用自定义次级引导加载程序时，才需要此更新。在将映像下载到 RadarSS/BSS 存储器之前，次级引导加载程序必须使用 SOC 门控/取消门控 API 来取消 BSS 时钟门控，如下所示。下图显示了 SDK 次级引导加载程序示例中的参考代码更新。</p> <p>File: C:\ti\mmwave_sdk_03_04_xx_xx\packages\ti\utils\sb\platform\sbl_xwr68xx.c 代码快照：请参阅节 5.4</p> |
| 10 | SDK 3.4 毫米波层对于 xwr6xxx 器件支持所有有效初始化和运行时校准 | MSS/DSS 启动代码 | <p>应用程序应在 mmWave_Open API 中传递 freqLimitLow 和 freqLimitHigh 的有效值，现在可以在 mmWave_Start API 中启用定期校准。下图显示 SDK 68xx 毫米波演示中的参考代码更新。</p> <p>文件： mmwave_sdk_03_04_00_03\packages\ti\demo\wxwr68xx\mmw\mss\mss_main.c 代码快照：请参阅节 5.8</p> |
| 11 | 物体检测 DPC 接受天线几何结构，以实现更宽的 Tx/Rx 天线配置 | DPC 配置 | <p>当编译为使用新的 AoA 2D 算法（在 xwr64xx AOP 毫米波演示中）时，只有基于 HWA 的物体检测 DPC 才必须使用此字段。对于基于 DSP 的 DPC 和使用标准 AoA DPU 的基于 HWA 的 DPC，不使用此字段。下图显示了 SDK 64xx 毫米波演示中的参考代码。</p> <p>文件： mmwave_sdk_03_04_00_03\packages\ti\demo\wxwr64xx\mmw\main.c 代码快照：请参阅节 5.14</p> |
| 12 | 物体检测 HWA DPC 现在接受范围 FFT 缩放参数 | DPC 配置 | <p>基于范围 HWA 的 DPU 和基于物体检测的 DPC 现在允许您设置螺旋形的缩放值，并将内部 24 位输出转换为 16 位输出。下图显示了 SDK 64xx 毫米波演示中的参考代码。</p> <p>文件： mmwave_sdk_03_04_00_03\packages\ti\demo\wxwr64xx\mmw\main.c 代码快照：请参阅节 5.9</p> |
| 13 | 物体检测范围 HWA DPC 现在允许用户指定雷达立方体格式 | DPC 配置 | <p>ObjDetRangeHWA DPC 允许用户指定雷达立方体格式，以便灵活地集成各种基于 DSP 的算法/处理链。</p> <p>注意：毫米波演示仅支持 DPIF_RADARCUBE_FORMAT_</p> <p>下图显示了 SDK 68xx 毫米波演示中的参考代码。</p> <p>文件： mmwave_sdk_03_04_00_03\packages\ti\demo\wxwr68xx\mmw\mss\mss_main.c 代码快照：请参阅节 5.10</p> |
| 14 | 与保存/恢复器件校准参数（相移校准参数）相关的更新 | 有关此更新和其他校准相关更新的更多详细信息，请参阅 迁移说明 中的 MMWAVE-SDK 3.4.0 版本说明。 | |

3.2.2 xWR6843AoP ES2.0 - AoA 软件更新

图 3-2 和图 3-3 比较了 xWR6843AOP ES1.0 和 xWR6843AOP ES2.0 的天线几何结构。

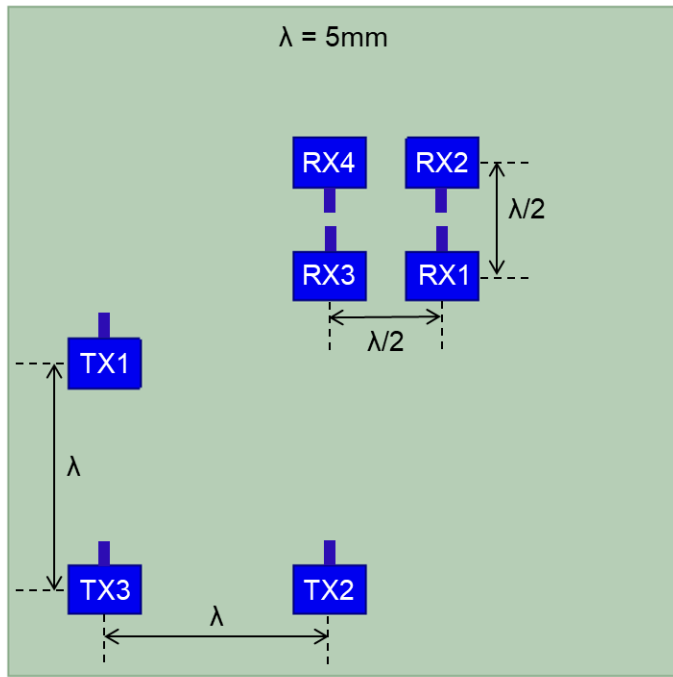


MIMO Virtual Antenna Array

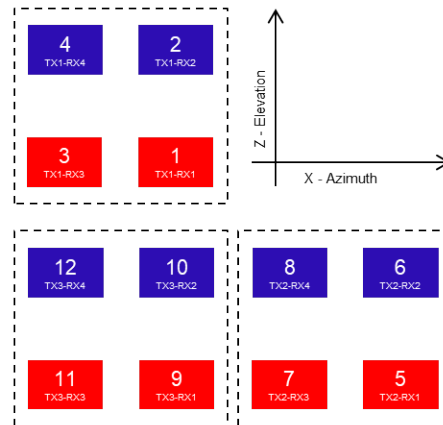


RX1 and RX3 are 180° out of phase with respect to RX2 and RX4. Because of this, a 180° phase inversion needs to be applied in software processing for the corresponding virtual RX channels (highlighted in Red)

