

参考如下：

Xiaojie雷达之路---profile、chirp、advframe、frame的设置及关系

/mss/Source/system/hal/RF/cfg_Ext.h

```
extern void Cfg_AdvFrameCfgInitParams (r1AdvFrameCfg_t* ptrAdvFrameCfg);
extern void Cfg_FrameCfgInitParams (r1FrameCfg_t* ptrFrameCfg);
extern void Cfg_ProfileCfgInitParams (uint8_t profileNum, r1ProfileCfg_t*
ptrProfileCfg);
extern void Cfg_ChirpCfgInitParams (uint32_t chirpNum, r1ChirpCfg_t* ptrChirpCfg);
extern void Cfg_LowPowerModeInitParams (r1LowPowerModeCfg_t* ptrLowPowerMode);
extern void Cfg_ChannelCfgInitParams (r1ChanCfg_t* ptrChannelCfg);
extern void Cfg_ADCOutCfgInitParams (r1AdcOutCfg_t* ptrADCOutCfg)
```

/MSS_2944/Source/system/hal/RF/cfg.c

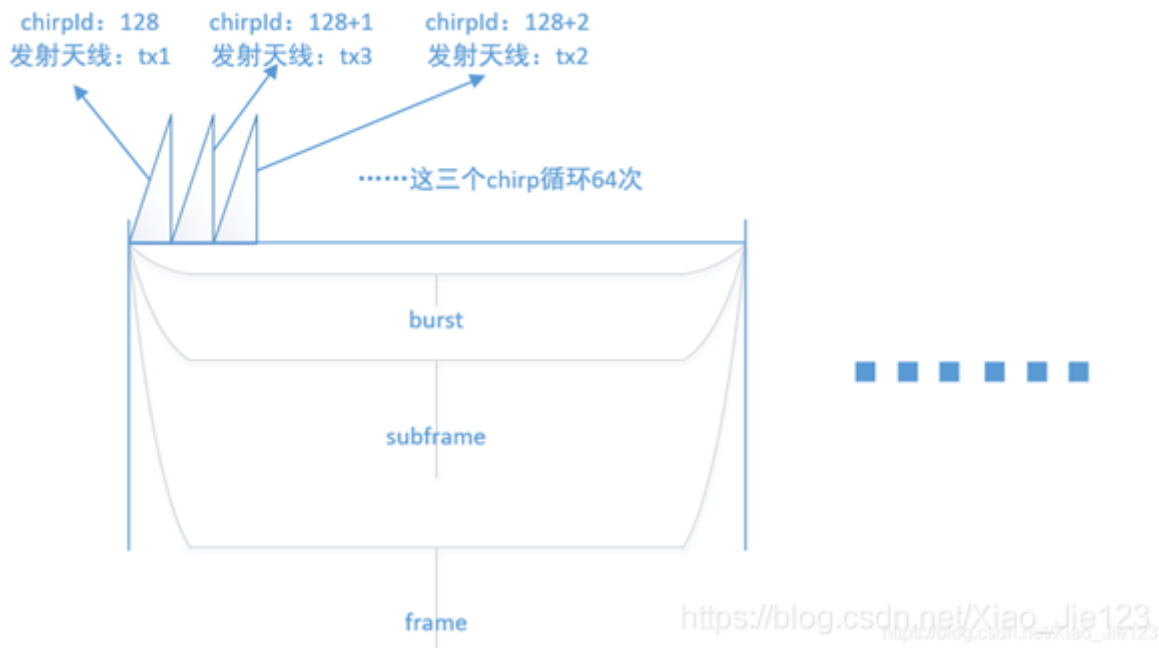
AdvFrameCfg

```
/**
 * @b Description
 * @n
 * The function initializes the frame configuration with the default
 * parameters.
 *
 * @param[out] ptrAdvFrameCfg
 * Pointer to the advance frame configuration
 *意思是：高级帧配置数据的指针容器
 * @retval
 * Not applicable
 */
void Cfg_AdvFrameCfgInitParams (r1AdvFrameCfg_t* ptrAdvFrameCfg)
```

函数功能：

这个函数允许在mmWave前端配置高级帧。高级帧是一系列的chirp以及这个序列如何随着时间的推移而重复。==用户首先需要定义一个profile和一系列的chirps(与profile相关)。然后，该函数定义如何对这些chirp进行排列。==多个chirp被循环产生一个脉冲。可以将多个burst分组以创建

sub-frame。多个sub-frame(最多4个)可以分组以创建高级帧。该函数可以定义高级帧的属性，如子帧中的burst的数量、burst中的chirps和loops的数、要发送的sub-frame序列、要发送的frames的数量，帧的周期性和触发方法。这个API在内部要调用两个API，一个调用RadarSS及进行传感器配置，另一个调用MasterSS进行数据路径的配置。



