****



**YICHIP提示音模块**

单击或点击此处输入文字。

Yichip Microelectronics

©2022

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 2022-01-11 | Mengke.Feng | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 提示音模块说明 4](#_Toc30379)

[1.1 提示音需求 4](#_Toc22557)

[1.2 结构优势 4](#_Toc2967)

[1.2.1 兼容性 4](#_Toc12009)

[1.2.2 快速性 4](#_Toc31775)

[1.3 结构劣势 4](#_Toc23978)

[2 提示音的生成 5](#_Toc20851)

[2.1 SBC压缩参数 5](#_Toc4360)

[2.2 实测数据 6](#_Toc5358)

[2.2.1 不同采样率下测试 6](#_Toc18171)

[2.2.2 不同压缩值测试 6](#_Toc9863)

[3 提示音的存储 6](#_Toc23033)

[4 软件提示音 8](#_Toc28624)

[1、 硬件提示音 8](#_Toc27256)

[5 业务调度流程 8](#_Toc8168)

图目录

**错误！未找到图形项目表。**

表目录

**错误！未找到图形项目表。**

# 提示音模块说明

提示音是TWS耳机产品中重要的功能需要，可以极大的提高用户操作体验。在不同的使用场景下，耳机通过播放不同的提示音，以告知用户当前耳机的工作状态，提高人机交互的体验。

根据芯片在使用过程中的资源分配（flash空间），提示音是在代码烧录的过程中和代码一起烧录到芯片中，为了使用的简洁性，有一段特定的空间作为提示音的存储空间。提示音在flash的起始地址保存在变量mem\_vp\_addr中。

## 提示音需求

在耳机使用的过程中，用户可以通过按键实现人机交互，在交互的过过程中，如何确保已经当前指令已经完成，当前的业务场景处于何种模式，这时就需要一个提示性的功能，于是就有了提示音这一功能。

## 结构优势

### 兼容性

为了配合第三方工具（OTA和配置工具）的使用，我们将提示音的在flash中存储的其实位置保存在一个变量当中，第三方工具使用只要找到该变量即可知道地址，提高兼容性。

### 快速性

用户在使用耳机进行OTA升级时，用户要求OTA速度足够快速。如果将提示音放在代码中进行处理(音频数据定义为const也可以实现功能)，这样做代码在每次编译的时候都会编译提示音，同时代码的体积也会变大；但在实际使用过程中，提示音的改变是很少的，每次编译都会增加编译时间，体验不是很好。

为配合OTA，这种结构设计可以选择不升级提示音，从而减少了代码体积，提高了OTA升级的快速性。

## 结构劣势

这种结构设计中提示音的分布是一次分布的，如果在提示音的生成过程中某一个提示音的长度或者地址的偏移量(都是4byte)丢失了一位，就会导致后面的所有数据都会出现错误。

在下一代产品中，可以针对这种问题调整分布结构，可以在每个提示音的头部加入特定的handle等。以后再讨论

# 提示音的生成

在选择可用的提示音资源时，可以选择的音源格式有pcm格式、MP3格式、WAV格式以及SBC格式等。由于原始格式的音频占用的空间比较多，芯片内部的空间有限，为了节约空间，在实际使用的过程当中可能会对数据进行先压缩，使用过程中再解压的过程。在TWS中暂时是将其格式的音频先转换为pcm格式，在将pcm格式的音频经过SBC编码，将编码后的音频存入芯片的flash中。为了使用方便，音频格式的转换已经写好了脚本，根据需要在运行脚本是输入不同的参数即可。

