Technical Article

DDMA 波形配置在 AWR1843 雷达传感器上的实现详解

Ken Zhang

AWR1843 作为第一款基于 RF-CMOS 工艺的毫米波雷达 SOC，在单芯片上集成了 3TX /4Rx 射频电路，PLL，AD 模数转换器，ARM R4F，HWA 以及 C674 DSP 处理器，极大的降低了毫米波雷达的开发难度和量产成本，是当前最为普及的毫米波雷达芯片之一。本文介绍了 DDMA（多普勒分集发射）波形在 AWR1843 上的配置和实现，与 TDMA(时间分集发射)波形相比，借助多根天线同时发送的优势，增加了信号发送能量，从而提升 SNR 和探测距离。用户可以通过本文了解 AWR1843 的射频配置背景知识以及 DDMA 的射频配置示例。

DDMA 原理简介

TDMA 由于其实现方式简单的优势，在实际的开发中被大部分客户所采用，这种实现方式在每个 chirp 发射时，有且仅有单个 Tx 通道进行发送，其他 Tx 通道的发射功率并未被利用。DDMA 则会同时使能所有 Tx 通道，并在每个 Tx 通道附加特定的频率（相位）偏移，从而能够在 Doppler 域上区分不同 Tx 通道的信号。

在 AWR1843 实现 DDMA 波形，可以为雷达系统的发射增益提升 10log(3) dB，有助于增加雷达系统的最大探测距离，但 DDMA 也会引入最大不模糊速度区间下降的问题，为此 TI 在 AWR2944 SDK 中引入了基于 Empty-band（空带）的解模糊策略，这种策略同样也适用于 AWR1843。

DDMA 的相位调制

通常对于 3 根发射天线的 DDMA 调制，如果使用 Empty-band 的调制策略，则需要将整体的 2D doppler 热图平均分成若干个宽度相同的 band，到 AWR1843 实际应用中，总的 chirp 数量（Doppler 热图的宽度）多为 4 的整数倍，因此推荐的 Empty-band 数量为 1，加上 3 根发射天线对应的 band 数量 Nt，总的 band 数量 N 为 4。在这个前提下，我们便可以根据公式（1）得到每个发射天线的理论发射相位 ωk，并将结果汇总在表 1 中。

$$\omega_k = \frac{2\pi(k-1)}{N}, \quad k = 1, \ldots N_t$$

表 1.

TX. No	调制相位 ω
Tx1	0
Tx2	π /2
Tx3	π

由此我们可以得出单帧内每个 chirp 中 Tx1 至 Tx3 的每个发射天线的相位，汇总在表 2 中

表 2.

	Chirp0	Chirp1	Chirp2	Chirp3	Chirp4	Chirp5	…	
Tx1 相位	0	0	0	0	0	0	…	
Tx2 相位	0	π /2	π	3 π /2	0	π /2	…	
Tx3 相位	0	π	0	π	0	π	…	

观察上表我们发现每隔 4 个 chirp， chirp 的发射相位会和和前四个 chirp 完全一致，即 chirp 的相位在以 4 为单位进行循环. 因此在实际 AWR1843 的配置中，仅需要配置 4 个 chirp 及其发射相位即可，并同样依次循环，构建单帧中的数百个 chirp 序列。

Chirp (Profile)，Loop，Frame

AWR1843 的射频配置中，一次收发的最基本的单元为 chirp，用户会首先创建若干个 profile 模板，设定采样率，起始频率，扫频时长等射频收发的各项基本参数，随后在 profile 的基础上，微调或直接继承 profile 各项参数，并将微调后最终的 chirp 参数存入芯片内部的 RAM 中，至多可存储 512 组 chirp 参数(Index 0 至 Index 511), 在上述 Index 0 至 Index 511 的参数集中，选取一段连续的参数集构成一次循环单元 Loop， 将 Loop 连续循环后便形成了最终的 Frame. 用户可参考 TI 官方文档 Programming Chirp Parameters in TI Radar Devices (Rev. A) 以获取更多背景知识。

AWR1843 LDO bypass

DDMA 的本质是同时使能多根发射天线并给每根天线赋予特定的相位，提升发射能量的同时也会额外增加射频电源轨的功耗，AWR1843 内部的 LDO 的功率有限，因此需要软件调用 ar1.RfLdoBypassConfig(0x3)以使能"LDO bypass"模式，调用 ar1.SetMiscConfig(1)以使能 per chirp phase shift。 在硬件连接上，需要将 VOUT_PA 连接至 RF2 电源轨.

