

一.AWR1843产品与架构介绍

参考:

视频参考:

文档参考:

[AWR1843](#)

(1) 产品定义:

集成 DSP、MCU 和雷达加速器的单芯片 76GHz 至 81GHz 汽车雷达传感器。

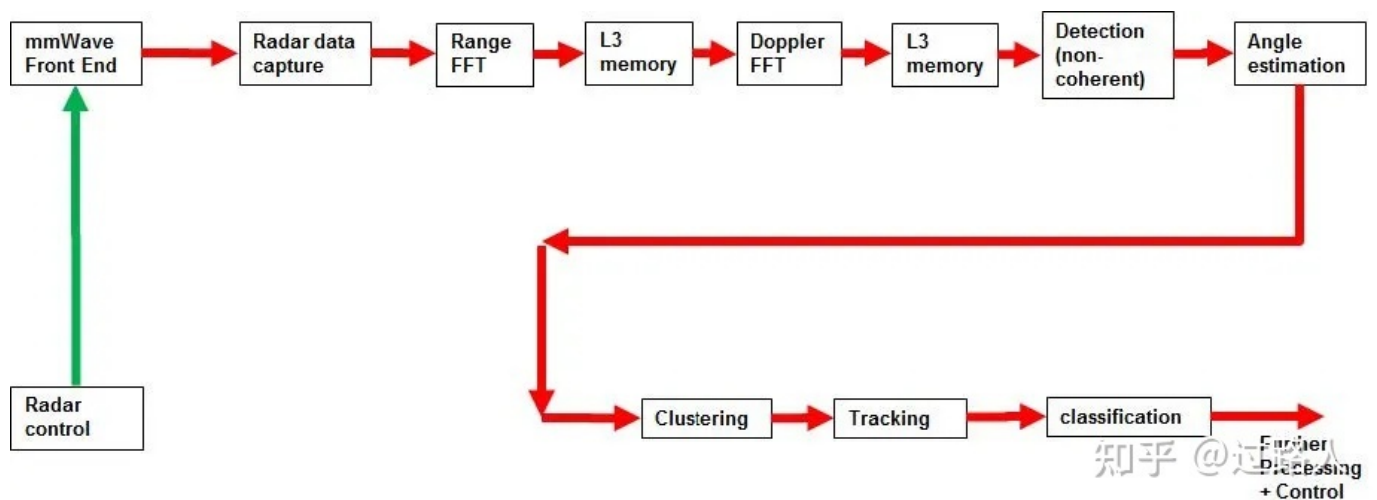
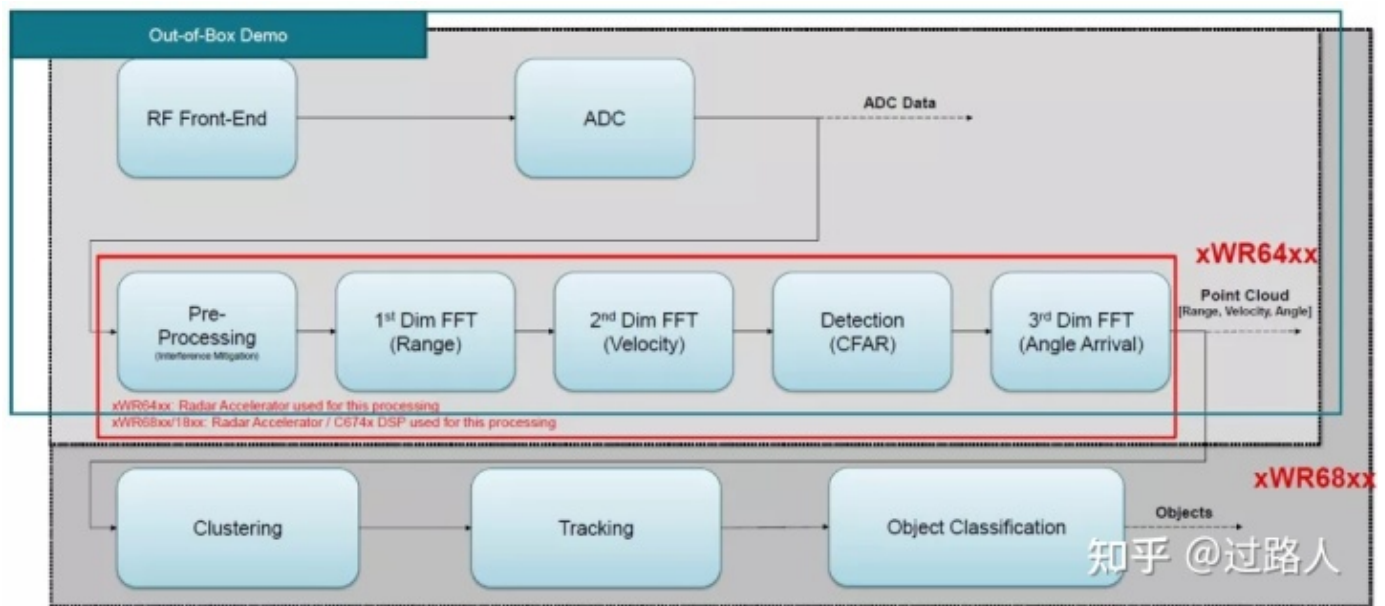
集成了所有的射频和模拟功能，包括VCO、PLL、PA、LNA、混频器和多个TX/RX通道的ADC集成到一个芯片中。 AWR1843是毫米波雷达片上设备，**包括4个接收通道和3个发射通道，另外还有一个用于信号处理的Cortex R4F和 用于 FMCW 信号处理的 C674x DSP。 **该器件包含一个 BIST 处理器子系统，该子系统负责无线电配置、控制和校准。