图 3-2. xWR6843AoP ES1.0 天线几何结构和产生的 MIMO 虚拟天线阵列



MIMO Virtual Antenna Array



RX1 and RX3 are 180° out of phase with respect to RX2 and RX4, similar to ES1. Because of this, a 180° phase inversion needs to be applied in software processing for the corresponding virtual RX channels (highlighted in Red)

图 3-3. xWR6843AoP ES2.0 天线几何结构和产生的 MIMO 虚拟天线阵列

xWR6843AOP ES2 中的关键天线更新 (如上所示) 包括:

- **RX** 天线: RX1 和 RX2 在 xWR6843AOP ES2 上交换。同样, RX3 和 RX4 交换
- **TX** 天线: TX2 和 TX3 在 xWR6843AOP ES2 上交换。

- **线路馈送**：xWR6843AoP ES2 上的 RX 线路馈送与 ES1 上的 RX 线路馈送相同，即 RX1 和 RX2 从相反的两端馈送，这会导致 RX1 和 RX2 之间产生 180° 的相位差。同样，RX3 和 RX4 相位差为 180°。要补偿反向线路馈送，需要在软件处理中为相应的虚拟通道应用 180° 相位反转，如图 3-3 所示。

MMWAVE-SDK 3.2.0.6 和 MMWAVE-SDK 3.4 包括 AoA2dProc DPU，后者使用硬件加速器对 xWR6843 AoP 天线阵列执行到达角处理。AoA2dProc DPU (数据路径处理单元) 在 xWR64xx AoP 毫米波演示中用于进行到达角处理。

要了解 xWR6843AoP ES2 所需的 AoA 更新，建议了解 AoA2dProc DPU 中定义的天线几何结构概念。

1. 导航至 C:\ti\mmwave_sdk_03_04_xx_xx\docs 并在浏览器中打开文件 mmwave_sdk_module_documentation.html。
2. 点击下图中突出显示的“AoA using 2D FFT method”链接：

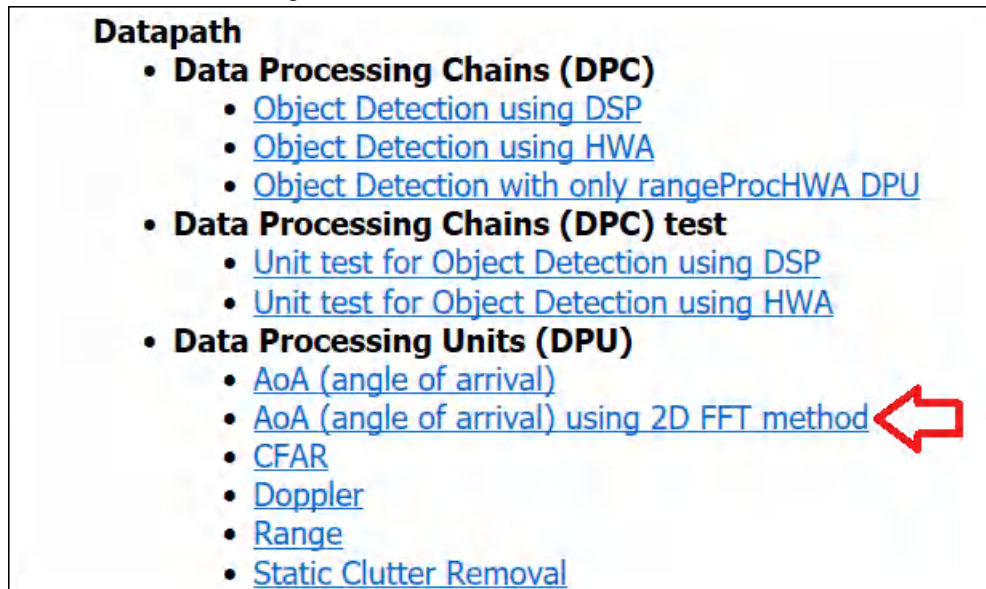


图 3-4. AoA2dProc HTML 文档

3. 向下滚动到名为“Antenna Geometry Definition”的部分，其中解释了 HWA AoA2dProc DPU 代码如何定义和使用通用天线几何结构。特定天线 (例如，xWR6843AoP ES2.0) 的天线几何结构在相应的 C 结构 mmwave_sdk_03_04_xx_xx\packages\ti\board\antenna_geometry.c 中定义。

下图显示了 xWR6843AoP ES2.0 与 MMWAVE-SDK 3.2.0.6 中的 xWR6843AoP ES1.0 相比的天线几何结构更新。

代码快照：请参阅节 5.13

天线几何结构在 mmwave_sdk_03_04_xx_xx\packages\ti\demo\xwr64xx\mmw\main.c 中初始化期间传递给物体检测 DPC

