

---

## STM32 微控制器上的 PDM 音频软件解码

---

### 1 简介

本应用笔记以 ST MP45DT02 MEMS 麦克风与 STM32 微控制器的连接为例，介绍了 PDM 信号解码和音频信号重建时使用的软件优化算法和架构。该软件可直接采集麦克风输出的脉冲密度调制 (PDM) 数据，并将其转换为 16 位脉冲编码调制 (PCM) 格式。

本文档还提供了快速入门信息，介绍如何通过基于 STM32F4 微控制器和 STM32F4DISCOVERY 板的 I2S 对 PDM 库进行单麦克风采集。

有关这一过程的详细信息，请参见 AN3997 *使用 STM32F4DISCOVERY 进行音频回放和录音*。

# 目录

<b>1</b>	<b>简介</b> .....	<b>1</b>
<b>2</b>	<b>PDM 信号简介</b> .....	<b>4</b>
<b>3</b>	<b>硬件接口：麦克风连接和采集</b> .....	<b>5</b>
<b>4</b>	<b>软件接口：数字信号处理</b> .....	<b>6</b>
	4.1 PDM 数字滤波和采样 .....	6
	4.2 数字信号调节 .....	6
<b>5</b>	<b>PDM 音频软件解码库说明</b> .....	<b>7</b>
	5.1 PDM_Filter_Init .....	8
	5.2 PDM_Filter_xx_xx .....	8
<b>6</b>	<b>版本历史</b> .....	<b>9</b>

## 图片索引

图 1.	麦克风与 STM32 的连接框图 .....	5
图 2.	数字信号处理 .....	6

## 2 PDM 信号简介

脉冲密度调制 (PDM) 是一种用于表示数字域中的模拟信号的调制形式。

在 PDM 信号中，不会像在 PCM 中一样将具体的幅值编码到脉冲中。而是对与模拟信号幅值相应的、一系列脉冲的相对密度进行编码。

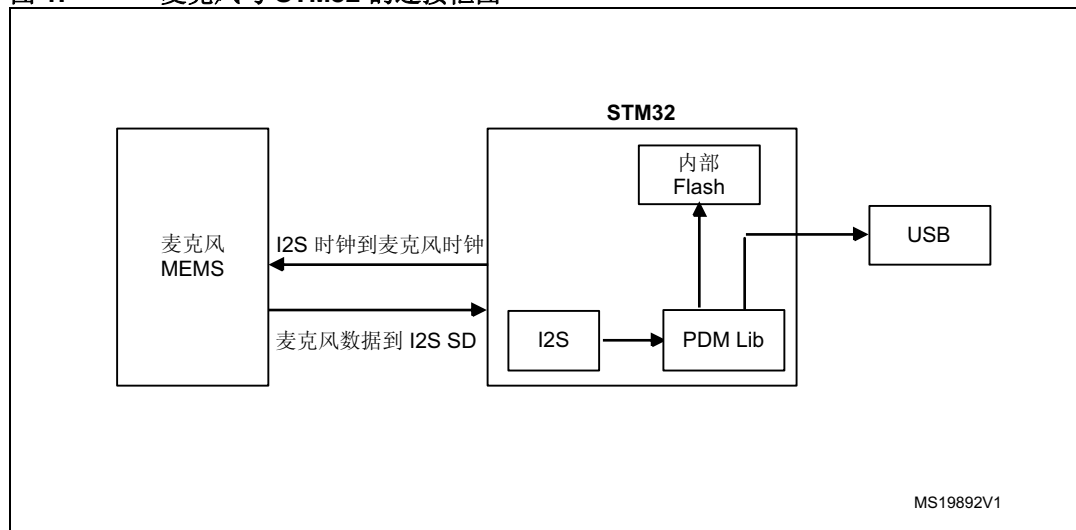
要从 PDM 比特流中获取数据帧，通常使用采样滤波器。采样过程第一阶段要降低采样频率，然后使用高通滤波器去除信号直流偏移。

### 3 硬件接口：麦克风连接和采集

MP45DT02 MEMS 麦克风会输出一个 PDM 信号，该信号具有高频（1 到 3.25 MHz）1 位数字采样流。

该输出信号通过 STM32 微控制器的同步串口（SPI 或 I2S）以 8 个样本为一组进行采集。麦克风的 PDM 输出信号与其输入时钟同步；因此，可通过一个 STM32 SPI/I2S 外设生成麦克风的时钟信号。

