



前言

目前基于智能门锁的开发，需要用到语音，但是由于如果用语音 IC 的话，声音内容就不方便调整更改，所以我们打算用单片机的 DAC 功能直接读取 WAV 文件，再用功放 IC 放大来实现智能语音。本应用笔记将以 KF32L530 为例介绍如何如何利用 DAC 播放 WAV 格式的音频

文档及相关代码下载地址：

Github: <https://github.com/passager123/Application-note>

Gitee: <https://gitee.com/passenger321/project-application-notes>

目录

| | |
|-----------------------------------|---|
| 1. WAV 文件格式..... | 3 |
| 2. WAV 文件解码..... | 4 |
| 3. 将 wav 格式文件去掉文件头转换成 BIN 文件..... | 6 |
| 4. 版本历史..... | 8 |

1. WAV 文件格式

1、音频基础知识

声道数（通道数）

即声音的通道的数目。很好理解，有单声道和立体声之分，单声道的声音只能使用一个喇叭发声（有的也处理成两个喇叭输出同一个声道的声音），立体声的 PCM 可以使两个喇叭都发声（一般左右声道有分工），更能感受到空间效果。

2、采样位数

即 采样值或取样值（就是将采样样本幅度量化）。它是用来衡量声音波动变化的一个参数，也可以说是声卡的分辨率。它的数值越大，分辨率也就越高，所发出声音的能力越强。

在计算机中采样位数一般有 8 位和 16 位之分，但有一点请大家注意，8 位不是说把纵坐标分成 8 份，而是分成 2 的 8 次方即 256 份；同理 16 位是把纵坐标分成 2 的 16 次方 65536 份。

3、采样频率

即取样频率，指 每秒钟取得声音样本的次数。采样频率越高，声音的质量也就越好，声音的还原也就越真实，但同时它占的资源比较多。由于人耳的分辨率很有限，太高的频率并不能分辨出来。在 16 位声卡中有 22KHz、44KHz 等几级，其中，22KHz 相当于普通 FM 广播的音质，44KHz 已相当于 CD 音质了，目前的常用采样频率都不超过 48KHz。

既然知道了以上三个概念，就可以由下边的公式得出 PCM 文件所占容量：

存储量 = (采样频率 * 采样位数 * 声道 * 时间) / 8 (单位：字节数)。

4、什么是 WAV 和 PCM?

WAV: wav 是一种无损的音频文件格式，WAV 符合 PIFF(Resource Interchange File Format)规范。所有的 WAV 都有一个文件头，这个文件头音频流的编码参数。WAV 对音频流的编码没有硬性规定，除了 PCM 之外，还有几乎所有支持 ACM 规范的编码都可以为 WAV 的音频流进行编码。

PCM:PCM (Pulse Code Modulation----脉冲编码调制)。所谓 PCM 编码就是将声音等模拟信号变成符号化的脉冲列，再予以记录。PCM 信号是由[1]、[0]等符号构成的数字信号，而未经过任何编码和压缩处理。与模拟信号比，它不易受传送系统的杂波及失真的影响。动态范围宽，可得到音质相当好的影响效果。PCM 数据是最原始的音频数据完全无损。

简单来说：wav 是一种无损的音频文件格式，pcm 是没有压缩的编码方式。

5、WAV 和 PCM 的关系

WAV 是使用 PCM 编码的一种音频格式。但是这不表示 WAV 只能使用 PCM 编码，MP3 编码同样也可以运用在 WAV 中，和 AVI 一样，只要安装好了相应的 Decode，就可以欣赏这些 WAV 了。在 Windows 平台下，基于 PCM 编码的 WAV 是被支持得最好的音频格式，所有音频软件都能完美支持，由于本身可以达到较高的音质的要求，因此，WAV 也是音乐编辑创作的首选格式，适合保存音乐素材。因此，基于 PCM 编码的 WAV 被作为了一种中介的格式，常常使用在其他编码的相互转换之中，例如 MP3 转换成 WAV。

简单来说：pcm 加上 wav 头文件就变成了 wav。

6、wav 文件格式

在文件的前 44 字节放置标头(header)，使播放器或编辑器能够简单掌握文件的基本信息，其内容以区块(chunk)为最小单位，每一区块长度为 4 字节。

2. WAV 文件解码

WAV: wav 是一种无损的音频文件格式, WAV 符合 RIFF(Resource Interchange File Format)规范。所有的 WAV 都有一个文件头, 这个文件头音频流的编码参数。