代码快照：请参阅节 5.14

RX 通道相位补偿：为了补偿反向线路馈送 (如节 5.12 所示)，使用毫米波演示中提供的 compRangeBiasAndRxChanPhase CLI 命令，对相应的 RX 通道 (包括虚拟通道) 应用 180° 相位反转。

MMWAVE-SDK 用户指南中的图 3-5 解释了此命令的结构。

| | | | |
|------------------------------------|---|--|---|
| <p>compRangeBiasAndRxChanPhase</p> | <p>Command for datapath to compensate for bias in the range estimation and receive channel gain and phase imperfections. Refer to the procedure mentioned here</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p> | <p><rangeBias> Compensation for range estimation bias in meters</p> <p><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)></p> <p>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</p> | <p>supported</p> <p>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.</p> <p>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)</p> |
|------------------------------------|---|--|---|

图 3-5. RX 通道相位补偿 : CompRangeBiasAndRxChanPhase CLI 命令

要了解在 MMWAVE-SDK 中提供的示例 AOP 配置文件配置中配置的 CompRangeBiasAndRxChanPhase 值，请参阅节 5.15。

4 有用资源

以下资源提供了 xWR6843 ES2.0 和 xWR1843 器件的示例源代码、makefile 和 CCS 工程。

| 资源名称 | 文件系统路径/Web URL | 内容参考 |
|--------------------------|--|---|
| MMWAVE-SDK 3.3 毫米波演示 | 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 | 源代码、Makefile、配置文件 (.cfg) |
| MMWAVE-SDK 3.4 毫米波演示 | 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 | |
| 毫米波工业工具箱 | 毫米波工业工具箱 68xx ISK - 毫米波 SDK 演示 - DSP 版本 64/68xx AoP - 毫米波 SDK 演示 68xx AoP 18xx - 毫米波 SDK 演示 以及工业工具箱中包含的各种其他演示 | 参考适用于毫米波 SDK 毫米波演示和其他应用特定演示的 CCS Projectspec。 |

5 代码快照

本节提供前几节所述迁移说明的代码快照。

5.1 针对 MMWave_open 的 SDK 3.3 API 变更

MMWAVE-SDK 3.2.1 与 MMWAVE-SDK 3.3.0

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

```

C:\ti\mmwave_sdk_03_02_01_02\packages\ti\demo\xwr68xx\mmw\mss_main.c
8/7/2019 11:27:49 AM 163,102 bytes C,C++,C# Source ANSI PC
3232 /* Open mmwave module, this is only done once */
3233 /* Setup the calibration frequency:
3234 * TODD: Presently DFP does not support these for 68xx platform,
3235 * need to change when DFP is updated with the support */
3236 gMmWssPCB.cfg.openCfg.freqLimitLow = 0U;
3237 gMmWssPCB.cfg.openCfg.freqLimitHigh = 0U;
3238
3239 /* start/stop async events */
3240 gMmWssPCB.cfg.openCfg.disableFrameStartAsyncEvent = false;
3241 gMmWssPCB.cfg.openCfg.disableFrameStopAsyncEvent = false;
3242
3243 /* No custom calibration: */
3244 gMmWssPCB.cfg.openCfg.useCustomCalibration = false;
3245 gMmWssPCB.cfg.openCfg.customCalibrationEnableMask = 0x0;
3246
3247
3248 /* Open the mmwave module: */
3249 if (MmWave_open (gMmWssPCB.ctrlHandle, &gMmWssPCB.cfg.openCfg, NULL, &errCode) < 0)
3250 {
3251 /* Error: decode and Report the error */
3252 MmWave_decodeError (errCode, &errorLevel, &mMwaveErrorCode, &subsysErrorCode);
3253 System_printf ("Error: mmwave Open failed [Error code: %d Subsystem: %d]\n",
3254 mMwaveErrorCode, subsysErrorCode);
3255 return -1;
3256 }
3257
C:\ti\mmwave_sdk_03_03_00_02\packages\ti\demo\xwr68xx\mmw\mss_main.c
9/3/2019 6:16:41 PM 164,371 bytes C,C++,C# Source ANSI PC
3246 /* Open mmwave module, this is only done once */
3247 /* Setup the calibration frequency:
3248 * TODD: Presently DFP does not support these for 68xx platform,
3249 * need to change when DFP is updated with the support */
3250 gMmWssPCB.cfg.openCfg.freqLimitLow = 0U;
3251 gMmWssPCB.cfg.openCfg.freqLimitHigh = 0U;
3252
3253 /* start/stop async events */
3254 gMmWssPCB.cfg.openCfg.disableFrameStartAsyncEvent = false;
3255 gMmWssPCB.cfg.openCfg.disableFrameStopAsyncEvent = false;
3256
3257 /* No custom calibration: */
3258 gMmWssPCB.cfg.openCfg.useCustomCalibration = false;
3259 gMmWssPCB.cfg.openCfg.customCalibrationEnableMask = 0x0;
3260
3261 /* calibration monitoring base time unit
3262 * setting it to one frame duration as the demo doesnt support any
3263 * monitoring related functionality
3264 */
3265 gMmWssPCB.cfg.openCfg.callMonTimeUnit = 1;
3266
3267 /* Open the mmwave module: */
3268 if (MmWave_open (gMmWssPCB.ctrlHandle, &gMmWssPCB.cfg.openCfg, NULL, &errCode) < 0)
3269 {
3270 /* Error: decode and Report the error */
3271 MmWave_decodeError (errCode, &errorLevel, &mMwaveErrorCode, &subsysErrorCode);
3272 System_printf ("Error: mmwave Open failed [Error code: %d Subsystem: %d]\n",
3273 mMwaveErrorCode, subsysErrorCode);
3274 return -1;
3275 }
    
```

