Technical Article

Detailed Implementation of DDMA Waveform Configuration on the AWR1843 Radar Sensor

TEXAS INSTRUMENTS

Ken Zhang

This document was translated from a simplified Chinese source. (ZHCTA17)

As the first millimeter-wave radar SoC based on RF-CMOS technology, the AWR1843 integrates on a single chip 3TX/4RX RF circuits, PLL, AD converters, an ARM R4F core, HWA, and a C674 DSP processor. This significantly reduces the development complexity and mass-production cost of millimeter-wave radar systems, making it one of the most widely adopted millimeter-wave radar chips available today. This article describes the configuration and implementation of DDMA (Doppler Division Multiple Access) waveforms on the AWR1843. Compared with TDMA (Time Division Multiple Access) waveforms, DDMA leverages simultaneous transmission from multiple antennas to increase transmit energy, thereby improving SNR and extending detection range. Readers can use this document to understand the background of RF configuration on the AWR1843 as well as a specific DDMA RF configuration example.

Brief Introduction to DDMA Principles

TDMA is adopted by most customers in practical development due to its straightforward implementation. In this approach, only a single Tx channel is active during each chirp transmission, while the transmit power of other Tx channels remains unused. DDMA, by contrast, enables all Tx channels simultaneously and applies a specific frequency (phase) offset to each Tx channel, allowing signals from different Tx channels to be distinguished in the Doppler domain.

Implementing DDMA waveforms on the AWR1843 can provide a transmit gain improvement of 10log(3) dB for the radar system, which helps increase the maximum detection range. However, DDMA also introduces the issue of reduced maximum un-ambiguous velocity range. To address this, TI introduced an empty-band based de-ambiguation strategy in the AWR2944 SDK, which is also applicable to the AWR1843.

Phase Modulation in DDMA

For DDMA modulation with three transmit antennas, if the empty-band modulation strategy is adopted, the entire 2D Doppler heatmap is evenly divided into a number of bands of equal width. In practical applications on the AWR1843, the total number of chirps (i.e., the Doppler heatmap width) is typically an integer multiple of 4. Therefore, the recommended number of empty bands is 1. With the three bands corresponding to the three transmit antennas ($N_t = 3$), the total number of bands N becomes 4. Under this condition, the theoretical transmit phase ω_k for each transmit antenna can be derived from Equation (1), and the results are summarized in Table 1.

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

Table 1.

Tx No.	Modulation Phase ω
Tx1	0
Tx2	π/2
Tx3	π

From this, we can obtain the phase for each transmit antenna (Tx1 to Tx3) in every chirp within a single frame, as summarized in Table 2

Table 2.

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

Observing the table above, we notice that every 4 chirps, the transmission phases repeat identically to those of the first 4 chirps—i.e., the chirp phase cycles with a period of 4. Therefore, in actual AWR1843 configuration, it is sufficient to configure only 4 chirps and their transmit phases, and then repeat this cycle sequentially to construct the hundreds of chirps in a single frame.

Chirp (Profile), Loop, and Frame

In the RF configuration of the AWR1843, the most basic transmit-receive unit is the chirp. The user first creates one or more profile templates, specifying fundamental RF parameters such as sampling rate, start frequency, and ramp duration. Based on these profiles, the user can fine-tune or directly inherit the profile parameters to define the final chirp parameters, which are then stored in the chip's internal RAM. Up to 512 sets of chirp parameters (Index 0 to Index 511) can be stored. From this parameter set (Index 0 to Index 511), a contiguous subset is selected to form a single loop, and this loop is repeated continuously to form the final frame. For more background information, users are referred to the TI official document Programming Chirp Parameters in TI Radar Devices (Rev. A).

AWR1843 LDO Bypass

The essence of DDMA is to enable multiple transmit antennas simultaneously with specific phase assignments for each antenna. While this increases transmit energy, it also adds extra power demand on the RF power supply rail. The internal LDO of the AWR1843 has limited power capability. Therefore, software must call ar1.RfLdoBypassConfig(0x3) to enable "LDO bypass" mode and call ar1.SetMiscConfig(1) to enable per-chirp phase shifting. In terms of hardware connections, VOUT_PA must be connected to the RF2 power rail.