AWR1843 器件是一款能够在 76 至 81GHz 频带中运行的集成式单芯片 FMCW 雷达传感器。该器件采用 TI 的低功耗 45nm RFCMOS 工艺进行构建，并且在超小封装中实现了出色的集成度。 AWR1843 是适用于汽车领域中的低功耗、自监控、超精确雷达系统的理想解决方案。

AWR1843 器件是一种自包含 FMCW 雷达传感器单芯片解决方案，能够简化 76 至 81GHz 频带中的汽车雷达传感器实施。它基于 TI 的低功耗 45nm RFCMOS 工艺构建，从而实现了一个具有内置 PLL 和 ADC 转换器的单片实施 3TX、4RX 系统。它集成了 DSP 子系统，该子系统包含 TI 用于雷达信号处理的高性能 C674x DSP。该器件包含一个 BIST 处理器子系统，该子系统负责无线电配置、控制和校准。此外，该器件还包含用于汽车连接的用户可编程 ARM R4F。硬件加速器区块 (HWA) 可执行雷达处理，并且有助于以更高级的算法在 DSP 上节省 MIPS。简单编程模型更改可支持各种传感器实施（近距离、中距离和远距离），并且能够进行动态重新配置，从而实现多模式传感器。此外，该器件作为完整的平台解决方案进行提供，其中包括 TI 参考设计、软件驱动程序、示例配置、API 指南以及用户文档。

(2) 流程与原理框图

1.图所示为毫米波雷达的信号处理链。



首先，雷达前端（RF Front-End）完成信号的发送、接收任务，并将信号传递给模数转换器（ADC）进行数字化处理。

然后，对数字信号进行预处理（Pre-Processing）抑制信号中的干扰，并分别进行一维FFT(测距)、二维FFT(多普勒)求解目标距离、速度信息，设置CFAR值来检测目标是否存在(非相干检测)，再进行三维FFT（角度）求解目标的角度信息，检测到的点生成点云数据。