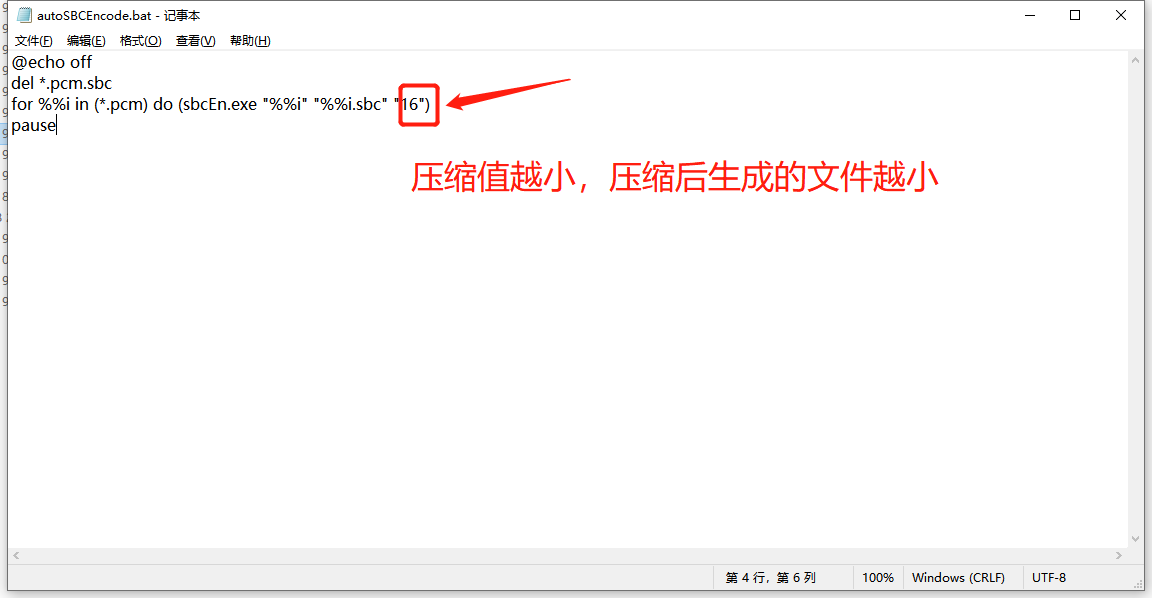
autoCreatPCM.bat可以将目录下的MP3以及WAV格式的音频转换成PCM，autoSBCEncode.bat可以将目录下的pcm文件进行SBC编码，压缩以节省空间,最终音频数据会生成一个满足芯片烧录的.dat文件。利用Perl脚本自动计算编码后的各个提示音的大小，然后按照特性的格式生成.dat文件。

注：使用autoCreatPCM.bat需要安装好FFmpeg，FFmpeg的使用查看FFMPEG之音频文件格式转换与PCM解压缩.docx。

## SBC压缩参数

压缩脚本autoSBCEncode.bat和sbcEn.exe以及被压缩的xx.pcm文件要放在同一文件夹下，双击脚本文件autoSBCEncode.bat，会生成压缩文件xx.pcm.sbc。

编辑脚本文件autoSBCEncode.bat，可设置压缩值，压缩值越小，生成的压缩文件越小，音质也越差，所以我们在压缩时要在压缩值与音质之间找一个平衡点。



## 实测数据

### 不同采样率下测试

表 1摇篮曲不同采样测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| 摇篮曲，压缩值为22,mp3格式大小835KBytes | | | |
| 采样率 | 音乐情况 | 音质 | .sbc文件大小 |
| 8k | 人声清晰 | 优 | 大小：679KBytes |
| 4k | 人声清晰 | 优 | 大小：339KBytes |
| 2k | 人声不清晰 | 一般 | 大小：169KBytes |
| 1.5k | 人声不清晰 | 一般 | 大小：127KBytes |
| 1.2k | 人声不清晰 | 一般 | 大小：101KBytes |
| 1k | 没有人声 | 一般 | 大小：84KBytes |
| 0.5k | 没有人声 | 差 | 大小：42KBytes |

### 不同压缩值测试

表 2摇篮曲4k采样率

|  |  |  |
| --- | --- | --- |
| 摇篮曲，4k采样率 | | |
| 压缩值 | .sbc文件大小 | 效果 |
| 16 | 260KBytes | 音质一般，有人声，轻微沙哑背景音 |
| 11 | 195KBytes | 音质差，有人声，有沙哑背景音 |
| 5 | 117KBytes | 音质差，无人声，声音卡顿严重 |
| 3 | 91KBytes | 音质差，无人声，声音卡顿严重 |
| 1 | 65KBytes | 无声音 |