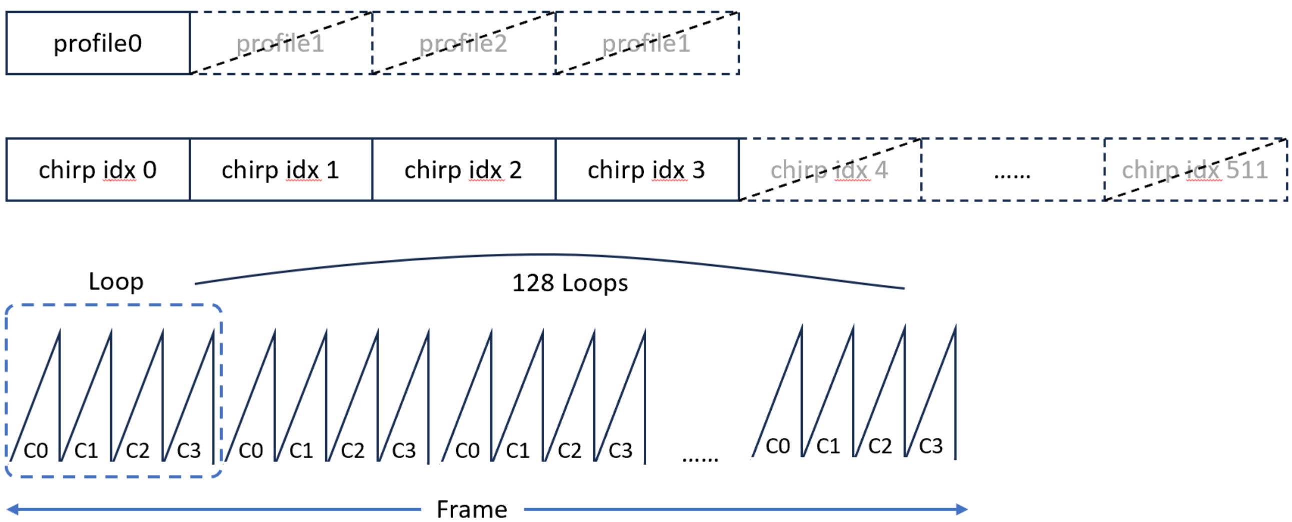
DDMA Configuration Example in mmWave Studio

An example of DDMA configuration using a Lua script file in mmWave Studio is shown below. Users can copy this file to the directory C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\Scripts, modify the port numbers in ar1.Connect as needed, and then execute it.

```
 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()
```

As shown in Figure 1, ar1.ProfileConfig adds profile 0, and all subsequent chirps inherit and use the parameters of profile 0, whose Tx phase for all three Tx antennas is initialized to 0. ar1.ChirpConfig uses the profile 0 parameters as a template, enables all three transmit antennas, and writes them into chirp RAM at chirp Index 0 through Index 3, storing a total of 4 sets of chirp parameters. Then ar1.SetPerChirpPhaseShifterConfig is called 4 times to modify the transmit phase of each chirp parameter set according to the results in Table 2 (each LSB corresponds to 5.625°). Finally, ar1.FrameConfig is called to select chirp Index 0 through Index 3 as a loop, and these 4 sets of chirp parameters are repeated 128 times, resulting in a frame containing 512 chirps.