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00000000 | 52 | 49 | 46 | 46 | 76 | 43 | 01 | 00 | 57 | 41 | 56 | 45 | 66 | 6D | 74 | 20 |
| 00000010 | 12 | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 80 | 3E | 00 | 00 | 00 | 7D | 00 | 00 |
| 00000020 | 02 | 00 | 10 | 00 | 00 | 00 | 64 | 61 | 74 | 61 | 50 | 3D | 01 | 00 | 00 | FF |
| 00000030 | 80 | FF | 20 | 00 | 80 | 00 | F0 | 00 | 30 | 01 | 20 | 01 | 90 | 01 | 10 | 02 |
| 00000040 | 20 | 02 | 00 | 02 | 60 | 02 | 80 | 02 | 40 | 02 | 30 | 02 | 00 | 02 | 90 | 01 |
| 00000050 | 80 | 01 | A0 | 01 | 30 | 01 | 90 | 00 | 20 | 00 | 00 | 00 | F0 | FF | E0 | FF |
| 00000060 | B0 | FF | 20 | FF | E0 | FE | 20 | FF | F0 | FE | A0 | FE | 80 | FE | A0 | FE |
| 00000070 | D0 | FE | 00 | FF | D0 | FE | 90 | FE | E0 | FE | 70 | FF | 90 | FF | 40 | FF |

WAV 文件头部解析:

00-03 52 49 46 46: RIFF 的标志

04-07 76 43 01 00 : 文件的长度(此大小加上前面四个字节(RIFF 的标志)即文件的总大小)

08-0B 57 41 56 45 : WAVE 的标志

0C-0F 66 6D 74 20 : fmt 的标志

10-13 12 00 00 00 : 编码的格式类别, 10H 代表 PCM 形式。

14-15 01 00 : 字块总数

16-17 01 00 : 通道数 1 为单声道, 2 为双声道

18-1B 80 3E 00 00 : 采样率: 每秒采集的样本个数。

1C-1F 00 7D 00 00 : 每秒播放的字节数 = 通道数 × 每秒采集的样本数 × 采集数据的位数 / 8。

20-21 02 00 : 每个样点的字节数, 表示块对齐的内容。数据块的调整数(按字节算的), 其值为通道数×每样本的数据位值 / 8。播放软件需要一次处理多个该值大小的字节数据, 以便将其值用于缓冲区的调整。

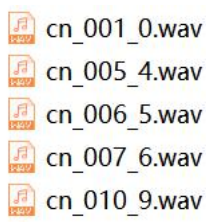
22-23 10 00 : 每个样点的数据位数(小端存储, 即 0x0010, 1000 表示 16 位存储, 0800 表示 8 位存储)

24-27 00 00 64 61 : data 的标志

28-2B 74 61 50 3D : 音频数据的大小(实际结尾处会多一些数据, 可不用管, 估计是一些时间信息)

2C 开始就是音频数据了,

用工具 winHex 可以直接将 WAV 文件打开, 内容如下所示



The screenshot displays the WinHex v20.0.1.0 application window. The title bar reads "WinHex - [cn_001_0.wav]". The menu bar includes File(F), Edit(E), Search(S), Position(P), View(V), Tools(T), Special Tools(I), Options(O), Window(W), and Help(H). The toolbar contains various icons for file operations, editing, and viewing.

The main editing area shows the following details:

- File Name:** cn_001_0.wav
- Offset:** 0
- File Size:** 6.3 KB (6,484 bytes)
- Default Editing Mode:** Original (原始)
- Cancellation Level:** 0
- Cancellation Inverse:** n/a
- Create Time:** 2021/05/29 11:41:56
- Last Write Time:** 2021/05/29 11:41:56
- Attributes:** A
- Icons:** 0
- Mode:** 16-bit
- Character Set:** ANSI ASCII
- Encoding:** 16-bit
- Bytes/Page:** 30x16=480
- Window #:** 2
- Window Number:** 6
- Clipboard:** Available (可用)
- Temporary File Path:** C:\ProgramData\Local\Temp

The hex editor view shows the following data:

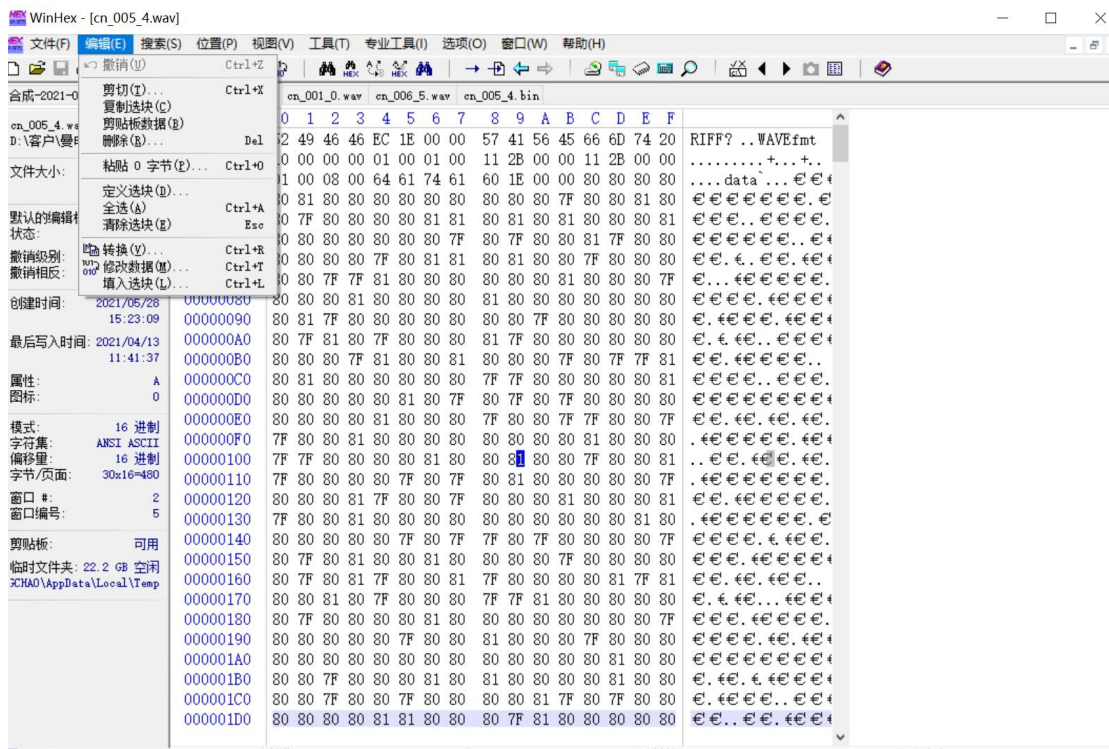
| Offset | Hex | ASCII | Comment |
|----------|-------------------------|---------------|-----------------|
| 00000000 | 52 49 46 46 4C 19 00 00 | | RIFFL...WAVEfmt |
| 00000010 | 10 00 00 00 01 00 01 00 | |+..+ |
| 00000020 | 01 00 08 00 64 61 74 61 |data? | |
| 00000030 | 80 80 7F 81 80 80 7F 80 | €..€.€€.€€€€ | |
| 00000040 | 7F 80 80 81 80 80 80 80 | .€€€€€€€..€ | |
| 00000050 | 80 80 80 7F 80 80 80 80 | €€€€€€€€..€ | |
| 00000060 | 80 80 80 80 80 80 80 80 | €€€€€€€€€€ | |
| 00000070 | 80 80 80 80 80 7F 80 80 | €€€€€€€€€€ | |
| 00000080 | 80 80 80 80 80 80 80 80 | €€€€€€€€€€ | |
| 00000090 | 80 81 81 80 80 80 80 80 | €.€€€€€€€€.. | |
| 000000A0 | 80 80 80 80 81 80 80 80 | €€.€€.€€.€€.€ | |
| 000000B0 | 7F 7F 80 80 80 81 80 80 | ..€€.€€.€€.€€ | |
| 000000C0 | 81 80 80 7F 7F 80 80 80 | .€€.€€€€€€€.€ | |
| 000000D0 | 80 80 80 80 7F 80 80 80 | €€.€€€€€€€€ | |
| 000000E0 | 80 80 80 80 80 81 80 80 | €€€€.€€€€€€.. | |
| 000000F0 | 80 80 80 80 80 80 80 80 | €€€€€€.€€€€€€ | |
| 00000100 | 80 80 81 80 81 80 80 80 | €.€.€€€€.€€€€ | |
| 00000110 | 80 7F 80 80 80 81 80 80 | €€.€€.€€.€€€€ | |
| 00000120 | 80 80 80 80 81 80 80 81 | €€.€€€€€€€€ | |
| 00000130 | 80 80 80 80 80 81 81 81 | €€€€....€.€.€ | |
| 00000140 | 80 7F 80 7F 81 80 80 80 | €€.€€€.€€€€. | |
| 00000150 | 7F 80 80 80 7F 80 80 80 | .€€.€€€€€€€€ | |
| 00000160 | 7F 80 80 80 80 80 80 80 | .€€€€€.€.€€.€ | |
| 00000170 | 80 80 80 81 81 80 81 80 | €.€€.€.€.€€.€ | |
| 00000180 | 81 80 80 7F 80 80 80 80 | .€€€€€€€€€.€ | |
| 00000190 | 80 80 80 80 80 7F 80 80 | €€€€€.€€.€€€ | |
| 000001A0 | 80 7F 80 80 80 80 80 80 | €€€€€€€€€€ | |
| 000001B0 | 80 80 80 80 80 80 81 80 | €€€€€€.€€.€ | |
| 000001C0 | 80 81 80 80 81 80 80 80 | €€.€€€€€€€€ | |
| 000001D0 | 80 80 7F 80 80 80 80 80 | €.€€€€€€€€€ | |

