US282F--重构技术简介

陈亮

2017-4-11

重构需求

序号	市场需求	子序号	技术需求	实现状态	说明
1	加快客户开发	1.1	系统内存管理重构	√	应用空间统一改成2KB大小的bank。不再用宏定义划分内存使用区域，统一使用malloc申请
		1.2	Makefile简化	√	240个makefile文件减为4个。用户不再需要写makefile
		1.3	算法分配的内存空间管理	√	算法内存空间使用前先申请，不使用时用来做应用cache
		1.4	良好的代码规范	30%	修改及重写部分做了规范，未对整个SDK做修改
		1.5	IC各模块管理	√	重新移植了各驱动
		1.6	各模块间去耦	√	Bt切分成btservice和btstack，key,led,rtc分开，Common瘦身等
		1.7	软件分层	√	按调用关系重新调整Sdk目录
		1.8	MIPS与DSP的交互界面完善	√	Mips与dsp地址转化做了澄清与更新
		1.9	MIPS与DSP之间的角色定义及模块耦合	-	未修改。Mips仍做从属角色
		1.10	I2C驱动重写	√	
		1.11	Common模块优化	√	
		1.12	接口统一，说明完整	√	面向应用层的接口按posix标准做了整理，如文件系统

炬芯 Actions

重构需求

序号	市场需求	子序号	技术需求	实现状态	说明
2	规格更全面	2.1	歌曲兼容性测试	√	保持
		2.2	常规音箱 + TWS + Waves音效 + 新架构 + actions音效	√	TWS + Waves音效 OK
		2.3	TWS实现Linein	-	未移植
		2.4	微信语音支持	√	
		2.5	微信硬件（协议）支持（通过公众号读取蓝牙设备数据）	√	
		2.6	USB Audio支持24bit,96K及以上	√	需要接24bit DAC子板（通过I2S接口）
3	系统更稳定	3.1	约束或边界 的定义与测试	√	做了更多的边界条件测试，如蓝牙，文件系统，插拔等，发现的缺陷部分282a同存在
		3.2	自动化稳定性测试(长时间nor测试，避免掉码)	50%	NOR兼容性已完成，未做老化测试
		3.3	硬开关方案工作状态随机关机和快速开关机的防护	√	重写nor驱动，增加日志块机制
		3.4	ESD恢复流程(走恢复流程--继续做重启前的事情)	√	代码加载已变为2KB/次* n，此时有条件对2KB大小代码增加校验
		3.5	EMI降低（打电话，卡播歌，推歌,TWS场景，换歌操作菜单可切bank）	√	保持

炬芯 Actions

重构需求

序号	市场需求	子序号	技术需求	实现状态	说明
4	方便客户Debug	4.1	调试工具修改，可在win7运行	50%	ATD工具跟踪到使用第三方库时出现兼容问题，因无源码无法跟踪下去。当前在win7系统的电脑曾成功安装并使用ATD工具，需继续花时间找出与不成功机器的差异
		4.2	T卡实现串口打印	-	T卡在音箱中使用频率高不太有机会出让给打印，所以实用性不强。281已实现usb打印可考虑移植。
		4.3	增加串口输入控制台来debug（console）	√	可以查看os task，stack，中断，bank切换情况，内存，重启…
5	让量产简单高效	5.1	ATT更好支持	√	保持
6	让升级更方便	6.1	SDK前后版本兼容	√	引入Posix接口，保持sdk更新时接口不变
		6.2	OTA升级支持差分升级	50%	282f是混合编译无法实现差分，仍需完整包升级。不过对磁盘分区与升级机制做了更新，速度和安全性有提升
		6.3	EMI降低（打电话，卡播歌，推歌,TWS场景，换歌操作菜单可切bank）	√	保持
7	降低BOM成本	7.1	固件大小减小	√	降低20%
		7.2	Norflash驱动考虑可开放，客户增加小众nor支持	√	重写nor驱动，增加逻辑层，需要可开放

重构需求

序号	市场需求	子序号	技术需求	实现状态	说明
8	工具更好用	8.1	提升音效工具ASET易用性	√	ASET增加另存cfg_dae.txt功能，MODIFY增加把config.txt和cfg_dae.txt合并成config.bin功能，如此可减少客户调定音效后需发cfg_dae.txt给我们AE，打包出固件回发他们的过程，较大提升效率
		8.2	改善W公司音效的推广	-	需加大调音人员培训
		8.3	音效的推广case by case	-	每家客户对音效要求不同，需具体谈
		8.4	ATT工具进一步优化	-	保持
		8.5	ASQT针对TWS+Waves调整	-	保持
		8.6	手机APP工具	√	对OTA体验做了改善，且增加ihome的睡眠曲升级支持
9	支持工具可配，不用修改代码	9.1	可配置化（按键，显示等）	-	ROM级可配置化已不做需求，其他可配置化 保持