图 5-1. 针对 MMWave_open 的 SDK 3.3 API 变更

5.2 针对 ADCBuf_open 的 SDK 3.3 API 变更

MMWAVE-SDK 3.2.1 与 MMWAVE-SDK 3.3.0

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

```

C:\ti\mmwave_sdk_03_02_01_02\packages\ti\demo\utils\mmwdemo_adconfig.c
8/7/2019 11:27:49 AM 7,583 bytes C,C++,C# Source ANSI UNIX
70 /* Fail NULL
71 ADCBuf_Handle MmDemo_ADCBufOpen(void)
72 {
73 ADCBuf_Params ADCBufparams;
74 ADCBuf_Handle ADCBufHandle = NULL;
75
76 /* Initialize the ADCBUF */
77 ADCBuf_init();
78
79 /* Start ADCBUF driver:
80 * Start ADCBUF driver:
81
82 /* ADCBUF Params initialize */
83 ADCBuf_Params_init(&ADCBufparams);
84 ADCBufparams.chirpThresholdPing = 1;
85 ADCBufparams.chirpThresholdPong = 1;
86 ADCBufparams.continuousMode = 0;
87
88 /* Open ADCBUF driver */
89 ADCBufHandle = ADCBuf_open(0, &ADCBufparams);
90
91 return ADCBufHandle;
92 }

C:\ti\mmwave_sdk_03_03_00_02\packages\ti\demo\utils\mmwdemo_adconfig.c
9/3/2019 6:16:41 PM 7,644 bytes C,C++,C# Source ANSI UNIX
70 /* Fail NULL
71 ADCBuf_Handle MmDemo_ADCBufOpen(SOC_Handle socHandle)
72 {
73 ADCBuf_Params ADCBufparams;
74 ADCBuf_Handle ADCBufHandle = NULL;
75
76 /* Initialize the ADCBUF */
77 ADCBuf_init();
78
79 /* Start ADCBUF driver:
80 * Start ADCBUF driver:
81
82 /* ADCBUF Params initialize */
83 ADCBuf_Params_init(&ADCBufparams);
84 ADCBufparams.chirpThresholdPing = 1;
85 ADCBufparams.chirpThresholdPong = 1;
86 ADCBufparams.continuousMode = 0;
87 ADCBufparams.socHandle = socHandle;
88
89 /* Open ADCBUF driver */
90 ADCBufHandle = ADCBuf_open(0, &ADCBufparams);
91
92 return ADCBufHandle;
93 }
    
```

图 5-2. 针对 ADCBuf_open 的 SDK 3.3 API 变更

5.3 针对 CANFD_init 的 SDK 3.3 API 变更

MMWAVE-SDK 3.2.1 与 MMWAVE-SDK 3.3.0

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

```

C:\ti\mmwave_sdk_03_02_01_02\packages\ti\drivers\canfd\test\xwr68xx\main.c
8/7/2019 11:27:49 AM 79,834 bytes C,C++,C# Source ANSI UNIX
394
395 static int32_t mcanLoopbackTest()
396 {
397     CANFD_Handle canHandle;
398     CANFD_MsgObjHandle txMsgObjHandle;
399     CANFD_MsgObjHandle rxMsgObjHandle;
400     int32_t retVal = 0;
401     int32_t errCode = 0;
402     CANFD_OptionTLV optionTLV;
403     uint8_t value;
404     CANFD_MCANInitParams mcanCfgParams;
405     CANFD_MCANBitTimingParams mcanBitTimingParams;
406     CANFD_MCANMsgObjCfgParams txMsgObjectParams;
407     CANFD_MCANMsgObjCfgParams rxMsgObjectParams;
408     CANFD_MCANLoopbackCfgParams mcanloopbackParams;
409     CANFD_MCANMsgObjStats msgObjStats;
410
411     gTxDoneFlag = 0;
412
413     MCANAppInitParams (&mcanCfgParams);
414
415     /* Initialize the CANFD driver */
416     canHandle = CANFD_init(&mcanCfgParams, &errCode);
417     if (canHandle == NULL)
418     {
419         System_printf ("Error: CANFD Module Initialization failed [Error
420         return -1;
421     }
422
C:\ti\mmwave_sdk_03_03_00_02\packages\ti\drivers\canfd\test\xwr68xx\main.c
9/3/2019 6:16:41 PM 80,192 bytes C,C++,C# Source ANSI UNIX
396
397 static int32_t mcanLoopbackTest()
398 {
399     CANFD_Handle canHandle;
400     CANFD_MsgObjHandle txMsgObjHandle;
401     CANFD_MsgObjHandle rxMsgObjHandle;
402     int32_t retVal = 0;
403     int32_t errCode = 0;
404     CANFD_OptionTLV optionTLV;
405     uint8_t value;
406     CANFD_MCANInitParams mcanCfgParams;
407     CANFD_MCANBitTimingParams mcanBitTimingParams;
408     CANFD_MCANMsgObjCfgParams txMsgObjectParams;
409     CANFD_MCANMsgObjCfgParams rxMsgObjectParams;
410     CANFD_MCANLoopbackCfgParams mcanloopbackParams;
411     CANFD_MCANMsgObjStats msgObjStats;
412
413     gTxDoneFlag = 0;
414
415     MCANAppInitParams (&mcanCfgParams);
416
417     /* Initialize the CANFD driver */
418     canHandle = CANFD_init(gInstanceId, &mcanCfgParams, &errCode);
419     if (canHandle == NULL)
420     {
421         System_printf ("Error: CANFD Module Initialization failed [Error
422         return -1;
423     }
    
```

