

引言

数字MEMS（微机电系统）麦克风专为需要小尺寸、高音质、高可靠性和高经济效益的音频应用而设计。

它们的封装尺寸小且噪声低，可在单个器件中实现多个麦克风，并通过提供免提人机界面、噪声消除和高质量音频捕获，促进了工业和消费类应用中音频技术的不断发展。

STM32 32位Arm® Cortex® MCU和STM32 Arm® Cortex® MCU提供各种音频功能和丰富的连接功能，包括串行接口和增强型声音捕获接口，使用户能够轻松地基于麦克风的应用构建解决方案。

本文档针对具有脉冲密度调制（PDM）输出的数字MEMS麦克风，描述了如何在单声道和立体声配置中使用SPI/I2S、SAI和DFSDM外设将麦克风连接到STM32 MCU和MPU。本应用笔记提供指南和基于CubeMX的示例，这些示例展示了如何正确地配置STM32器件，以便采集并处理来自麦克风的原始数据，并将其转换为标准音频数据。

目录

1	PDM数字麦克风概述	7
1.1	声音采集概述	7
1.2	PDM数字麦克风框图	8
1.3	基础数字麦克风连接	9
1.4	PDM和PCM信号	12
1.4.1	脉冲密度调制 (PDM)	12
1.4.2	脉冲编码调制 (PCM)	12
1.4.3	PDM到PCM的转换	13
1.5	声学参数	13
1.5.1	灵敏度	13
1.5.2	信噪比 (SNR)	13
1.5.3	声学过载点 (AOP)	14
1.5.4	电源抑制比 (PSRR)	14
1.6	数字麦克风的附加价值	14
1.7	可用的ST数字麦克风	15
2	将PDM数字麦克风连接到STM32 MCU和MPU	16
2.1	串行外设接口 (SPI) /集成电路内置音频总线 (I2S)	16
2.1.1	单声道配置	16
2.1.2	立体声配置	18
2.2	串行音频接口 (SAI)	20
2.2.1	使用单个子模块	20
2.2.2	使用两个同步SAI子模块	23
2.2.3	使用PDM接口	24
2.3	数字滤波器, 用于 $\Sigma\Delta$ 调制器 (DFSDM)	27
2.3.1	立体声配置	28
2.4	时钟方面的考虑	28
2.4.1	数字麦克风时钟	28
2.4.2	外设时钟	29
2.5	GPIO数量的考虑	31
3	数字信号处理	32
3.1	PDM音频软件解码库	32

3.1.1	概述	32
3.1.2	数字数据流	32
3.1.3	数字信号处理	33
3.2	数字信号处理用DFSDM滤波器	34
3.2.1	数字数据流：采集和处理	34
4	基于STM32CubeMX的配置示例	35
4.1	示例1：使用I2S、SPI或者单个SAI模块，连接单声道或者立体声模式下的数字麦克风与I2S、SPI或单个SAI子模块	35
4.1.1	使用STM32CubeMX配置硬件	35
4.1.2	添加PDM软件解码库中间件文件	47
4.2	示例2：使用两个同步SAI子模块，连接立体声模式下的数字麦克风。两个同步子模块连接数字麦克风与SAI	48
4.2.1	使用STM32CubeMX配置SAI	48
4.2.2	添加PDM软件解码库中间件文件	53
4.3	示例3：使用SAI的PDM接口，连接立体声模式下的数字麦克风。PDM接口连接数字麦克风与SAI	53
4.3.1	使用STM32CubeMX配置SAI	53
4.3.2	添加PDM软件解码库中间件文件	57
4.4	示例4：使用DFSDM连接数字麦克风	58
4.4.1	使用STM32CubeMX配置DFSDM	58
5	结论	64
6	版本历史	65

表格索引

表1.	DOUT信号模式选择	9
表2.	引脚说明	9
表3.	数字麦克风的附加价值	14
表4.	ST数字麦克风	15
表5.	推荐IO线与要连接的数字麦克风数量	24
表6.	应用示例和相关的麦克风时钟频率	29
表7.	用于连接一个数字麦克风的硬件	31
表8.	用于连接两个数字麦克风的硬件	31
表9.	用于连接四个数字麦克风的硬件	31
表10.	I2S2时钟配置和精度	36
表11.	SPI时钟配置和精度	39
表12.	时钟配置和精度	42
表13.	SAI时钟配置和精度	49
表14.	DFSDM滤波阶数	60
表15.	DFSDM时钟配置精度值	63
表16.	文档版本历史	65
表17.	中文文档版本历史	65

图片目录

图1.	音频应用中声音采集的示例	7
图2.	典型PDM数字MEMS麦克风框图	8
图3.	单声道配置 - 在右侧通道上生成数据	9
图4.	右侧通道数据模式	10
图5.	单声道配置 - 在左侧通道上生成数据	10
图6.	左侧通道数据模式	10
图7.	立体声配置：共享一条主数据线	11
图8.	立体声配置数据模式	11
图9.	PDM 信号	12
图10.	PCM 信号	13
图11.	在单声道配置中将一个数字麦克风连接至SPI或I2S	17
图12.	在立体声配置中将两个数字麦克风连接至SPI模块	18
图13.	立体声模式时序图	19
图14.	在单声道配置中将一个数字麦克风连接至SAI	20
图15.	在立体声配置中使用SAI子模块和定时器将两个数字麦克风连接至SAI	22
图16.	在立体声配置中使用两个同步SAI子模块将两个数字麦克风连接至SAI	23
图17.	能够连接最多四对麦克风的PDM接口	25
图18.	使用SAI PDM接口时的数据格式（具有32位槽和8个麦克风）	26
图19.	使用SAI PDM接口时的数据格式（具有8位槽和8个麦克风）	26
图20.	能够连接最多4个数字麦克风的DFSDM	27
图21.	立体声配置	28
图22.	SPI的总线和内核时钟拓扑	30
图23.	SAI和DFSDM的总线和内核时钟拓扑	30
图24.	数字数据采集和处理（框图）	33
图25.	数字信号处理	33
图26.	使用DFSDM采集和处理数字数据（框图）	34
图27.	I2S GPIO引脚配置	36
图28.	I2S2时钟配置	37
图29.	I2S配置	37
图30.	I2S参数设置	38
图31.	I2S DMA设置	38
图32.	SPI GPIO和引脚配置	39
图33.	SPI时钟配置	39
图34.	SPI 配置	40
图35.	SPI参数设置	40
图36.	SPI DMA设置	41
图37.	SAI GPIO和引脚配置	41
图38.	单声道模式下16 kHz时的SAI时钟配置	42
图39.	SAI配置	43
图40.	SAI参数设置	44
图41.	SAI DMA设置	45
图42.	DMA请求设置	45
图43.	TIM GPIO和引脚配置	46
图44.	TIM配置	46
图45.	TIM参数设置	47
图46.	SAI GPIO和引脚配置	48
图47.	SAI时钟配置	49
图48.	SAI参数设置	51

图49.	SAIB参数设置	52
图50.	SAI DMA设置	52
图51.	SAI GPIO和引脚配置	53
图52.	16 kHz时2个麦克风的SAI时钟配置	54
图53.	SAI配置	54
图54.	SAI参数设置	55
图55.	SAI DMA设置	56
图56.	CORTEX_M7配置	56
图57.	Cortex M7参数设置	57
图58.	立体声模式下的DFSDM GPIO配置	58
图59.	DFSDM引脚配置	59
图60.	DFSDM通道1配置	59
图61.	DFSDM通道0配置	60
图62.	DFSDM滤波器0配置	61
图63.	DFSDM滤波器1配置	61
图64.	DFSDM输出时钟配置	62
图65.	DFSDM DMA设置	62
图66.	DFSDM时钟配置	63

1 PDM数字麦克风概述

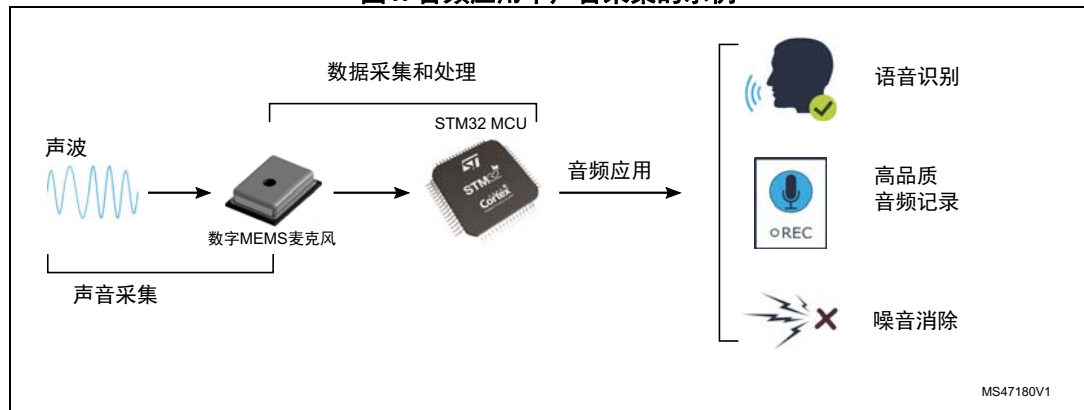
本章提供PDM数字麦克风的简述和将其与STM32器件相连的基本案例。STM32 MCU和MPU是基于Arm^{®(a)}的器件。



1.1 声音采集概述

数字MEMS麦克风是将声压波转换为数字信号的传感器。STM32 MCU和MPU通过特定外设从麦克风采集数字数据，这些数据经过处理并被转换为标准音频数据。然后，由微控制器按照目标音频应用处理音频数据。

