

Andy Yin

Central mmWave FAE

摘要

AWR2188 作为 TI 最新一代高性能支持 8 发 8 收的 MMIC (单片微波集成电路)，其波形配置方式与上一代 MMIC AWR2243 有所差异。本文描述 AWR2188 与 AWR2243 的射频及波形配置差异，并就 AWR2188 的波形配置给出详细介绍，便于客户能快速配置生成相应的波形。

内容

1 前言.....	2
2 FMCW 波形介绍.....	4
3 AWR2188 FMCW 波形配置.....	6
3.1 配置关系.....	6
3.2 帧配置.....	7
3.3 子帧配置.....	8
3.4 chirp 配置.....	9
3.5 profile 配置.....	10
4 总结.....	11
5 参考文献.....	11

插图清单

图 1-1. AWR2188 芯片框图.....	2
图 2-1. 典型 FMCW chirp.....	4
图 2-2. 典型 FMCW 帧结构.....	4
图 2-3. FMCW 高级帧结构.....	5
图 3-1. 波形配置存储位置及关联.....	7
图 3-2. 配置 API 流程.....	7
图 3-3. 帧配置例程.....	8
图 3-4. 子帧中 chirp 参数 LUT 配置使用例程.....	8
图 3-5. 子帧配置参考代码.....	9
图 3-6. image10_chirp 参数 LUT 配置参考代码.....	10
图 3-7. profile 配置参考代码.....	10

表格清单

表 1-1. AWR2188 和 AWR2243 的参数对比.....	3
表 3-1. AWR2188 与 AWR2243 波形配置对比.....	6
表 3-2. 子帧配置中 chirp 参数 LUT 配置说明.....	8

商标

所有商标均为其各自所有者的财产。

1 前言

为应对市场对高性能成像雷达的需求，德州仪器(TI)推出了新一代高性能毫米波雷达 AWR2188。如 图 1-1 所示，AWR2188 是一款基于 RF-CMOS 工艺的单芯片微波集成电路(MMIC)，内部包括以下子系统：

- 射频模拟子系统：射频及模拟相关的电路，包括合成器(synthesizer)，功率放大器(PA)，低噪声放大器(LNA)，混频器(mixer)，模数转换器(ADC)，晶体振荡器(VCO)，温度传感器等电路。单芯片支持 8 个发射通道 8 个接收通道同时工作。
- 雷达前端控制器：射频控制器的固件运行于内置的 cortex R4F，用于控制射频前端波形参数的配置、校准、监控，并对采样信号进行抽取滤波。
- 主机接口控制模块：内置 cortex M4F，运行 TI 提供的固件，用于配置 SPI 或者 I2C 接口与外部处理器进行通信，并配置 CSI 接口，用于 ADC 数据的传输。

