



前言

本应用笔记描述了 ADPCM 音频固件编解码器，并提供了演示固件来说明如何在使用 STM32F103xx I²S 时，利用外部的 DAC 来播放 ADPCM 文件。

本应用笔记基于“AN2739：如何利用具有外部 I²S 音频编解码器的高容量 STM32F103xx 微控制器播放音频文件”。由于关于 SPI、I²S 和外部 DAC 的部分在 AN2739 中已有说明，此处不再赘述。因此，为了能充分理解本应用笔记，建议您参考 AN2739。

目录

1 **ADPCM 算法** **5**

 1.1 概述 5

 1.2 ADPCM 算法实现 5

 1.3 ADPCM 算法函数 6

 1.3.1 ADPCM_Encode 函数 6

 1.3.2 ADPCM_Decode 函数 6

2 **实现示例** **8**

 2.1 说明 8

 2.2 ADPCM 文件生成 8

 2.3 ADPCM 文件加载 9

 2.4 ADPCM 文件播放11

3 **结论** **13**

4 **修订历史** **14**



表格索引

表 1. ADPCM 算法函数 6

表 2. ADPCM_Encode 函数 6

表 3. ADPCM_Decode 函数 7

表 4. 文档修订历史 14

图片索引

图 1. ADPCM 文件生成器 8

图 2. DFU 文件管理器 10

图 3. DFU 文件管理器源浏览器 10

图 4. DfuSe 软件 11



1 ADPCM 算法

1.1 概述

自适应差分脉冲编码调制，或简称为 ADPCM，是波形编码的音频算法，它通过以前的信号值预测当前的信号值，并且仅发送真实值和预测值的差值。

在普通的脉冲编码调制（PCM）中，发送的是真实或实际信号值。

ADPCM 的优势在于通常预测信号值和真实信号值的差值相当小，这意味着与相应的 PCM 值相比，它可以用更少的位数来表示。

根据需要的质量和压缩比，差分信号按照 4（2 位）、8（3 位）、16（4 位）或 32（5 位）水平量化。

ADPCM 算法的实现方法有很多。它们在量化和预测模式上有所不同。

在本应用笔记中，我们提供了由交互式多媒体协会（IMA）开发的 4 位量化算法 IMA ADPCM。

选择 IMA ADPCM 有多种原因：

- 它可以应用于 8 kHz 到 44.10 kHz 之间的不同采样率
- 它能在较低的 CPU 的使用率和较少的内存占用情况下保证音频的播放质量
- 它具有广泛的实现方式，例如在 Windows 和 Mac 操作系统中

IMA 数字音频对焦和技术工作组公布的一份文件对 IMA ADPCM 算法进行了充分说明：“*在多媒体系统中增强数字音频兼容性的建议措施*” 版本 3。

1.2 ADPCM 算法实现

本应用笔记提供的 IMA ADPCM 算法用于对具有下述规范的音频文件进行编码：

- 音频格式：PCM
- 音频采样大小：16 位
- 通道：1（单声道）
- 音频采样率：8 kHz 到 44.1 kHz

每个 16 位 PCM 采样被编码为 4 位 ADPCM 采样，压缩比达到 1/4。

IMA ADPCM 算法实现包含两个函数，一个函数编码音频采样，另一个函数解码音频采样。

ADPCM 固件由两个文件组成：

- a) `adpcm.c`：它包括执行编码和解码的两个 ADPCM 函数的源代码。
- b) `adpcm.h`：它是 `adpcm.c` 的头文件。它应该包含在调用 ADPCM 函数的文件中。

1.3 ADPCM 算法函数

表 1 描述了 ADPCM 函数。

表 1. ADPCM 算法函数

函数名称	说明
ADPCM_Encode	将一个 16 位 PCM 采样值编码为一个 4 位 ADPCM 采样值。
ADPCM_Decode	将一个 4 位 ADPCM 采样值解码为一个 16 位 PCM 采样值。

1.3.1 ADPCM_Encode 函数

表 2 描述了 ADPCM_Encode 函数。

表 2. ADPCM_Encode 函数

函数名称	ADPCM_Encode
原型	uint8_t ADPCM_Encode(int16_t sample);
函数说明	将一个 16 位 PCM 采样值编码为 4 位 ADPCM 采样值。
输入参数	采样值：一个 16 位的 PCM 采样值
输出参数	无
返回参数	将编码后的 ADPCM4 位采样值加载进一个字节中