图1. 音频应用中声音采集的示例



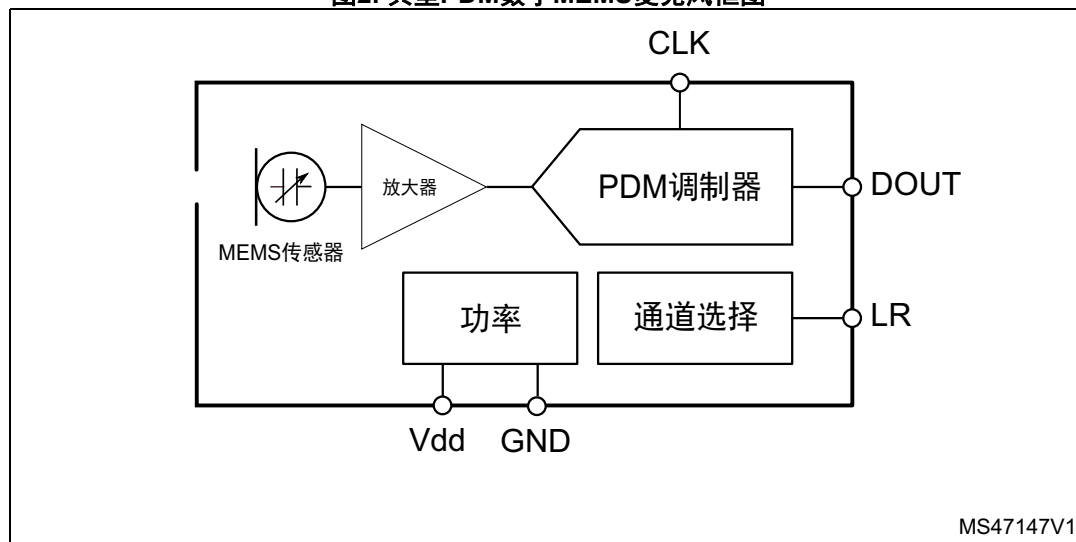
MS47180V1

a. Arm是Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

1.2 PDM数字麦克风框图

数字麦克风中的主要部件是MEMS传感器、放大器和PDM调制器。

图2. 典型PDM数字MEMS麦克风框图



MEMS传感器

MEMS传感器是一个可变电容，将声波导致的空气压力变化转换为电压。

放大器

放大器对MEMS传感器提供的电压进行缓冲，并向PDM调制器提供足够强的信号。

PDM调制器

PDM调制器将缓冲模拟信号转换为串行脉冲密度调制信号。时钟输入（CLK）用于控制PDM调制器。ST数字麦克风的时钟频率范围在1 MHz至3.25 MHz之间。该频率定义生成离散时间表示（PDM比特流）的放大器模拟输出信号采样频率。

通道选择

麦克风的输出被驱动至所选时钟边沿上的合适电平，然后进入高阻抗状态并持续另外半个时钟周期。通道选择定义数字麦克风输出有效数据的时钟边沿。LR引脚必须连接到Vdd或GND。

表 1显示了如何选择DOUT信号模式。

表1. DOUT信号模式选择

LR	DOUT	
	CLK低	CLK高
GND	有效数据	高阻抗
Vdd	高阻抗	有效数据

电源

电源为不同数字麦克风组件供应Vdd和GND。由于任何波动都可能在输出上生成噪声，因此应提供正确的电源。

引脚说明

表2. 引脚说明

引脚名称	功能	方向
Vdd	3.3 V电源	输入
GND	0 V	输入
LR	左/右选择	输入
CLK	同步时钟	输入
DOUT	左/右PDM数据输出	输出

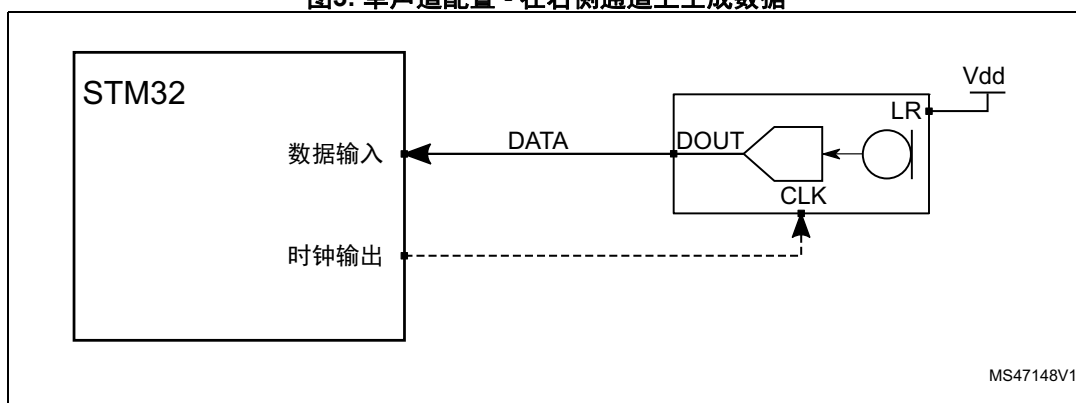
1.3 基础数字麦克风连接

单声道模式

在该模式下，LR引脚可以连接到Vdd或GND。

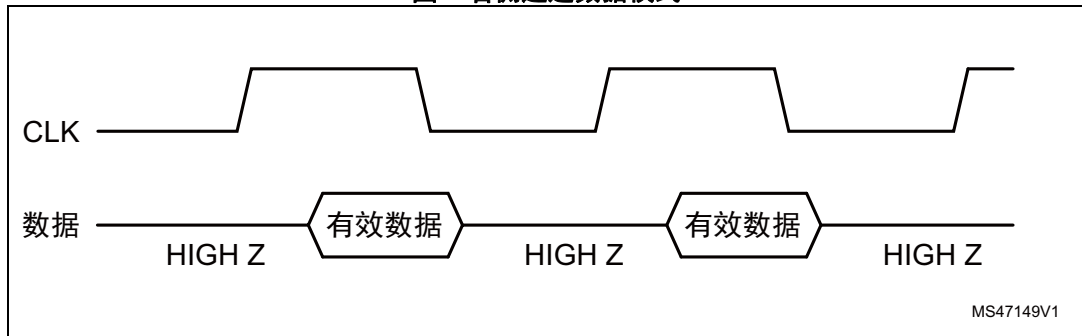
LR引脚连接到Vdd

图3. 单声道配置 - 在右侧通道上生成数据



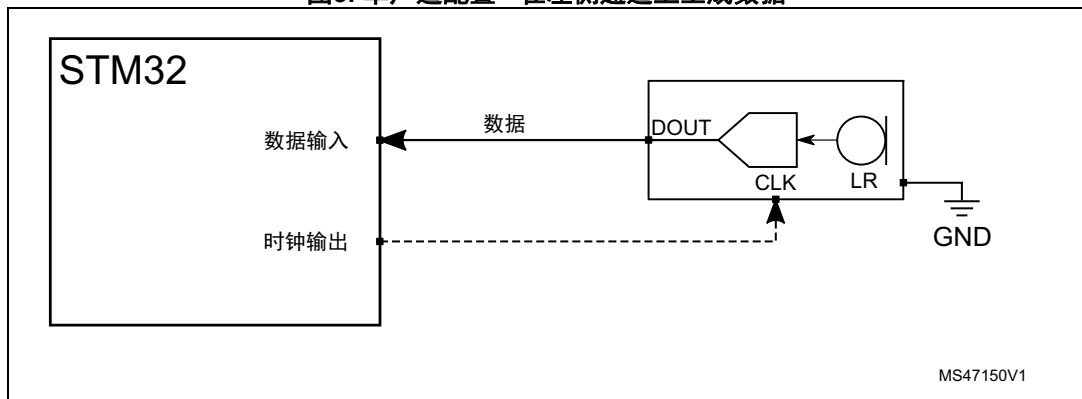
在时钟的上升沿，麦克风在一半时钟周期生成有效数据，然后进入高阻抗状态并持续另一半时钟周期。

图4. 右侧通道数据模式



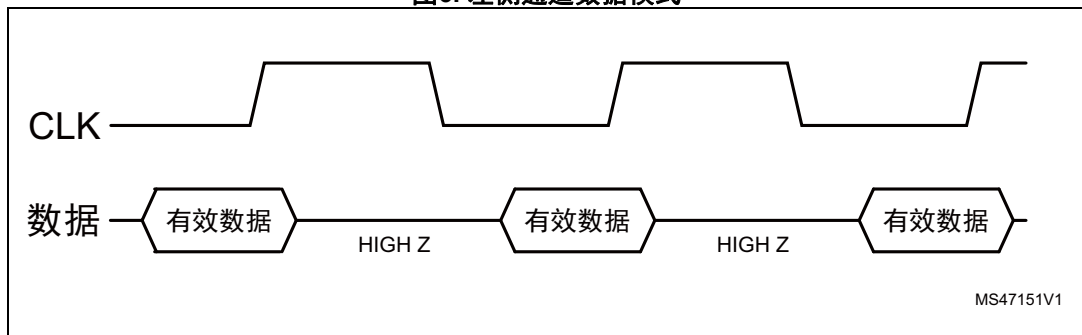
LR引脚连接到GND

图5. 单声道配置 - 在左侧通道上生成数据



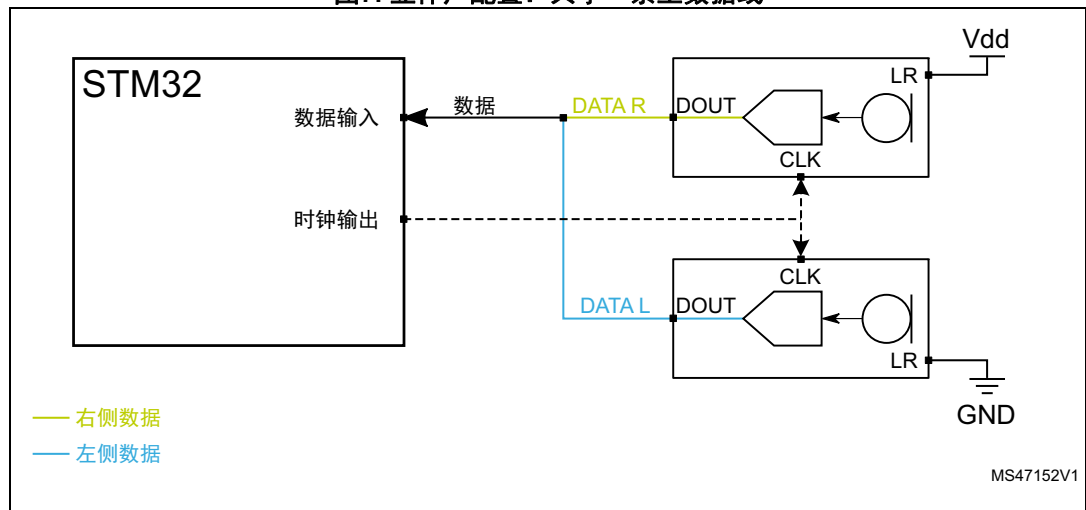
在时钟的下降沿，麦克风在一半时钟周期生成有效数据，然后进入高阻抗状态并持续另一半时钟周期。