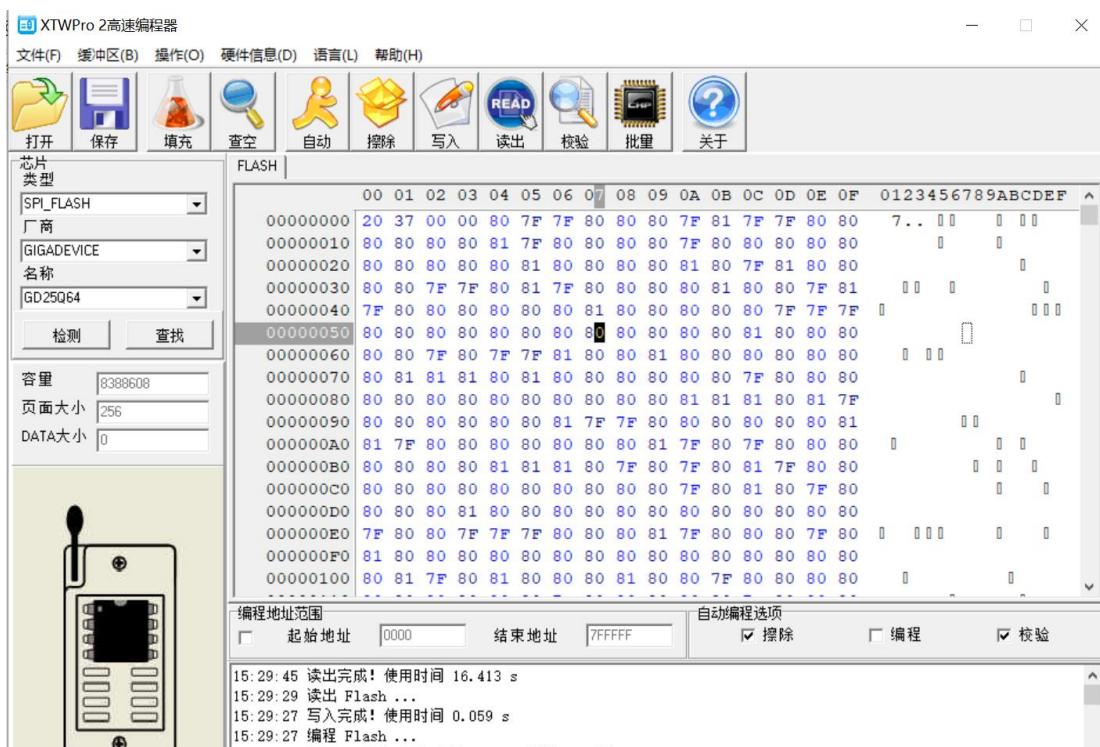
此内容中前 44 个字节是对 WAV 文件编码格式的一些描述，具体查看上面文件编码格式一一对应，第 5 个字节开始的四个字节记录本 WAV 文件数据的长度(不包括 RIFF)，如图中 4C190000，本个 WAV 文件数据大小为 $0x194C = 6476$ 字节，此数据不包括前面的 4 个字节，加上前面的 4 个字节，即为 $6476 + 4 = 6480 = 0x1950$ ，如下图，0x18C0 为音频数据的大小，实际从 0x2C 处开始到结尾处的数据量不止 18C0，估计结尾处是一些时间信息(这里只是猜测，没查到具体说明)，在代码中获取数据时可只获取 18C0 个数据播放。