ADPCM_Encode 函数返回一个包含 4 位 ADPCM 采样值的字节。软件将 2 个 ADPCM 采样值存入一个字节以便节约内存空间。

示例：

```
// 输入: pcm_sample1 和 pcm_sample2； 2 个 16 位 PCM 采样值。
// 输出: adpcm_byte； 存入一个字节的两个 4 位 ADPCM 采样值。

uint8_t code;
/* 编码第一个 16 位采样值 */
code = ADPCM_Encode(pcm_sample1);
/* 存储第一个 4 位采样值 */
adpcm_byte = code;
/* 编码第二个 16 位采样值 */
code = ADPCM_Encode(pcm_sample2);
/* 存储第二个 4 位采样值 */
adpcm_byte |= (code << 4);/* adpcm_byte 包含两个 4 位 ADPCM 采样值 */
```

1.3.2 ADPCM_Decode 函数

表 3 描述了 ADPCM_Decode 函数。



表 3. ADPCM Decode 函数

函数名称	ADPCM_Decode
原型	int16_t ADPCM_Decode(uint8_t code);
函数说明	将一个 4 位 ADPCM 采样值解码为一个 16 位 PCM 采样值。
输入参数	代码：一个 8 位的数据项，其 4 个 LSB 包含编码后的 ADPCM 采样值
输出参数	无
返回参数	一个 16 位的 PCM 采样值

ADPCM_Encode 函数的输入是一个包含 4 位 ADPCM 采样值的字节。在调用 ADPCM_Decode 函数之前，该软件必须提取出 4 位 ADPCM 数据并将其存入一个字节。

示例:

```
// 输入: adpcm_byte; 存入一个字节的两个 4 位 ADPCM 采样值。
// 输出: pcm_sample1 和 pcm_sample2; 2 个 16 位 PCM 采样值。

uint8_t code;
/* 提取第一个 ADPCM4 位采样值 */
code = (adpcm_byte & 0x0F);
/* 解码第一个 ADPCM 采样值 */
pcm_sample1 = ADPCM_Decode(code);
/* 提取第二个 ADPCM 采样值 */
code = (adpcm_byte >> 4);
/* 解码第二个 ADPCM 采样值 */
pcm_sample2 = ADPCM_Decode(code);
```

2 实现示例

2.1 说明

可以在 STM3210E-EVAL 板上运行提供的例子。它是一个典型的音频应用，包括：

- 首先用 PC 软件将一个 PCM 文件编码为 ADPCM 格式（这个软件名为 **muse.exe**，是由 ST 开发的。它实现了与 STM32F103xx 增强型系列产品相同的 ADPCM 算法编码器。**muse.exe** 在本应用笔记附带的压缩软件包中：请参考 [第 2.2 章节](#) 了解如何应用这个工具。）
- 然后将编码文件加载到 Flash
- 使用大容量 STM32F103xx 微控制器解码文件，驱动 DAC 播放音频。

在 STM3210E-EVAL 板上，128 兆位 NOR Flash 存储 ADPCM 文件和 STM32F103xx 器件上运行的固件解码器。然后用它解码该数据流，并且通过应用 I²S 外部 DAC 回放音频采样。