炬芯 Actions

技术特点

1. ap与os统一编译，ap起始地址不重叠
2. makefile优化
3. bank大小固定为2KB
4. 内存划分为core区，bank区，cache区，堆栈区，DSP区
5. 新增VDFS文件系统
6. 新增snapshot现场保存
7. 实现5层中断嵌套
8. OTA升级后加快启动速度
9. 增加fprof与编译预处理实现bank自动重排
10. rodata（const变量或字符串）指针传递检测及报警
11. 实现完整功能的malloc和free
12. bt重新整理分层

技术特点

13. btstack使用事件驱动（event）代替循环查询
14. 重写蓝牙回连机制
15. 标准C库实现
16. 重写nor驱动
17. 实现标准printf()
18. 增加console控制台增强调试能力
19. 增加mpu内存保护
20. 重写fat和exfat文件系统
21. 重写文件选择器增加歌曲序号选歌
22. 增加CPU,DSP动态调频
23. TWS对齐精度提升

技术特点说明

1. ap与os统一编译，ap起始地址不重叠

- 混编后地址依序增加，无大段跳空
  - ✓ 地址跳空会增加学习难度
- 减少编译**时间**
  - ✓ 282a是5分30秒，282f是**3分08秒**
- 减少代码量
  - ✓ 282a因ap可以独立编译，所以函数可以同名，但代价是某些函数在多个ap中要用到时可能ap间会各存在一份
  - ✓ 省去了每个ap加载处理（如缺页）的相关代码
  - ✓ 282a固件是1.7MB，282f是**1.3MB**
- ap与驱动地址唯一，可以使用mips-elf-insight单步调试
  - ✓ 282a因ap地址复用，用insight设置断点会重复断停
- 轻微加快运行效率
  - ✓ 统一编译减少了中间层级的调用和时间

技术特点说明

2. makefile优化

- 用户开发**不用再写一句脚本文件**
  - ✓ 让makefile对用户透明
  - ✓ 自动搜索新增源文件，加入编译
  - ✓ 只编译修改文件及依赖其的文件，减少编译时间
- 大幅降低学习成本
  - ✓ 脚本文件晦涩难懂，软件基础要求高
  - ✓ 不再需要了解并熟记整个SDK的**内存分布**才能动手开发
- 大幅降低开发成本
  - ✓ 新建应用或驱动不用添加脚本文件
  - ✓ 精致控制代码运行地址，会让开发受限并暗藏**隐患**（如发生内存覆盖不自知）
  - ✓ 一键编出固件，不用分步骤敲命令行
  - ✓ 支持板型选择
  - ✓ 支持发布包生成
- 更友好的编译界面
  - ✓ 简洁提示编译阶段，让用户了解编译进程，最后提示编译时间
  - ✓ **代码检查**提示