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------|
| 00001790 | 80 | 80 | 81 | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | 80 | 7F | 7F | €. .€ .€€€ .€€€.. |
| 000017A0 | 7F | 80 | 7F | 7F | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | .€ .€€€€ .€€€€. |
| 000017B0 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 81 | 81 | 80 | 80 | 80 | 80 | 81 | 80 | 80 | €€€€€€ .€€€€. |
| 000017C0 | 80 | 81 | 80 | 80 | 80 | 7F | 80 | 81 | 80 | 81 | 81 | 81 | 80 | 81 | 80 | 80 | €€€€€€ ..€€€€. |
| 000017D0 | 80 | 80 | 80 | 80 | 81 | 80 | 7F | 80 | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 80 | €€€€ .€€€€€€ €€€€. |
| 000017E0 | 80 | 80 | 80 | 81 | 7F | 81 | 80 | 80 | 7F | 80 | 80 | 81 | 80 | 80 | 80 | 80 | €€€€.. €€€€€€ €€€€. |
| 000017F0 | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 81 | 80 | 7F | 80 | 80 | 7F | €€€€€€€€ .€€€€. |
| 00001800 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | €€€€€€€€ €€€€. |
| 00001810 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 7F | 80 | 80 | 80 | 80 | 7F | €€€€€€€€ .€€€€. |
| 00001820 | 80 | 7F | 7F | 80 | 80 | 7F | 7F | 80 | 80 | 80 | 81 | 81 | 7F | 80 | 80 | 7F | €€ .€€€€€€ ..€€€€. |
| 00001830 | 80 | 80 | 80 | 80 | 81 | 80 | 81 | 80 | 7F | 80 | 81 | 80 | 80 | 80 | 80 | 80 | €€€€€€€€ ..€€€€. |
| 00001840 | 80 | 80 | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 7F | 80 | 80 | €€€€€€€€ ..€€€€. |
| 00001850 | 80 | 80 | 80 | 7F | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | 80 | €€€€€€€€ ..€€€€. |
| 00001860 | 7F | 80 | 80 | 80 | 80 | 7F | 80 | 80 | 80 | 7F | 80 | 7F | 80 | 81 | 80 | 80 | .€€€€€€€€€€€€€€. |
| 00001870 | 81 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | €€€€€€€€€€€€€€. |
| 00001880 | 80 | 80 | 81 | 80 | 80 | 81 | 80 | 80 | 80 | 81 | 7F | 80 | 80 | 80 | 81 | 7F | €€ .€€€€€€ .€€€€.. |
| 00001890 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | 7F | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | €€ .€€€€€€ .€€€€.. |
| 000018A0 | 80 | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | 80 | 7F | 80 | 80 | 80 | €€ .€€€€€€ .€€€€.. |
| 000018B0 | 80 | 7F | 80 | 81 | 80 | 80 | 80 | 80 | 80 | 81 | 7F | 80 | 80 | 80 | 81 | 80 | €€€€€€€€ .€€€€. |
| 000018C0 | 7F | 7F | 7F | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 7F | 80 | 80 | ...€€€€€€€€ .€€€€. |
| 000018D0 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 81 | 80 | 80 | 80 | €€€€€€€€ .€€€€. |
| 000018E0 | 7F | 80 | 80 | 80 | 80 | 7F | 7F | 80 | 7F | 80 | 80 | 81 | 49 | 44 | 33 | 73 | .€€€€€ .€€€€ ID3€ |
| 000018F0 | 5F | 00 | 00 | 00 | 49 | 44 | 33 | 04 | 00 | 00 | 00 | 00 | 00 | 55 | 54 | 53 |ID3.....UTX* |
| 00001900 | 53 | 53 | 00 | 00 | 00 | 12 | 00 | 00 | 03 | 74 | 69 | 6D | 65 | 5F | 72 | 65 | XX.....time_re* |
| 00001910 | 66 | 65 | 72 | 65 | 6E | 63 | 65 | 00 | 30 | 80 | 54 | 44 | 52 | 43 | 00 | 00 | ference.0.TDRC..* |
| 00001920 | 00 | 0C | 00 | 00 | 03 | 32 | 30 | 31 | 39 | 2D | 30 | 36 | 2D | 31 | 37 | 00 |2019-06-17.* |
| 00001930 | 54 | 53 | 53 | 45 | 00 | 00 | 00 | 0F | 00 | 00 | 03 | 4C | 61 | 76 | 66 | 35 | TSSE.....Lavf5* |
| 00001940 | 38 | 2E | 32 | 32 | 2E | 31 | 30 | 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 8. 22. 100.....* |
| 00001950 | 84 | 02 | BB | 00 | | | | | | | | | | | | | ? * |