硬件环境基于应用笔记“AN2739：如何使用高密度 STM32F103xx 微控制器与外部 I²S 音频编解码器播放音频文件”，请参考该应用笔记以获得更多信息。

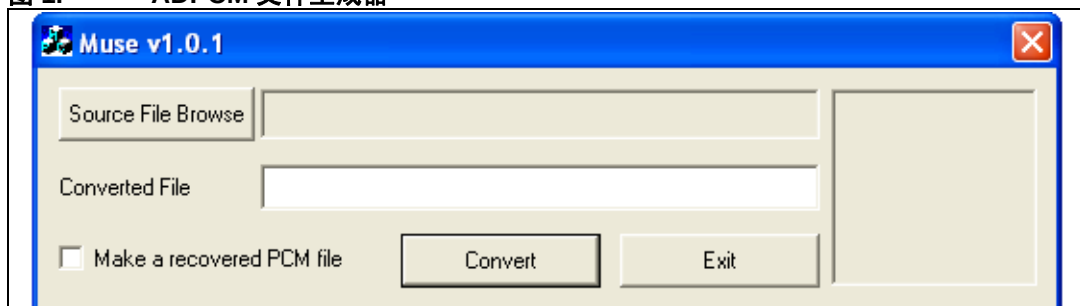
要运行这个演示示例，需要进行三个步骤：

1. 生成 ADPCM 文件
2. 将 ADPCM 文件载入 NOR Flash
3. 播放 ADPCM 文件

2.2 ADPCM 文件生成

本应用笔记包括一个名为 Muse 的 PC 软件，该软件可以将 PCM 文件编码为 ADPCM 格式。

图 1. ADPCM 文件生成器



运行 *Muse.exe* 程序并按 **Source File Browse** 按钮，出现 **File Selection** 对话框，提示您选择一个 WAV 源文件。如前所述，WAV 文件应该满足以下要求：

- 音频格式：PCM
- 音频采样大小：16 位
- 通道：1（单声道）
- 音频采样率：8 kHz 到 44.1 kHz

如果您的 WAV 文件不能满足这些条件，就必须先用一个音频转换工具转换，比如 *sndrec32.exe*，它是 Microsoft® 为音频记录提供的默认程序。

一旦选择了待转换的文件，就会出现 **New File Selection** 对话框，提示选择 ADPCM 转换文件的存储地址。

可以通过勾选 **Make a recovered PCM file** 复选框获取编码的音频文件的 PCM 格式。在这种情况下，生成的 ADPCM 文件被解码并格式化为 WAV 文件。这个 WAV 文件名为 *Recover.wav*，它生成在与 WAV 源文件相同的目录下。您可以用这种方式听原始的 WAV 文件和恢复的文件，以比较声音质量。

2.3 ADPCM 文件加载

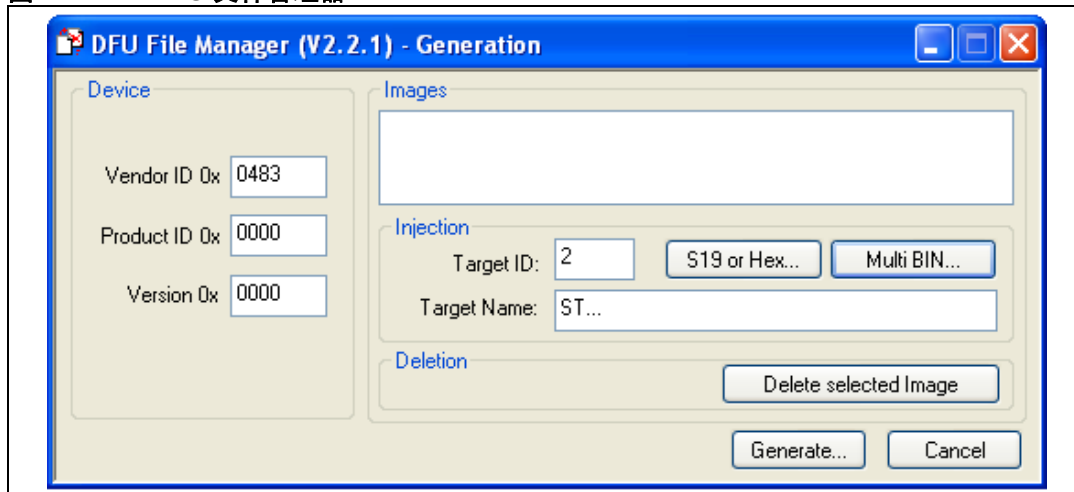
用 DfuSe 软件将 ADPCM 文件载入 NOR Flash 为此，需要进行两个步骤：