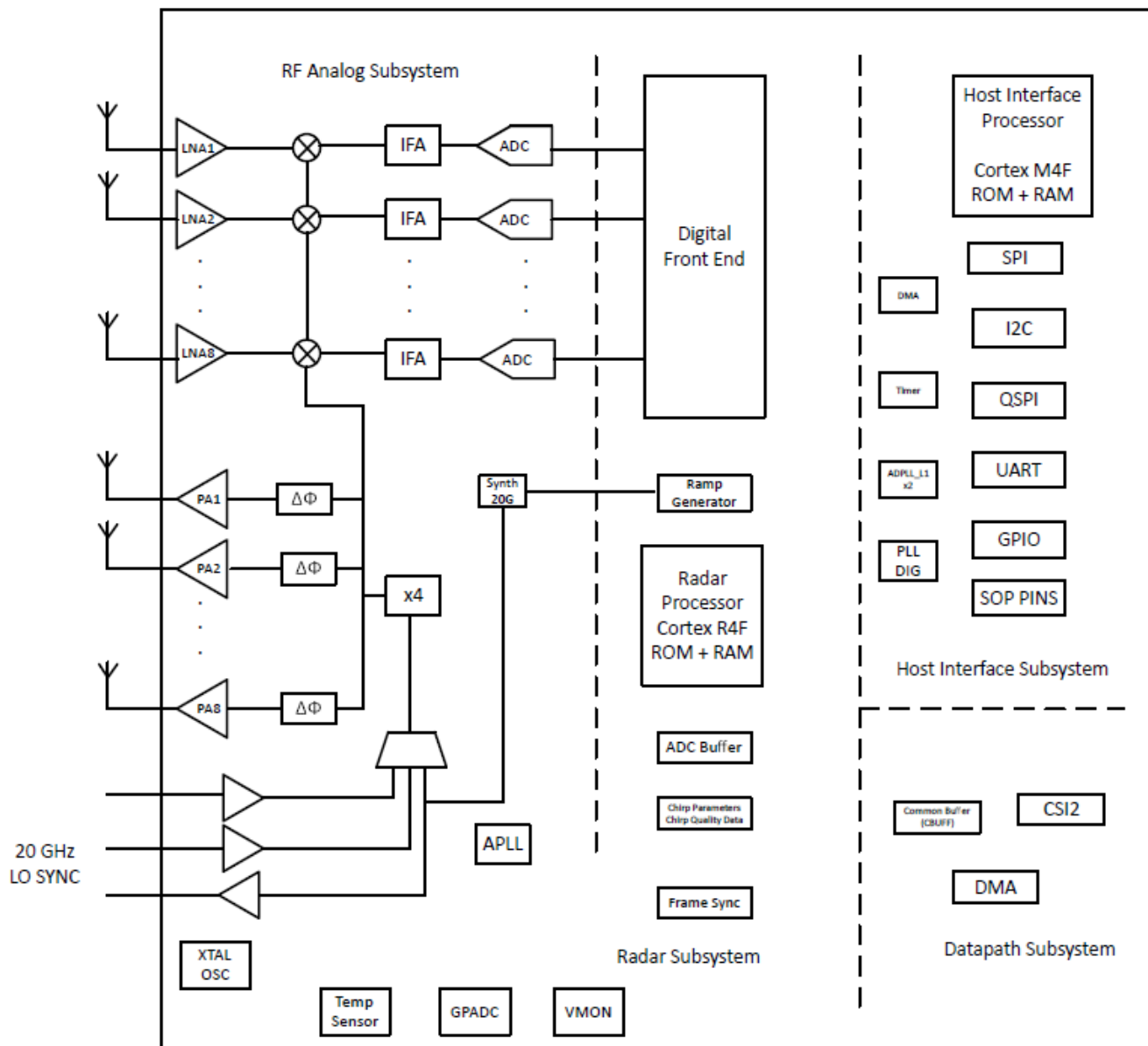


图 1-1. AWR2188 芯片框图

AWR2188 射频性能相比上一代 MMIC AWR2243 也有提升，具体如下表 1-1 所示。

表 1-1. AWR2188 和 AWR2243 的参数对比

参数	AWR2243	AWR2188
收发天线数	3T4R	8T8R
典型发射功率	13dBm	13.5dBm
典型接收噪声系数	13db	10db
移相器位宽	6bit	8bit
移相器精度	5 度	3 度
中频带宽	20MHz	30MHz
ADC 采样率	45Msps	66.67Msps
ADC 分辨率	12bit	16bit
CSI2 速率	2.4Gbps	6.8Gbps

从表 1-1 可见，AWR2188 单芯片可以支持 8T8R，发射功率、噪声系数、采样率等指标的提高，可以大大提高探测性能。单芯片物理天线数及射频性能均优于 2 片 AWR2243 级联，可以简化成像雷达系统硬件设计，降低成本，提升探测能力。因此 AWR2188 方案的推出，将给成像毫米波雷达市场带来一次技术的飞跃，加速成像雷达的市场渗透率。

为了满足不同客户系统的设计，提供更强灵活性，AWR2188 对调频连续波(FMCW)的配置与 AWR2243 有所区别，这势必给客户的软件开发带来新的挑战。本文将对 AWR2188 的 FMCW 波形配置进行详细介绍，加速客户波形配置相关软件的开发。

2 FMCW 波形介绍

FMCW(Frequency Modulated Continuous Wave)调频连续波，是一种通过发射频率随时间线性变化的连续波信号，并利用回波与发射信号的频率差来计算目标距离，结合多普勒效应可以计算速度。该技术具有硬件处理相对简单、成本低、高距离分辨率、抗干扰能力强以及体积小、功耗低等优势。FMCW 技术广泛应用于汽车雷达及高级驾驶辅助系统 (ADAS)，用于实现盲点检测、自动巡航控制等功能。FMCW 的波形最小单元是 chirp，如下图 2-1 所示，一个典型的 chirp 由 Idle time，TX start time，ADC start time，ADC sampling rate，Freq slope 等参数描述。