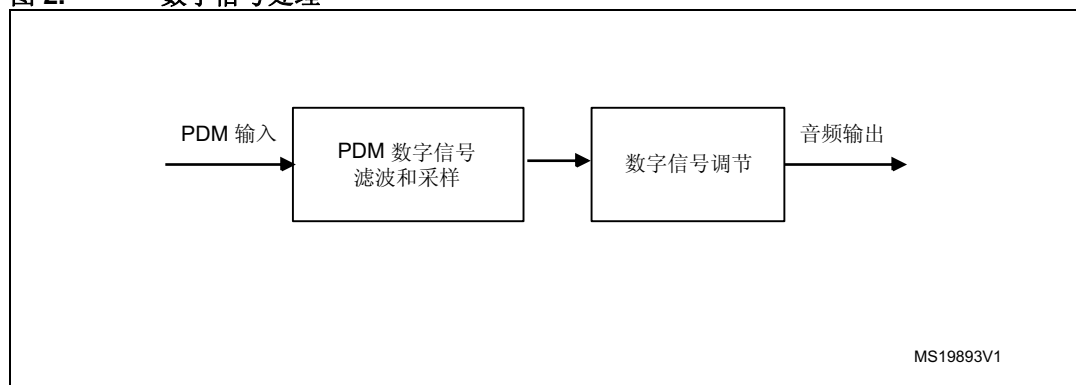
图 1. 麦克风与 STM32 的连接框图



## 4 软件接口：数字信号处理

麦克风输出的数据发送到采样过程，该过程包括两个组成部分：一个用于将 1 位 PDM 数据转换为 PCM 数据的采样滤波器，还有两个可单独配置的 IIR 滤波器（低通和高通滤波器）。转换后的音频采用 16 位脉冲编码调制 (PCM) 格式。完成转换之后会产生原始数据，可根据实际应用程序对这些数据进行处理（以波形/压缩数据形式存储到大容量存储介质，通过 I2S 外设传输到外部音频编码器 DAC...）。

图 2. 数字信号处理



### 4.1 PDM 数字滤波和采样

为了获得所需频率和分辨率的声音信号，需要对来自麦克风的 PDM 信号进行滤波和采样。

麦克风输出的 PDM 数据的频率（等于麦克风的时钟输入信号频率）必须是系统所需的最终音频输出信号的整数倍。例如，若要执行 80 倍采样以得到 30 kHz 输出频率，则需要为麦克风提供一个 2.4 MHz 的时钟频率。

滤波器是一个 16 位的输出管道流，同相增益 (0 dB) 的输出范围为 [-32768, 32767]。

### 4.2 数字信号调节

上述滤波管道流中产生的数字音频信号会经过进一步处理，以便用于信号调节。

此信号调节的第一阶段是使用高通滤波器，专门去除信号直流偏移。该过程通过所含截止频率低于可听频率范围的 IIR 滤波器实现，从而保证信号质量。

第二阶段通过使用 IIR 滤波器的低通滤波器实现。

这两个滤波器均可通过滤波器初始化函数使能/禁止和配置（截止频率）。相关增益由外部整型变量 (MicGain) 控制，如以下公式所示： $G = \text{MicGain}/64$ 。

## 5 PDM 音频软件解码库说明

PDM 库由四个 PDM 滤波函数的结构和实施过程组成。该库使用两个缓冲区，即 PDM 输入缓冲区和 PCM 输出缓冲区；应用程序必须在主程序中定义这两个缓冲区。

- 输入缓冲区（数据）是一个 `uint8` 变量，其长度至少等于  $(\text{输出频率} / 1000 * \text{采样因子} * \text{麦克风输入通道} / 8)$ 。
- 输出缓冲区（数据输出）是一个 `uint16` 变量，其长度至少等于  $(\text{输出频率} / 1000 * \text{麦克风输出通道})$ 。

`pdm_filter.h` 文件中定义了库结构，该结构用于配置滤波器；其构成如下所示：

```
typedef struct {  
    uint16_t Fs;  
    float LP_HZ;  
    float HP_HZ;  
    uint16_t Out_MicChannels;  
    char InternalFilter[34];  
} PDMFilter_InitStruct;
```

