目 录

[摘 要 I](#_Toc482733943)

[Abstract II](#_Toc482733944)

[1 绪论 1](#_Toc482733945)

[1.1 主要任务 1](#_Toc482733946)

[2 系统总体方案及硬件设计 1](#_Toc482733947)

[2.1 设计思路 1](#_Toc482733948)

[2.2 工作原理 2](#_Toc482733949)

[2.3 系统硬件设计 2](#_Toc482733950)

[2.3.1 电源设计 3](#_Toc482733951)

[2.3.2 DDR2电路设计 4](#_Toc482733952)

[2.3.3 声卡电路设计 4](#_Toc482733953)

[3 系统软件设计 5](#_Toc482733954)

[3.1 AWorks简介 5](#_Toc482733955)

[3.2 emWin简介 6](#_Toc482733956)

[3.2.1 一般特性 6](#_Toc482733957)

[3.2.2 图形库 7](#_Toc482733958)

[3.2.3 字体 7](#_Toc482733959)

[3.2.4 字符串、值输出例程 7](#_Toc482733960)

[3.2.5 窗口管理器(WM) 7](#_Toc482733961)

[3.2.6 PC界面外观的可选小工具 7](#_Toc482733962)

[3.2.7 支持触摸屏和鼠标 8](#_Toc482733963)

[3.2.8 PC工具 8](#_Toc482733964)

[3.3 系统任务设计 8](#_Toc482733965)

[3.3.1 GUI任务 8](#_Toc482733966)

[3.3.2 触摸屏任务 8](#_Toc482733967)

[3.3.3 音乐播放任务 9](#_Toc482733968)

[3.4 音频解码 9](#_Toc482733969)

[3.4.1 WAV格式 9](#_Toc482733970)

[3.4.2 MP3格式 11](#_Toc482733971)

[3.4.3 FLAC格式 13](#_Toc482733972)

[3.4.4 音乐播放流程 14](#_Toc482733973)

[3.5 LRC歌词解析 15](#_Toc482733974)

[3.5.1 LRC格式简介 15](#_Toc482733975)

[3.5.2 LRC歌词解析 15](#_Toc482733976)

[参考文献 17](#_Toc482733977)

摘 要

本课题拟设计一种基于AWorks嵌入式系统软件开发平台，使用i.MX283处理器，搭载emWin界面库的高性能音乐播放器。

该播放器通过软件解码音频文件，codec芯片做模数转换实现音乐的播放，除支持WAV、MP3与FLAC格式的播放外，还能播放USB设备以及SD卡内的音乐文件，且有美观的界面以及优良的人机交互性能，如播放、暂停、快退、快进、增加音量、减小音量和循环播放等操作的支持。

除以上功能外，还支持歌词同步显示以及多国语言支持。

关键词：音乐播放器；AWorks；emWin；WAV；MP3；FLAC

Abstract

This paper further divides incentive……

*小四号、Times New Roman字体*

**Key words:** Wage Incentive……

*小四号、Times New Roman字体、加粗 小四号、Times New Roman字体*

# 绪论

随着时代的发展，人们生活水平日益上升，对娱乐活动的需求越来越高。电子技术发展也非常迅猛，手机从以前的功能机到现在的智能手机，整合了手机、相机以及音乐播放器等设备的功能，方便人们的使用。但是从前随处可见的音乐播放器由于功能单一，储存空间小，播放效果一般。例如，采用低成本8位单片机的方案虽然成本低廉，但是性能受到限制。使用FPGA加速的方案性能高的同时成本也高，且开发不是很灵活。使用高性能32位单片机软件解码的方案软件灵活，成本适中，但是想要扩展更多的功能比较困难。导致市场占有率越来越低，迫切需要升级换代来适应时代的发展。

嵌入式系统播放MP3等有损压缩格式音乐的技术已经比较成熟，但是对于播放无损压缩音乐却鲜有报道。所以迫切需要一种能为用户提供高品质音乐享受，且体积小巧，功能强大的音乐播放器，占领随身设备的高端市场。针对上述问题，本课题拟设计一种基于AWorks嵌入式系统软件开发平台的高性能音乐播放器。

## 主要任务

本次课程设计的主要任务是设计基于i.MX283处理器的音乐播放器。主要功能如下：

1)使用i.MX283处理器设计；

2)基于AWorks软件平台；

3)使用emWin界面库；

4)支持多语言显示；

5)支持SD卡播放；

6)支持U盘播放；

7)支持WAV、MP3、FLAC格式播放；

8)TFT液晶屏显示；

9)简单的人机交互；

10)支持歌词同步显示；① ②③④⑤⑥⑦⑧⑨⑩

# 系统总体方案及硬件设计

本系统主要由i.MX283处理器、DDR2存储器、TFT液晶屏、电阻式触摸屏、串口、USB主机、音频接口、储存接口、时钟源、调试与下载接口和ES9018K2M声卡组成。其系统框架如图1所示。

## 设计思路

本设计是基于i.MX283处理器的音乐播放器，采用32位的i.MX283处理器，把数字部分和模拟部分分离开来，从而降低电磁干扰，获得较好的音效品质[[1]](#endnote-1)。开发环境为开源的集成开发环境eclipse cdt，程序采用GNU C语言编程。首先自动将音频文件并添加到播放列表中，用户可以通过播放列表选择指定音乐播放，也可以设置单曲循环、顺序播放和随机播放三种不同的播放模式来播放不同的音乐。播放完成后自动根据不同的播放模式切换到下一首歌并继续播放。在播放音乐的同时，还要在屏幕上显示歌手的图片以及当前播放的歌词。本设计使用触摸屏进行人机交互，通过点击界面上不同的按钮完成不同的操作。