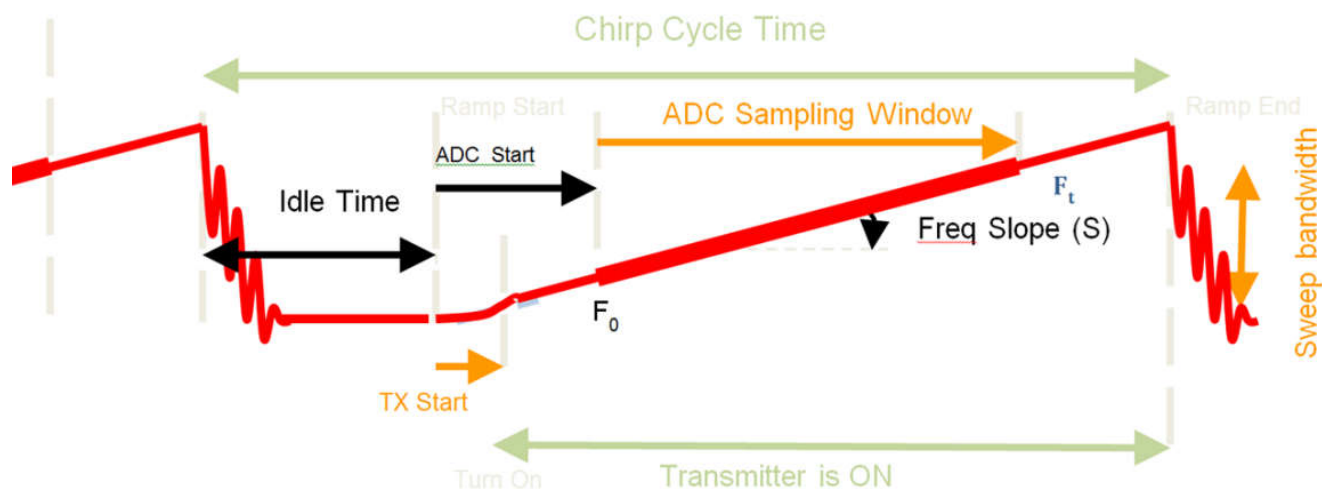


图 2-1. 典型 FMCW chirp

如下图 2-2 是一个典型的 FMCW 帧结构，由多个 chirp 及帧间 idle time 组成。

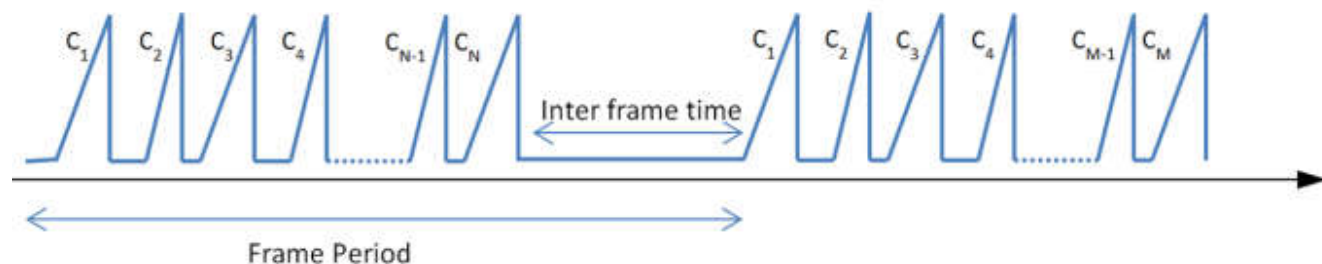


图 2-2. 典型 FMCW 帧结构

为了在一个帧里支持多种检测模式，引入了高级帧结构，如下图 2-3 所示 FMCW 高级帧结构。相比图 2-2 典型 FMCW 帧结构，引入了子帧(subframe)及 burst 的概念，一个高级帧由多个子帧构成，一个子帧可以由多个 burst 构成，一个 burst 可以由图 2-2 中一系列 chirp 组成。

