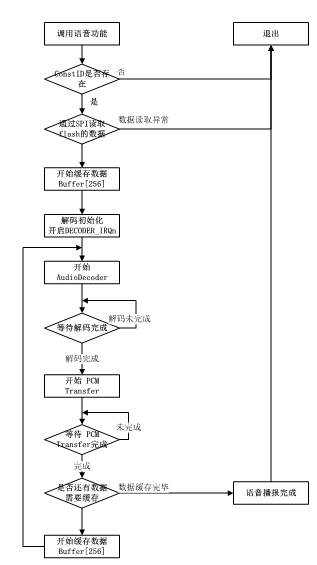
**语音提示功能 软件详细设计**

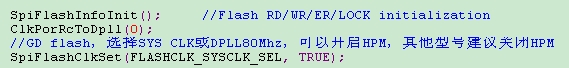
**版本更新记录**

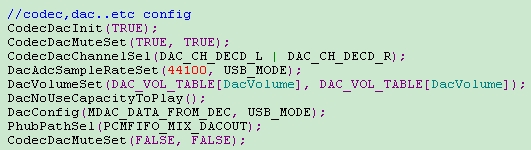
|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本号 | 更新内容 | 修改人 |
| 2016-5-10 | V0.1 | 1. 初版； 2. 在Audio Decoder的基础上，实现了语音提示功能； | KK |
| 2016-6-2 | V0.2 | 1. 重新梳理功能模块的代码，重新整理文档； 2. 新增SD卡升级，通过升级将语音文件写入Const区域，便于进行测试； | KK |
|  |  |  |  |

1. **功能开发说明**
2. 基于O18B SDK的基础上，实现了Sound Remind功能，整理输出独立的Example；以便后续在其他项目上可直接进行移植；
3. 由于语音文件保存在CONST DATA区域，在程序下载时，不会对该区域进行烧录，所以代码中增加了SD卡升级功能，将语音文件制作成.MVA文件，可通过SD卡进行升级；
4. **Example软件测试方式**
5. 通过UART发送指令：
6. ‘r’：插入SD卡，读取SD卡的升级文件，并升级程序；
7. ‘n’：播放提示音(总共6条[扫地机项目]，可以循环播放)
8. 测试前提，需要先将语音文件烧录到FLASH中；
9. **语音提示功能涉及范围**
10. 从FLASH中对语音包的读取；
11. 语音文件的解码；
12. DAC转换输出；
13. 文件升级；
14. **Sound Remind 流程图**



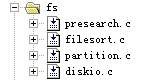
1. **语音升级文件制作方法**
2. 语音文件可以通过 “文字转语音工具” 进行录制，然后通过GoldWave工具进行剪切；对录制的语音文件进行调整，做成 .mp3 文件；
3. 将MP3文件导入CONST数据包，制作成MVA程序，可以参考 MVAssistant Help相关文件；
4. **软件移植说明**
5. 导入sound\_remind.c和sound\_remind.h；
6. 上电时，需要同步初始化SPI，ADC，DAC，采样率，音量等设置；

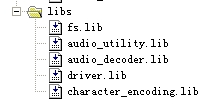




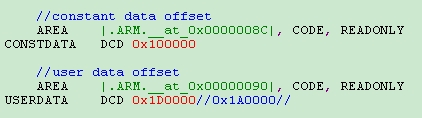
以及 DecoderTaskInit()；

1. 在使用语音功能时，直接调用 SoundRemind 函数接口；
2. 主循环需要调用 DecoderTaskEntrance()；[对音频流进行解码处理]；
3. 工程中需要包含如下文件和库；

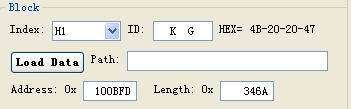


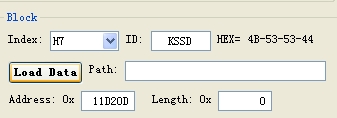


1. 语音包存放在Const Data区域，由于根据语音包的实际大小，调整constant data区域的大小，修改user data offset的值，以防覆盖User Data数据和调用出错的问题出现。



1. constant data所占用的实际空间，可以在MVAssistant工具中获取，如下图：





1. 语音MVA文件，不能直接删除语音问题，可能会导致文件损坏，但是可以直接在上面增加语音；
2. 在修改语音文件时，需要通过的更新代码中的 SoundID 和 SoundTable 数据；