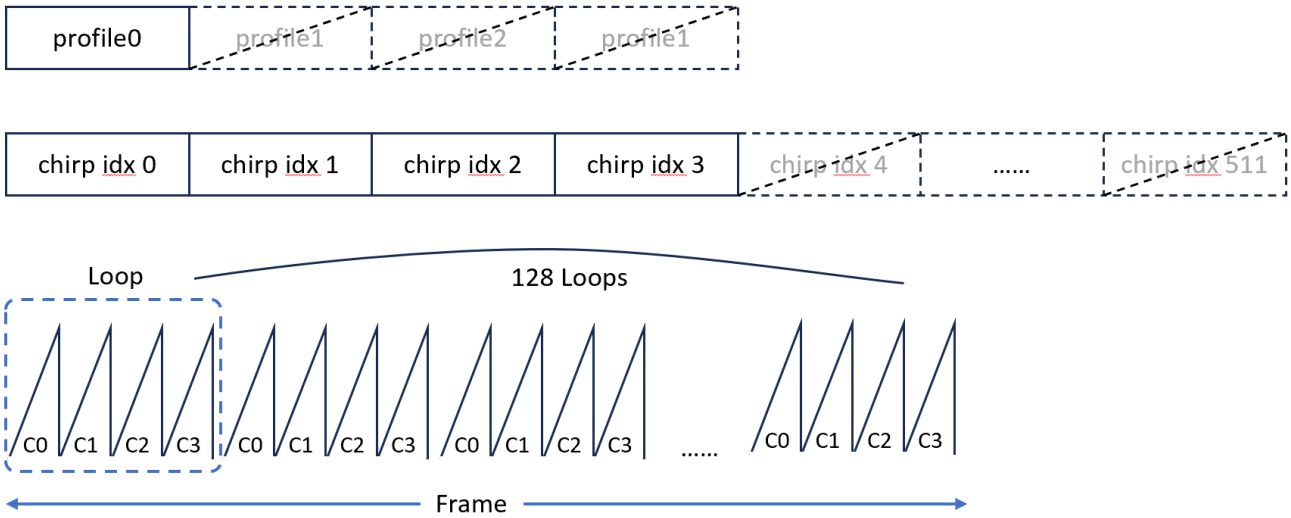
mmwave studio 的 DDMA 设置示例

在 studio 中，使用 lua 脚本文件配置 DDMA 的一个示例如下, 用户可将该文件拷贝至 C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\Scripts 目录下，修改 ar1.Connect 中的串口编号后执行.

```
 ar1.FullReset()
 ar1.SOPControl(2)
ar1.Connect(56,921600,1000)
ar1.Calling_IsConnected()
ar1.SelectChipVersion("AR1642")
ar1.SelectChipVersion("AR1642")
ar1.deviceVariantSelection("XWR1843")
ar1.frequencyBandSelection("77G")
ar1.SelectChipVersion("XWR1843")
ar1.DownloadBSSFw("C:\\ti\\mmwave_studio_02_01_01_00\\rf_eval_firmware\\radarss\
\xwr18xx_radarss.bin")
ar1.DownloadMSSFw("C:\\ti\\mmwave_studio_02_01_01_00\\rf_eval_firmware\\masterss\
\xwr18xx_masterss.bin")
ar1.PowerOn(0, 1000, 0, 0)
ar1.SelectChipVersion("AR1642")
ar1.SelectChipVersion("XWR1843")
ar1.RfEnable()
ar1.ChanNAdcConfig(1, 1, 1, 1, 1, 1, 1, 2, 1, 0)
ar1.RfLdoBypassConfig(0x3)
ar1.LPModConfig(0, 0)
ar1.SetMiscConfig(1)
ar1.RfInit()
ar1.DataPathConfig(513, 1216644097, 0)
ar1.LvdsClkConfig(1, 1)
ar1.LVDSLaneConfig(0, 1, 1, 0, 0, 1, 0, 0)
ar1.ProfileConfig(0, 77, 100, 6, 60, 0, 0, 0, 0, 0, 0, 35.003, 0, 256, 10000, 0, 0, 30)
ar1.ChirpConfig(0, 3, 0, 0, 0, 0, 0, 1, 1, 1)
ar1.SetPerChirpPhaseShifterConfig(0, 0, 0, 0, 0)
ar1.SetPerChirpPhaseShifterConfig(1, 1, 0, 16, 32)
ar1.SetPerChirpPhaseShifterConfig(2, 2, 0, 32, 0)
ar1.SetPerChirpPhaseShifterConfig(3, 3, 0, 48, 32)
ar1.FrameConfig(0, 3, 1, 128, 200, 0, 0, 1)
ar1.CaptureCardConfig_StartRecord("C:\\ti\\mmwave_studio_02_01_01_00\\mmWaveStudio\\PostProc\
\adc_data.bin", 1)
ar1.StartFrame()
```

如图一所示，ar1.ProfileConfig 添加了 profile 0, 后续的所有 chirp 均继承和使用 profile 0 的参数，profile 0 的三个 Tx 的发射相位均初始化为 0. ar1.ChirpConfig 则选取 profile 0 的参数作为模板，使能全部的三个发射天线后，填充至 chirp Index 0 至 Index 3 的 chirp RAM 中， 共计存储 4 组 chirp 参数。接着调用 4 次 ar1.SetPerChirpPhaseShifterConfig，按照表 2 的结果修改每组 chirp 参数的发射相位(每个 LSB 代表 5.625°)。最

后调用 ar1.FrameConfig，选取 chirp Index 0 至 Index 3 作为一个 loop，并将这 4 组 chirp 参数循环 128 次，最终一帧中包含 512 个 chirp。