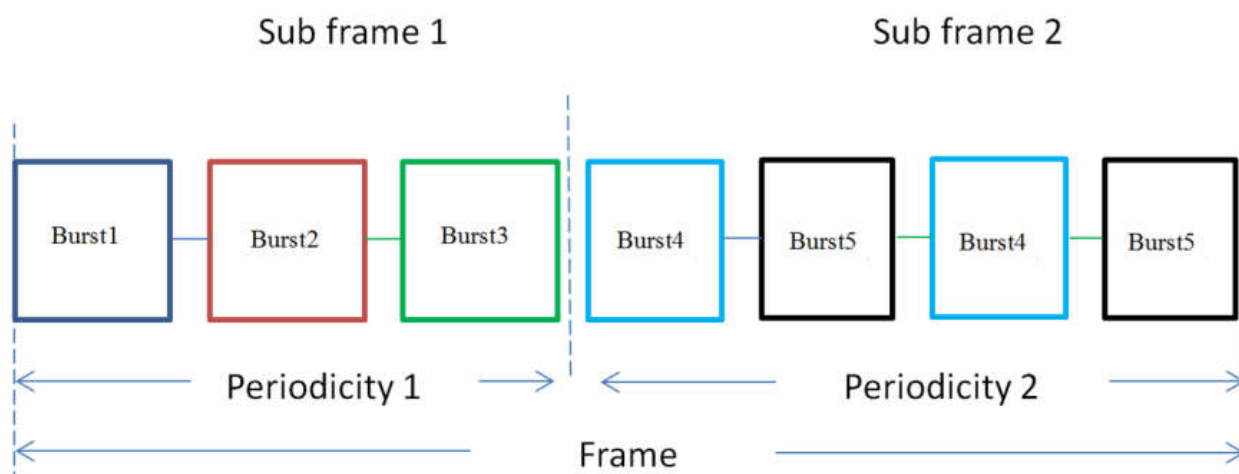


图 2-3. FMCW 高级帧结构

3 AWR2188 FMCW 波形配置

TI 毫米波芯片 FMCW 波形，包括 profile，chirp，子帧(subframe)，帧(frame)的配置。Profile 配置是定义基本的波形参数，包括起始频率(start frequency)、空闲时间(idle time)、扫描频率斜率(frequency slope)、采样率(ADC sampling rate)、采样点数(sampling number)等参数。Chirp 配置是在 profile 基础上，做参数的调整，得到不同的 chirp。子帧配置，是定义一个子帧内部的 burst，每个 burst 中 chirp 的构成，包括 burst 空闲时间及子帧空闲时间。帧配置则是由子帧组成，包括帧空闲时间。AWR2188 的波形配置相对 AWR2243 有所变化，相对来说更加简化，配置典型帧及高级帧均采用同一流程。如下表 3-1 列出了 AWR2188 与 AWR2243 波形配置的对比。

表 3-1. AWR2188 与 AWR2243 波形配置对比

	AWR2243	AWR2188
Profile	存储于 profile RAM，最大支持 4 个 profile	存储于 profile RAM，最大支持 8 个 profile。配置方式没有变化
Chirp	支持两种独立的配置方式，可以根据需要任选一种，两者的配置流程存在差异。 1. 典型 chirp: 存储于 chirp RAM，最大支持 512 个 chirp 2. 高级 chirp: 存储于 LUT(look-up-table 查找表格) RAM, 查表方式，更灵活，最大支持 2048chirp	只有一种 chirp LUT 方式，chirp 相关的参数差异构成一个 LUT，存储在 chirp RAM，通过查表方式获取并形成一系列 chirp。任意波形的配置必须配置 chirp LUT。
子帧(Subframe)	需要调用高级帧配置，最大支持 4 个子帧	跟 chirp 一样，存储于 chirp RAM，不区分典型帧和高级帧，最大支持 8 个子帧。
帧(Frame)	支持典型帧和高级帧，需要调用不同的 API 配置	统一配置，不区分典型帧和高级帧。

从表 3-1 可知，相对 AWR2243，AWR2188 的波形配置更加灵活统一，不再区分典型帧和高级帧。下面将对 AWR2188 的配置展开介绍。

3.1 配置关系

AWR2188 的波形配置包括以下几步：

- Profile 配置：包括基本的参数配置，每个 chirp 必须与一个 profile 相对应，存储在特定的 profile RAM。
- Chirp 参数 LUTs：chirp 可调的参数都会定义成一个查找表格(look-up-table LUT)，存储在特定的 chirp RAM。
- 子帧配置：包括 chirp 数、burst 及时间信息，及跟子帧相关的 chirp LUT 参数，包括需要使用 chirp LUT 的参数项，每个参数在 chirp LUT 的地址、大小、查表方式等。子帧相关的配置也会跟 chirp 参数 LUTs 一样存储在 chirp RAM。
- 帧配置：包括该帧需要使用的子帧在 chirp RAM 中的起始地址，子帧数等参数。