图6. 左侧通道数据模式



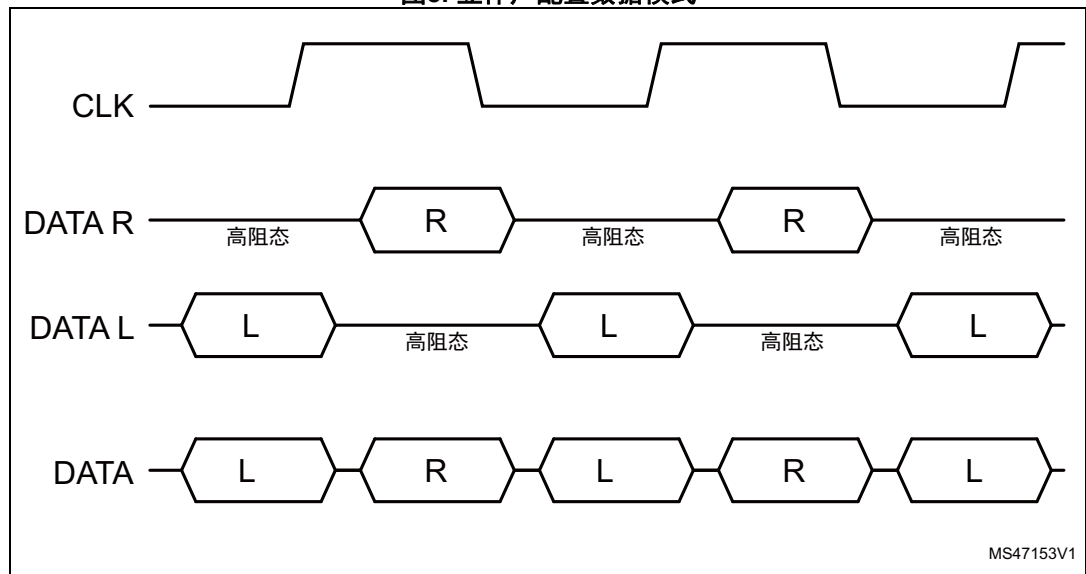
立体声配置

图7. 立体声配置：共享一条主数据线



两个不同的数字MEMS麦克风连接在同一条数据线上，通过将LR引脚设置为Vdd将第一个数字MEMS麦克风配置为在时钟上升沿生成有效数据，并通过将LR引脚设置为GND将另一个数字MEMS麦克风配置为在下降沿生成有效数据。

图8. 立体声配置数据模式

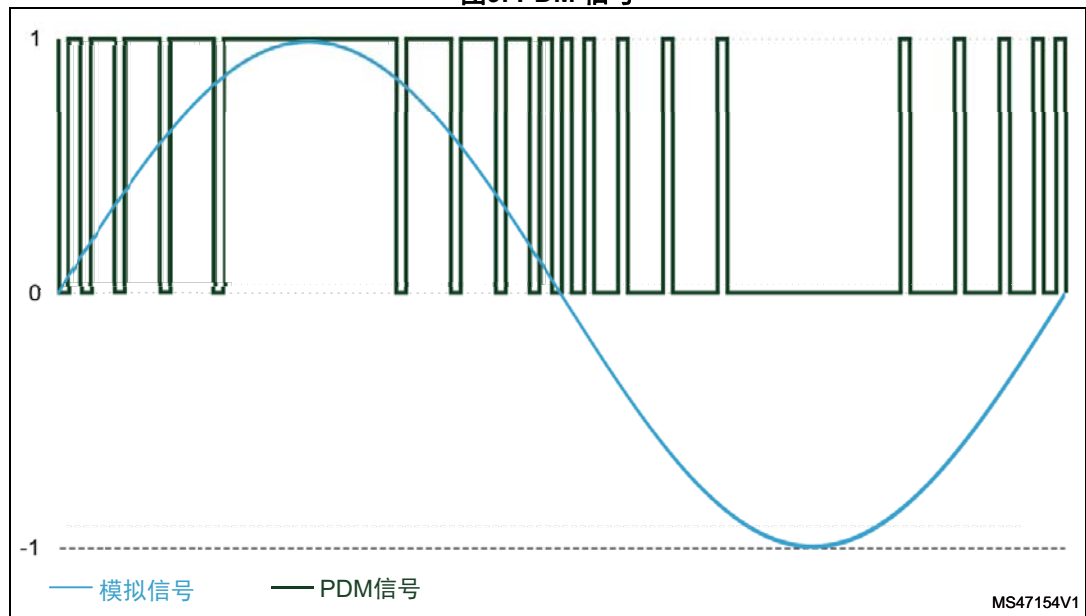


1.4 PDM和PCM信号

1.4.1 脉冲密度调制 (PDM)

PDM是一种调制形式，用于表示数字域中的模拟信号。它是1位数字采样的高频数据流。在PDM信号中，脉冲的相对密度对应于模拟信号的幅度。大量的'1'对应于高（正）幅度值，而大量的'0'对应于低（负）幅度值，交替的'1'和'0'对应于幅度值0。

图9. PDM 信号



1.4.2 脉冲编码调制 (PCM)

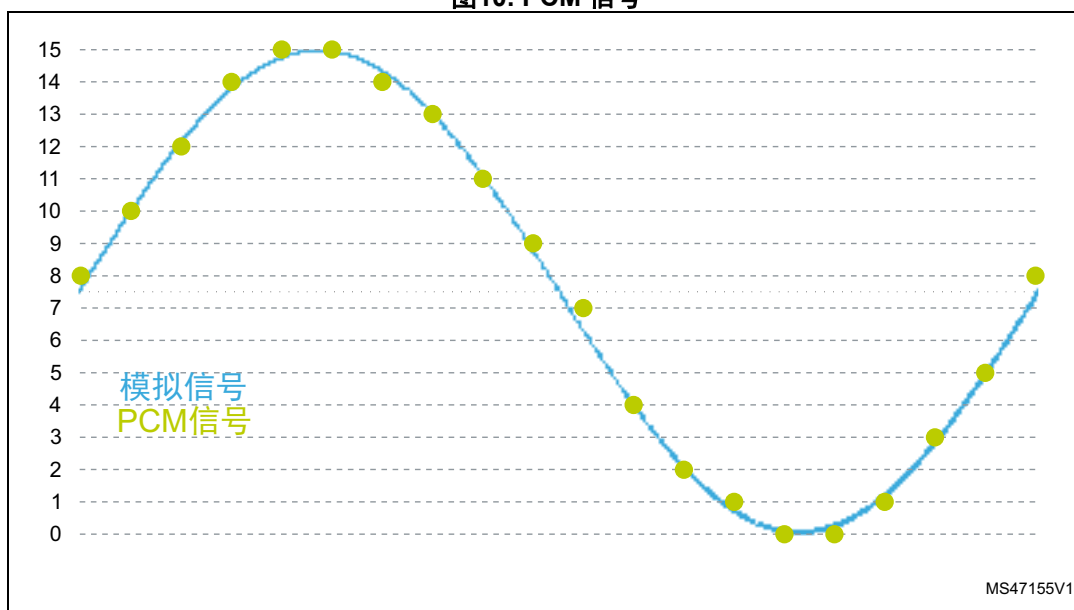
在PCM信号中，具体的幅度值被编码为脉冲。

PCM流有两个基本属性，它们决定了流相对于原始模拟信号的保真度：

- 采样率
- 位深

采样率是为了以数字方式表示信号，每秒采集的信号样本数。位深决定了每个样本中信息的位数。

图10. PCM 信号



1.4.3 PDM到PCM的转换

为了将PDM流转换为PCM样本，需要对PDM流进行滤波和抽取。

在抽取阶段，PDM信号的采样率降低至目标音频采样率（例如，16 kHz）。通过在每M个样本中选择一个，将采样率降低为1/M。因此，PDM数据频率（麦克风时钟的频率）是应用所需的目标音频采样频率的M倍，其中M为抽取因子。