图 5-3. 针对 CANFD_init 的 SDK 3.3 API 变更

5.4 SDK 3.3 68xx 次级引导加载程序更新

MMWAVE-SDK 3.2.1 与 MMWAVE-SDK 3.3.0

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

```

C:\ti\mmwave_sdk_03_02_01_02\packages\ti\utils\sbl\platform\sbl_xwr68xx.c
8/7/2019 11:27:49 AM 19,865 bytes C,C++,C# Source ANSI UNIX
432
433     offset = (uint32_t)SBL_BSS_SHARED_MEM_TCMB_OFFSET;
434
435     else if ((sectionPtr + sectionLen) <= SBL_BSS_SECTION_END_ADDR
436     {
437         offset = (uint32_t)SBL_BSS_SHARED_MEM_OFFSET;
438     }
439     else
440     {
441         offset = 0U;
442         gSblMcb.errorStatus |= SBL_RPRC_PARSER_BSS_FILE_OFFSET_MIS
443     }
444
445     /* Configure the MPU settings for BSS section */
446     if (gSblMcb.bssMpuInit == 0)
447     {
448         gSblMcb.bssMpuInit = 1U;
449
450
451     /* Enable the regions */
452     SBL_mpuConfigBSS(true);
453 }
454
C:\ti\mmwave_sdk_03_03_00_02\packages\ti\utils\sbl\platform\sbl_xwr68xx.c
9/3/2019 6:16:41 PM 20,010 bytes C,C++,C# Source ANSI UNIX
433
434     offset = (uint32_t)SBL_BSS_SHARED_MEM_TCMB_OFFSET;
435
436     else if ((sectionPtr + sectionLen) <= SBL_BSS_SECTION_END_ADDRESS)
437     {
438         offset = (uint32_t)SBL_BSS_SHARED_MEM_OFFSET;
439     }
440     else
441     {
442         offset = 0U;
443         gSblMcb.errorStatus |= SBL_RPRC_PARSER_BSS_FILE_OFFSET_MISMATCH;
444     }
445
446     /* Configure the MPU settings for BSS section */
447     if (gSblMcb.bssClockMpuInit == 0)
448     {
449         gSblMcb.bssClockMpuInit = 1U;
450
451         /* ungate clock */
452         SOC_ungateClock(gSblMcb.socHandle, SOC_MODULE_BSS, &errCode);
453     }
454
455     /* Enable the regions */
456     SBL_mpuConfigBSS(true);
457 }
    
```

图 5-4. SDK 3.3 68xx 次级引导加载程序更新

5.5 SDK 3.3 16xx 与 68xx : 校准频率更新

MMWAVE-SDK 3.3.0 毫米波演示 (16xx 与 68xx)

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

图 5-5. SDK 3.3 16xx 与 68xx : 校准频率更新

5.6 SDK 3.3 16xx 与 68xx : SoC 定义更新

MMWAVE-SDK 3.3.0 毫米波演示 (16xx 与 68xx)

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

图 5-6. SDK 3.3 16xx 与 68xx : SoC 定义更新

5.9 SDK 3.4 物体检测 HWA DPC 范围 FFT 缩放

MMWAVE-SDK 3.3 与 MMWAVE-SDK 3.4

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

```

C:\ti\mmwave_sdk_03_03_00_03\packages\ti\demo\xwr64xx\mmw\main.c
9/16/2019 1:01:15 PM 132,920 bytes C,C++,C# Source ANSI UNIX
1667 staticCfg->numVirtualAntAzim = RfparserOutParams.numVirtualAntAzim;
1668 staticCfg->numVirtualAntElev = RfparserOutParams.numVirtualAntElev;
1669 staticCfg->numVirtualAntennas = RfparserOutParams.numVirtualAntennas;
1670 staticCfg->rangeStep = RfparserOutParams.rangeStep;
1671 for (i = 0; i < RfparserOutParams.numRxAntennas; i++)
1672 {

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
1705 staticCfg->numVirtualAntAzim = RfparserOutParams.numVirtualAntAzim;
1706 staticCfg->numVirtualAntElev = RfparserOutParams.numVirtualAntElev;
1707 staticCfg->numVirtualAntennas = RfparserOutParams.numVirtualAntennas;
1708 staticCfg->rangeStep = RfparserOutParams.rangeStep;
1709
1710 /* Current 64xx/68xx SOC has higher receive level as compared to 18xx and hence use
1711 * fftOutputDivShift to avoid overflow when converting from 24-bit to 16-bit
1712 * TODO: Future RadarSS firmware should be evaluated to assess if these settings
1713 */
1714 if (RfparserOutParams.numRangeBins >= 1022)
1715 {
1716     staticCfg->rangeFFTTuning.fftOutputDivShift = 1;
1717     /* scale only 2 stages */
1718     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 2;
1719 }
1720 else if (RfparserOutParams.numRangeBins==512)
1721 {
1722     staticCfg->rangeFFTTuning.fftOutputDivShift = 2;
1723     /* scale last stage */
1724     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 1;
1725 }
1726 else
1727 {
1728     staticCfg->rangeFFTTuning.fftOutputDivShift = 3;
1729     /* no scaling needed as ADC data is 16-bit and we have 8 bits to grow */
1730     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 0;
1731 }
1732
1733 for (i = 0; i < RfparserOutParams.numRxAntennas; i++)
1734 {
    
```