如下图 3-1 为各配置在 RAM 的存储位置及关联。

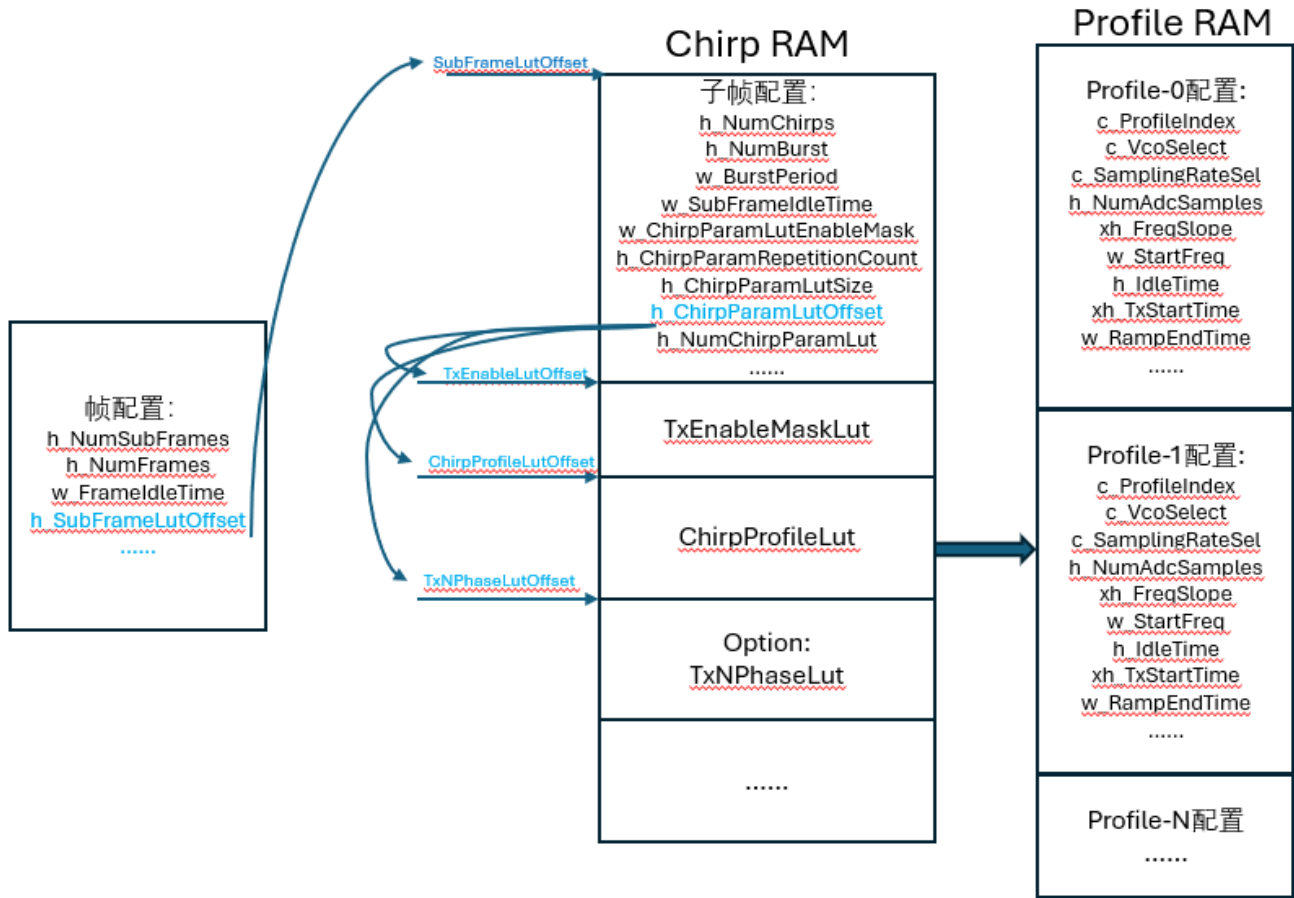


图 3-1. 波形配置存储位置及关联

如图 3-1 所示，每一帧的波形生成，先找到帧的配置，根据帧中的 h_SubFrameLutOffset 去 chirp RAM 中找到相应子帧配置；每个子帧根据各自定义的 chirp LUT 参数去 chirp RAM 索引 profile、TxEnable 等参数，生成相应的 chirp。

对于 AWR2188 波形的配置，基本的 API 配置流程如下图 3-2 所示。其中 chirp 参数 LUT 提供了很强的灵活性，可以支持不同波形的生成。



图 3-2. 配置 API 流程

下面将对帧、子帧、chirp 参数 LUT 及 profile 做展开介绍，详细的配置参数结构体、API 等说明见参考文献^[5]。

3.2 帧配置

帧配置中除了包含子帧数、发射帧数及帧时间相关参数外，跟 AWR2243 区别之处，在于包含了一个参数 h_SubFrameLutOffset，这个参数对应子帧配置在 chirp RAM 中的地址偏移。使用这个参数可以在 chirp RAM 中找到相应的子帧参数。如果一个帧包括多个不同的子帧，则多个子帧配置连续存储在 h_SubFrameLutOffset 开始的地址。可以调用 API ml_sensorFrameConfig 进行 frame 的配置。

如下截图 3-3 为帧配置例子，从图中可以看出，帧的配置包括子帧数，发波的帧数，帧空闲时间，子帧配置在 chirp RAM 的地址偏移，及校准监控相关的参数表在 chirp RAM 的位置。

```

/*Frame Config*/
T_ML_SENSOR_FRAME framecfg =
{
    .h_NumSubFrames = 1,
    .h_NumFrames = 0,
    .w_FrameIdleTime = 100000U * LINK_FRC_1US,
    .h_SubFrameLutOffset = 0x200,
    .h_CalMonScheduleLutOffset = 0x1000,
    .h_CalMonScheduleLutLength = 1,
};
/*sensorFrameConfig*/
status = ml_sensorFrameConfig (DEVICE_INDEX, &framecfg);