PDM频率 = 音频采样频率 × 抽取因子

抽取因子通常处于48至128的范围之内。

抽取阶段之前有一个低通滤波器，用于避免混叠导致的失真。

1.5 声学参数

1.5.1 灵敏度

灵敏度是数字麦克风就给定声学参考信号输出的电信号的电平（单位为dBFS）。

通常使用1 kHz、1 Pa（或94 dB SPL）的音调作为参考信号给出麦克风的灵敏度。

1.5.2 信噪比（SNR）

SNR表示在麦克风输出处，参考信号（94 dB SPL@1kHz）与残余噪声量之比。

更高SNR可提供更高的语音清晰度和远端（免提听筒）可理解性。

1.5.3 声学过载点（AOP）

AOP是麦克风在失真可接受的情况下能够捕获的最大声学信号（一些规范允许声学过载点有最多10%的失真）。

1.5.4 电源抑制比（PSRR）

PSRR参数量化了麦克风对电源变化所产生的噪声的抑制能力。

1.6 数字麦克风的附加价值

表3. 数字麦克风的附加价值

特征	附加价值
抗RF噪声和抗电磁干扰（EMI）能力	– 集成工作量更小
无需进行模拟信号调节	– 简化应用设计 – 直接连接具有数字麦克风接口的编解码器 – 立体声模式只需要一条主数据线 – 在系统中有更多麦克风时显著节约PCB面积 – 为应用添加额外麦克风的灵活性
稳健的数字传输	– 在应用系统上轻松布置MEMS – 标准数字调节 – 允许音频增强整合，用于立体捕获、噪声消除和波束成形

1.7 可用的ST数字麦克风

表 4所示为可用的ST数字麦克风。

表4. ST数字麦克风

产品编号	顶部/底部端口	电源电压 (V)	信噪比 (dB)	灵敏度 (dBFS)	声学过载点 (dBSPL)
MP34DB02	底部	1.64至3.6	62.6	-26	120
MP34DT01-M	顶部	1.64至3.6	61	-26	120
MP34DT02	顶部	1.64至3.6	60	-26	120
MP34DT04	顶部	1.6至3.6	64	-26	120
MP34DT04-C1	顶部	1.6至3.6	64	-26	120
MP34DT05	顶部	1.6至3.6	64	-26	122.5
MP45DT02-M	顶部	1.64至3.6	61	-26	120

2 将PDM数字麦克风连接到STM32 MCU和MPU

本章描述如何将数字MEMS麦克风连接至单声道和立体声配置中的STM32 MCU和MPU中嵌入的SPI/I2S、SAI和DFSDM外设。

2.1 串行外设接口（SPI）/集成电路内置音频总线（I2S）

STM32 MCU和MPU提供名为SPI的串行外设接口模块。其中的一些SPI模块还能使用集成电路内置音频总线协议（I2S）。此外，STM32器件提供两个版本的SPI模块，在本文档中，最早的版本称为SPI-V1，最新的版本称为SPI-V2。当我们提到SPI模块时不特指SPI-V1或SPI-V2。

STM32H7系列目前提供的是SPI-V2。

通过使用SPI或I2S协议，可以将一个或两个数字麦克风连接至SPI模块。

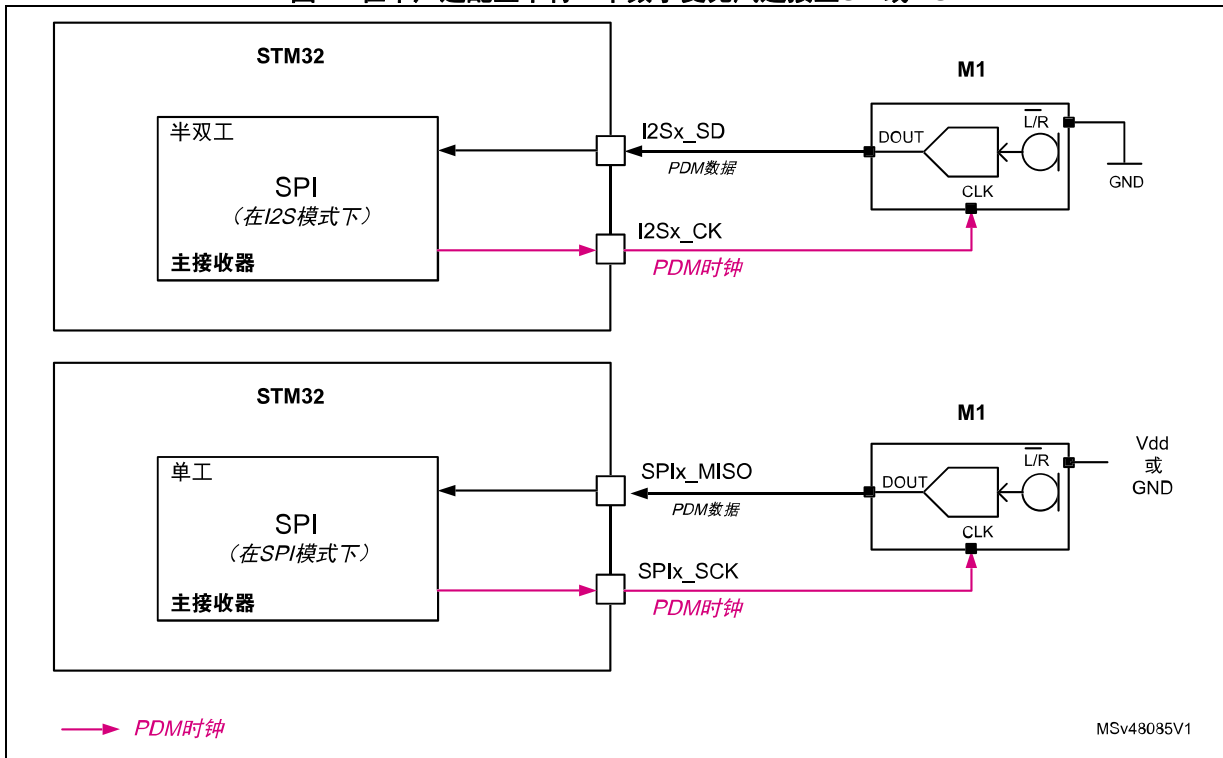
- SPI协议提供简单的通信接口，允许微控制器与外部器件通信。
- I2S协议广泛用于从微控制器/DSP（数字信号处理器）传输音频数据到音频编解码器，以播放音乐或从麦克风捕获声音。

2.1.1 单声道配置

单个数字麦克风连接到SPI模块。SPI模块可以配置为SPI或I2S模式。

两种情况下，SPI模块均配置为主接收器模式。在该模式下，外设为数字麦克风提供时钟。通过串行数据引脚获取音频采样。

图11. 在单声道配置中将一个数字麦克风连接至SPI或I2S



如果使用SPI协议，麦克风的L/R通道选择（LR）引脚可以连接至Vdd或GND。SPI时钟极性应与L/R输入的配置相一致。

- 如果L/R = GND，则SPI应使用SPIx_SCK的上升沿对传入数据执行采样，
 - 如果L/R = Vdd，则SPI应使用SPIx_SCK的下降沿对传入数据执行采样，
- 如果使用I2S协议，建议将麦克风的L/R通道选择（LR）引脚设置为GND。默认情况下，I2S协议使用I2Sx_CK的上升沿对传入数据执行采样。请注意，SPI-V2模块还提供为I2S协议配置采样边沿的能力。

数据格式

SPI模块在I2S或SPI模式下采集的样本可通过DMA或发送中断信号保存到存储器中。接收数据寄存器（SPIx_DR）提供从麦克风收到的连续位数据，如下面16位格式的示例所示：

位：	15	14	13	12	11	...	0
内容：	M1_b _N	M1_b _{N+1}	M1_b _{N+2}	M1_b _{N+3}	M1_b _{N+4}	...	M1_b _{N+15}

M1_b_{xx}表示数字麦克风1的数据位，M1_b_N是较早的数据位。

注：如果将接口设定为LSB优先而不是MSB优先，则可以颠倒已接收采样的位序。外设通常支持各种不同的数据大小，在此不予赘叙。



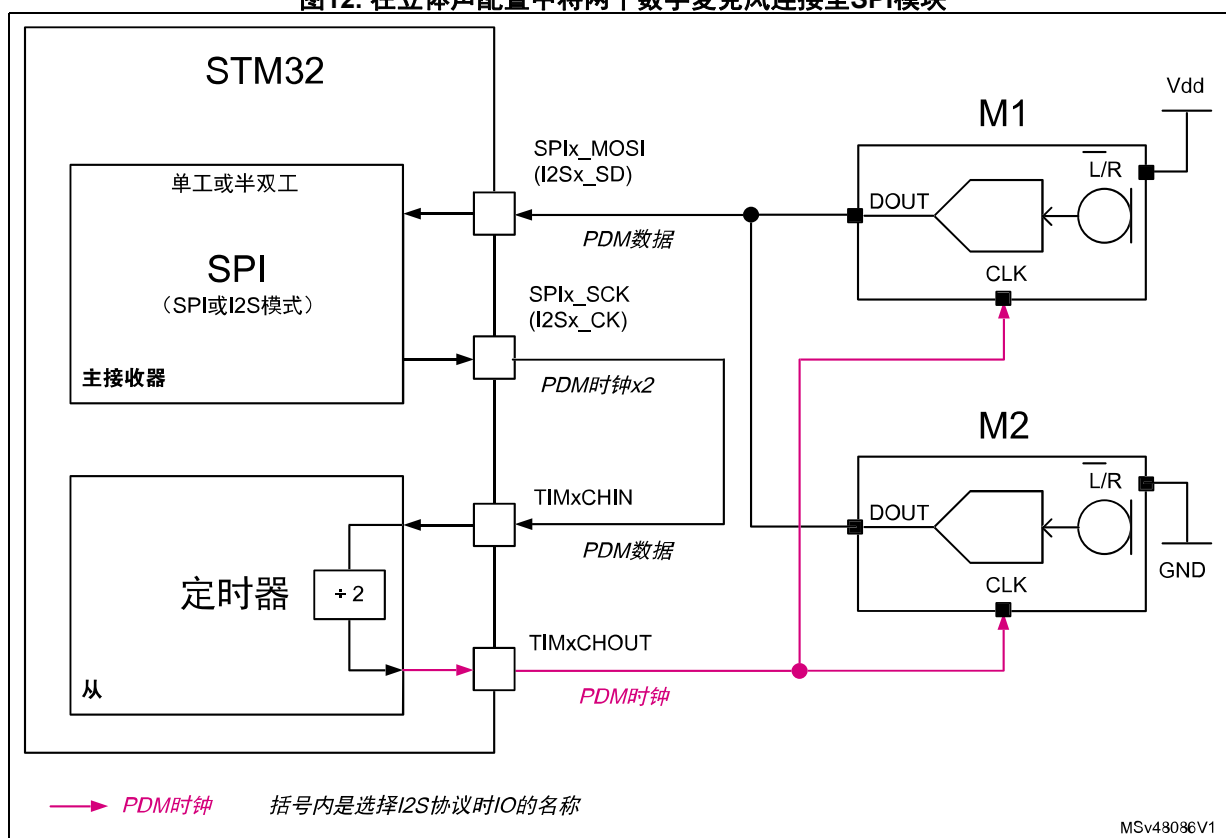
2.1.2 立体声配置

两个数字麦克风可使用定时器连接到SPI模块。SPI模块可以配置为SPI或I2S模式。

两种情况下，SPI模块均配置为主接收器模式。在这种配置下，SPI外设以两倍于麦克风频率的频率工作，以在其时钟的下降沿读取两个麦克风提供的数据。这允许两个麦克风共享一条数据线。