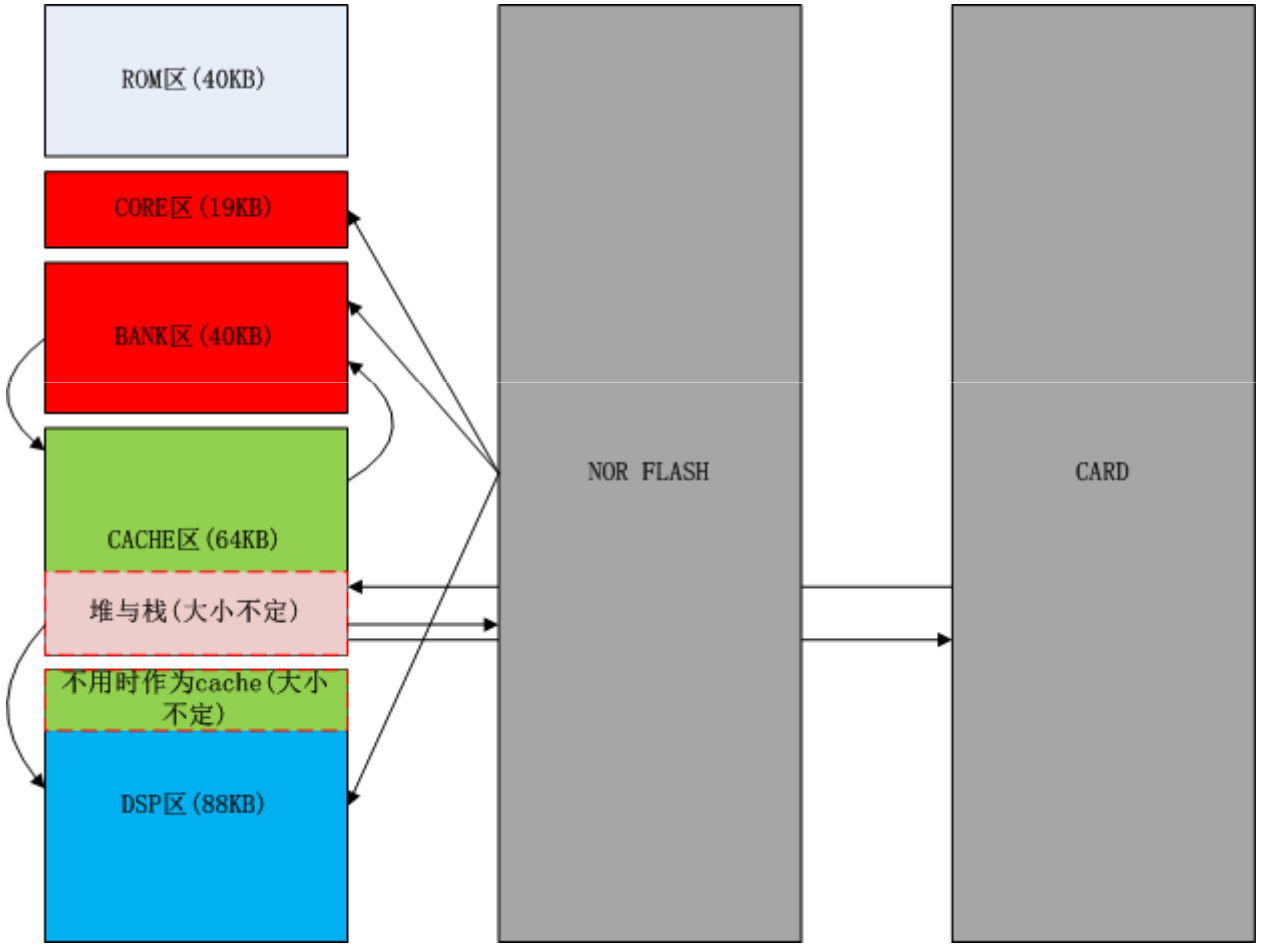
技术特点说明

3. bank大小固定为2KB

- 系统统一管理bank切换
  - ✓ 减少bank大小的多样性，保持管理程序简洁
  - ✓ ap和驱动不用再自行管理bank切换（不用担心哪些bank可以使用，哪些bank不能使用），减少学习与开发时间
  - ✓ 用到时才加载，一开始不需把过多代码load进内存
  - ✓ 代码以2KB为单位load进内存，使为**代码增加校验**成为可能，增强**抵抗ESD**干扰能力
- 22个page Reg最大支持22*2KB=44KB的代码运行区空间
- 实现**软件cache**
  - ✓ mips与dsp没用到的内存全部以2KB为单位转化为cache page来管理，提高内存使用效率
- 要求单文件源码编译后不能超过2KB
  - ✓ 超过的需要拆分源码为多个文件（比如以函数为单位拆）
  - ✓ 应用层代码一般90%以上不会超过2KB，算法代码较多见
  - ✓ core代码常驻，没有这个限制

技术特点说明

4. 内存划分为core区，bank区，cache区，堆栈区，DSP区



技术特点说明

4. 内存划分为core区，bank区，cache区，堆栈区，DSP区

- 系统代码和全局变量运行和存放于core区
- ap和驱动都运行于bank区
  - ✓ 以2KB为单位，称为1个page
- 根据将运行代码的地址从nor搬代码进入运行区的相应位置
- 原位置代码备份到cache区
  - ✓ 使用**最近最少使用**替换算法
  - ✓ 2KB ram 2 ram拷贝一般耗时**25.6us**(mips跑100MHz)，消耗较小
- 堆与栈与cache区复用，从高地址往低地址方向申请page来使用
  - ✓ cache从低地址往高地址申请page来使用
- DSP区使用前需申请
  - ✓ 不使用时系统自动拿来当cache page使用