```

图 3-3. 帧配置例程

3.3 子帧配置

子帧的配置，这里特别说明跟 chirp 参数 LUT 相关的配置，其他参数说明见参考手册^[5]。Chirp 相关的参数是通过查表得到，支持查表的 chirp 参数说明见参考手册^[5]，w_ChirpParamLutEnableMask 决定哪些 chirp 参数需要由查表获取，其中 profile 和 TxEnable 相应的 bit 必须使能。对于需要查表的参数，其表格存储地址由 h_ChirpParamLutOffset 决定，表格的具体查询使用方式由 h_ChirpParamRepetitionCount、h_ChirpParamLutSize、h_NumChirpParamLut 决定。如下表 3-2 对子帧中的 chirp 参数 LUT 配置说明。

表 3-2. 子帧配置中 chirp 参数 LUT 配置说明

w_ChirpParamLutEnableMask	每个 bit 对应一个 chirp 参数，总共支持 17 个参数，其中 profile 和 TxEnable 必须使能
h_ChirpParamLutOffset	Chirp 参数 LUT 在 chirp RAM 中的存储起始地址偏移
h_ChirpParamRepetitionCount	Chirp LUT 中每个参数每次被读取后被使用的次数
h_ChirpParamLutSize	每个 burst 中 chirp LUT 大小
h_NumChirpParamLut	定义 burst 数量用于 Chirp LUT 的复位，即间隔几个 burst 后，chirp LUT 参数从起始地址重新索引

图 3-4 给出了子帧中 chirp 参数 LUT 相关配置的使用例程，在 chirp LUT 中写入了一组配置[A B C D E E]，h_ChirpParamRepetitionCount=2 说明每个参数取出来后连续使用了 2 次，即作用于 2 个连续 chirp。h_ChirpParamLutSize=3 说明每个 burst 中使用 LUT 的 3 个参数。h_NumChirpParamLut=2 说明该子帧在使用 LUT 中的参数项时，在每 2 个 burst 发送完后，从 LUT 的第一个开始循环。

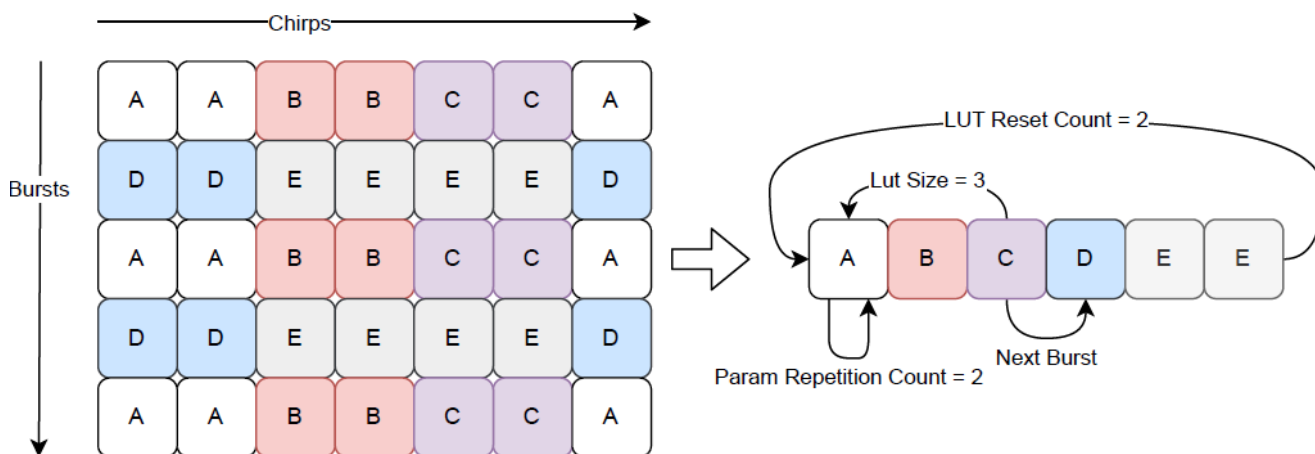


图 3-4. 子帧中 chirp 参数 LUT 配置使用例程

子帧的配置可以调用 `ml_sensorSubFrameConfig`，每次将一个子帧的配置写入 chirp RAM，对于多个子帧则需要调用多次。

如果要配置多个子帧，可以调用 `ml_sensorChirpRamWrite` 一次向 chirp RAM 写入多个子帧配置，可以提高配置效率。

如下图 3-5 为子帧配置参考代码，子帧配置中定义了 burst 数量，每个 burst 包含的 chirp 数，burst 周期，子帧空闲时间，及 chirp 参数 LUT 相关得配置。