1. 生成 .dfu 映像
2. 将该 .dfu 映像载入 NOR Flash

生成 .dfu 映像

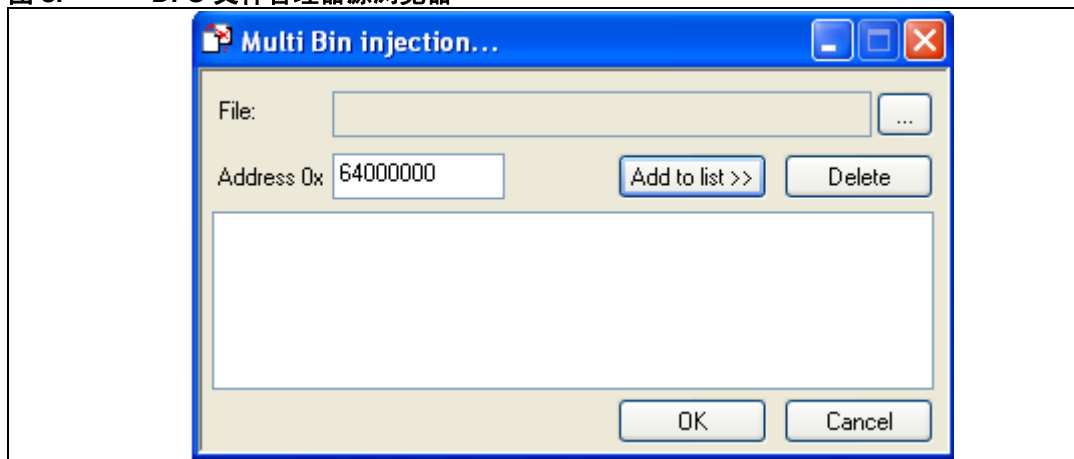
为了生成 .dfu 映像，必须使用 DfuSe 包中提供的 DFU File Manager 软件。那么，打开 DFU File Manager 程序并选择 DFU 文件生成选项。[图 2](#) 中所示的窗口打开。必须在选择目标 ID 时选择 2，这意味着文件将被载入 NOR Flash。然后，点击 **Multi BIN...** 按钮打开源浏览器并选择您的 ADPCM 文件。

图 2. DFU 文件管理器



该 NOR Flash 从地址 0x6400 0000 映射到 FSMC 接口上，这正是 ADPCM 文件存储区域的起始地址。所以，必须将 .dfu 映像的地址设置为 0x6400 0000（参考 图 3）。最后，点击 **Add to list**，然后点击 **Generate** 完成 DFU 文件生成。