系统框架如下：

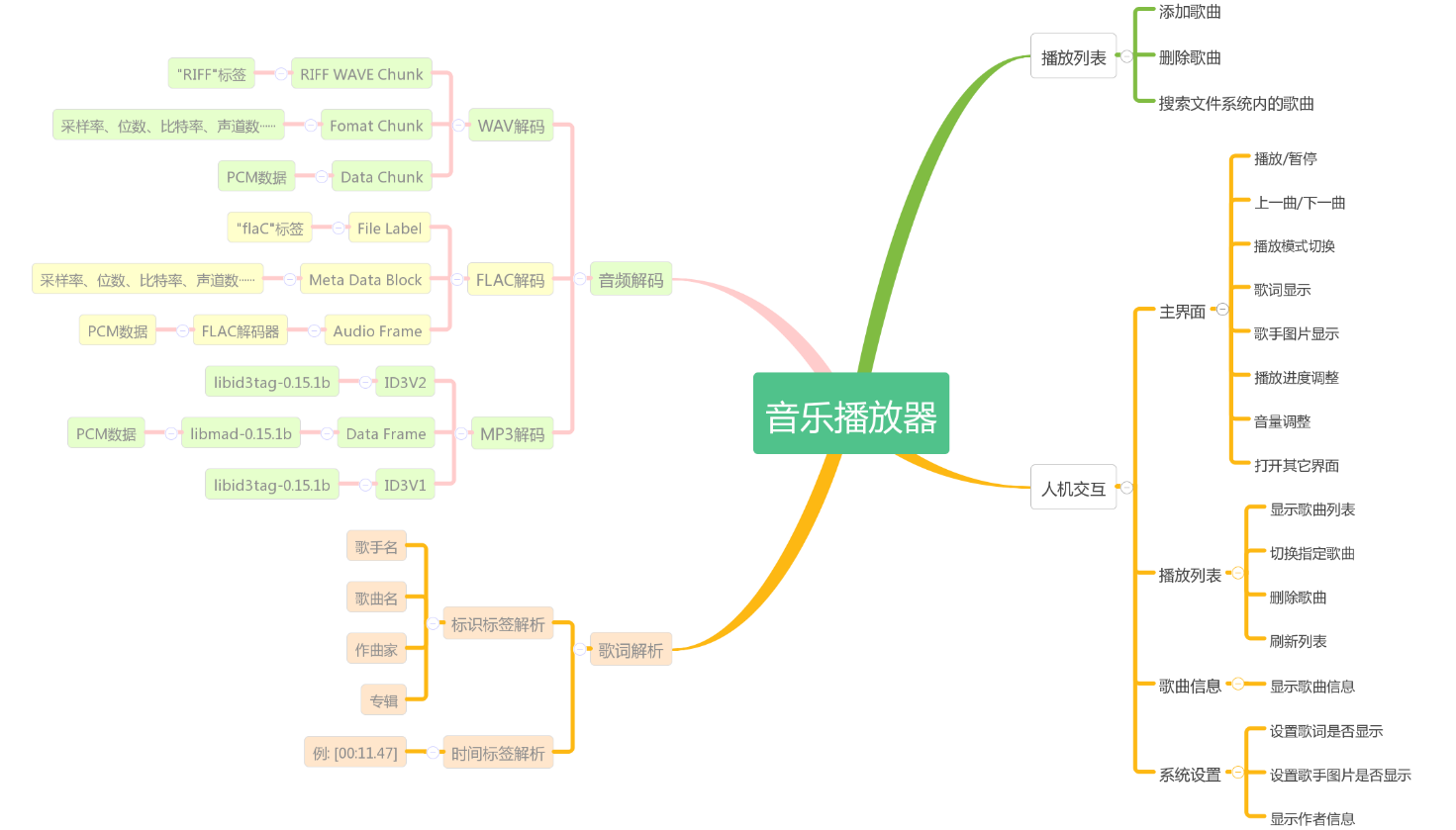


图1 系统框架思维导图

## 工作原理

首先扫描SD卡和U盘内的音频文件并添加到播放列表中。确定需要播放的音乐文件后，首先解析音频的比特率、采样率、位数等基本信息，然后开始解码音频数据并通过IIS接口送入声卡中播放。在播放音乐的同时，将与歌曲同路径且同文件名的JPG格式图片显示到屏幕上作为歌手头像，而与歌曲同路径且同文件名的LRC文件识别为歌词，并在屏幕上同步显示当前播放音乐的歌词。

## 系统硬件设计

本系统采用Freescale的i.MX283处理器（基于ARM926EJ-S内核），具有丰富的硬件资源，提供了完善的Linux软件支持包、开发工具和丰富的实用范例，大大降低了Linux学习门槛和开发难度，可以帮助用户在短期内实现产品功能验证和开发。

i.MX283是一款低功率、高性能的应用处理器，专为通用嵌入式工业控制和消费电子市场而优化。i.MX283内核采用飞思卡尔快速的、久经验证的、高能效ARM926EJ-S内核，频率高达454MHz。i.MX283适用于构建一系列现今最具创新性的应用，如智能能源网关、手持打印机和扫描仪及便携式医疗设备，具有一套综合的软件和工具助您目前快速开始启动。