SPI模块为嵌入式定时器提供时钟，该定时器将串行接口时钟（SPIx_SCK或I2Sx_CK）除以2。分频后的时钟被提供给数字麦克风。通过I2S外设从数字麦克风数据输出引脚获取音频采样。

图12. 在立体声配置中将两个数字麦克风连接至SPI模块



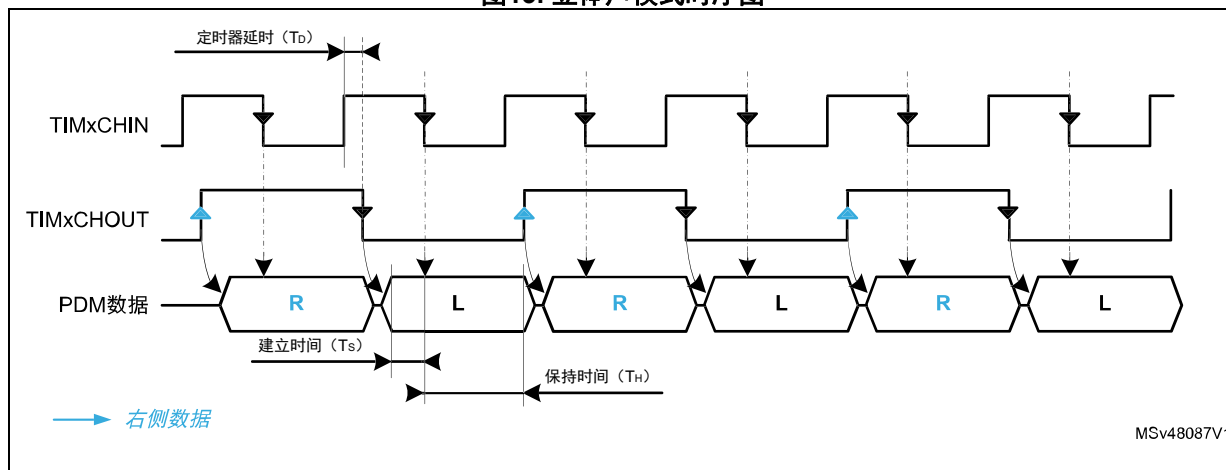
使用定时器作为时钟生成器

当定时器用于生成数字麦克风的时钟时，必须考虑以下两点：

- 应用必须确保在定时器执行时钟分频引起延迟的情况下，仍能保证麦克风所提供样本的建立时间 (T_S) 中的余量。为此，定时器应使用尽可能快的时钟。输入 (TIMxCHIN) 和输出时钟 (TIMxCHOUT) 之间的定时器所引起的最大延迟 (T_D) 为定时器参考时钟的5个时钟周期。定时器通常使用其APB时钟或其APB时钟的倍数作为参考。请参见图 13。

- 应用必须确保向TIMxCHIN输入提供时钟的外设及用于分频的定时器使用的是相同参考时钟。如不遵守此规则，数字麦克风会不时收到具有更长或更短周期的时钟。这类抖动会降低麦克风模数转换的质量。

图13. 立体声模式时序图



数据格式

SPI模块在I2S或SPI模式下采集的样本可通过DMA或发送中断信号保存到存储器中。

在这种配置下，将按位交叉存取从麦克风读取的数据。数据交叉存储到SPIx_DR寄存器中，如下面16位格式的示例所示：

位：	15	14	13	12	11	...	0
内容：	M1_b _N	M2_b _N	M1_b _{N+1}	M2_b _{N+1}	M1_b _{N+2}	...	M2_b _{N+7}

M1_b_{xx}表示数字麦克风1的数据位，M1_b_N是较早的数据位。M2_b_{xx}表示数字麦克风2的数据位，M2_b_N是较早的数据位。

SPI-V1和SPI-V2提供多种数据格式，例如在将接口设定为LSB优先而不是MSB优先时，可以颠倒已接收采样的位序，但必须注意的是，来自麦克风1（M1）的数据位和来自麦克风2的数据位（M2）总是交叉存储。

需使用软件去交叉模块将两个麦克风数据流的信号分开，以便进行进一步处理，例如PDM至PCM的转换。

2.2 串行音频接口 (SAI)

STM32 MCU和MPU中集成的SAI提供的接口允许微控制器与外部音频设备（例如放大器、麦克风、扬声器或音频处理器）通信。SAI包含两个可同步或异步操作的独立子模块。每个子模块都具有自己的音频时钟生成器。

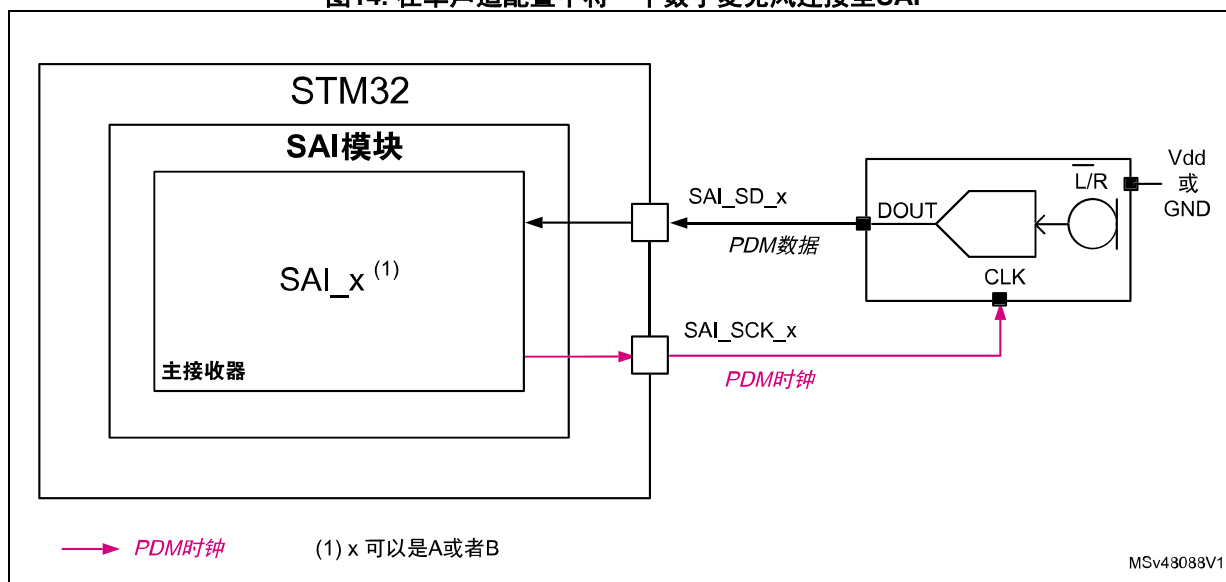
一些SAI还提供专用PDM接口，可支持最多8个数字麦克风。

2.2.1 使用单个子模块

单声道配置

数字麦克风连接到单声道配置中串行音频接口（SAI）的子模块之一。SAI子模块配置为主接收器模式。在这种配置下，SAI子模块为数字麦克风提供时钟。音频采样由SAI子模块通过串行数据（SD）引脚从数字麦克风数据输出（DOUT）引脚采集。

图14. 在单声道配置中将一个数字麦克风连接至SAI



麦克风的L/R通道选择（LR）引脚可以连接至Vdd或GND。传入时钟信号的上升或下降沿的麦克风输出数据取决于选择的通道。应相应地配置SAI模块的采样边沿。

注： 另一个SAI子模块完全独立，可用于另一目的：例如，可以连接至外部音频编解码器。

数据格式

SAI子模块采集的样本可通过DMA或发送中断信号保存到存储器中。

接收数据寄存器（SAI_ADR、SAI_BDR）提供从麦克风收到的连续数据位，如下面的示例所示：

位：	k	k-1	k-2	k-3	k-4	...	0
内容：	M1_b _N	M1_b _{N+1}	M1_b _{N+2}	M1_b _{N+3}	M1_b _{N+4}	...	M1_b _{N+k}