图 3. DFU 文件管理器源浏览器

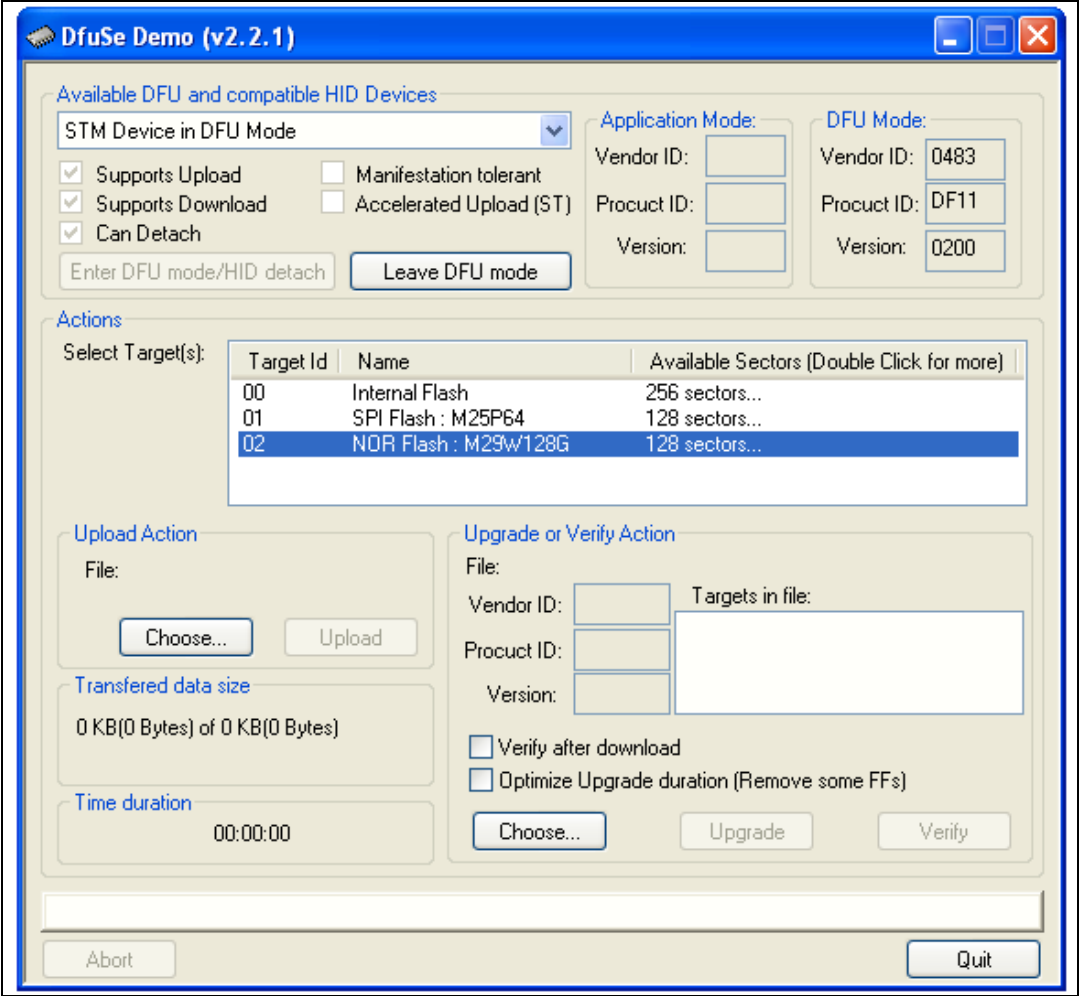


载入 DFU 映像

为了将 DFU 映像载入 NOR Flash，必须：

1. 使用 USB 线将 STM3210E-EVAL 板连接到 PC
2. 在 STM3210E-EVAL 板上下载并运行器件固件升级（DFU）演示。您可以在 STM3210xxx USB 开发工具中找到这个固件
3. 在 PC 上开始 DfuSe 软件（参考 图 4）

图 4. DfuSe 软件



选择 NOR Flash 作为目标，然后点击 **Choose** 按钮选择您的 ADPCM DFU 映像，该按钮位于 **Upgrade or Verify Action** 区域。最后，点击 **Upgrade** 执行下载。

2.4 ADPCM 文件播放

此时，您已成功地将 ADPCM 文件载入 NOR Flash。现在可以开始播放演示了。请在应用笔记包打开 *ADPCM_AN* 文件夹。打开文件夹项目，选择合适的项目工作区并在您的工具链中打开它。建立所有源文件，加载项目映像，然后运行程序。

STM3210E-EVAL 板上的 LCD 会显示音频指令，而摇杆和按键将被用于控制演示。允许的音频指令：播放，暂停，停止，前进，后退和音量控制。

您可以选择音频输出接口，可以是 STM3210E-EVAL 板提供的扬声器或耳机。耳机能实现更好的音频质量。您可以通过音频插孔将它们连接到 STM3210E-EVAL 板。

需要注意的是音频文件的后退和前进，必须按下述使用摇杆按钮：

- 向右推摇杆来前进
- 向左推摇杆来后退

3 结论

本应用笔记描述了如何实现用于 STM32F103xx 器件的 IMA ADPCM 固件音频编解码器。它也提供了一个应用实例，解码一个存储于 Flash 的 ADPCM 编码的音频流，并回放解码后的采样。

本应用笔记应用了高容量 STM32F103xx，它具有 I²S 接口，可以与 STM3210E-EVAL 评估板上使用的外部 DAC 进行通信。

但是其他不具备 I²S 外设的 STM32F10xxx 器件也可以利用 PWM 输出运行这个 ADPCM 固件编解码器和播放音频文件。

在这种情况下，您只需更新 STM3210B-EVAL 评估板的解码示例，并用 PWM 输出进行音频回放。

4 修订历史

表 4. 文档修订历史

日期	修订	变更
2009 年 3 月 04 日	1	初始版本。
2009 年 4 月 30 日	2	在代码方面， u8 和 s16 分别被更新为 uint8_t 和 int16_t。



重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2015 STMicroelectronics - 保留所有权利