- **Fs:** 定义了滤波器的输出频率（单位为 Hz）。
- **LP\_HZ:** 定义了低通滤波器的截止频率。若该频率为 0，则表示已禁止低通滤波器。
- **HP\_HZ:** 定义了高通滤波器的截止频率。若该频率为 0，则表示已禁止高通滤波器。
- **In\_MicChannels:** 定义了输入流中的麦克风个数。此参数用于指定输入缓冲区中的麦克风交错。PDM 样本按每组八个进行分组，并采用 `u8` 格式（8 位）。
- **Out\_MicChannels:** 定义输出流中的麦克风数量；此参数用于交错输出缓冲区中的不同麦克风。每个样本即为一个 16 位值。
- **InternalFilter:** 定义 PDM 采样期间解码库内部使用的一个 34 字节内存。

PDM 音频软件解码库中含有一个头文件 `pdm_filter.h` 和以下平台所使用的二进制/目标代码：

- `libPDMFilter_IAR.a` : 针对 IAR 编译器
- `libPDMFilter_Keil.lib` : 针对 ARM 编译器
- `libPDMFilter_GCC.a` : 针对 GNU 编译器

可在“STM32F4-DISCOVERY 板固件包”（v1.10 及更高版本）的“Utilities\STM32F4-Discovery”路径下找到此解码库。

## 5.1 PDM\_Filter\_Init

此函数用于初始化单信号输出通道的滤波器。其定义如下：

```
Void PDM_Filter_Init (PDMFilter_InitStruct * Filter)
```

该函数会按照 PDMFilter\_InitStruct 中指定的参数初始化 PDM 滤波器。

下面以单个麦克风在 16 kHz 下初始化 PDM 滤波器为例：

```
PDMFilter_InitStruct Filter;  
Filter.LP_HZ = 8000;  
Filter.HP_HZ = 10;  
Filter.Fs = 16000;  
Filter.Out_MicChannels = 1;  
Filter.In_MicChannels = 1;
```

```
PDM_Filter_Init ((PDMFilter_InitStruct *) Filter);
```

在调用 PDM\_Filter\_Init() 函数之前，应用程序代码必须使能 STM32 微控制器 CRC 外设的时钟（相关配置在 RCC 寄存器中完成）。

否则，PDM\_Filter\_Init() 函数调用将失败，并进入无限循环。

## 5.2 PDM\_Filter\_xx\_xx

这些函数用于处理单个麦克风中一毫秒内输出的 PDM 数据。完成处理后会返回 PCM 样本的个数，该值等于滤波器初始化中定义的频率除以 1000。若频率值不是 1000 的倍数（如 44100 Hz 或 22050 Hz），则函数返回的样本数等于频率的除 1000 后向下取整的结果（44 和 22）。

函数定义如下：

```
PDM_Filter_XX_XX(uint8_t* data, uint16_t* dataOut, uint16_t MicGain,  
PDMFilter_InitStruct * Filter)
```

- **data:** 指含有 PDM 的输入缓冲区；应用程序必须将要处理的麦克风的首个输入样本指针传递给函数。
- **dataOut:** 指由 PDM\_Filter 函数处理的输出缓冲区。应用程序必须将要获取的通道的首个样本指针传递给函数。
- **Volume:** 指用于指定麦克风增益的一个数值，在 0 到 64 之间。
- **Filter:** 指含有用户使用 PDM\_Filter\_Init 函数指定的所有滤波器参数的结构。

此函数具有四种不同的实现方式，具体取决于采样因子（64 或 80）以及输入缓冲区的 LSB 或 MSB 表达式。



## 6 版本历史

表 1. 文档版本历史

日期	版本	变更
2011 年 10 月 27 日	1	初始版本。

**请仔细阅读：**

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本档中信息的提供仅与ST产品有关。意法半导体公司及其子公司（“ST”）保留随时对本档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有ST产品均根据ST的销售条款出售。

买方自行负责对本文所述ST产品和服务的选择和使用，ST概不承担与选择或使用本文所述ST产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本档任何部分涉及任何第三方产品或服务，不应被视为ST授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在ST的销售条款中另有说明，否则，ST对ST产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且/或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的ST产品如有不同于本档中提出的声明和/或技术特点的规定，将立即导致ST针对本文所述ST产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大ST的任何责任。

ST和ST徽标是ST在各个国家或地区的商标或注册商标。

本档中的信息取代之前提供的所有信息。

ST徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2013 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

[www.st.com](http://www.st.com)