M1_b_{xx}表示数字麦克风1的数据位，M1_b_N是在SAI_ADR/BDR寄存器中的较早的数据位。

例如，如果将DS[2:0]设为“010”（8位），则“k”等于7。

存储到SAI_ADR/BDR寄存器中的数据大小取决于设定的数据大小（DS[2:0]），此外，SLOTSZ[1:0]必须强制为0。

请注意，如果将接口设定为LSB优先而不是MSB优先，则可以颠倒已接收采样的位序。

注： SAI_ADR/BDR寄存器中的数据总是右对齐。

立体声配置

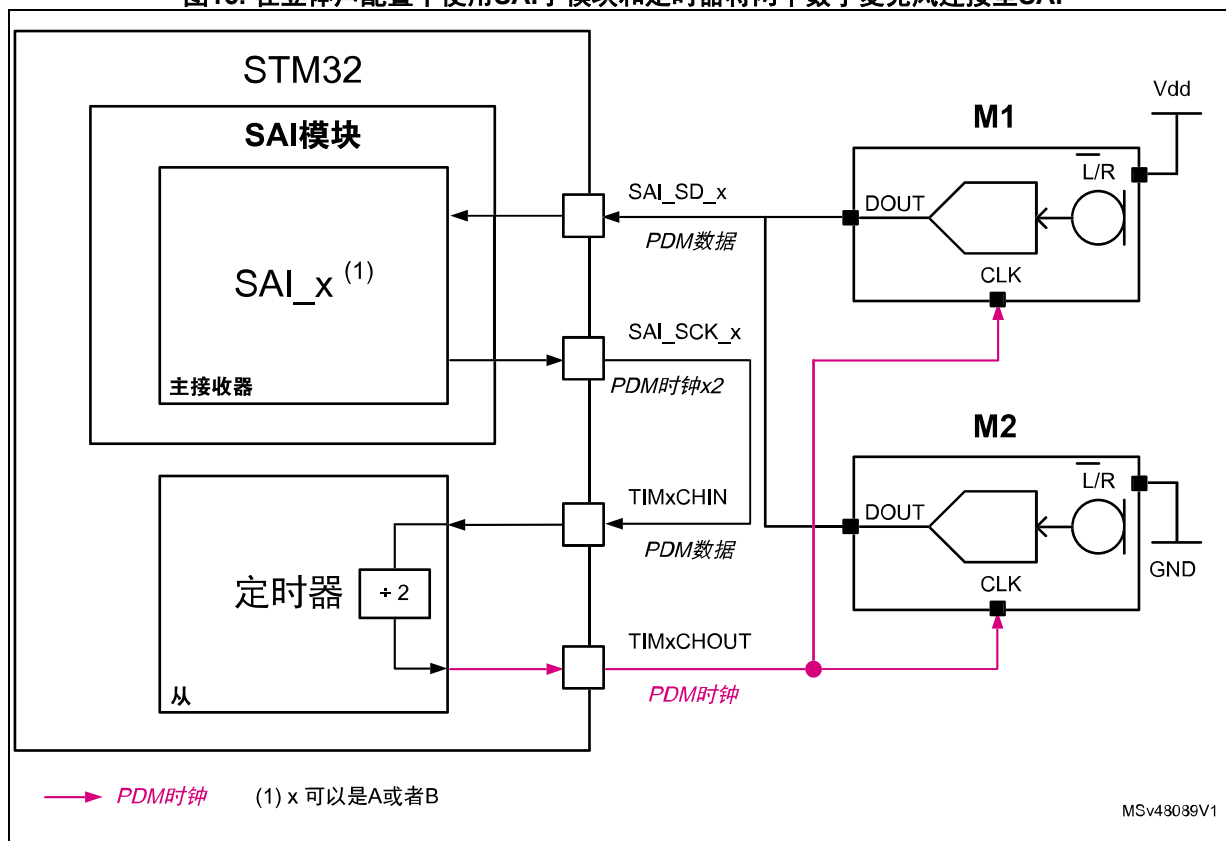
使用内部定时器可以将两个数字麦克风连接到一个SAI子模块。SAI子模块配置为主接收器模式。在这种配置下，SAI子模块以两倍于麦克风频率的频率工作，以在其时钟的相同边沿读取两个麦克风提供的数据。这允许两个麦克风共享一条数据线（请参考图 15）。

SPI子模块为嵌入式定时器提供时钟（SAI_SCK_x），该定时器将时钟除以2。分频后的时钟被提供给数字麦克风。

注： 另一个SAI子模块完全独立，可用于另一目的：例如，可以连接至外部音频编解码器。

图 13所示的时序图对本例同样有效。另请参考使用定时器作为时钟生成器获取关于定时器使用的建议。

图15. 在立体声配置中使用SAI子模块和定时器将两个数字麦克风连接至SAI



数据格式

SPI子模块采集的样本可通过DMA或发送中断信号保存到存储器中。

在这种配置下，将按位交叉存取从麦克风读取的数据。数据交叉存储到SAI_ADR/BD寄存器中，如下面的示例所示：

位：	k	k-1	k-2	k-3	...	1	0
内容：	M1_b _N	M2_b _N	M1_b _{N+1}	M2_b _{N+1}	...	M1_b _{N+(K-1)/2}	M2_b _{N+(K-1)/2}

M1_b_{xx}表示数字麦克风1的数据位，M1_b_N是较早的数据位。M2_b_{xx}表示数字麦克风2的数据位，M2_b_N是较早的数据位。

“k”是数据位在SAI_ADR/BDR寄存器中的位置，例如：如果将DS[2:0]设置为“100”（16位），则“k”等于15。

存储到SAI_ADR/BDR寄存器中的数据大小取决于设定的数据大小（DS[2:0]），此外，SLOTSZ[1:0]必须强制为0。

注： 如果将接口设定为LSB优先而不是MSB优先，则可以颠倒已接收采样的位序，但必须注意的是，来自麦克风1（M1）的数据位和来自麦克风2的数据位（M2）总是交叉存储。

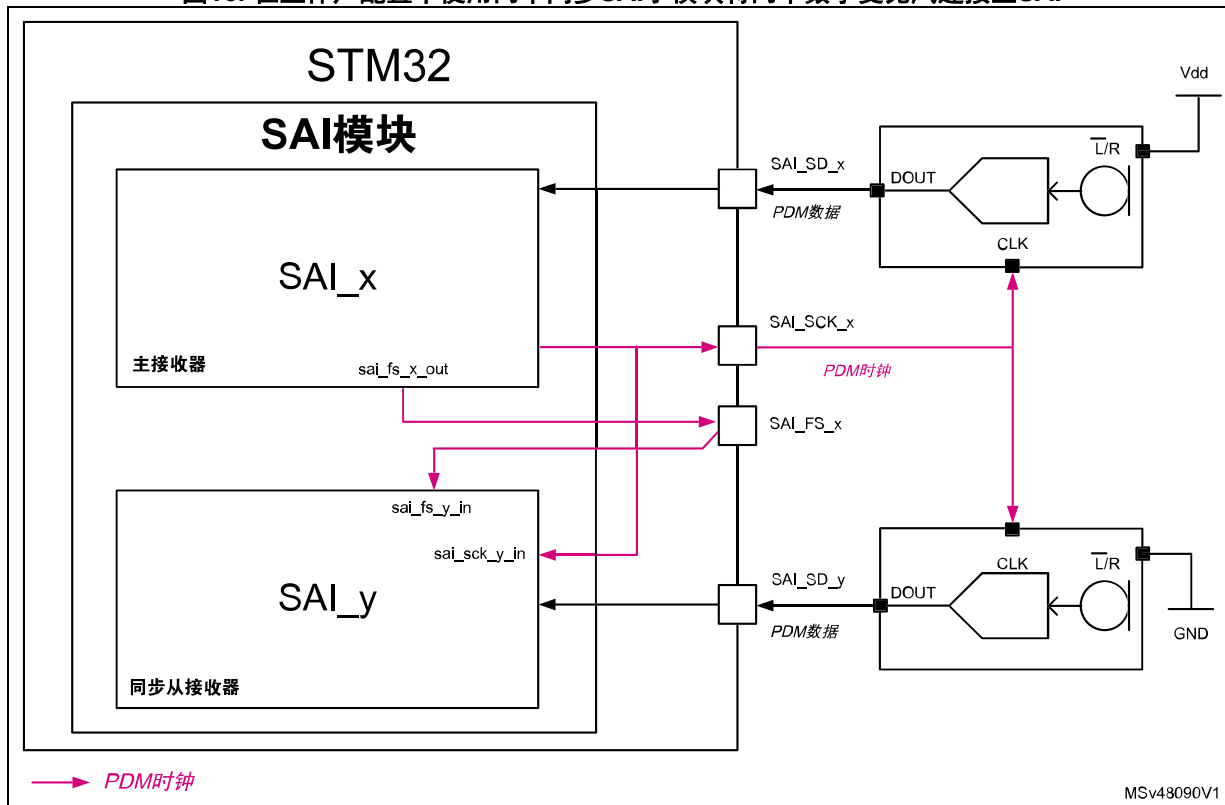
SAI_ADR/BDR寄存器中的数据总是右对齐。

需使用软件去交叉模块将两个麦克风数据流的信号分开，以便进行进一步处理，例如PDM至PCM的转换。

2.2.2 使用两个同步SAI子模块

通过同步两个SAI子模块，可以将两个数字麦克风连接到SAI外设，无需使用嵌入式定时器。每个麦克风连接到一个SAI子模块。一个SAI子模块配置为主器件接收模式，另一个子模块配置为同步从器件接收模式。在这种配置下，SAI子模块配置为主模式，将时钟提供给数字麦克风和其他SAI子模块。两个SAI子模块从麦克风同步读取数据。

图16. 在立体声配置中使用两个同步SAI子模块将两个数字麦克风连接至SAI



由于两个麦克风不共享一条主数据线，麦克风的L/R通道选择（LR）引脚可以连接至Vdd或GND。传入时钟信号的上升或下降沿的麦克风输出数据取决于选择的通道。应相应地配置每个SAI子模块的时钟极性。