图 3-5 示例中，使能了 profile 及 TxEnable 参数项，profile 参数存储在地址偏移 0x0 的 chirp RAM，TxEnable 参数存储在地址偏移 0x100 开始的 chirp RAM，在 chirp RAM 中均只存储了一个参数项，每个参数适用于所有 chirp。代码中调用 `ml_sensorSubFrameConfig` 将子帧配置写入地址偏移 0x200 的 chirp RAM，注意这个地址偏移必须与 3.2 节描述帧配置的 `h_SubFrameLutOffset` 相匹配。

```
/*Subframe Config*/
T_ML_SENSOR_SUB_FRAME subframecfg =
{
    .c_SubFrameCpIndex = 0,
    .h_NumChirps = 256,
    .h_NumBurst = 1,
    .w_BurstPeriod = ((70U * 256U) + 5000U) * LINK_FRC_1US, // (rampTime + idleTime) * numChirp + burstIdle
    .w_SubFrameIdleTime = 10000U * LINK_FRC_1US,
    .w_ChirpParamLutEnableMask = M_ML_SENSOR_CHIRP_LUT_MASK_PROFILE | M_ML_SENSOR_CHIRP_LUT_MASK_TX_ENABLE_MASK,
    .h_ChirpParamRepetitionCount = {
        [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE] = (UINT8)1U,
        [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_ChirpParamLutSize = {
        [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE] = (UINT8)1U,
        [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_ChirpParamLutOffset = {
        [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE] = (UINT16)0x0,
        [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT16)0x100,
    },
    .h_NumChirpParamLut = {
        [M_ML_SENSOR_CHIRP_LUT_INDEX_PROFILE] = (UINT8)1U,
        [M_ML_SENSOR_CHIRP_LUT_INDEX_TX_ENABLE_MASK] = (UINT8)1U,
    },
    .h_MiscControlMask = 0,
    .h_InterChirpPowerSaveCtrl = 0,
};
/*SensorsubFrameConfig*/
status = ml_sensorSubFrameConfig(DEVICE_INDEX, 0, 0x200, &subframecfg);
```

图 3-5. 子帧配置参考代码

3.4 chirp 配置

AWR2188 支持每个 chirp 的参数有所变化，这些 chirp 参数以查找表格(look-up-table LUT)的方式存储在特定的 chirp RAM，具体的 chirp 参数见参考文献^[5]。其中 profile LUT 及 TxEn LUT 是必须使能，分别指定 chirp 对应的 profile，及 chirp 对应的发射天线。

每个 chirp 参数 LUT 有各自的格式和单位，具体见参考文献^[5]。可以使用 `ml_sensorChirpRamWrite` 将 chirp 参数 LUT 写入指定地址偏移的 chirp RAM。注意，chirp RAM 总共可用空间是 31KB，包括 chirp 参数 LUT 和子帧的配置。

如下图 3-6 所示，定义了 profile 和 TxEnable LUT，并通过调用 `ml_sensorChirpRamWrite` 将其分别写入偏移地址为 0 和 0x100 的 chirp RAM，注意这里的写入地址偏移必须与 3.3 节描述子帧配置中的 `h_ChirpParamLutOffset` 相匹配。

```
uint8_t lutparam_profile[4] = {0};
uint8_t lutparam_Txenable[8] = {M_ML_SENSOR_MASK_ALL_TX};

/*LUT write for Profile parameter Mandatory*/
status = ml_sensorChirpRamWrite(DEVICE_INDEX, 0x0, 4, lutparam_profile);
while(status!=0);

/*LUT write for TX Enable Mask parameter Mandatory*/
status = ml_sensorChirpRamWrite(DEVICE_INDEX, 0x100, 4, lutparam_Txenable);
while(status!=0);
```

图 3-6. image10_chirp 参数 LUT 配置参考代码

3.5 profile 配置

profile 的配置跟 AWR2243 没有区别，其中定义了基本的射频参数，包括采样率、采样点数、接收增益、发射功率衰减、扫描斜率等参数，具体见参考文献^[5]，可以使用 ml_sensorProfileConfig 进行 profile 配置。