i.MX283主要功能列举：

①带有触摸屏功能的LCD控制器；

②NAND支持–SLC/MLC和eMMC4.4可管理；

③硬件BCH(高达20位校正)；

④200MHz 16位DDR2，LV-DDR2，mDDR外部储存器支持；

⑤带有嵌入式PHY的高速双USB；

⑥8个通用12位ADC通道和单个2MspsADC通道；

⑦带温度传感器，提供热保护；

⑧多种连接端口(UARTs，SSP，SDIO，SPI，I2C，I2S)；

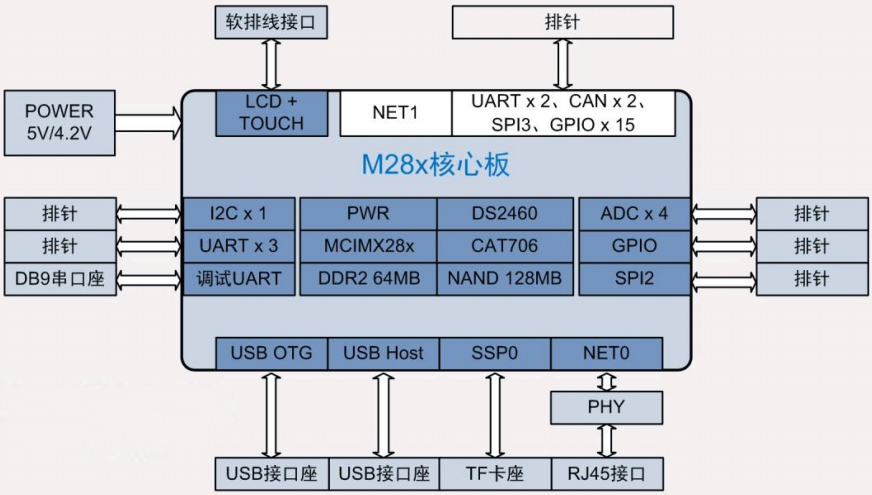


图2 i.MX283系统板功能框图

### 电源设计

i.MX283多媒体应用处理器内部集成有高效的电源管理单元（PMU）。它由DCDC、若干个线性稳压器及电池充电模块组成。外部只需给CPU提供5V或4.2V电源（电池电源），PMU就能自动产生CPU正常工作所需的所有电压。这不仅极大降低了硬件成本，同时还降低了系统电源设计的难度，使得i.MX283非常适用于电池供电的便携设备。i.MX283 PMU内部框图详见图3。

PMU可采用3种供电方式，分别是：

①5V单独供电：如USB供电，5V电源适配器或5V线性稳压器供电；

②4.2V单独供电：如电池供电，或通过线性稳压器、DCDC产生的4.2V电源供电；

③电池供电：即采用电池供电，5V电源作为电池充电电源；

本设计中，为了使用方便，直接通过随处可见的USB接口给系统板供电。

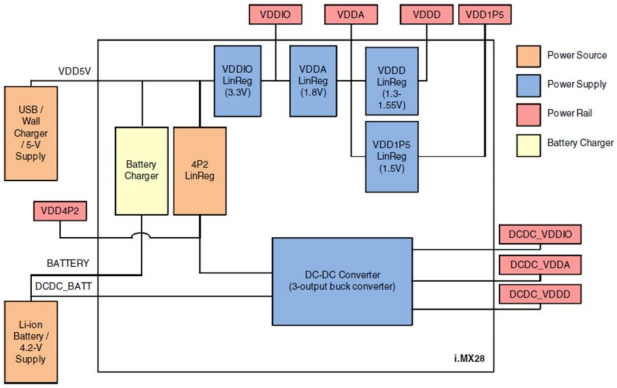


图3 PMU内部结构框图

### DDR2电路设计

i.MX283支持16bit mobile DDR、DDR2、和LV-DDR2，时钟频率高达205MHz。本设计的i.MX283系统板上选用的是DDR2，具体选用Nanya公司的NT5TU32M16DG-AC，内存大小64MB，完全满足系统应用需求。DDR2参考电路详见图4。

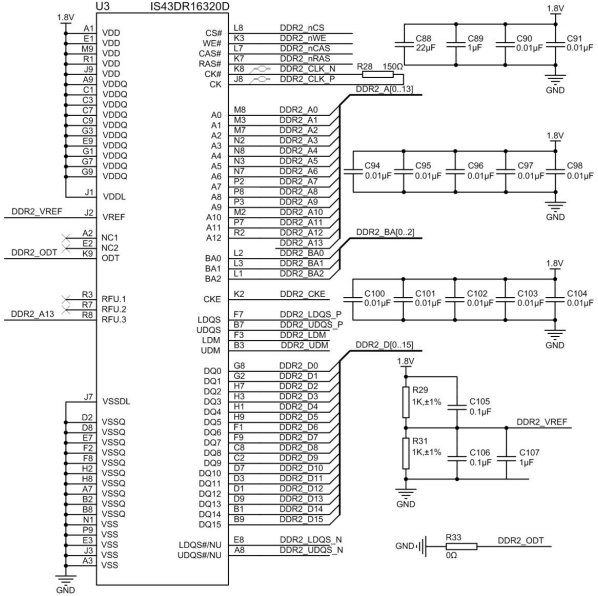


图4 DDR2典型应用电路

### 声卡电路设计

ES9018K2M是一款高性能的32位2通道音频DAC解决方案，设计用于手机和数字音乐播放器等发烧级便携式应用，消费类应用如蓝光播放器，音频前置放大器和A/V接收器，以及专业应用如录音系统，调音台和数字音频工作站。

ES9018K2M采用专利的32位Hyperstream DAC架构和时域抖动消除器,可提供高达127dB的信噪比(DNR)和-120dB的THD+N，性能水平将满足最苛刻的音频爱好者。ES9018K2M通过I2S接口可以输入多达32位384kHz的PCM数据，DSD接口则能够输入11.2MHz的数据。而且支持最高性能应用的单声道模式。支持同步和异步采样率转换模式。正常工作模式下的功耗低于40mW。