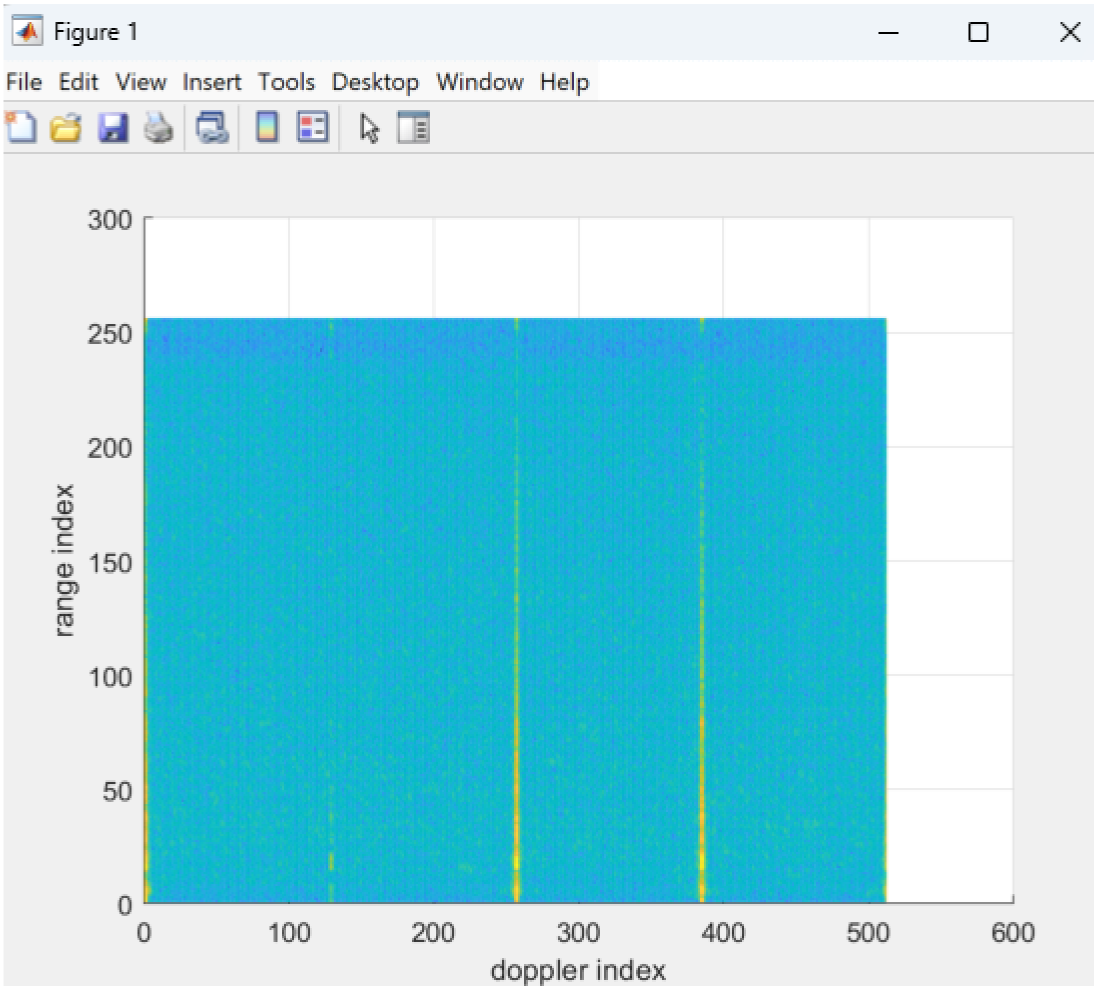
The built-in post-processing module in mmWave Studio does not support DDMA. Users are advised to write their own post-processing code or refer to the following MATLAB post-processing program to process the acquired adc_data.bin file. After applying Hann windows along both the range and Doppler dimensions and performing a 2D FFT, the resulting R-D heatmap of the captured raw data is shown below. It can be observed that the R-D heatmap is evenly divided into 4 bands, containing three bands with relatively strong energy and one empty band used for de-ambiguation.

DDMA Verification and Post-Processing

In summary, we have completed a case study of DDMA RF implementation on the AWR1843.

```
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;
```



References:

1. Programming Chirp Parameters in TI Radar Devices (Rev. A)
2. mmwave Radar Device ADC Raw Data Capture (Rev. B)
3. Principles and Implementation of Automotive Radar DDMA Waveform Based on AWR2944

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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 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, TI’s General Quality Guidelines, 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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