# 提示音的存储

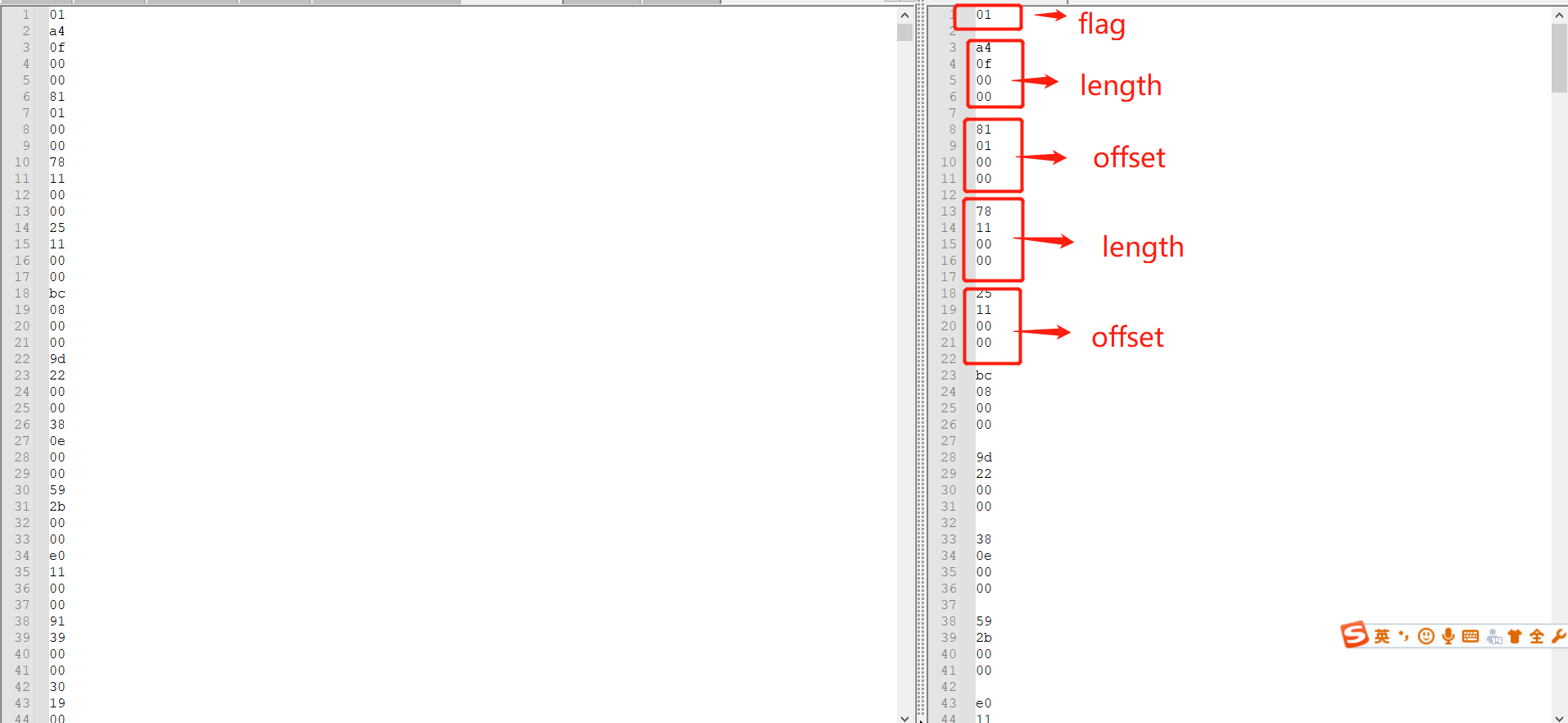


提示音在Flash的存储格式如上图所示。包含1Bytes的flag，4Bytes的长度（length，代表该文件总的长度）和4Bytes 的偏移量(offset,代表文件的起始地址相对于提示音存储的base\_addr的偏移量)，根据提示音的个数，后面依次是该提示音的length、offset，直到结束。

表 3提示音分布说明

|  |  |  |  |
| --- | --- | --- | --- |
| 变量名 | 大小 | 数值 | 说明 |
| Flag | 1byte | 01 | SBC格式 |
| 02 | PCM格式 |
| 03 | MP3格式 |
| 04 | WAV格式 |
| Length | 4byte | 0xxxxxxxxx | 提示文件的长度 |
| Offset | 4byte | 0xxxxxxxxx | 提示音的偏移量 |
| .... | .... | .... | .... |
| 提示音文件1 | Length bytes |  |  |
| .... | .... | .... | .... |

实例说明：

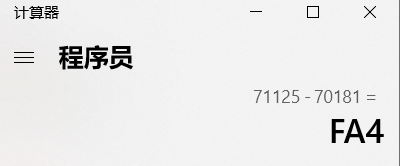


提示音在flash存储的开始地址在变量mem\_vp\_addr中，该变量的值为0x70000.