技术特点说明

5. 新增VDFS文件系统

- norflash中除代码区以外的空间统一用vdfs管理起来
  - ✓ vdfs与sdfs相似，每目录项占 32字节，其中名称使用7字节。有字段记录文件**读写属性**
  - ✓ **vram**是vdfs中的**一个文件**
  - ✓ 客户定制文件如ihome的催眠曲是vdfs中的一个或多个文件
  - ✓ vdfs的文件**可用OTA替换**
  - ✓ 增加方案定制的灵活性

6. 新增snapshot现场保存

- 正常关机需要
  - ✓ 不需蓝牙应用关注关机前的现场保存
  - ✓ 开机后迅速恢复关机前现场
  - ✓ 原理是对malloc出去的内存进行镜像（约50KB+）
  - ✓ 使用**lzo**压缩算法（压缩率约为50%）
  - ✓ 存于norflash中某一物理块中
  - ✓ 耗时约500ms

技术特点说明

7. 实现5层中断嵌套

- 为了提高系统响应实时性
  - ✓ L0: 任务n |----------|
  - ✓ L1: 普通中断 |----------|
  - ✓ L2: 缺页异常 |----------|
  - ✓ L3: 不可屏蔽中断 |----------|
  - ✓ L4: 缺页异常 |----------|
  - ✓ L5: 蓝牙中断 |---------|

  - ✓ 其中L3: 不可屏蔽中断----Time0中断，**Usound中断**
  - ✓ 282a为了保证usound实时性，在bank切换函数中增加回调函数

炬芯 Actions

技术特点说明

8. OTA升级后加快启动速度

- OTA使能条件
  - ✓ Norflash容量 > 2倍代码容量，如1.5MB代码需使用4MB nor
- 代码从nor尾巴往前保存
  - ✓ 蓝牙传过来的代码从4-1.5=2.5M位置开始保存
  - ✓ brec写到4M nor最后一个扇区
  - ✓ 启动时读nor第一个扇区和最后一个扇区，根据新旧决定运行哪份代码，**耗时为0**
  - ✓ 282a OTA蓝牙传过来的代码先保存到后2MB，启动时再拷贝前2MB，耗时约40s。如果拷贝过程掉电，下次上电重新开始拷贝。

技术特点说明

9. 增加fprof与编译预处理实现bank自动重排

- 整个SDK代码以2KB为单位依次占用bank号往下分配运行地址
  - ✓ bank号与运行地址可直接换算
- bank的合理排布决定运行时的性能
  - ✓ 把某一场景运行时所需的代码（简称**热代码**）尽量放在同一个bank内，其次放在相邻bank内
  - ✓ 热代码错开bank分配，能减少运行时的bank切换
  - ✓ 发生bank切换时首先从cache区寻找，miss后再到nor读
- 通过cygwin命令行：**make fprof=1**打开profile功能
- 在预处理时在每一个函数头插桩，以计数函数被调用次数
- 操作DVB进入需优化场景(如蓝牙播放，卡播放)，正常使用10分钟，统计结果通过打印输出到PC
- 修改串口打印接收工具SerialUtility，增加接收计数统计结果并生成文件的功能
- 把文件拷贝到SDK指定目录，正常make即可生成经过优化的bank排布

技术特点说明

10. rodata（const变量或字符串）指针传递检测及报警

- rodata访问miss问题
  - ✓ 该问题较容易发生且隐匿性强，存在随机性，不易debug
- 问题形成原因
  - ✓ rodata通常被当成代码与.text段连续放在一起
  - ✓ 存在bank切换的系统理论上都存在rodata miss问题
  - ✓ 其机理是caller把本bank内的rodata指针作为参数去调用callee，当callee所处bank的低位地址与caller所处bank的低位**地址相同**时，就会发生访问miss
  - ✓ callee按参数传入地址去访问rodata，获取的数据是不确定的
- 解决方法
  - ✓ 282f新增**编译预处理**脚本，对将编译源码先做rodata检查，当发现上述问题时，自动做出修正并给出提示，再继续正常编译。开发者可根据提示自行决定是否在代码中修正
  - ✓ 修正方法是把rodata变量转化为**局部变量**（使用栈空间），传递指向栈的指针。代价是对栈空间有消耗。注：带d-cache的硬件系统可以完美解决此问题