最后，在之后使用更高层的算法（例如聚类、跟踪、分类）对点云数据进行后处理以表示现实世界中的目标。

2.毫米波雷达的功能框图如下所示：

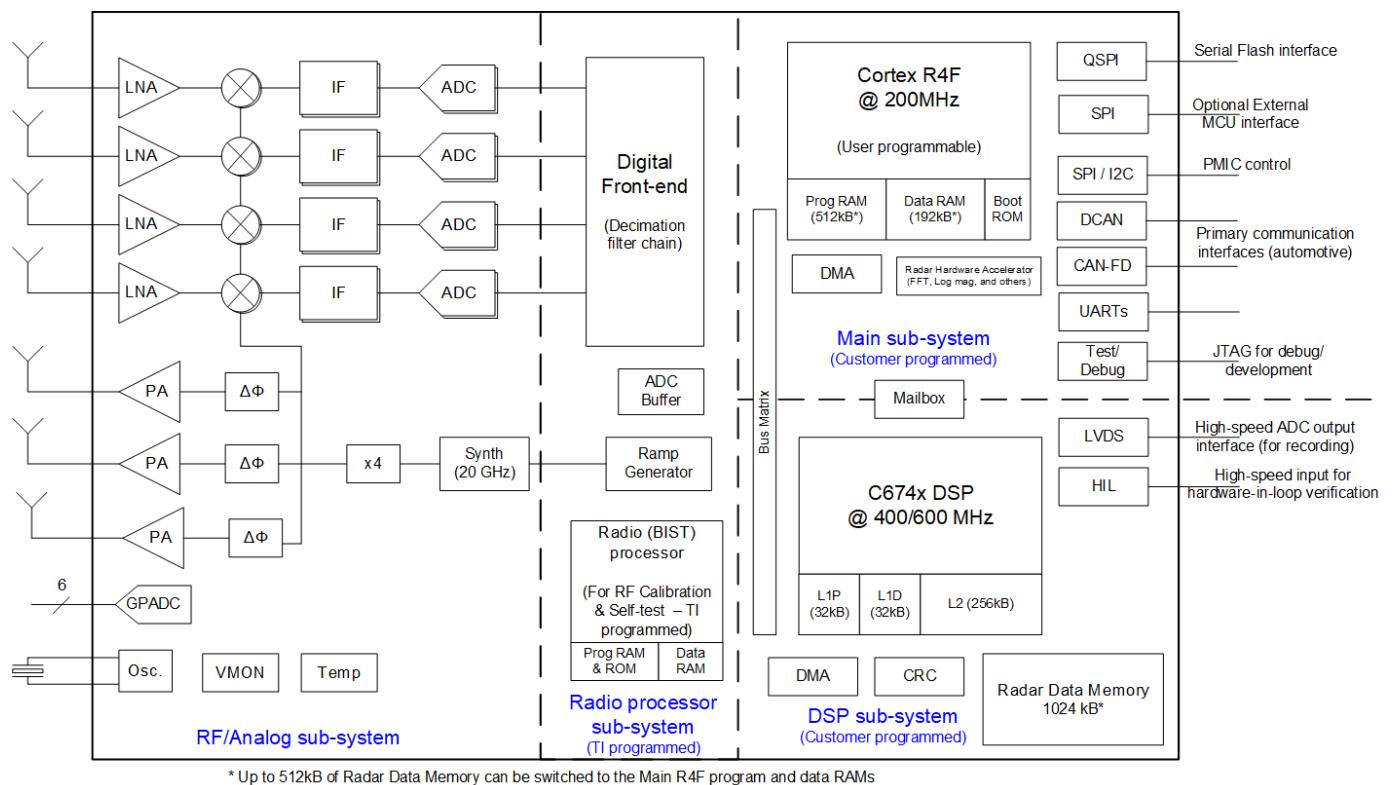
其主要包含

射频/模拟子系统（RF/Analog Sub-System）、

射频处理子系统（Radio Processor Sub-System），也称雷达子系统（Radar Sub-System）：包含时钟子系统、发射子系统、接收子系统。雷达子系统包括DFE（数字前端）和斜坡发生器。