第一个提示音的长度为：0xfa4,提示音的起始地址为 0x70000+0x0181 = 0x70181

第二个提示音的长度为：0x1178,提示音的其实地址为 0x70000+0x1125 = 0x71125

验证一下第一个提示音的长度是否正确：0x71125 - 0x70181 = 0xfa4,符合data文件中的数据。



# 软硬件件提示音

## 软件提示音

软件提示音的播放流程全部是软件行为，是可控的。在使用软件提示音进行播放时，由于芯片只有一路DAC，所以播放提示音时需要对DAC进行独占使用，这样就会带来一个问题，在播放提示音时，如果其他业务正在使用DAC的资源，这时就需要先停止其他资源使用DAC。如播放歌曲过程中需要将歌曲停止（丢弃这段时间的音频流），播放完提示音后再恢复歌曲播放。同样对应通话也是一样的情况，这样会牺牲用户体验。此外播放完提示音后，需要恢复其他被打断业务。

## 硬件提示音

硬件提示音的效果跟软件提示音的效果基本相同，两者最大的区别是硬件提示音可以弥补软件提示音的部分缺陷：可以解决软件提示音机制需要打断正常音乐播放的问题。但是播放硬件提示音时，提示音的采样率必须和当前播歌时的dac的采样率必须相同。在TWS音频的同步问题上，使用硬件提示音可以使当前的音频流不被打断，很大程度上能够避免由于播放提示音导致的主从耳音频不同步问题。

### 软硬件提示音各自优劣

#### 软件提示音优势

在任何场景下可以独自播放，不受任何限制。

#### 软件提示音劣势

1. 软件提示音播放时需要打断当前DAC正在进行的业务。
2. 软件提示音的播放会对TWS的同步业务造成影响。

#### 硬件提示音优势

1. 硬件提示音的播放不会打断当前DAC正在进行的业务。
2. 硬件提示音对TWS的同步业务几乎没有影响。

#### 硬件提示音劣势

1. 硬件提示音播放时当前DAC必须正在工作（听歌或者通话），单独不能正常工作。
2. 硬件提示播放时的DAC采样率必须要与当前正在进行业务（听歌或者通话）DAC的采样率保持一致。
3. 在硬件提示音播放过程中，如果听歌或者通话结束，提示音没有播放完时此刻也会停止播放，会导致提示音的播放性不完整。

# 业务调度流程



在业务调度上，软件提示音和硬件提示音的调度方式都一样，都是通过轮询的方式进行的：

，

轮询检测是否有提示音的数据，性判断提示音对应的队列是否为空。在使用的时利用接口

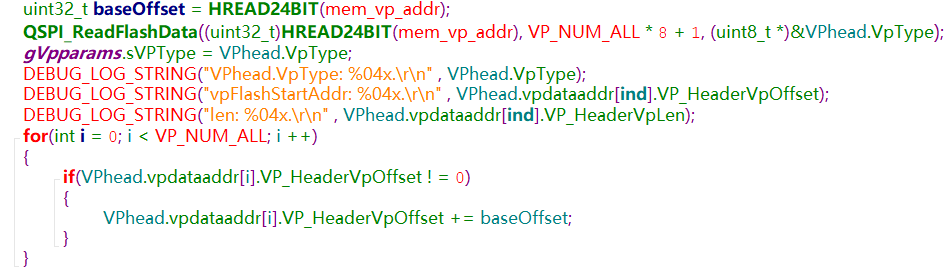
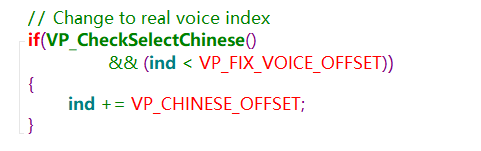


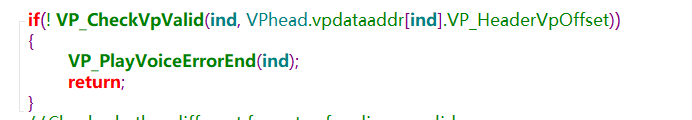
传入提示音的索引值，将提示音的信息做入队操作。在进行提示音的播放时，先进行提示音的中英文判断，如果当前提示音支持的模式为中文模式，则根据提示音的索引值



