

参考如下:

视频参考:

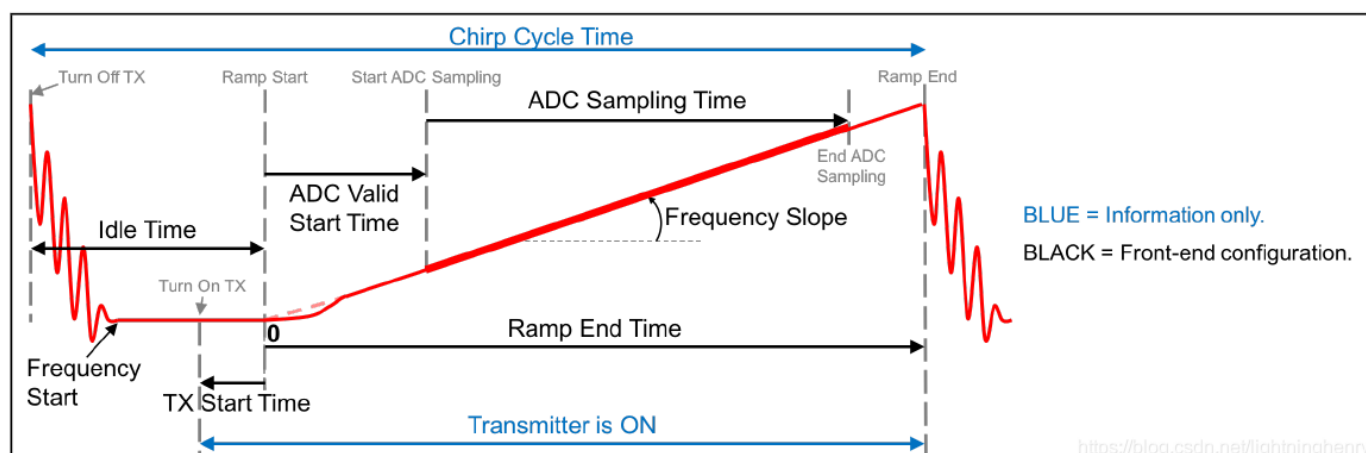
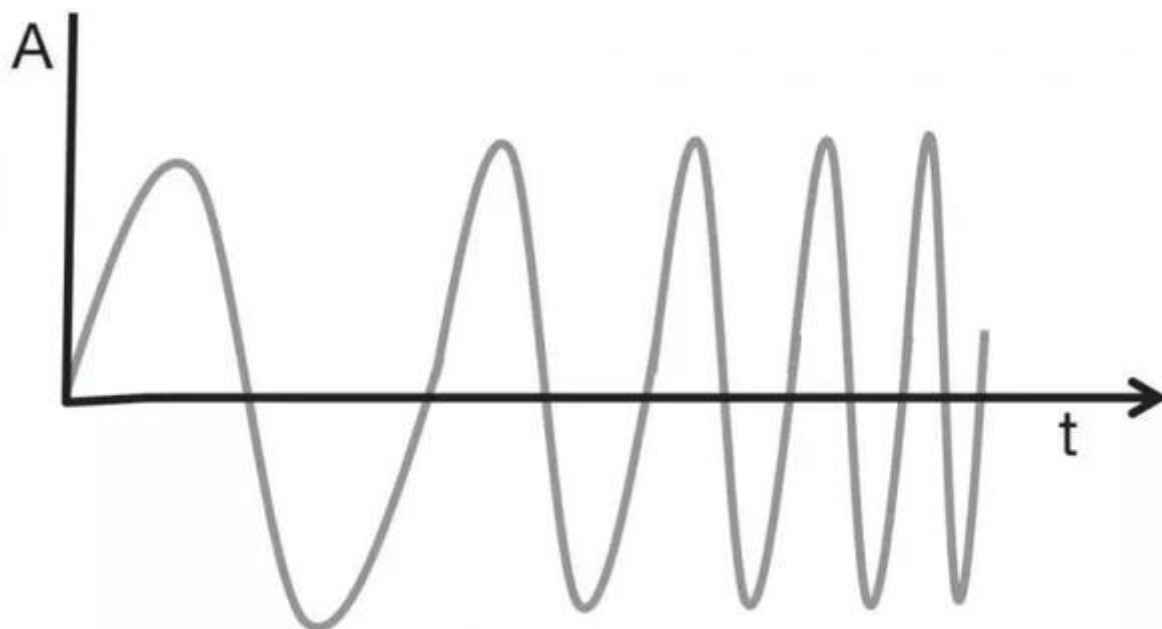
文档参考:

[TI实战笔记---chirp、profile、frame、advframe](#)

[

一、什么是chirp?

chirp是雷达发射的一种信号，频率随着时间线性增加，最多可以定义512个chirp，chirp斜坡的各种参数(如频率斜率、扫描带宽等等)会影响系统的性能，如下图所示：



Chirp Timing Parameters:

Idle time: 上一次chirp结束时间和下一次chirp开始时间之间的时间 Tx start time: 从发射器打开到斜坡开始的时间 ADC start time: 从斜坡开始到ADC开始对数据进行采集这段时间

Ramp end time: 从斜坡开始到chirp持续上升的时间。在这段时间之后，合成器的频率就被重置为下一个chirp的起始频率

可以看到ADC Samping Time并不是全程采样，还有idel time，这个时间会很长；Turn on TX后也不是马上就能发射雷达波

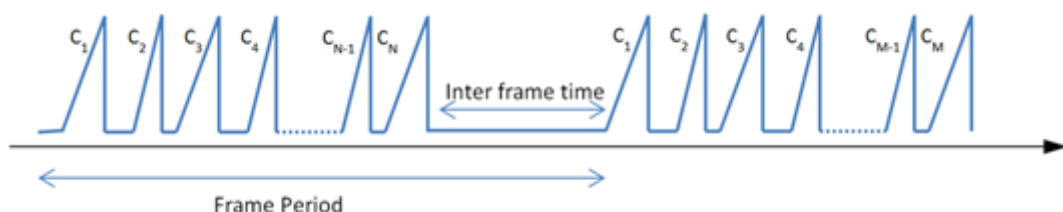
1D处理需要关心的参数主要位于 `profileCfg 0 77 429 7 57.14 0 0 70 1 256 5209 0 0 30` 这一行，破译结果如下

name	profile cfg
startFreq	77GHz
idleTime	429us
adcStartTime	7us
rampEndTime	57.14us
freqSlopeConst	70MHz/us
txStartTime	1us
numAdcSamples	256
digOutSampleRate	5209ksps
rxGain	30dB

我们知道1843最大支持的带宽为77~81GHz，第一行起始频率为77GHz，对上了。再看斜率为70MHz/us 乘以调频时间57.14us，大家可以算一下，结果就是4GHz，带宽和我们设置的也吻合。

二、什么是frame?

frame是一组chirp的集合，有多个chirp组成，这个表示"快速调频连续波"调制，其中每个chirp的持续时间通常为10us，如下图：



三、什么是profile?

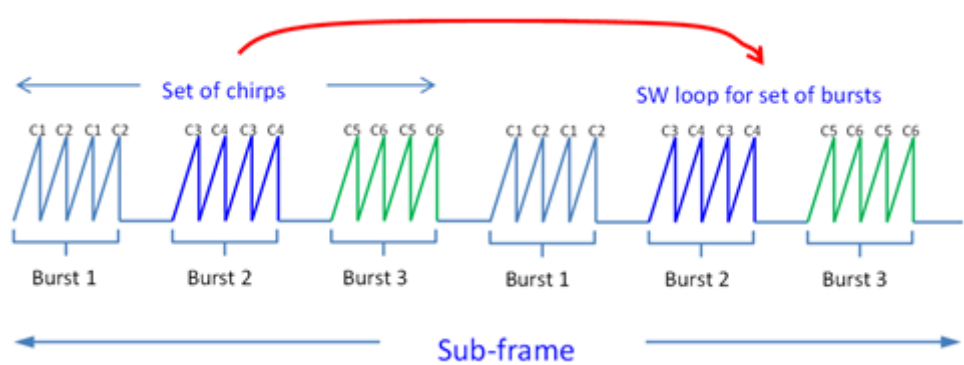
Profile是chirp的时序模板，用于定义一个或多个定义中存在显示差异的chirp变量参数(开始频率、斜率、空闲时间等)。最多可以定义4个profile