3. 将 wav 格式文件去掉文件头转换成 BIN 文件

用 WINHEX 软件打开 WAV 文件，复制全部内容，新建一个记事本，文件格式为 h，粘贴到里面，再用 winhex 工具打开 h 文件，将 h 文件的内容转换为二进制(如图编辑->转换，选二进制)，转换好后，再保存为 BIN 文件，就可以利用下载工具下载到 flash 器件中了，**注意下载的时候要把外部 flash 里面的数据擦除掉 (编程工具为 XTW PRO)**，另将 MCU 里面代码更新为空代码(不然若代码里面有操作外部 flash 会造成冲突)再下载，否则下载进去的数据和读出来的数据不一样，下载工具为 XTW PRO。





利用 DAC 播放语音时，注意如果是 12 位 DAC，PCM 无论是 8 位数据位还是 16 位数据位，都要处理成 12 位数据格式，16 位数据位时要丢弃低 4 位的精度。8 位数据位的可丢可不丢。

之前对有符号 WAVs 数据的操作是（直接 16 位数据转 12 位数据送 DAC）

```
uint16 uiDacValue = 0;
```

```
uiDacValue = VoiceTestTable[uiCnt*2+1];
```

```
uiDacValue = uiDacValue << 8;
```

```
uiDacValue = uiDacValue + VoiceTestTable[uiCnt*2];
```

```
uiDacValue = uiDacValue >> 4;
```

```
DacOutVolSet(uiDacValue, 2);
```

修改后的代码：（注意有符号数据送 DAC 时候的处理方式）

```
int16 uiDacValue = 0;
```

```
///-----方法一
```

```
uiDacValue = VoiceTestTable[uiCnt*2+1];
```

```
uiDacValue = uiDacValue << 8;
```

```
uiDacValue = (int16)((uint16_t)uiDacValue + VoiceTestTable[uiCnt*2]);
```

```
uiDacValue = uiDacValue - 0x8000;
```

```
uiDacValue = uiDacValue >> 4;
```

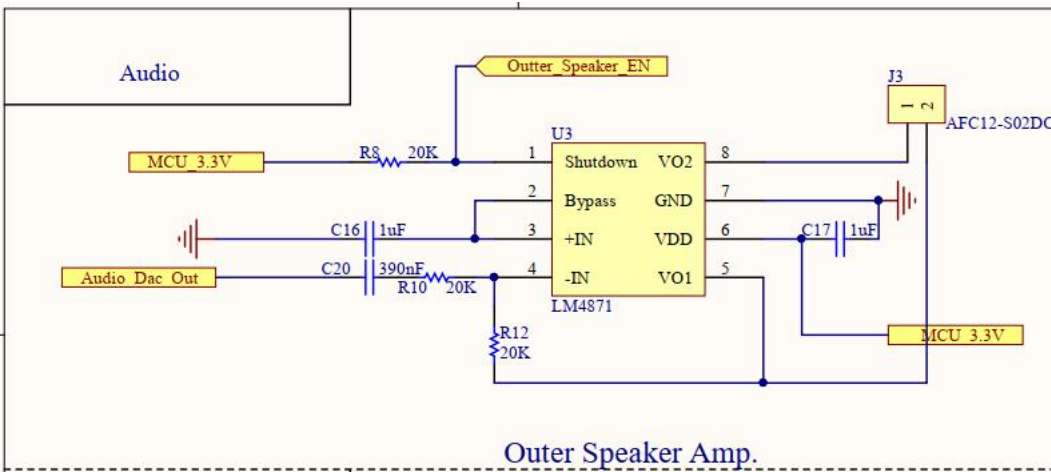
```
///-----方法二
```

```
///uiDacValue=((u8)(VoiceTestTable[uiCnt*2+1]-0x80)<<4)|(VoiceTestTable[uiCnt*2]>>4); //去尾
```

```
DacOutVolSet(uiDacValue, 2);
```

再次注意：有符号数据的模拟量波形是在 0 值附近上下波动的，所以需要向上平移 0x8000，后再取高 12 位的数据送到 12 位的 DAC 里面。

喇叭驱动用的 LM4871，如下图。



4. 版本历史

文档版本历史

| 日期 | 版本 | 变更 |
|-----------------|------|-------|
| 2021 年 1 月 20 日 | V1.0 | 初始版本。 |