技术特点说明

11. 实现完整功能的malloc和free

- 采用buddy算法管理内存
  - ✓ 采用位图记录，分配与回收**非常快**，不存在**碎片**问题
  - ✓ 分配最小单位是16bytes
  - ✓ 内存管理损耗是128/2048 = 6.25%
- 堆栈与cache区复用ram
  - ✓ 先向cache区申请出一个page(2KB)，再继续在该page上分配堆与栈
  - ✓ 函数运行完释放堆与栈，归还该page成为cache
  - ✓ cache，堆，栈充分复用ram，**大大提升内存使用效率**
- 增加内存使用**监测**
  - ✓ 堆需malloc与free配套使用，不malloc就使用指针，或不注意free，或free了别人申请的指针都容易造成内存泄漏，内存非对齐访问，代码区被写等问题
  - ✓ 为此282f增加了多种内存使用监测（后述），包括申请者，申请量，使用量，剩余量，非法释放，0地址访问等，出错即打印现场，有效保证了debug效率

炬芯 Actions

技术特点说明

12. bt重新整理分层

- 自上而下划分为btservice，btstack，btdriver三层
  - ✓ btservice为协议栈之上的面向服务的层次
  - ✓ btstack为蓝牙协议层（IVT库）
  - ✓ btdriver为蓝牙controller驱动层
  - ✓ 282a把btdriver放到了case，现转放到platform (psp)中
  - ✓ 282a把btservice和btstack混合到一起叫btstack，现拆开
- 层次分明减少耦合，方便debug
- 层次分明易于应对**开放需求**
- 分层是实现btstack（IVT库）**状态机自转**的需要（见下一小节）

技术特点说明

13. btstack使用事件驱动（event）代替循环查询

- 282f为btstack新开了独立线程响应btservice需求
  - ✓ 282a是在btservice的loop循环中查询标志来调用btstack
  - ✓ 282f btstack内可使用消息队列及事件维持自身状态机运转
  - ✓ 事件响应能解决任务循环内阻塞问题
  - ✓ 最大好处是提升了**响应实时性**
- 优化内存使用
  - ✓ btstack内有**大量**小内存（几字节到几十字节）使用需求，正好可以通过malloc，free满足，而不是静态分配，提升了内存使用效率
  - ✓ btstack内的收发大buffer通过动态分配**减少了一半的内存**使用（282f是3发3收每个628B共4KB，282a是H5:4KB+service:4KB）。282f通过直接传递buffer指针也减少了内存搬运

技术特点说明

14. 重写蓝牙回连机制

- 蓝牙断开后回连**过快**与**过慢**皆有问题
  - ✓ 回连过慢会使使用体验下降
  - ✓ 回连过快碰到某些手机也会自动回连时将发生冲突，造成回连不成功，出现**手机兼容性**问题
- 回连前增加“过滤”处理
  - ✓ 282c耳机方案做过优化，已同步到282f上

技术特点说明

15. 标准C库实现

- 字串操作接口
- 标准输入输出接口
- 类型转换接口
- 变参接口
- 文件系统接口
- 信号量接口
- 线程创建删除接口
- 标准C库接口有利于普通开发者使用SDK
  - ✓ 学校学习时使用过
  - ✓ 开发界面友好增强**开发者信心**，减少理解歧义与错误发生
- ✓ 各接口代码占用nor空间
  - ✓ 由于SDK重构接口变化，原放于5116 **BROM**的代码90%已不适合使用，一些本应常驻的代码在282f仍跑在BANK区，占用内存空间，存在被切换可能，如能把这些代码转化为ROM CODE（改一版BROM），相信282f**性能**还会有所提升

炬芯 Actions

技术特点说明

16. 重写nor驱动

- 增加逻辑层
  - ✓ 实现逻辑物理映射表
  - ✓ 使用日志快做写时备份，增强写安全
- 重新设计了nor分区
  - ✓ 只划分为物理区和逻辑区
  - ✓ 物理区用于存放代码，逻辑区用vdfs(自有文件系统)管理
  - ✓ 过去常用的vram区将变成是vdfs中的一个文件
  - ✓ 只要nor空间有剩余，可以通过OTA给nor更新任何预存文件，比如可以通过OTA更新出厂时的崔眠曲，曲数目不限（ihome规格。282a只允许更新2首）
- 新增snapshot功能
  - ✓ ESD造成重启的现场恢复处理
- BANK代码校验
  - ✓ 为每一个2KB大小的代码增加checksum
  - ✓ 从nor加载代码进内存时做校验，发现不一致即从nor重读
- nor擦除延时处理
  - ✓ nor擦除耗时300ms，运行时响应会影响实时性，统一放到每次开机处理