上面介绍的是最基本的frame，也就是basic frame

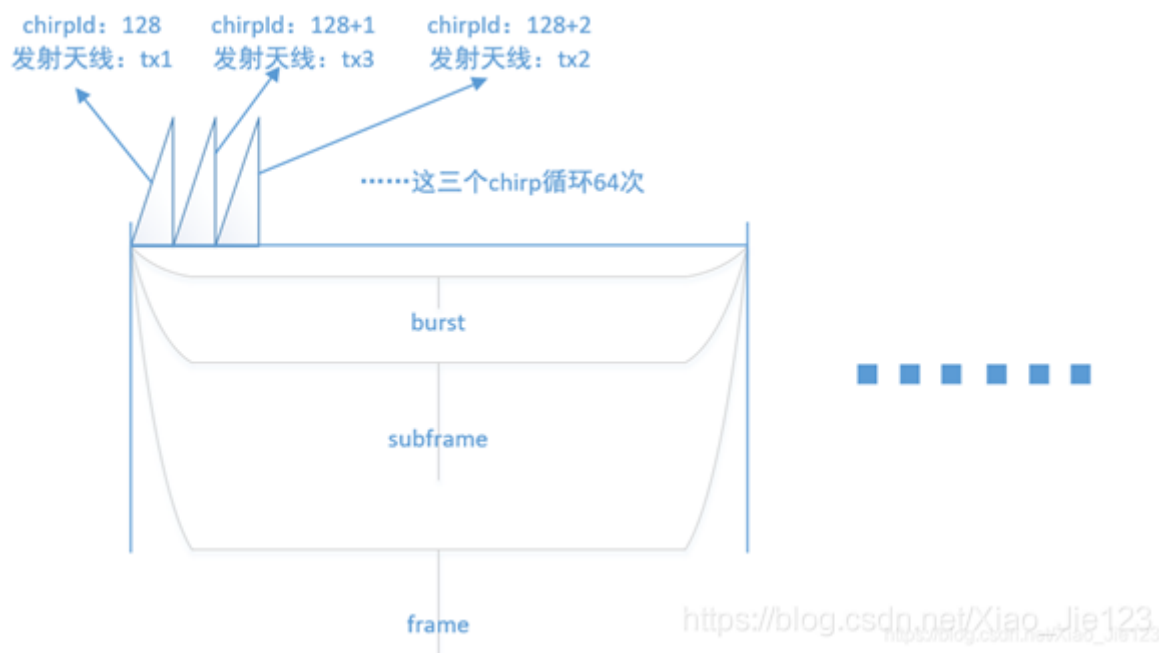
下面要介绍advanced frame的相关概念

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sub-frame，每个sub-frame由多个burst组成，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，sub-frame中的一组burst可以定义为循环多达64次。