图 5-9. SDK 3.4 物体检测 DPC FFT 范围缩放配置

5.10 SDK 3.4 物体检测范围 HWA DPC 雷达立方体格式

MMWAVE-SDK 3.3 与 MMWAVE-SDK 3.4

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

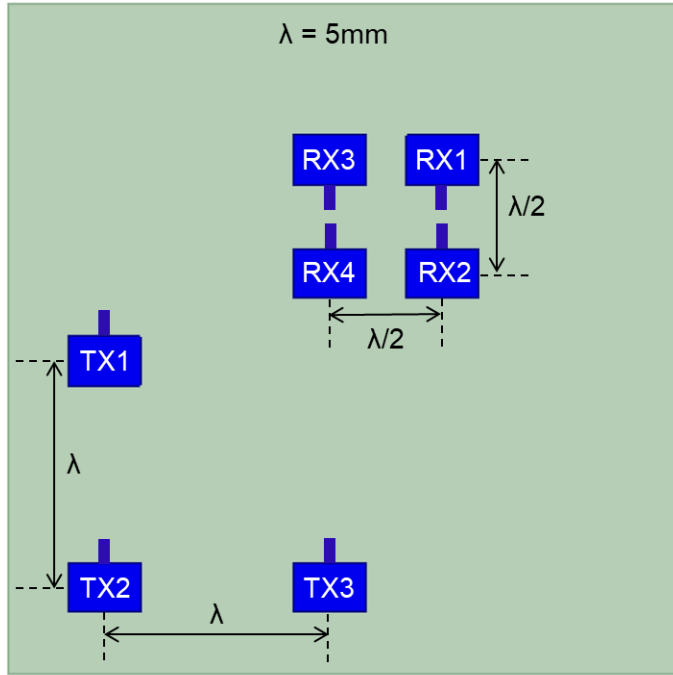
```

C:\ti\mmwave_sdk_03_03_00_03\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c
9/16/2019 1:01:15 PM 164,371 bytes C,C++,C# Source ANSI PC
2078 staticCfg->numRangeBins = RfparserOutParams.numRangeBins;
2079 staticCfg->numTxAntennas = RfparserOutParams.numTxAntennas;
2080 staticCfg->numVirtualAntennas = RfparserOutParams.numVirtualAntennas;
2081
2082
2083 /* Fill dynamic configuration for the sub-frame */
2084 objDetPreStartR4fCfg.dynCfg = subFrameCfg->objDetDynCfg.r4fDynCfg;

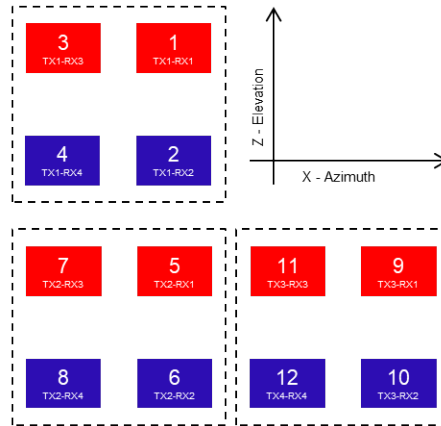
C:\ti\mmwave_sdk_03_04_00_03\packages\ti\demo\xwr68xx\mmw\mss\mss_main.c
3/30/2020 6:14:33 PM 167,431 bytes C,C++,C# Source ANSI PC
2110 staticCfg->numRangeBins = RfparserOutParams.numRangeBins;
2111 staticCfg->numTxAntennas = RfparserOutParams.numTxAntennas;
2112 staticCfg->numVirtualAntennas = RfparserOutParams.numVirtualAntennas;
2113
2114 /* Current 68xx SOC has higher receive level as compared to 18xx and hence use
2115 * fftOutputDivShift to avoid overflow when converting from 24-bit to 16-bit
2116 * TODO: Future RadarSS firmware should be evaluated to assess if these settings
2117 */
2118 if (RfparserOutParams.numRangeBins >= 1022)
2119 {
2120     staticCfg->rangeFFTTuning.fftOutputDivShift = 1;
2121     /* scale only 2 stages */
2122     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 2;
2123 }
2124 else if (RfparserOutParams.numRangeBins==512)
2125 {
2126     staticCfg->rangeFFTTuning.fftOutputDivShift = 2;
2127     /* scale last stage */
2128     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 1;
2129 }
2130 else
2131 {
2132     staticCfg->rangeFFTTuning.fftOutputDivShift = 3;
2133     /* no scaling needed as ADC data is 16-bit and we have 8 bits to grow */
2134     staticCfg->rangeFFTTuning.numLastButterflyStagesToScale = 0;
2135 }
2136
2137 /* objectdetection DSP DPC needs radacube in format DPIX_RADARCUBE_FORMAT_1 */
2138 staticCfg->radarCubeFormat = DPIX_RADARCUBE_FORMAT_1;
2139
2140 /* Fill dynamic configuration for the sub-frame */
2141 objDetPreStartR4fCfg.dynCfg = subFrameCfg->objDetDynCfg.r4fDynCfg;
    
```