数据格式

每个SAI子模块采集的样本均可通过DMA或发送中断信号保存到存储器中。这种情况下，需要最多2个通道。

每个SAI子模块的数据格式与SAI单声道配置的数据格式相似。

2.2.3 使用PDM接口

PDM接口为数字麦克风提供支持，最多可并联4对数字麦克风。PDM接口还提供延迟线，以便在每个传入比特流中执行微延迟，从而简化波束成形应用。每个延迟单元的深度为8个比特流采样。

PDM功能旨在与配置为时分复用（TDM）主模式的SAI_A子模块配合使用。它不能与SAI_B子模块搭配使用。

为减少存储器占用量，用户可以选择应用需要的麦克风数量。用户可以选择2、4、6或8个麦克风。

例如，如果应用需要使用3个麦克风，则用户必须选择4个麦克风。在这种情况下，通过SAI_D1和SAI_D2采集PDM数据。SAI_D1从第一对麦克风接收数据，而SAI_D2从第三个麦克风接收数据。

表 5所示为推荐IO线与要连接的数字麦克风数量。

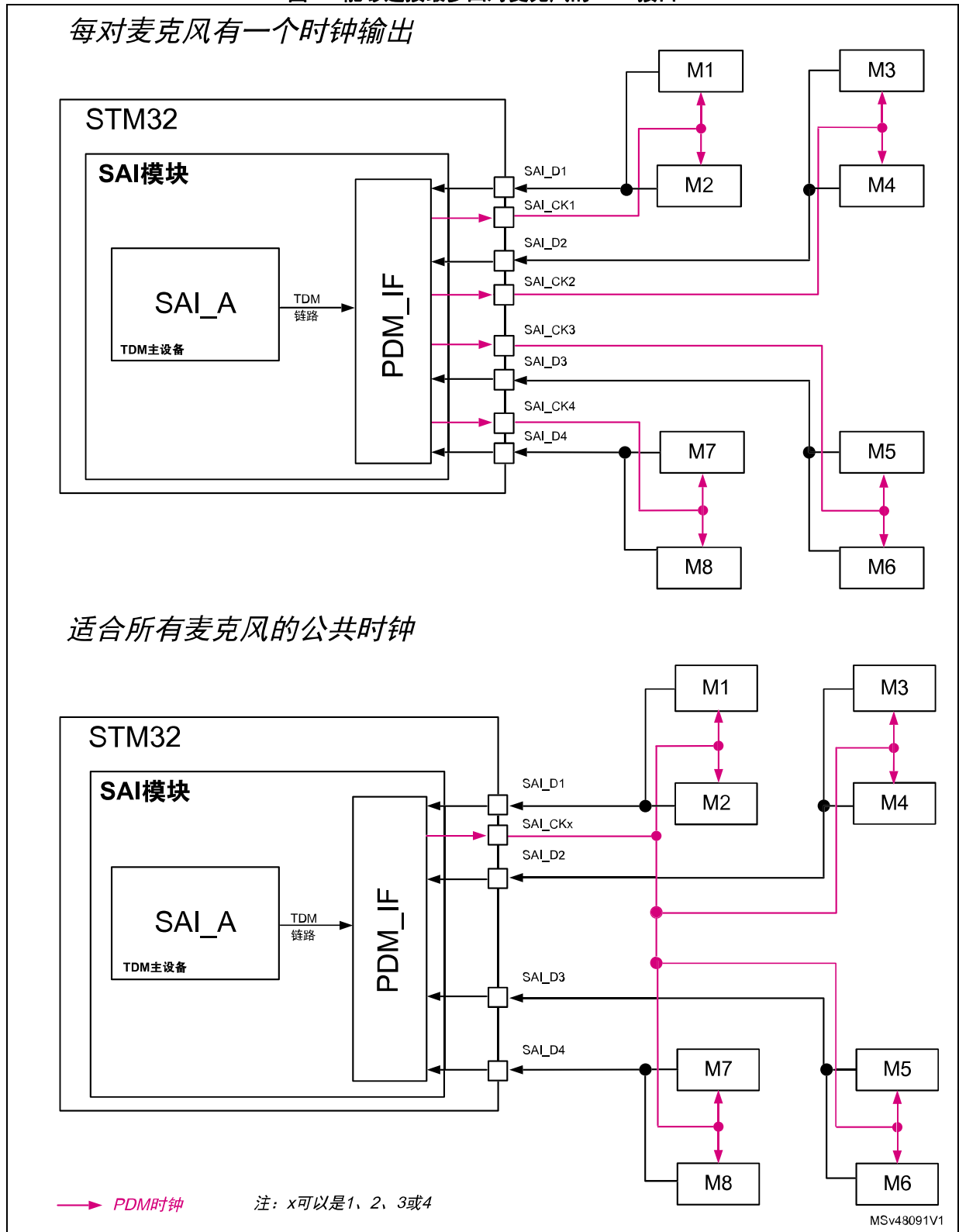
表5. 推荐IO线与要连接的数字麦克风数量

麦克风数量	推荐的数据IO线	推荐的时钟IO线
1或 2	SAI_D1	SAI_CK1、2、3或 4
3或 4	SAI_D1和SAI_D2	
5或 6	SAI_D1、SAI_D2和SAI_D3	
7或 8	SAI_D1、SAI_D2、SAI_D3和SAI_D4	

麦克风的时钟既可以从同一SAI_CKx（x=0...3）提供，也可以单独从不同SAI_CK提供，这给予了用户根据具体应用使能或禁用特定麦克风的音频采集的灵活性。

SAI的工作频率是选择的麦克风数量（2、4、6或8）乘以麦克风频率，这样才能从应用中的所有麦克风读取数据。

图17. 能够连接最多四对麦克风的PDM接口



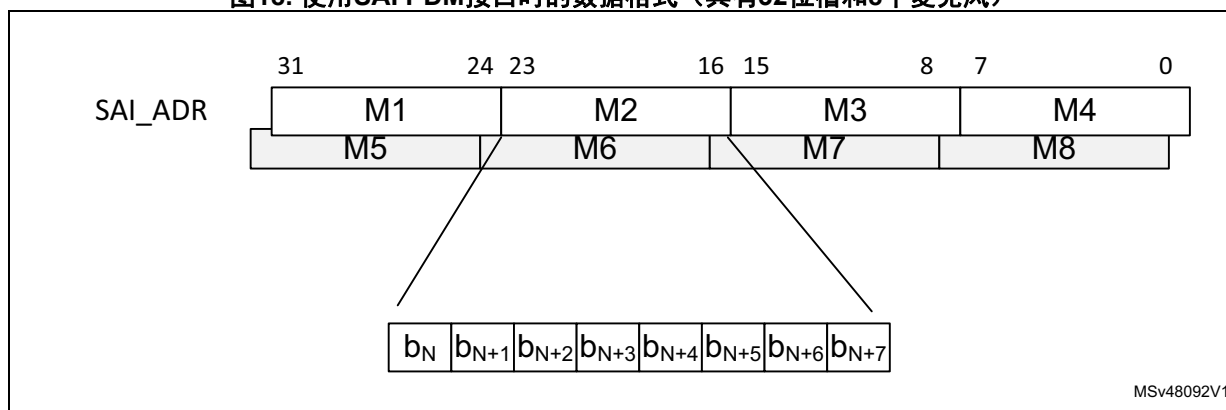
SAI的PDM接口提供对数字麦克风的最优连接，尽可能地节约IO。此外，PDM接口还可以按字节分离每个麦克风的数据，避免去交叉操作。