主子系统（Master Sub-System）包括基于 ARM Cortex-R4F 的无线电控制、信号处理子系统及相关接口。

数字信号处理子系统（DSP Sub-System）主要为用于高级信号处理的 C674x DSP系统。



(3)专业术语

BSS: 指的是mmWave Front End(雷达前端)

MSS: 指的是Master Sub-system(Cortex R4F)

DSS: 指的是DSP Sub-system(C674x core)

为什么会有两个工程呢？

这是因为在AWR1843里面有两个处理器：一个是ARM的Cortex_R4F，对应于mss工程，另一个是DSP的C674x，对应于dss工程。

UART:

串口的作用就是在两个设备之间进行通信。在这个案例中(基于Automated Parking这个案例)用到了两个串口，一个串口的作用是通过上位机将信息发送到AWR1843，另一个串口就是将雷达采集到的数据通过DSS处理以后在发送到PC。

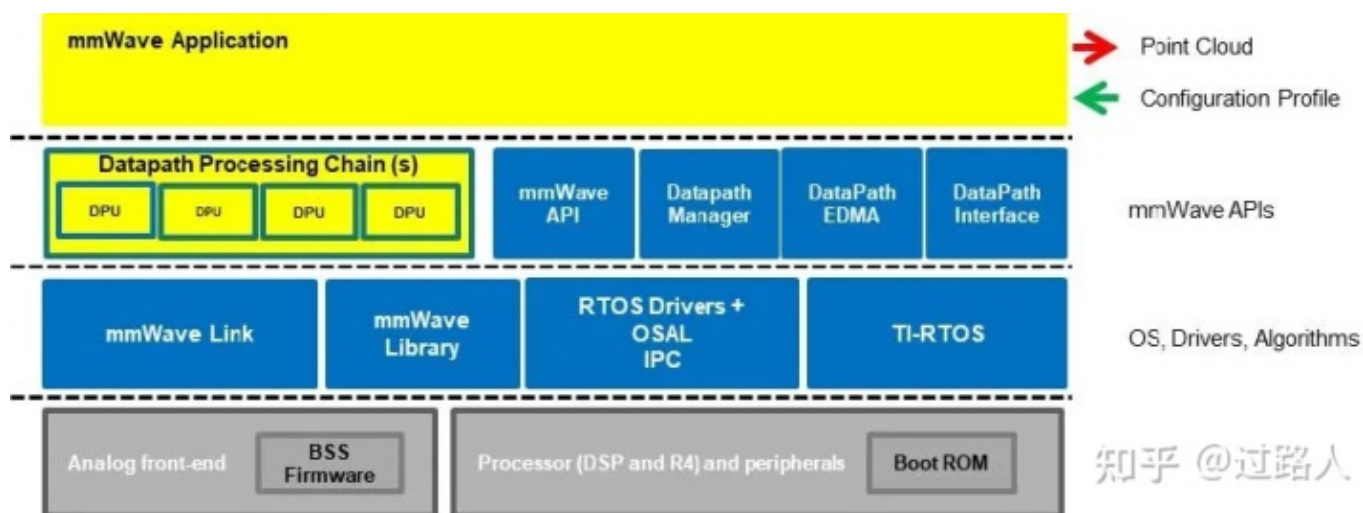
MailBox:

邮箱，在我们的日常生活中，也就是来进行通信的工具，在这里也不例外，邮箱的作用就是在 MSS 和 DSS 之间建立连接。

Semaphore:

信号量的作用就是防止资源竞争，就是协调各个线程，保证他们能够正确、合理的是使用公共资源。

(4) mmWave SDK 及其分层结构的组件

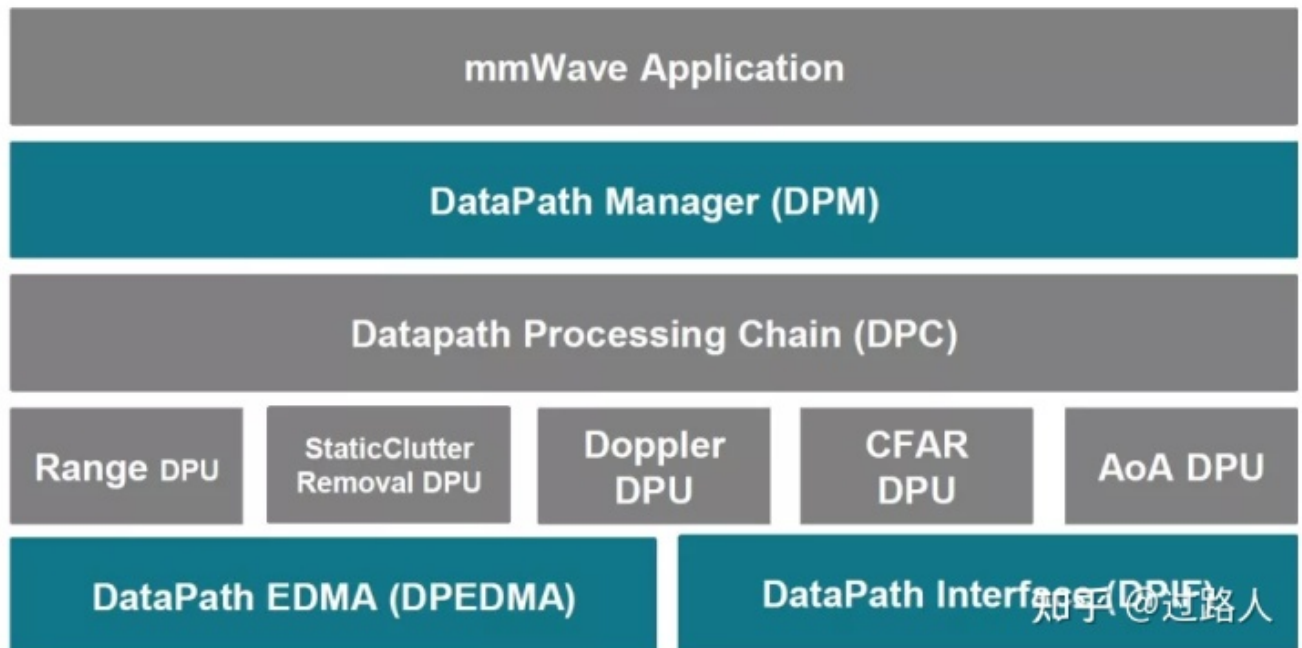


毫米波雷达的 SDK 结构如上图所示。其主要包含：

- 支持 SOC（系统级芯片 DSP 和 R4）的基础组件—— TI_RTOS、驱动程序（Drivers）、mmWaveLink、mmWaveLib
- RTOS——实时操作系统（Real Time Operating System），是保证在一定时间限制内完成特定功能的操作系统。mmWave 主系统运行在 RTOS 架构上。所有的驱动程序都附带 OSAL，可以将这些驱动程序移植到不同的 RTOS 系统
- OSAL——操作系统抽象层（Operating System Abstraction Layer），它并不是一个传统意义上的操作系统，但是实现了部分类似操作系统的功能。主要是一种任务分配资源的机制，从而形成了一个简单多任务的操作系统
- mmWave Link——用于 RSS（Radar Sub-System）固件的驱动程序，并提供用于控制前端中每个硬件块的底层 API，通过 Mailbox 完成主/子系统之间的通信协议、毫米波前端的低电平控制；通过 Mailbox 与 BSS 通信（前端），实现 BIST（Built-in Self-Test，内部自检）子系统和主控子系统之间的通信协议
- mmWave Lib——处理关键算法程序，包含 FFT、CFAR-CA Detector、Angle estimation。提供典型雷达处理链功能；针对 C674x 进行了优化；加快客户开发速度，减少软件开发工作，实现有效的雷达处理链

- mmWave API——应用程序编程接口(Application Programming Interface)，是对 mmWave Link 的抽象层控制，并为应用程序提供用于配置前端雷达感测任务的简单 API。同时也处理 R4F MCU 和 DSP 子系统之间的同步和进程间通信
- Data Path - 数据通道配置模块