图 5-10. SDK 3.4 物体检测范围 HWA DPC FFT 雷达立方体格式

5.11 xWR6843AoP ES1.0 天线几何结构



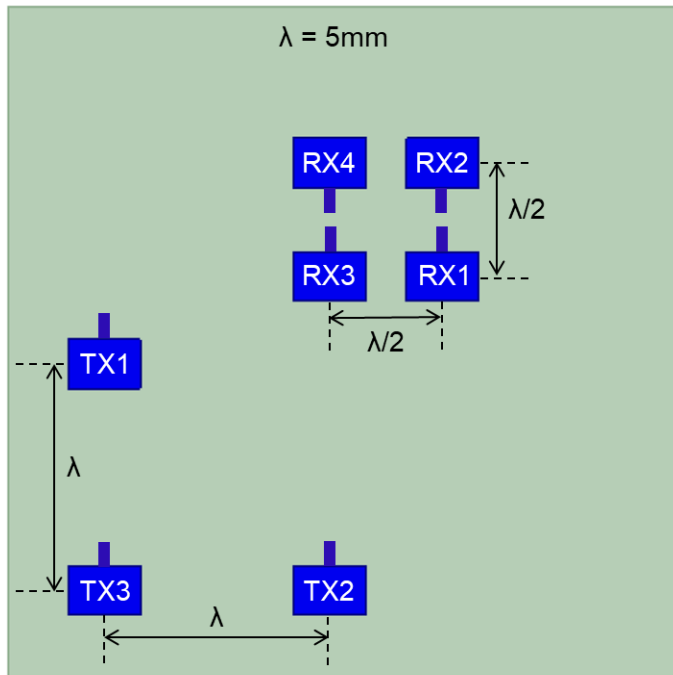
MIMO Virtual Antenna Array



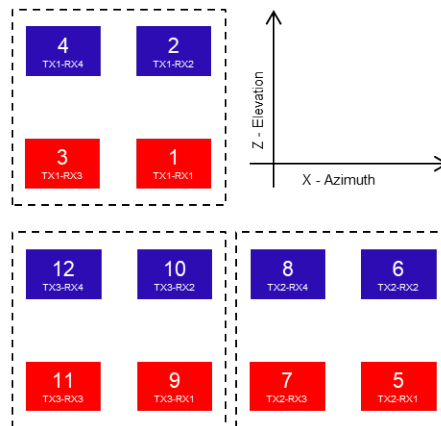
RX1 and RX3 are 180° out of phase with respect to RX2 and RX4. Because of this, a 180° phase inversion needs to be applied in software processing for the corresponding virtual RX channels (highlighted in Red)

图 5-11. xWR6843AoP ES1.0 天线几何结构

5.12 xWR6843AoP ES2.0 天线几何结构



MIMO Virtual Antenna Array



RX1 and RX3 are 180° out of phase with respect to RX2 and RX4, similar to ES1. Because of this, a 180° phase inversion needs to be applied in software processing for the corresponding virtual RX channels (highlighted in Red)

图 5-12. xWR6843AoP ES2.0 天线几何结构

5.13 xWR6843AoP ES2.0 天线几何结构代码更新

MMWAVE-SDK 3.2.0.6 与 MMWAVE-SDK 3.4

文件 : 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 3/31/2019 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 };
    
```

图 5-13. SDK 3.2.0.6 与 SDK 3.4 : xWR6843AoP ES2.0 的天线几何结构更新

5.14 毫米波演示中的天线几何结构用法

文件 : 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 /* Copy antenna geometry definition */
1563 if defined(XWR68XX_AOP_ANTENNA_PATTERN)
1564     extern ANTDEF_AntGeometry gAntDef_IWR6843AOP;
1565     dataPathObj->objDetCommonCfg.preStartCommonCfg.antDef = gAntDef_IWR6843AOP;
1566 else
1567     extern ANTDEF_AntGeometry gAntDef_default;
1568     dataPathObj->objDetCommonCfg.preStartCommonCfg.antDef = gAntDef_default;
1569 endif
1570
1571 /* DPC pre-start common config */
1572 errCode = DPM_ioctl (dataPathObj->objDetDpmHandle,
1573                     DPC_OBJDET_IOCTL_STATIC_PRE_START_COMMON_CFG,
1574                     &dataPathObj->objDetCommonCfg.preStartCommonCfg,
1575                     sizeof (DPC_ObjectDetection_PreStartCommonCfg));
1576
1577 if (errCode < 0)
1578 {
1579     System_printf ("Error: Unable to send DPC_OBJDET_IOCTL_STATIC_PRE_START_COMMON_CFG [Error:%d]\n", errCode);
1580     goto exit;
1581 }
    
```