本设计中，通过IIS接口向ES9018K2M传输音频数据，并使用I2C接口写入配置信息。

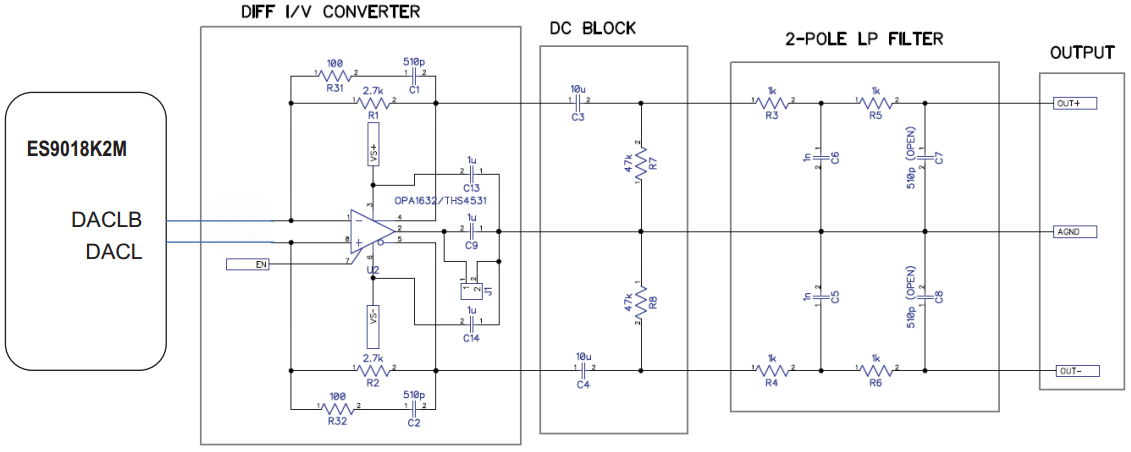


图5 ES9018K2M应用电路

# 系统软件设计

本音乐播放器的软件体系结构采用分层模式[[2]](#endnote-2)，总共包含四层：硬件层、设备驱动层、操作系统及应用层。硬件层包括音乐播放器的物理设备：USB接口、音频接口芯片ES9018K2M、4.3英寸的TFT液晶屏、电阻触摸屏等。设备驱动层包括上述各个设备的驱动程序。操作系统层为AWorks嵌入式系统软件开发平台，由操作系统统一管理各个硬件设备。最上层是应用层，使用emWin设计图形用户界面，通过解码器与控制器实现对音乐文件进行解码、播放、音量增减的控制。

## AWorks简介

AWorks是广州致远电子有限公司开发的一个创新的嵌入式软件平台，它把各种软件组件集成在一起，提供了数量庞大且高质量的服务。大量现成的软件不仅节省了用户的研发投入，还能提高产品的质量。通过简单的剪裁和配置之后，它甚至能够在只有几K内存的小资源平台上运行。尽管AWorks本身已经提供了众多高品质的可复用组件，AWorks同样也支持用户将自己的软件组件集成到平台中。AWorks对底层硬件做了良好的抽象和封装，最大程度上降低了上层应用与底层硬件的耦合。上层应用不再绑死在某款MCU上，有利于产品的升级和维护。

其关键特性如下：

* 1. 跨平台的应用编程接口；
  2. 物联网关键协议栈支持；
  3. 传感器支持；
  4. 电源管理；
  5. 网络通信；
  6. 丰富的扩展接口；
  7. 轻量级实时内核(Real-Time Kernel)；
  8. 图形配置工具；
  9. 多媒体支持；
  10. 提供I/O系统、文件系统、Shell等操作系统上常见的服务；

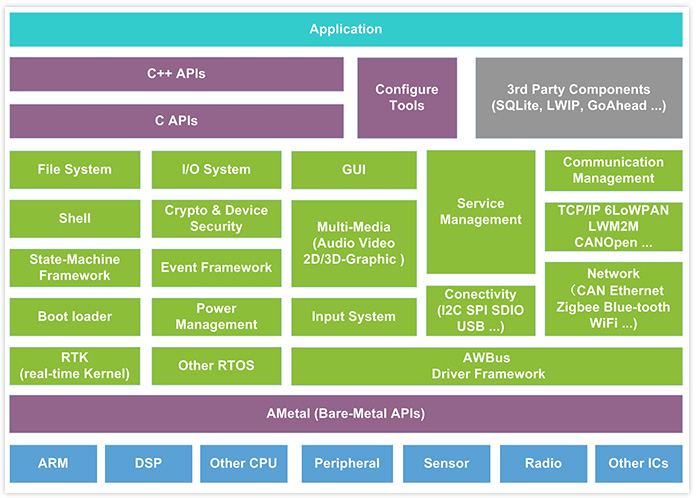


图6 AWorks架构

## emWin简介

emWin设计用于提供高效且独立于处理器和显示控制器的图形用户界面，用于任何使用图形显示进行操作的应用。它与单任务和多任务环境、专用操作系统或具有任何商业RTOS兼容，emWin的发货形式为C语言源代码。它可适用于任何尺寸的、具有任何显示控制器和CPU的物理和虚拟显示器。其特性包括：

### 一般特性

* 1. 支持使用任何控制器的任何（单色、灰度或彩色）显示器（如果有正确的驱动）；
  2. 在较小显示器上无显示控制器也可运行；
  3. 使用配置宏可支持任何接口；
  4. 显示尺寸可配置；
  5. 可在显示器上的任何点（而不仅仅是在偶数位字节地址）上写入字符和位图；
  6. 已针对尺寸和速度优化了各种例程；
  7. 利用编译时间切换可进行不同优化；
  8. 对于较慢的显示控制器，可在存储器中缓存显示，将存取操作减到最少，从而获得非常高的速度；
  9. 结构清晰；
  10. 支持虚拟显示，虚拟显示可大于实际显示；

### 图形库

* 1. 支持不同色深的位图；
  2. 可使用位图转换器；
  3. 绝对无浮点使用；
  4. 快速线、点绘制（不使用浮点）；
  5. 圆形、多边形绘制非常快速；
  6. 不同的绘图模式；