图 3-7 是 profile 参数及配置参考代码，定义了一个 profile，如果定义配置多个 profile，则在 profilecfg 中增加一组配置，并需要调用 ml_sensorProfileConfig 再配置一次 profile 即可。

```
/*Profile Config*/
T_ML_SENSOR_PROFILE profilecfg[MAX_NUM_OF_PROFILES] =
{
    {
        /*Profile 0*/
        .c_ProfileIndex = 0,
        .c_VcoSelect = M_ML_SENSOR_VCO_SELECT_AUTO,
        .h_MiscSettings = M_ML_SENSOR_HPF_FAST_INIT_ENABLE | M_ML_SENSOR_CAL_MON_OVERRIDE_ENABLE,
        .c_SamplingRateSel = 16,
        .c_RxGainSelect = M_ML_SENSOR_RF_GAIN_SEL_38dB | (UINT8)44U,
        .h_NumAdcSamples = 512,
        .c_RxRfInitCalDataSel = 0,
        .c_HpfCutoffSel = {[0U] = M_ML_SENSOR_HPF_CUTOFF_FREQ_350KHz, [1U] = M_ML_SENSOR_HPF_CUTOFF_FREQ_350KHz},
        .c_TxBackoffs = {[0U] = 0U, [1U] = 0U, [2U] = 0U, [3U] = 0U, [4U] = 0U, [5U] = 0U, [6U] = 0U, [7U] = 0U},
        .c_TxClpcCalEnableMasks = {[0U] = 0xFFU, [1U] = 1U, [2U] = 1U, [3U] = 1U, [4U] = 1U, [5U] = 1U, [6U] = 1U, [7U] = 1U},
        .c_TxRfInitCalDataSel = {[0U] = 0U, [1U] = 0U, [2U] = 0U, [3U] = 0U, [4U] = 0U, [5U] = 0U, [6U] = 0U, [7U] = 0U},
        .xc_TxPhase = {[0]=0, [1]=0, [2]=0, [3]=0, [4]=0, [5]=0, [6]=0, [7]=0},
        .xw_StartFreqAccDelta = 0,
        .xc_TxPhaseAccDelta = {[0]=0, [1]=0, [2]=0, [3]=0, [4]=0, [5]=0, [6]=0, [7]=0},
        .h_StartFreqAccPeriod = 0,
        .h_TxPhaseAccPeriod = {[0]=0, [1]=0, [2]=0, [3]=0, [4]=0, [5]=0, [6]=0, [7]=0},
        .w_StartFreq = LINK_CONST_GET_SYNTH_FREQ(76),
        .xh_FreqSlope = LINK_CONST_GET_CHIRP_SLOPE(15.998),
        .xh_TxStartTime = 0 * LINK_RAMPGEN_1US,
        .w_RampEndTime = 40U * LINK_RAMPGEN_1US,
        .h_IdleTime = 30U * LINK_RAMPGEN_1US,
        .h_AdcStartTime = 6U * LINK_RAMPGEN_1US,
        .h_AdcSkipSamples = 0,
        .c_LoopbackFreqSelA = (UINT8)30U,
        .c_LoopbackFreqSelB = (UINT8)6U,
        .c_LoopbackGainA = (UINT8)63U,
        .c_LoopbackGainB = (UINT8)0U,
        .c_LoopbackMode = M_ML_SENSOR_LOOPBACK_DISABLE,
        .w_CalMonStartFreq = LINK_CONST_GET_SYNTH_FREQ(78),
        .xh_CalMonFreqSlope = LINK_CONST_GET_CHIRP_SLOPE(0.1),
        //.h_ProgFilterLength = 4U,
        //.h_ProgFilterStartIndex = 0,
    },
    /* Other Profiles can be added if required*/
};

/*sensorProfileConfig*/
status = ml_sensorProfileConfig(DEVICE_INDEX, &profilecfg[0]);
```

图 3-7. profile 配置参考代码

4 总结

本文对比了 AWR2243 与 AWR2188 的波形配置差异，详细介绍了 AWR2188 波形配置流程，并给出了相应参考示例，帮助客户更好的理解掌握 AWR2188 波形配置方法，更快的生成产品所需的波形配置。

5 参考文献

1. SWRA553A, [Programming Chirp Parameters in TI Radar Devices](#)
2. SWRS223D, [AWR2243 Single-Chip 76- to 81-GHz FMCW Transceiver datasheet](#)
3. SWRS339, [AWR2188 Single Chip 8x8 Cascadable 76-to-81 GHz Transceiver in LOP package datasheet \(Rev. A\)](#)
4. [mmWave-Radar-Interface-Control.pdf](#)
(mmwave_mcuplus_sdk_04_06_01_02\mmwave_dfp_02_02_04_00\docs)
5. [mmWaveLink_4_Interface_Control_Document_1_0.pdf](#)(mmWave_DFP_04_00_06_02\docs)

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。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月