图 5-14. 毫米波演示中的天线几何结构用法

5.15 xWR6843AoP ES2.0 RX 通道相位补偿

MMWAVE-SDK 3.2.0.6 与 MMWAVE-SDK 3.4

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

| File Path | Timestamp | Size | Encoding | Line | Code |
|---|--------------------------------|---|-----------|------|------|
| C:\ti\mmwave_sdk_03_02_00_06_AOP\packages\ti\demo\xwr64xx\mmw\profiles\profile_3d_aop.cfg | 5/31/2019 9:58:33 AM | 2,717 bytes | <default> | ANSI | UNIX |
| C:\ti\mmwave_sdk_03_04_00_03\packages\ti\demo\xwr64xx\mmw\profiles\profile_3d_aop.cfg | 3/30/2020 6:14:33 PM | 2,717 bytes | <default> | ANSI | UNIX |
| 37 | flushCfg | | | | |
| 38 | dfeDataOutputMode | 1 | | | |
| 39 | channelCfg | 15 7 0 | | | |
| 40 | adcCfg | 2 1 | | | |
| 41 | adcbufCfg | -1 0 1 1 1 | | | |
| 42 | lowPower | 0 0 | | | |
| 43 | profileCfg | 0 60.25 7 3 24 0 0 156 1 256 12500 0 0 30 | | | |
| 44 | chirpCfg | 0 0 0 0 0 0 1 | | | |
| 45 | chirpCfg | 1 1 0 0 0 0 2 | | | |
| 46 | chirpCfg | 2 2 0 0 0 0 4 | | | |
| 47 | frameCfg | 0 2 32 0 100 1 0 | | | |
| 48 | guiMonitor | -1 1 1 1 0 0 1 | | | |
| 49 | cfarCfg | -1 0 2 8 4 3 0 15.0 0 | | | |
| 50 | cfarCfg | -1 1 0 4 2 3 1 15.0 0 | | | |
| 51 | multiObjBeamForming | -1 1 0.5 | | | |
| 52 | calibDcRangeSig | -1 0 -5 8 256 | | | |
| 53 | clutterRemoval | -1 0 | | | |
| 54 | | | | | |
| 55 | compRangeBiasAndRxChanPhase | 0.0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 | | | |
| 56 | measureRangeBiasAndRxChanPhase | 0 1. 0.2 | | | |
| 57 | | | | | |
| 58 | aoaFovCfg | -1 -90 90 -90 90 | | | |
| 59 | cfarFovCfg | -1 0 0.25 15 | | | |
| 60 | cfarFovCfg | -1 1 -13.39 13.39 | | | |
| 61 | extendedMaxVelocity | -1 0 | | | |
| 62 | | | | | |
| 63 | CQRxSatMonitor | 0 3 4 63 0 | | | |
| 64 | CQSigImgMonitor | 0 127 4 | | | |
| 65 | analogMonitor | 0 0 | | | |
| 66 | lvdsStreamCfg | -1 0 0 0 | | | |
| 67 | sensorStart | | | | |

图 5-15. SDK 3.2.0.6 与 SDK 3.4 : RX 通道相位补偿

6 参考文献

- 德州仪器 (TI) : [IWR1642 器件勘误表](#)
- 德州仪器 (TI) : [AWR1642 器件勘误表](#)
- 德州仪器 (TI) : [IWR1843 器件勘误表](#)
- 德州仪器 (TI) : [AWR1843 器件勘误表](#)
- 德州仪器 (TI) : [IWR6843 器件勘误表](#)
- 德州仪器 (TI) : [AWR6843 器件勘误表](#)
- [xWR1642BOOST 布局和设计文件](#)
- [xWR6843AOPEVM 原理图、装配和物料清单](#)
- 德州仪器 (TI) : [xWR1642 EVM \(xWR1642BOOST\) 单芯片毫米波传感解决方案用户指南](#)
- 德州仪器 (TI) : [MMWAVEICBOOST 和天线模块用户指南](#)
- [用于原理图审阅、布局审阅、启动/唤醒的 xWR6843 检查清单](#)
- [xWR1843BOOST 硬件文件](#)
- 德州仪器 (TI) : [xWR1843 评估模块 \(xWR1843BOOST\) 单芯片毫米波传感解决方案用户指南](#)
- [xWR6843 产品页面](#) (器件数据表、器件勘误表)
- [xWR6843AoP 产品页面](#) (器件数据表、器件勘误表)
- [xWR1843 产品页面](#) (器件数据表、器件勘误表)
- [xWR1642 产品页面](#) (器件数据表、器件勘误表)
- 德州仪器 (TI) : [IWR14xx/16xx/18xx/68xx 工业雷达系列技术参考手册](#)
- [MMWAVE-SDK 产品页面](#)
- [MMWAVE-SDK 3.3.0 下载页面](#) (版本说明、用户指南和 SDK 下载)
- [MMWAVE-SDK 3.4.0 下载页面](#) (版本说明、用户指南和 SDK 下载)

7 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

| Changes from Revision B (May 2020) to Revision C (October 2022) | Page |
|---|-------------|
| • 更新了整个文档中的表格、图和交叉参考的编号格式..... | 3 |
| <hr/> | |
| Changes from Revision * (November 2019) to Revision A (May 2020) | Page |
| • 对“器件比较”主题进行了更新..... | 4 |
| • 添加了新的 节 2.1.2 。..... | 5 |
| • 对 节 2.1.2.1 进行了更新..... | 5 |
| • 添加了新的 节 2.1.3 。..... | 5 |
| • 更新了“硬件从 xWR6843AoP ES1.0 更改为 xWR6843AoP ES2.0”主题..... | 9 |
| • 更新了“软件从 xWR6843AoP ES1.0 迁移到 xWR6843AoP ES2.0”主题..... | 11 |

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司