### 字体

* 1. 基本软件配备多种不同的字体：4\*6、6\*8、6\*9、8\*8、8\*9、8\*16、8\*17、8\*18、24\*32以及像素高度为8、10、13、16的比例字体；
  2. 可以定义新的字体并只需简单链接；
  3. 只有应用程序使用的字体才实际链接到生成的可执行程序，从而使ROM使用最小；
  4. 字体可分别在X和Y方向完全缩放；
  5. 可使用字体转换器；主机系统（即Windows）上有的任何字体都可以转换；

### 字符串、值输出例程

* 1. 可以十进制、二进制、十六进制、任何字体形式显示值；
  2. 可以十进制、二进制、十六进制、任何字体形式编辑值；

### 窗口管理器(WM)

* 1. 完整的窗口管理操作，包括裁剪，窗口的客户区以外的区域不可能被覆盖；
  2. 窗口可以移动和调整大小；
  3. 支持回调例程（可选择是否使用）；
  4. WM使用最小的RAM（每个窗口大约50字节）；

### PC界面外观的可选小工具

* 1. 提供各种小工具（窗口对象，也称为控件），它们通常自动操作并且简单易用；

### 支持触摸屏和鼠标

* 1. 对于按钮小工具等窗口对象，emWin提供触摸屏和鼠标支持；

### PC工具

* 1. 模拟脉冲查看器；
  2. 位图转换器；
  3. 字体转换器；

## 系统任务设计

在AWorks平台中的任务和其它操作系统中的线程是一样的，只是名称不一样，所以系统任务的设计即是系统线程的设计。

由于本系统需要完成音乐播放功能，且有比较丰富的人机交互，用户通过触摸屏进行操作的时候不能影响音乐的播放，而音频的解码也不能影响用户的操作。所以设计系统中任务如下：

### GUI任务

本任务中主要进行图形界面的初始化、处理界面相关的事件、轮询SD卡，U盘的插入与拔出。

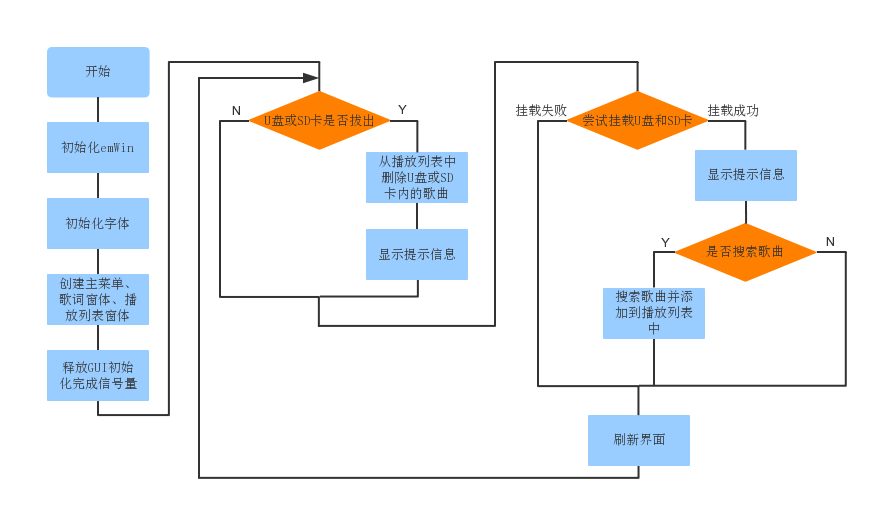


图7 GUI任务流程图

### 触摸屏任务

本任务中处理的事情比较少，只是将将触摸屏状态保存进emWin。

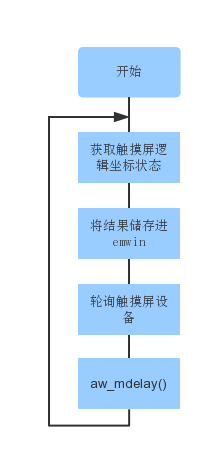


图8 触摸屏任务流程图

### 音乐播放任务

本任务完成音频的解码并播放以及音乐的切换。

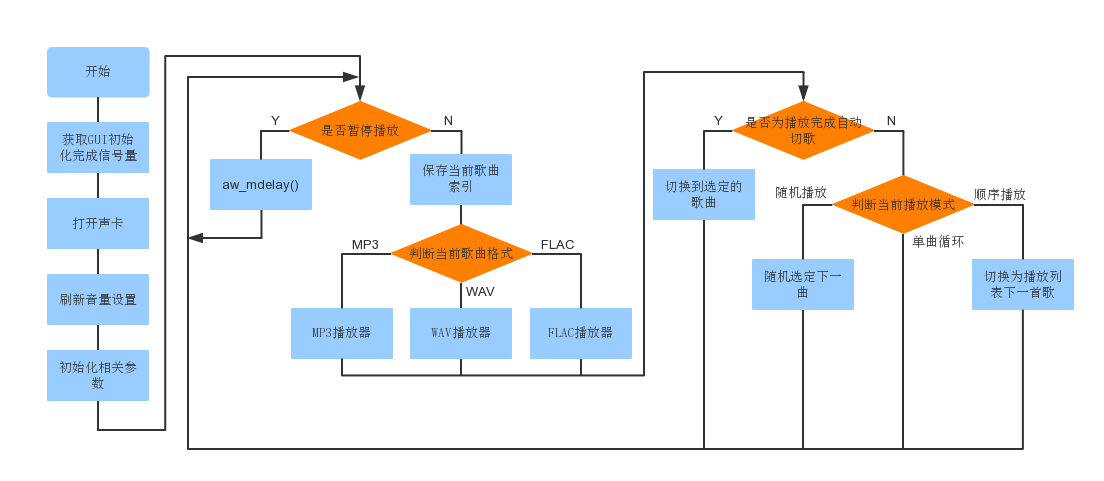


图9 音乐播放任务流程图

## 音频解码

本设计支持播放WAV、MP3、FLAC三种格式的音乐。其中WAV和FLAC为无损音乐，而MP3为有损压缩音乐[[3]](#endnote-3)。

### WAV格式

本设计中，最高能播放采样率为192KHz，量化位数为32位的立体声WAV音频。

在Windows环境下，大部分的多媒体文件都依循着一种结构来存放信息，这种结构称为资源互换文件格式，简称RIFF。例如声音的WAV文件、视频的AV1文件等等均是由此结构衍生出来的。RIFF可以看做是一种树状结构，其基本构成单位为chunk，犹如树状结构中的节点，每个chunk由辨别码、数据大小及数据所组成。