用于控制mmWave设备中的高速接口



上图为毫米波雷达数据路径层框图。其中：

DPM: DataPath Manage（数据路径管理器）——支持架构“可扩展性”方面的基础层

DPIF: DataPath Interface（数据路径接口）——定义了检测链中的标准接口点，如输入的 ADC数据、雷达数据立方体（RadarCube）、检测矩阵（Detection Matrix）、点云（Point Cloud）的格式标准

DPU: Data Processing Units（数据处理单元）——从一个接口点到另一个接口的数据转换功能

-距离处理（ADC 数据到雷达立方体） -多普勒处理（雷达立方体到检测矩阵） -CFAR 和 AoA（检测矩阵到点云）

DPC: Data Processing Chain（数据处理链）——“数据处理单元”的链称为数据处理链。例如：检测 DPC（从ADC 到点云）

(5) mmWaveLink 的 Modules

为了简单起见，TI的mmWaveLink框架能够划分为多个模块。这些功能包括设备控制、射频/模拟控制、ADC配置、数据路径(LVDS/CSI2)配置、FMCW线性调频脉冲配置等。下面列出了mmWaveLink框架中的各个模块：

1. Device - 控制毫米波雷达设备，包括：初始化，如：mmwave设备开机/关机，固件补丁下载，级联设备配置，如添加/连接多个mmWave设备
2. Sensor - RF/传感器配置模块>控制mmWave前端内的不同硬件块

mmWave前端有以下关键块：

1) Chirp时序器(雷达定时引擎) - 这个模块负责构建FMCW线性调频脉冲或帧的序列，并对定时引擎进行编程 2) Rx/Tx通道 - 这个定义了需要启用多少个Rx和Tx信道。并定义了成像雷达在级联模式下如何配置毫米波前端 3) Rx模拟链 - 定义接收信号的混合方式以及链中不同滤波器的配置方式 4) ADC和数字前端配置 - 这定义了中频数据如何数字化，以及如何在DSP或硬件加速器中对器进行采样以进一步处理。相同的ADC数据可以通过LVDS/CSI2接口发送到扩展处理器。

配置API可以进一步分类为：

mmwave静态配置，例如：Tx和Rx通道、ADC配置等 mmwave动态配置，例如：FMCW Chirp配置，profile配置 mmwave高级配置，例如：二进制相位调制、动态节能等 mmwave传感器控制，例如：帧的启动/停止