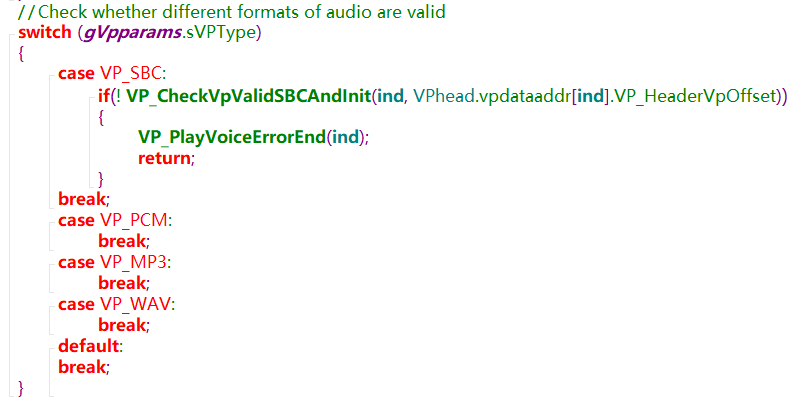
跳转到对应的中文提示音区域。(注意：默认的提示音分布是英文提示音、中文提示音以及不区分中引文的特殊提示音。)。

播放提示音的流程如下：

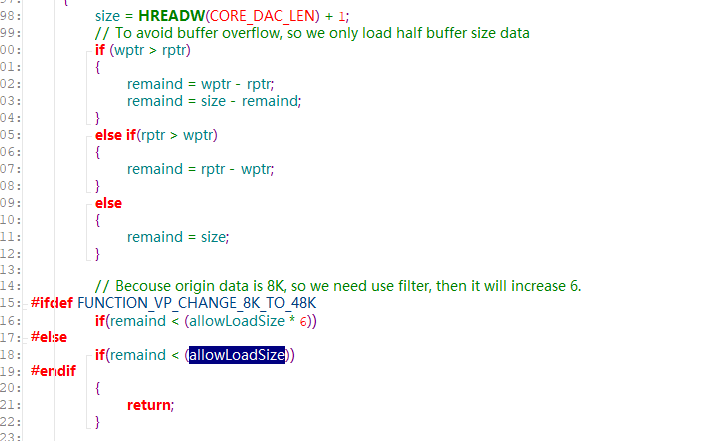
1. 从flash中读取存取的提示音头文件，即.dat文件头保存的提示音的编码格式flag、提示音的长度length以及提示音的偏移量offset。
2. 判断当前提示音播放处于何种模式：中文或者英文。如果是中文模式则根据中文提示音与英文提示音之间的偏移跳转到对应的中文提示音区域
3. 检查要播放的提示音是否存在及正确



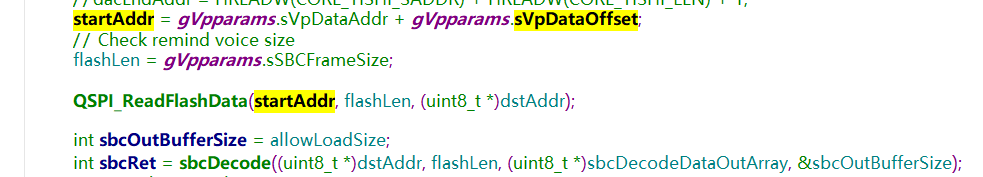
1. 根据读取的flag，检查当前文件是否满足该格式的数据包格式，为了以后的兼容使用，可能除了支持BSC格式，后续还需支持MP3、WAV等。



5、判断当前DAC剩余的buffer看是否放的下要送入DAC的数据，如果当前的空间不足，则继续等待，知道buffer足够再继续往DAC送数据。若播放的是48K的音频，他需要的buffer是8K音频buffer的6倍。



6、从flash相应的位置读取提示应的数据进行播放，将读取的数据进行播放。若走软件提示音时，应该先将目前正在占用DAC资源的应用停下，等播放完提示音后再回复之前的业务。



在现有的TWS耳机中，提示音的编码格式采用的是SBC格式的，在从flash读取数据时，每次读取的长度是由它的参数决定的，

在该函数中通过解析SBC数据包得到(注意：sbc编码后的数据每包都有包头)。

将读取到的数据进行解码，解码后的数据是pcm格式，直接可以通过DAC播放。



1. SBC的解码

根据读取到的数据包头，判断当前的数据是否为BSC格式，如果是BSC格式，就开始进行数据的解码操作