|  |
| --- |
| 块的标志符（4字节） |
| 数据大小（4字节） |
| 数据 |

表1 块的结构示意

辨别码由4个ASCII码所构成，数据大小则标示出紧跟其后数据的长度(单位为Byte)，而数据大小本身也用掉4个Byte，所以事实上一个chunk的长度为数据大小加8。一般而言，chunk本身并不允许内部再包含chunk，但有两种例外，分别为以RIFF及L1ST为辨别码的chunk。而针对此两种chunk，RIFF又从原先的数据中切出4个Byte。此4个Byte称为格式辨别码，然而RIFF又规定文件中仅能有一个以RIFF为辨别码的chunk。

|  |  |
| --- | --- |
| RIFF/LIST标志符 | |
| 数据大小 | |
| 数据 | 格式/列表类型 |
| 子数据 |

表2 RIFF/LIST块结构

只要依循此一结构的文件，我们均称之为RIFF档。此种结构提供了一种系统化的分类。如果和MS-DOS文件系统作比较，RIFF chunk就好比是一台硬盘的根目录，其格式辨别码便是此硬盘的逻辑分区，而L1ST chunk即为其下的子目录，其他的chunk则为一般的文件。至于在RIFF文件的处理方面，微软提供了相关的函数。windows下的各种多媒体文件格式就如同在磁盘机下规定仅能放怎样的目录，而在该目录下仅能放何种数据。

wav格式是微软公司开发的一种声音文件格式，也叫波形文件，是最早的数字音频文件格式，它具有RIFF(Resource Inter⁃change File Format)格式[[4]](#endnote-4)。用于保存Windows平台的音频信息资源，被Windows平台及其应用程序所广泛支持，也是其音乐发烧友中常用的指定规格之一。由于此音频格式未经过压缩，所以在音质方面不会出现失真的情况，但文件的体积因而在众多音频格式中较为大。该格式支持多种音频数字，取样频率和声道，标准格式化的WAV文件和CD格式一样，也是44.1K的取样频率，16位量化数字，因此在声音文件质量和CD相差无几。

由表3可知，RIFF块包含两个子块，这两个子块的ID分别是"fmt"和"data"，其中"fmt"子块由结构PCMWAVEFORMAT所组成，其子块的大小就是sizeof(PCMWAVEFORMAT)，数据组成就是PCMWAVEFORMAT结构中的数据。

|  |
| --- |
| 标志符（RIFF） |
| 数据大小 |
| 格式类型（"WAVE"） |
| "fmt" |
| sizeof(PCMWAVEFORMAT) |
| PCMWAVEFORMAT |
| "data" |
| 声音数据大小 |
| 声音数据 |

表3 WAVE文件结构

"data"子块包含WAVE文件的数字化波形声音数据，其存放格式依赖于"fmt"子块中wFormatTag成员指定的格式种类，在多声道WAVE文件中，样本是交替出现的。如16bit的单声道WAVE文件和双声道WAVE文件的数据采样格式分别如表4和表5所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 采样一 | | 采样二 | | …… |
| 低字节 | 高字节 | 低字节 | 高字节 | …… |

表4 16位单声道

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 采样一 | | | |  |
| 左声道 | | 右声道 | | …… |
| 低字节 | 高字节 | 低字节 | 高字节 | …… |

表5 16位双声道

### MP3格式

MP3全称MPEG audio player 3，其中MPEG（Moving Picture Experts Group）标准包括视频和音频标准，其中音频标准已制定出MPEG-1、MPEG-2、MPEG-2 AAC和MPEG-4。MPEG-1和MPEG-2标准使用同一个音频编码解码族：Layer1、Layer2、Layer3。它根据压缩质量和编码复杂程度进行划分的，分别对应MP1、MP2、MP3这三种声音文件，并根据不同的用途，使用不同层次的编码。MPEG音频编码的层次越高，编码器越复杂。采样频率为32KHz、44.1KHz、48KHz。Layer3增加了MDCT变换，使其频率分辨能力是Layer2的18倍，Layer3还使用了与MPEG Video类似的平均信息量编码，减少了冗余信息。MP3的音频质量取决于它的Bit rate和Sampling frequency，以及编码器质量。MP3对音频信号采用的是有损压缩方式，为了降低声音失真度，MP3采取了“感官编码技术”，即它丢掉脉冲编码调制音频数据中对人类听觉不重要得数据，从而达到了较高的压缩比，即编码时先对音频文件进行频谱分析，然后用过滤器滤掉噪音电平，接着通过量化的方式将剩下的每一位打散排列，最后形成具有较高压缩比的MP3文件，并使压缩后的文件在回放时能够达到比较接近原音源的声音效果。