为了提供frame中chirp的最大灵活性，advanced frame提供了将frame分解为不同的 subframe（最多四个？），每个sub-frame由多个burst组成（多达64次），每个burst最多可以由512个不同的chirp组成，每个chirp要与4个profile中的一个关联

FrameCfg

(未用到，用到了advframecfg)

```
/**
 * @b Description
 * @n
 * The function initializes the frame configuration with the default
 * parameters.
 *
 * @param[out] ptrFrameCfg
 * Pointer to the frame configuration
 *
 * @retval
 * Not applicable
 */
void Cfg_FrameCfgInitParams (r1FrameCfg_t* ptrFrameCfg)
```

函数功能：

此函数允许在毫米波前端配置FMCW帧。一帧基本上是一系列的chirp，以及这种序列如何随着时间的推移而重复。用户首先需要定义一个profile和一组chirp(与一个profile相关联)。然后，这个函数定义如何对这些chirps进行排序。同一个chirp可以简单的循环以创建一个大的FMCW帧，或者可以对多个独特的chirp进行排序来创建帧，Chirp Start和end Index定义了如何在一个帧中对它们进行排序。该API还允许配置要传输的帧数，帧的周期性和触发器方法。触发方法可以是基于SW-API的触发器或基于HW-SYNC-IN的触发器。这个API内部调用两个API，一个调用RadarSS进行传感器配置，另一个调用MasterSS进行数据路径配置

Profilecfg

```
/**
 * @b Description
 * @n
 *     The function initializes the profile configuration with the default
 *     parameters.
 *
 * @param[in] profileNum
 *     Profile number to be initialized
 * @param[out] ptrProfileCfg
 *     Pointer to the profile configuration
 *
 * @retval
 *     Not applicable
 *
 * 函数功能：
 * 这个函数用于设置mmWave Front end的chirp profilecfg，一个profile就像一个模板，
 * 其中包含有关FMCW信号的粗略信息，如起始频率、chirp斜率、chirp持续时间、发生功能等。
 * API允许通过将profile数组与profile计数一起传递来设置多个profiles
 */
void Cfg_ProfileCfgInitParams (uint8_t profileNum, rlProfileCfg_t* ptrProfileCfg)
```

Chirpcfg

```
/**
 * @b Description
 * @n
 *     The function initializes the chirp configuration with the default
 *     parameters.
 *
 * @param[out] chirpNum
 *     Chirp Number to be configured
 * @param[out] ptrChirpCfg
```

```

*      Pointer to the chirp configuration
*
*      @retval
*      Not applicable
*
*  函数功能:
*  此函数用于在chirp profile的顶部设置chirp到chirp的变化, 用于首先应该使用
*  Cfg_ProfileCfgInitParams定义一个profile。然后, 这个Cfg_ChirpCfgInitParams函数
*  通过将chirp与Cfg_ProfileCfgInitParams API
*  中定义的特定的profile关联起来配置chirp。此外, 用户能使用此API定义profile中的
*  参数的精细抖动。此配置使用的抖动只是Cfg_ProfileCfgInitParams中编程参数的附加值。
*  此API允许配置1个到512个chirp。它还允许配置每个chirp使用哪些发射通道
*/
void Cfg_ChirpCfgInitParams (uint32_t chirpNum, rlChirpCfg_t* ptrChirpCfg)

```

注意:

可以设置多达512个独特的chirp, 能被存储在毫米波前端的专用存储器中。因此, 用户不需要再运行时对chirp进行编程。也可以使用Cfg_FrameCfgInitParams再一个帧中对这些chirp进行排序, 以产生更大的FMCW信号。

四者(profile、chirp、advframe、frame)总结

如您所见, 根据所需的应用, chirp配置需要进行不同的设置。但是如果我们需要同时支持多种模式, 例如短程和中程, 怎么使用单一雷达设备? TI雷达的advanced frame配置允许在一个frame中可以有多个chirp配置的灵活性。frame可以使用“sub frame”序列, 其中每个sub-frame代表不同的雷达模式。这个rlSetAdvFrameConfig帮助启用这种配置。有关详细信息, 请参见在DPF包中的“毫米波雷达接口控制文件”

该帧由最多四个这样的sub frame构成, 并且每个子帧可以具有不同的chirps。不同的chirps也可以使用不同的发射机(可能有不同的天线配置)。图9显示了不同的chirp配置文件和子帧的示例可以形成一个帧。advanced frame的时序要求如下: inter-burst时间应 $\geq 50\mu\text{s}$, inter sub-frame时间应为 $\geq 100\mu\text{s}$, 帧间时间为 $\geq 200\mu\text{s}$ 。

profile是chirp的模板, profile只是对chirp进行粗略的配置, 而对chirp进行配置的时候, 可以通过指定profileId来确定要指定哪一个profile, advframe和frame的区别在于frame只能由chirp构成, 而advframe由subframe组成, 而subframe由burst组成, burst由chirp组成, 下图两幅图说明advframe和frame的区别。