(6)AUTOSAR代码架构

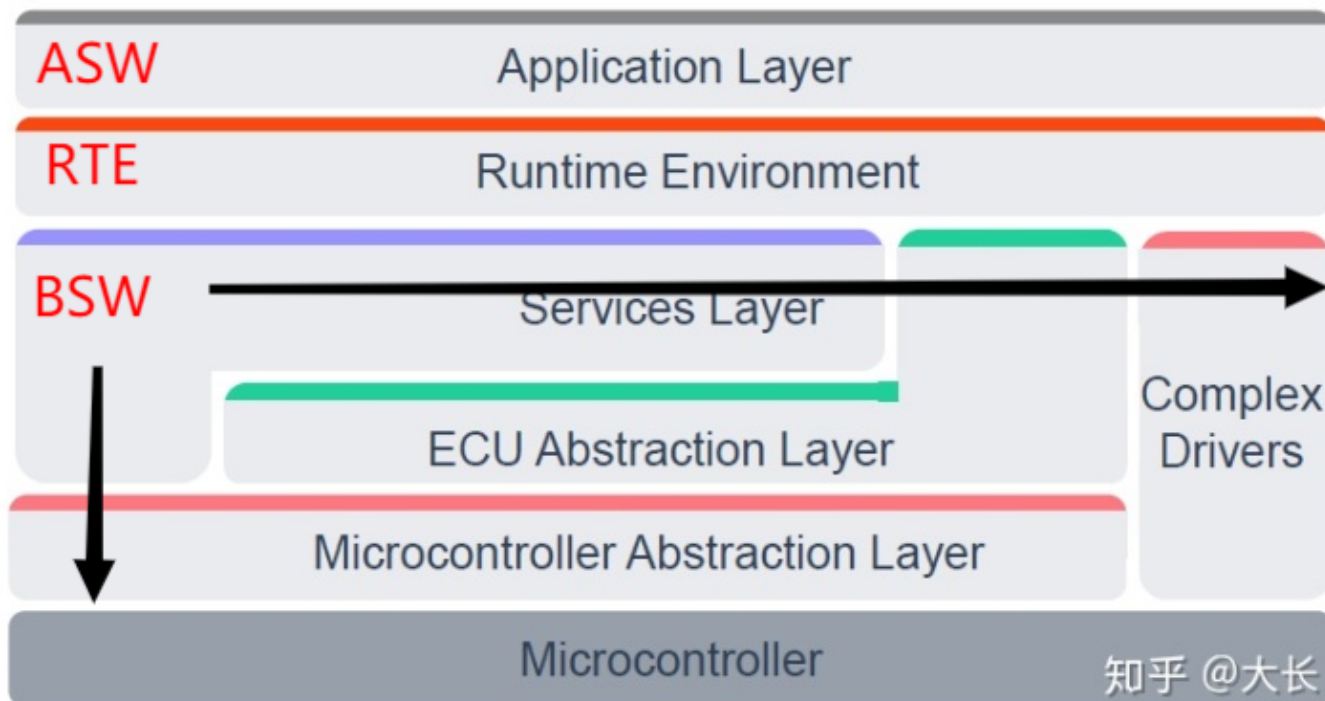
先来简单了解一下AUTOSAR软件架构，主要有三部分组成：ASW、RTE、BSW，举例来和大家说一下：

ASW大家可以理解为手机里的APP应用层；

RTE就是一个强大的转接器，用来衔接ASW和BSW的；

BSW可以理解为手机里的安卓操作系统，底层驱动；

这样分层的好处就在于做APP开发的不需要知道底层软件是如何工作的，不需要知道具体用什么芯片等硬件；从手机行业类比来看，BSW应该会很重要



Common							存放公共类型及接口	
Link							存放CMD文件	
Source							代码	
	Asw							
		Spm					信号处理算法模块	
			Fft	Fft.c, Fft.h	王菲	周喜明	FFT	10
			Cfar	Cfar.c, Cfar.h	王鹤雷		CFAR	1
			Range	Range.c, Range.h	王鹤雷	周喜明	测距计算模块	3
			Speed	Speed.c, Speed.h	王菲	周喜明	测速计算模块	3
			Doa	Doa.c, Doa.h	吴光智、王鹤雷		水平和俯仰角计算模块	10
			Rcs	Rcs.c, Rcs.h	王鹤雷、吴光智		RCS计算模块	15
		Dpm					数据处理模块	
			Cluster	Cluster.c, Cluster.h	王鹤雷		聚类算法模块	1
			Tracking	Tracking.c, Tracking.h	王鹤雷		跟踪算法模块	1
			SelfCal	Selfcal.c, Selfcal.h	王鹤雷		自校准算法模块	2
	Bsw							
		Task						
			Task_Init.c		秦念豪			
			Task_Chirp.c		秦念豪			
			Task_Frame.c		秦念豪			
		Cdd						
			Datapath	Datapath.c, Datapath.h	周喜明			2
			Adc	Adc.c, Adc.h	秦念豪			1
			Edma	Edma.c, Edma.h	周喜明			2
			Lvds	Lvds.c, Lvds.h	秦念豪			1
			MailBox	Mailbox.c, Mailbox.h	王开祥			1
			MemAlloc	Memalloc.c, Memalloc.h	周喜明			2
			Dem	Dem.c, Dem.h	王开祥			5
			Dsp				模式分发到各个接口	
				Dsp_Eol	吴光智			10
				Dsp_Normal	周喜明			10
		dss_main.c			秦念豪			
工程搭建		王开祥						