MP3文件是由帧构成，帧是MP3文件的最小组成单位。每帧都包含帧头，并可以计算帧的长度。根据帧的性质不同，文件主要分为三个部分，ID3V2标签帧，数据帧和ID3V1标签帧。并非每个MP3文件都有ID3v2，但是数据帧和ID3v1帧是必须的。ID3v2在文件头，以字符串”ID3”为标志，包含了演唱者，作曲，专辑等信息，长度不固定，扩展了ID3V1的信息量。ID3v1在文件结尾，以字符串”TAG”为标记，其长度是固定的128个字节，包含了演唱者、歌名、专辑、年份等信息。

|  |  |
| --- | --- |
| ID3V2 | 包含了作者，作曲，专辑等信息，长度不固定，扩展了ID3V1的信息量 |
| FRAME | 一系列的帧，个数由文件大小和帧长决定  每个FRAME的长度可能不固定，也可能固定，由位率bitrate决定  每个FRAME又分为帧头和数据实体两部分  帧头记录了mp3的位率，采样率，版本等信息，每个帧之间相互独立 |
| ID3V1 | 包含了作者，作曲，专辑等信息，长度为128BYTE |

表6 MP3文件结构

每个音频数据帧都有一个帧头Header，长度是4Byte，帧头后面可能有两个字节的CRC校验值，这两个字节的是否存在决定于Header信息的第16bit，为0则帧头后面无校验，为1则有校验，校验值长度为2个字节，紧跟在Header后面，接着就是帧的实体数据。Header格式如表7所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 位长 | | 说明 |
| 同步信息 | 11 | 第1、2字节 | 所有位均为1，第1字节恒为FF |
| 版本 | 2 | 00-MPEG2.5 01-未定义10-MPEG2 11-MPEG1 |
| 层 | 2 | 00-未定义 01-Layer3 10-Layer2 11-Layer1 |
| CRC校验 | 1 | 0-校验 1-不校验 |
| 位率 | 4 | 第3字节 | 取样率，单位是kbps  "free"表示位率可变"bad"表示不允许值 |
| 采样频率 | 2 | 采样频率，对于MPEG-1：00-44.1kHz01-48kHz10-32kHz11-未定义  对于MPEG-2：00-22.05kHz01-24kHz10-16kHz11-未定义  对于MPEG-2.5：00-11.025kHz01-12kHz10-8kHz11-未定义 |
| 帧长调节 | 1 | 用来调整文件头长度，0-无需调整，1-调整 |
| 保留字 | 1 | 没有使用 |
| 声道模式 | 2 | 第4字节 | 表示声道，00-立体声Stereo 01-Joint Stereo 10-双声道 11-单声道 |
| 扩充模式 | 2 | 当声道模式为01时才使用。 |
| 版权 | 1 | 文件是否合法，0-不合法 1-合法 |
| 原版标志 | 1 | 是否原版，0-非原版 1-原版 |
| 强调方式 | 2 | 用于声音经降噪压缩后再补偿的分类，很少用到，今后也可能不会用。  00-未定义 01-50/15ms 10-保留 11-CCITTJ.17 |

表7 MP3帧头字节说明

帧长度是压缩时每一帧的长度，包括帧头。它将填充的空位也计算在内。Layer I的一个空位长4字节，Layer II和Layer III的空位是1字节。当读取MPEG文件时必须计算该值以便找到相邻的帧。

从头中读取比特率，采样频率和填充的值后可以进行计算，

Lyaer I使用公式：

(3.1)

Lyer II和Lyaer III使用公式：

(3.2)

每帧的持续时间可以通过计算获得，下面给出计算公式：

(3.3)

### FLAC格式

FLAC为无损音频压缩编码。FLAC是一套著名的自由音频压缩编码，其特点是无损压缩。不同于其他有损压缩编码如MP3及AAC，它不会破坏任何原有的音频资讯，所以可以还原音乐光盘音质。2012年以来它已被很多软件及硬件音频产品（如CD等）所支持。

FLAC与MP3不同，MP3是音频压缩编码，但FLAC是无损压缩，也就是说音频以FLAC编码压缩后不会丢失任何信息，将FLAC文件还原为WAV文件后，与压缩前的WAV文件内容相同。这种压缩与ZIP的方式类似，但FLAC的压缩比率大于ZIP和RAR，因为FLAC是专门针对PCM音频的特点设计的压缩方式。而且可以使用播放器直接播放FLAC压缩的文件，就象通常播放你的MP3文件一样。

FLAC编码设置只对编码时间有较大影响而对解码影响不大。因为等级越高，编码器就会花越多的时间去寻找最佳的压缩算法，而解码器则根据给定的压缩算法直接解压。

FLAC中所有数值都是整形，大端模式，除非特别指出，否则数值都是unsigned的。

音频数据由一个或多个音频帧组成，每一帧包含一个帧头：同步字，块大小，采样率，声道数…然后是8bit CRC校验码。同时帧头还包含本帧第一个采样点的采样序号(blocksize变长的文件)或本帧的序号(blocksize定长的文件)，他们用于精确定位。接着是编码后的subframes，每个subframe代表一个声道。最后是一些有0填充的边界。每个subframe有他自己的帧头用于指出他是怎样被编码的。

一个音频帧被一个帧头和帧脚包围。帧头以同步字开始，包含了解码这帧的最小信息如采样率，采样位数等，同时包含了这帧里的分块数或采样数及一个8位CRC校验码。帧头可以用来进行再同步。帧尾包含一个16位CRC校验码。如果解码器检测到CRC错误将产生一个silent block。

|  |
| --- |
| 字符串“flaC” |
| 文件信息描述块 |
| 可选的其他描述信息块 |
| 一个以上的音频帧 |

表8 FLAC文件结构

|  |  |  |
| --- | --- | --- |
| 长度 | 功能 | 说明 |
| 14 | 同步字 | '11111111111110' |
| 1 | 保留 | 0：强制值 1：保留未来使用 |
| 1 | 分块策略 | 0：固定块大小，帧头包含帧的序号  1：块大小可变，帧头包含采样点序号 |
| 4 | 块内的采样数 | 块内的采样数 |
| 4 | 采样率 | 采样率 |
| 4 | 声道分配 | 声道分配 |
| 3 | 采样深度 | 采样深度 |