下图是Sub-frame形成的示例：

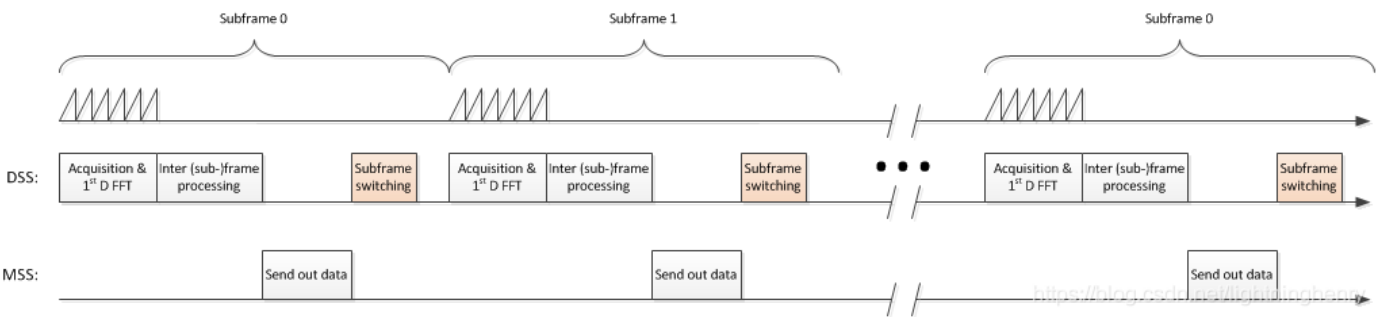


根据上面profile、chirp、advframe三段程序的配置可以得到下面的配置

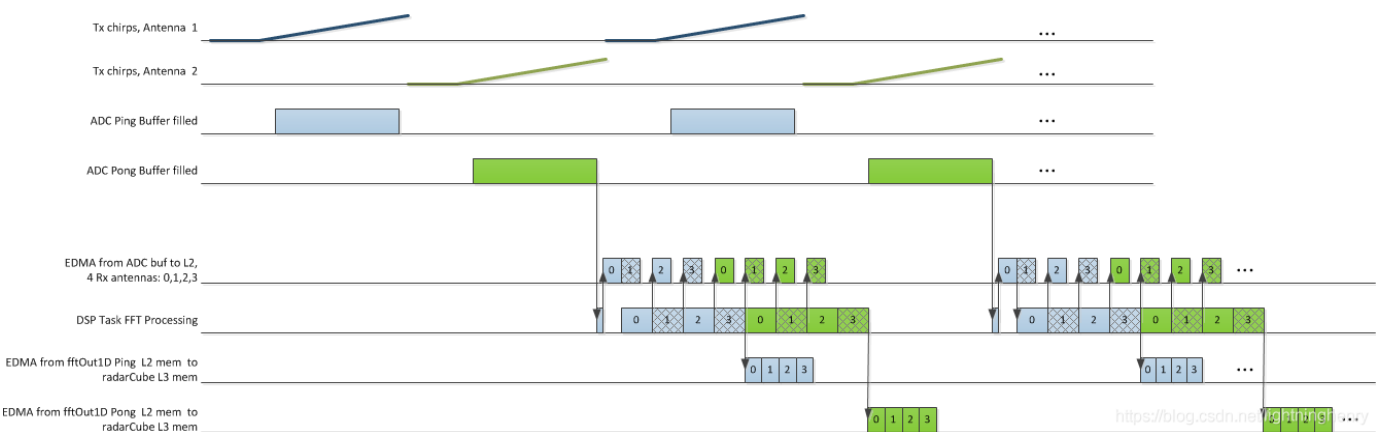


先看一个比较笼统的数据流，雷达信号处理是以帧为单位的，下图的横轴都是时间轴，每积累够一帧才能出一次结果，每帧里又有很多个chirp数据（比如256个），每个chirp又有很多和距离门