技术特点说明

17. 实现标准printf()

- 使用va宏簇实现变参函数
  - va_list，va_start()，va_arg ()，va_end ()
  - 开发人员使用频度最高的接口--**打印接口**
  - 282a的打印接口自定义，不止一个。需培养使用习惯
  - 实现了多级打印

18. 增加console控制台增强调试能力

- 打印用的uart tx功能，console用的是**uart rx功能**
  - 可以在PC串口工具中敲入命令行，实时了解OS的运行情况
  - 命令以#开头，目前实现有19条命令，具体可通过**#help**查看
  - 在现有BANK机制下，对这种**无实时性要求**的应用，可以异常简单地添加更多想加的命令，无任何约束
- 配套使用自写串口工具SerialUtility.exe
  - 可记忆输入的串口命令
  - 可根据打印的地址查找对应的函数
  - 可打印蓝牙收发的2进制数据

技术特点说明

19. 增加mpu内存保护

- mpu是5116的一个创新
  - mpu全称:Memory Protection Unit，用于对内存访问的**监控**，对于**异构多核**的代码运行与调试有重要作用
  - 指定一个区域纳入监控，并且只允许特定master访问，其他master访问即发生中断通知系统
  - 282f使能了mpu，并把0地址， bank区设为只读(load代码时打开一下写)等等，对于驱动调试，mips和dsp**联调**等起到了良好的作用（dsp写mips内存， mips写dsp内存造成无故跑飞在模块初期屡见不鲜）
  - mpu的使能有助于**下游开发者**更快的debug，收敛方案问题

炬芯 Actions

技术特点说明

20. 重写fat和exfat文件系统

- 修正旧有文件系统不合理地方
- 优化**性能**
  - ✓ 根据SD卡物理读写特性设计buffer，减少buffer间拷贝
  - ✓ 最大可以同时打开6个文件
- 实现标准文件系统接口

21. 重写文件选择器增加歌曲序号选歌

- 更简易的接口
  - ✓ 减少nor的使用，扫描后生成的数据存放在卡上或U盘上
  - ✓ **数字点歌**，上层不关心文件系统目录操作等，完全按文件序号点歌播放
  - ✓ **循环模式**不再由文件选择器提供，而是上层实现
  - ✓ 上层开发容易调整扫描文件总数，容易调整过滤的**文件后缀**名
  - ✓ 磁盘上可能有很多文件，扫描完需要比较长时间，需要提供一种**机制**扫描一小部分文件后，music可以开始播歌，扫描程序继续再完成扫描

技术特点说明

22. 增加CPU,DSP动态调频

- **CPU调频原理**
  - ✓ 快升：cpu idle < 20%时升到最高档频率
  - ✓ 慢降：cpu idle > 60%时降低一档频率
- **DSP调频原理**
  - ✓ DSP可通过dsp软件编写idle函数(或任务)，在里面记录dsp的忙闲，从而实现动态调频。这需要算法部新增模块有一定工作量，目前282f没有这样实现
  - ✓ 当前在mips端找到了能**间接表征**dsp忙闲的依据，即根据**asrc** buffer中的码流剩余量来推算dsp的忙闲，如果剩余很多数据说明dsp很闲（产出大于消耗），如果剩余很少数据说明dsp很忙
  - ✓ 根据忙闲动态调整dsp频率，达到既保证**性能**又降低**功耗**目的
  - ✓ 过去根据经验值设定dsp频率，大多时候能保证性能，但功耗非最优，个别时候性能也不一定能保证（遇到特殊码流时）

炬芯 Actions

技术特点说明

23. TWS对齐精度提升

- **根据送DAC采样点数同步**
  - ✓ 主，副箱送DAC点数超过24点差异时(假设为C)需reset中间件
  - ✓ 原“将送采样点数” 从DSP获取，假设为A，然后取$C=A_{主}-A_{副}$
  - ✓ 现通过读取DAC FIFO“剩余采样点数”假设为B，取$C=(A_{主}-B_{主})-(A_{副}-B_{副})$

炬芯 Actions

Thanks!

炬芯(珠海)科技有限公司
珠海总公司
地址：广东省珠海市高新区科技四路1号
电话：0756-3392353
传真：0756-3392251　邮编：519085