表9 Header结构

### 音乐播放流程

本设计在音乐播放任务中需要调用音乐播放函数进行一首曲目的播放，所以需要一个能够解码并播放音频文件的函数。在该函数中，首先打开音频文件并获取文件大小，然后读取文件内的音频信息，如采样率、量化位数、比特率等。获取到音频的相关信息后，通过这些参数初始化声卡。接下来读取到的就是音频数据，将音频数据送入解码器之后得到PCM数据，然后输入到声卡播放，直到当前文件播放完成退出播放。

解码并播放一首音乐的程序流程如图10所示。

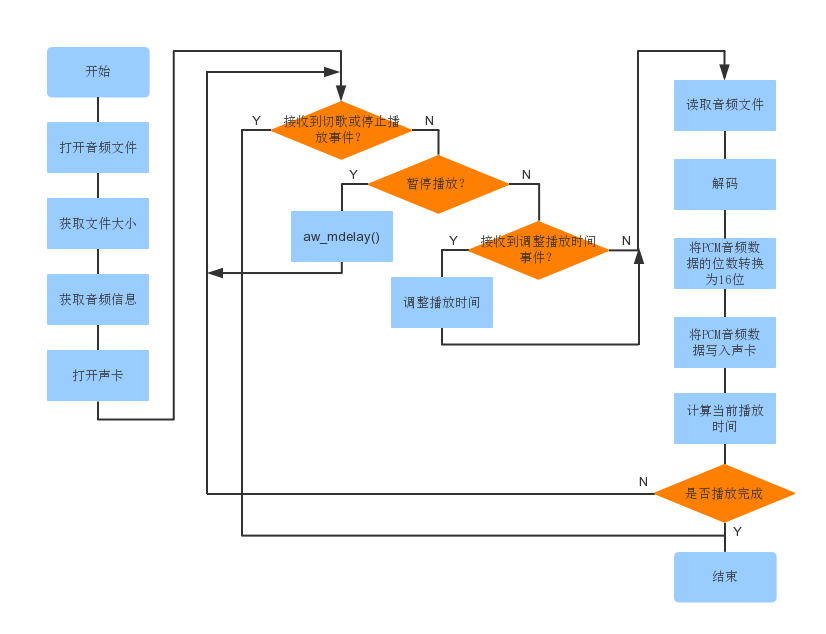


图10 音频文件播放流程图

## LRC歌词解析

### LRC格式简介

LRC是一个可以跟音乐文件做同步的文件格式。当一个音乐文件（如MP3、Vorbis或WMA等）被电脑音乐播放程序或现代的MP3随身听以及DVD播放机等设备播放时，歌词可以被同步显示出来。歌词文件通常和音乐文件有同样的文件名称，但是扩展名不同。例如：song.mp3和song.lrc。LRC格式是一种文字格式，与电视和电影的字幕档很相似。由于中、日、韩文歌词在ANSI格式里可能产生乱码，可以使用UTF-8或Unicode文字编码避免。

LRC歌词文本中含有两类标签：

一是标识标签，其格式为”[标识名:值]”主要包含以下预定义的标签：

[ar:歌手名]、[ti:歌曲名]、[al:专辑名]、[by:编辑者(指lrc歌词的制作人)]、[offset:时间补偿值]（其单位是毫秒，正值表示整体提前，负值相反。这是用于总体调整显示快慢的，但多数的MP3可能不会支持这种标签）。

二是时间标签，形式为”[mm:ss]”或”[mm:ss.ff]”，时间标签需位于某行歌词中的句首部分，一行歌词可以包含多个时间标签。当歌曲播放到达某一时间点时，MP3就会寻找对应的时间标签并显示标签后面的歌词文本，这样就完成了歌词同步的功能。

### LRC歌词解析

LRC歌词是以文本的形式存储在文件中的，解析LRC歌词文件时，通过C语言标准库函数内的标准输入函数可以按行读取文本数据，读取到一行数据后，判断该行内标示标签和时间标签的数量，然后分别解析标识标签和时间标签，然后存储到数组中，供其它函数使用。解析LRC歌词流程如图11所示。

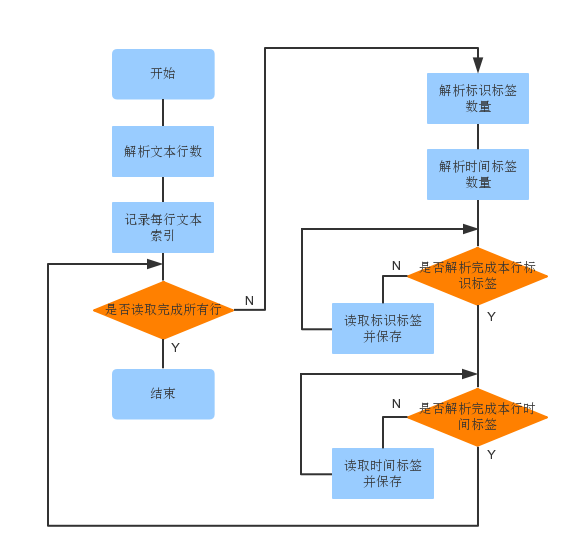


图11 歌词解析流程图

参考文献

1. 於少峰,严菊明,胡晨.基于AC97标准的嵌入式音频系统设计与实现[J].电子器件.2004,27(4):733-736. [↑](#endnote-ref-1)
2. 刘垣,李外云,赵嘉怡.基于STC单片机WAVE音乐播放器的设计与实现[J].科技创新与应用,2015(34):50-51. [↑](#endnote-ref-2)
3. 李伟,张真,范文豪.基于STM32微控制器的mp3播放器设计[J].现代电子技术,2015(4):118-120. [↑](#endnote-ref-3)
4. 罗海涛.wav音频文件格式分析与数据获取[J].计算机工程应用技术.2016,12(27):211-213. [↑](#endnote-ref-4)