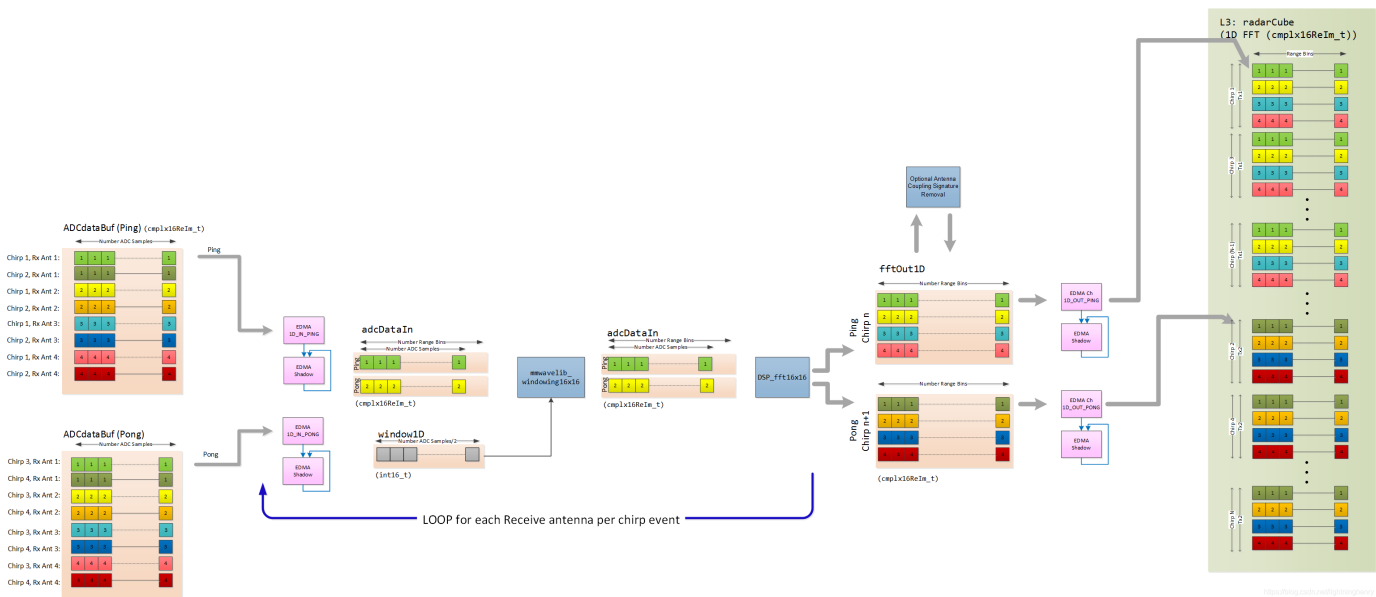
数据（比如1024个），每个数据均为复数。每来一个chirp数据就要做一次1D-FFT，见下图中的subframe里面



接着来看1D-FFT的处理。首先要说明下天线，我们看下面这张图（横轴是时间），开发板上有两个发射天线TX，但是这两个发射天线不是同时辐射电磁波，是在时间轴上交替发射的；而接收天线RX有4个，所以同一时刻雷达会同时收到4路回波信号，但是由于DSP、ARM只能顺序接收和处理数据，所以4个接收通道的数据也是在时间轴上交替发过来的。



每一个chirp里面都有4个通道的回波数据，读入第0通道的数据后紧接着开始做FFT，做第0通道的FFT时，EDMA读入第1通道的数据；做第1通道的FFT时，EDMA读入第2通道的数据，形成每个chirp内的小乒乓（下图中间部分）；两个发射天线产生的两组chirp回波数据形成大乒乓（下图右侧），在运算第二组chirp数据的FFT时，将第一组chirp的运算结果搬移到radarcube L3 memory1，循环往复直到一帧内所有数据都运算完。



上图展示了1D-FFT的整个处理流程，处理的过程中是乒乓流水的，每来一个回波就做一次全部的处理，从 ADC 采集到加窗到FFT，最后把处理好的数据缓存到radarcube L3 memory中，我们看下最右边的radarCube（存了一帧数据），放大看。

chirp1、chirp2、...、chirp(N-1)、chirp(N)是对雷达回波的编号，是按照时间顺序来编号的，结合本文第二张图来看，奇数表示接收到的是发射天线1发射的回波，偶数表示接收到的是发射天线2发射的回波，奇偶分开存储。代码就是通过判断chirp的编号的奇偶来进行乒乓操作的（大乒乓和小乒乓都是）。

右边这个radarCube其实就是一个二维矩阵，列数为range bin的总数（本工程中数值为256）。上图中重复出现的绿1-黄2-蓝3-粉4为4个接收天线分别接收到的数据，，上图中只显示了TX1，所以行数为 $4 \times N/2$ 。那么Doppler Bin的总数怎么算呢，每次发射都有 2×4 个回波产生，我们要把这8个回波分开来算多普勒频率，所以Doppler Bin的总数为 $N/2$ 。

综上，距离维就是这个二维矩阵的行方向，多普勒维就是这个二维矩阵的列方向！

相参积累：可以对n个回波进行累加，由于噪声是随机的，累加的结果是信号变强，而噪声因是随机的，强度反而变小，这样信号与噪声比就提高了。