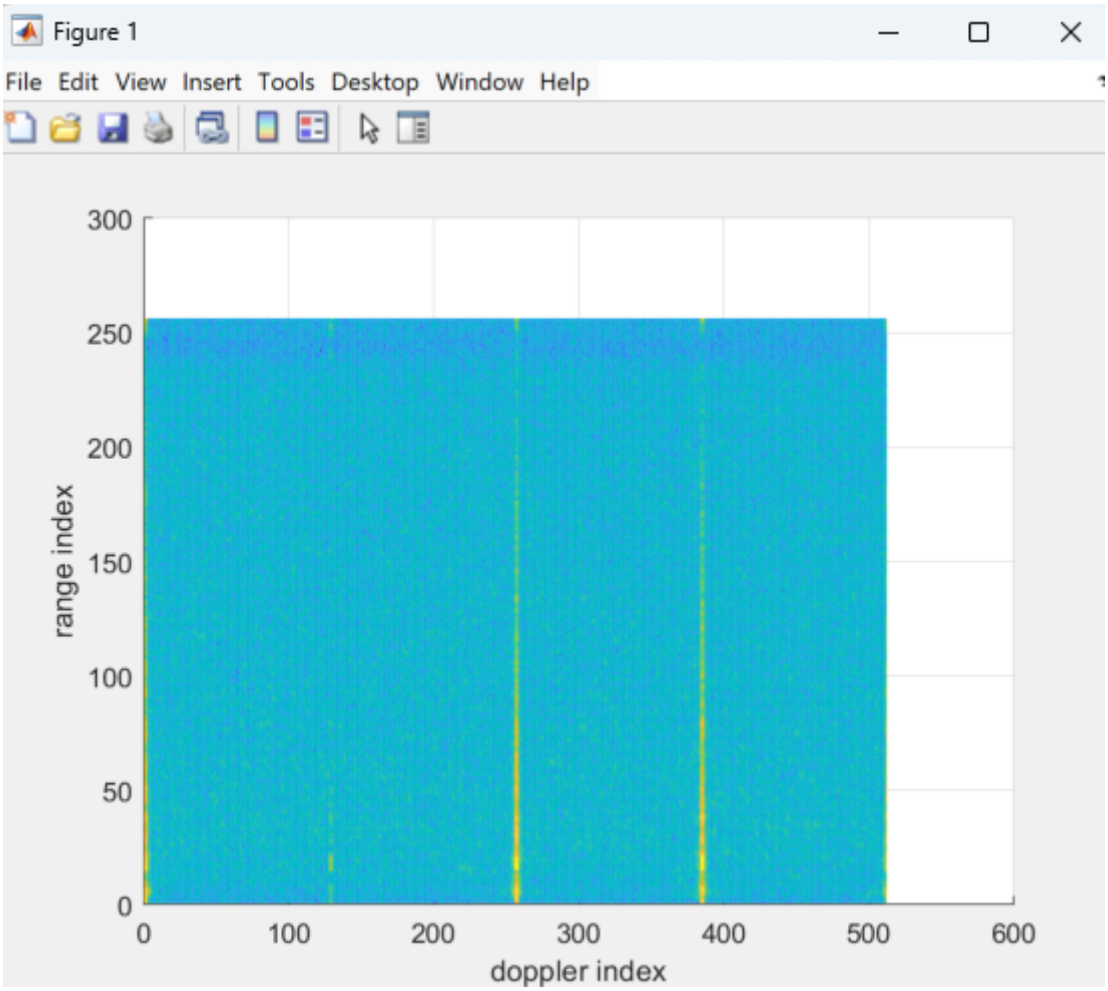


mmwave studio 内嵌的 post proc 后处理模块不支持 DDMA，建议用户自行编写，或参考如下 matlab 后处理程序对采集的 adc_data.bin 文件进行后处理，在对 range 维以及 doppler 维分别添加 hann 窗，并进行 2D FFT 处理后，采集原始数据的 R-D heatmap 如下图所示。可以看到 R-D heatmap 被均匀划分为 4 个 band，包含三个能量较强的 band 以及用于解调的 empty-band。

DDMA 的验证与后处理

综上，我们完成了 DDMA 在 AWR1843 上射频实现的案例。

```
clc;clear;close all;
RxNum = 4;
sampleNumPerChirp = 256;
chirpNumPerFrame = 512;
%% mmWave Studio ADC 数据排布为格式为：[samples,Rxs,chirps,frames]
data_input =
readDCA1000("C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\PostProc\adc_data.bin",RxNum,
sampleNumPerChirp,chirpNumPerFrame);
frameNum = size(data_input,4);
win_range = reshape(hann(sampleNumPerChirp).',[1,sampleNumPerChirp,1,1]);
win_range = repmat(win_range,[RxNum,1,chirpNumPerFrame,frameNum]);
win_doppler = reshape(hann(chirpNumPerFrame).',[1,1,chirpNumPerFrame,1]);
win_doppler = repmat(win_doppler,[RxNum,sampleNumPerChirp,1,frameNum]);
data_range = fft(win_range.*data_input,[],2);
data_doppler = fft(win_doppler.*data_range, [], 3);
figure;
mesh(20*log10(abs(squeeze(data_doppler(1,:,:,1)))));xlabel('doppler index');ylabel('range index');
return;
```



参考资料:

1. Programming Chirp Parameters in TI Radar Devices (Rev. A)
2. mmwave Radar Device ADC Raw Data Capture (Rev. B)
3. 基于 AWR2944 的汽车雷达 DDMA 波形的原理和实现

重要通知和免责声明

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出任何明示或暗示的担保，包括但不限于对适销性、与某特定用途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安全、安保法规或其他要求。

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 TI 销售条款)、TI 通用质量指南 或 ti.com 上其他适用条款或 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。 除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

最后更新日期：2025 年 10 月