数据格式

SAI_A子模块采集的样本可通过单个DMA通道或发送中断信号保存到存储器中。

接收数据寄存器（SAI_ADR）为每个麦克风提供8个连续位，如下面的示例所示：

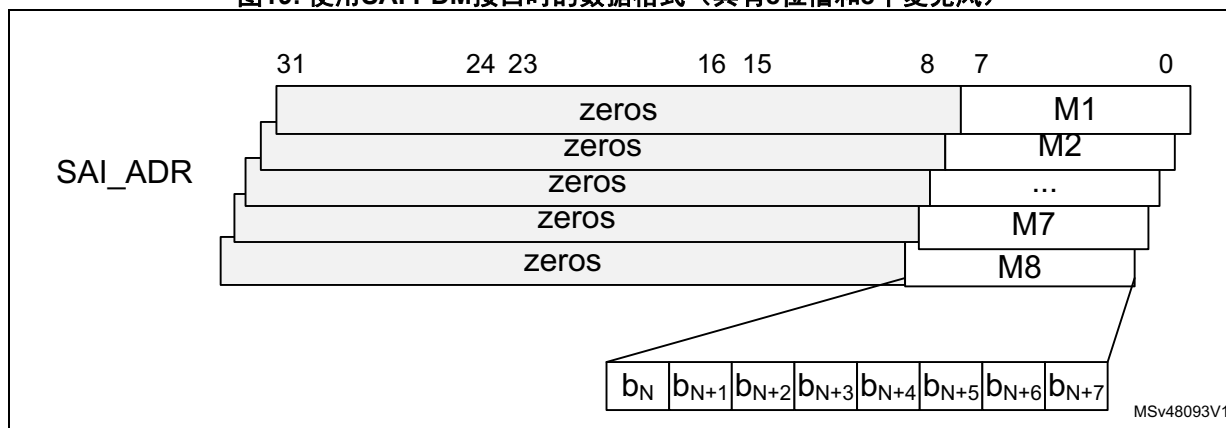
图18. 使用SAI PDM接口时的数据格式（具有32位槽和8个麦克风）



SAI_ADR中保存的数据量取决于多个SAI参数的设定，请参考SAI用户说明，以获取关于使用PDM接口时的数据格式的更多信息。

请注意，如果插槽尺寸设置为8位，则SAI_ADR只包含一个麦克风的一个数据字节。必须8次读取SAI_ADR才能获取8个麦克风的一个字节。

图19. 使用SAI PDM接口时的数据格式（具有8位槽和8个麦克风）



2.3 数字滤波器，用于ΣΔ调制器（DFSDM）

DFSDM是STM32MCU和MPU内部的数字外设。它的作用相当于速度/分辨率可调节并具有外部模拟前端的标准ADC。

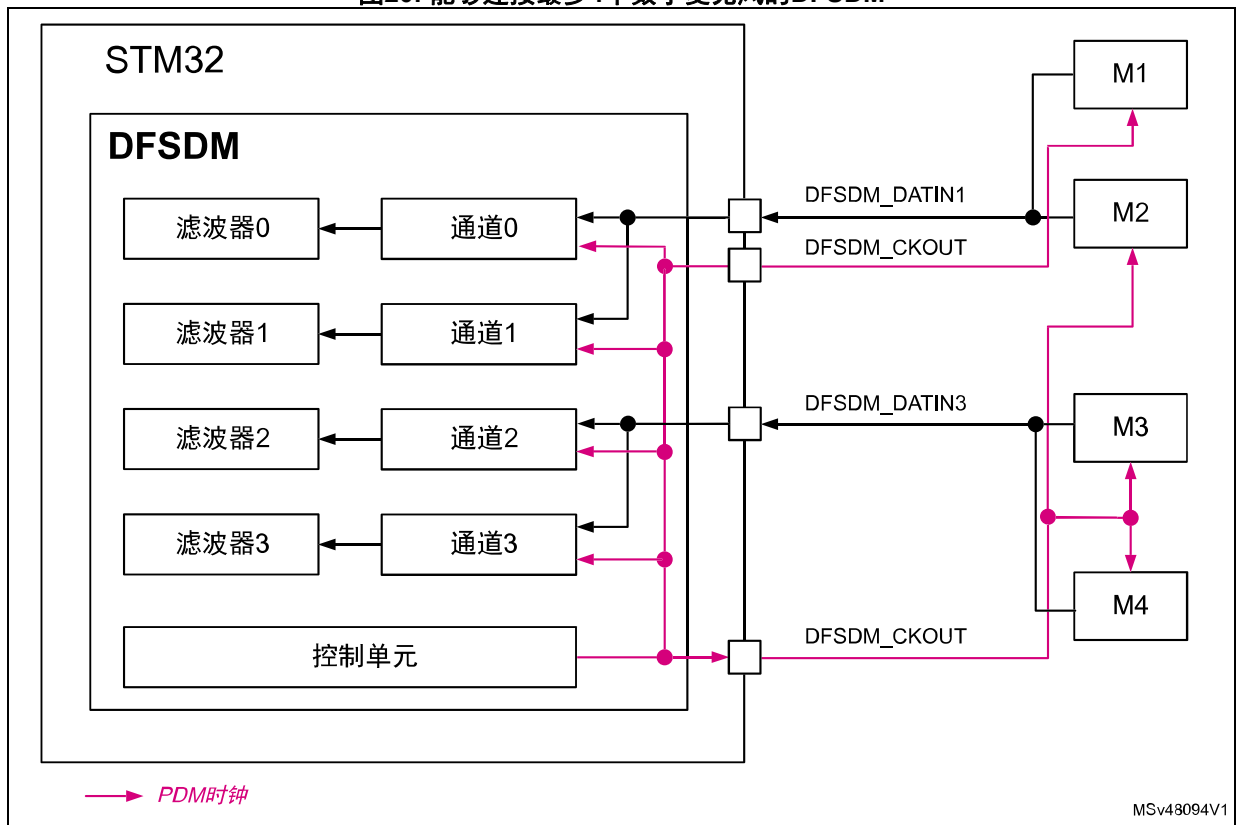
提供PDM输出数据格式的数字MEMS麦克风可直接连接到DFSDM。DFSDM提供经过滤波和抽取的采样。每个滤波器都具有自己的DMA通道，因此每个麦克风的采样都是分开的。这为应用免去了大量的滤波和去交叉操作。最后，一些DFSDM模块还提供延迟线，以便在每个传入比特流中执行微延迟，从而简化波束成形应用。每条延迟线的深度至少等于抽取比率。

可以将多个数字麦克风与一个DFSDM相连，这取决于集成的滤波器数量和接口数量。

DFSDM具有时钟输出信号（DFSDM_CKOUT），用于驱动数字麦克风。时钟输出具有可调节的分频因子。DFSDM_CKOUT可以是不同IO的输出，这给予了用户根据具体应用使能或禁用特定麦克风的音频采集的灵活性。图 20：能够连接最多4个数字麦克风的DFSDM所示配置通常用在低功耗应用中：数字麦克风M1可以工作，而其他麦克风则在不提供时钟时切换至低功耗模式。为了节约更多能源，还可以断开M3和M4的电源。

当需要激活所有麦克风时，通过两个不同的PAD向所有麦克风提供相同时钟。

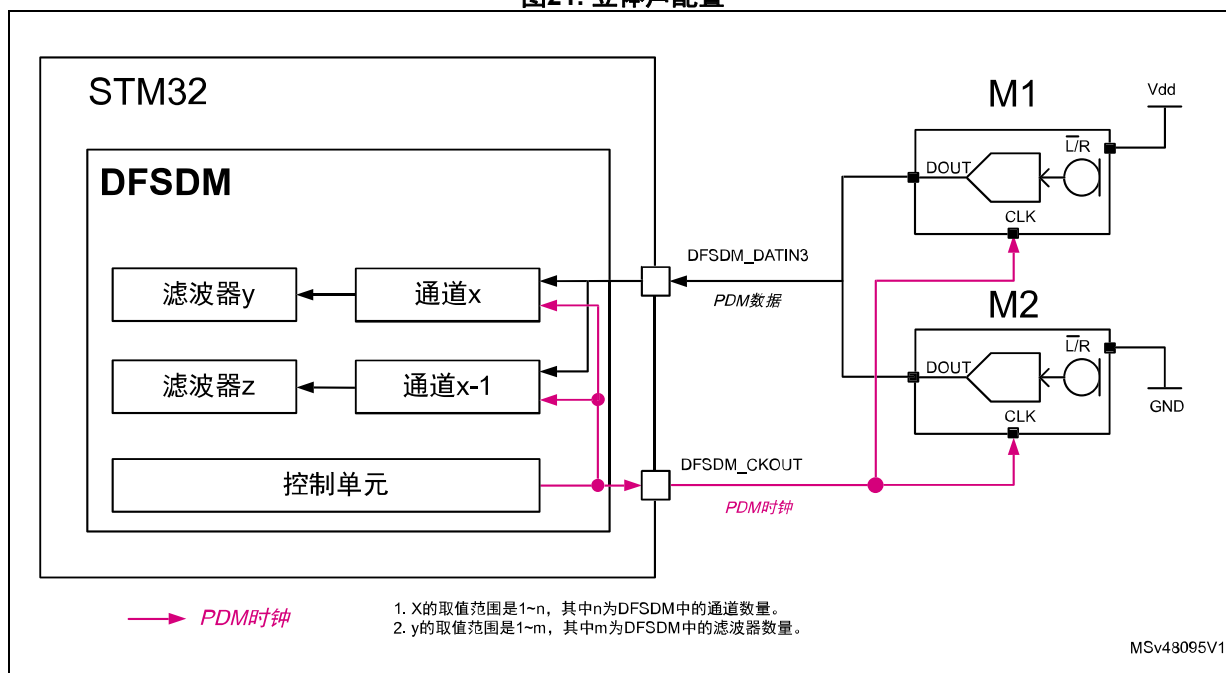
图20. 能够连接最多4个数字麦克风的DFSDM



2.3.1 立体声配置

在立体声配置中，有两个数字麦克风被连接至DFSDM。DFSDM必须使能两个由内部时钟驱动连续通道。DFSDM外设提供外部时钟（DFSDM_CKOUT），用于驱动数字麦克风。在这种配置下，设定DFSDM以使通道x和x-1从DFSDM_DATINx引脚接收数据。每个通道在不同的时钟边沿读取数据，这允许两个麦克风共享一条数据线。然后，每个通道采集的数据重定向至不同的DFSDM滤波器进行处理。

图21. 立体声配置



数据格式

每个数字滤波器采集的样本可通过专用DMA通道或发送中断信号保存到存储器中。无交叉，每个滤波器提供一个麦克风的转换后的采样值。需要的DMA通道数量等于激活的数字滤波器数量。

2.4 时钟方面的考虑

2.4.1 数字麦克风时钟

提供给数字麦克风的时钟有以下几个作用：

- 当缺少时钟或时钟频率过低时（请参考产品数据手册），数字麦克风将切换至低功耗模式。
- 当时钟频率过低（通常介于400和800 kHz之间）时，麦克风以低功耗模式工作。这意味着其性能的少量下降，目的是尽可能多地降低功耗。并非所有麦克风都具有这一特性。时钟应尽可能干净，低抖动。

- 最后，当时钟频率较高（约1至4.8 MHz）时，麦克风以标称模式工作。时钟应尽可能干净，低抖动。
请注意，可接受的频率范围取决于麦克风，请参考产品数据手册了解详情。

还需注意的是，当数字麦克风刚刚因其时钟激活或处于可接受的频率范围之内而退出低功耗模式时，要等待若干毫秒才能提供达到预期质量的采样。

表 6为一些应用示例和相关的麦克风时钟频率。

表6. 应用示例和相关的麦克风时钟频率

应用案例	提供给数字麦克风的时钟频率
声音捕获	400至800 kHz之间
语音应用	1至1.5 MHz之间
高品质音频应用	2.4至4.8 MHz之间

2.4.2 外设时钟

为了选择实现，应用还应考虑微控制器向SPI、SAI或DFSDM提供时钟的可能性。此外，还必须核实将时钟生成器嵌入SPI、SAI和DFSDM的能力。

外设音频模块通常有两个时钟输入：

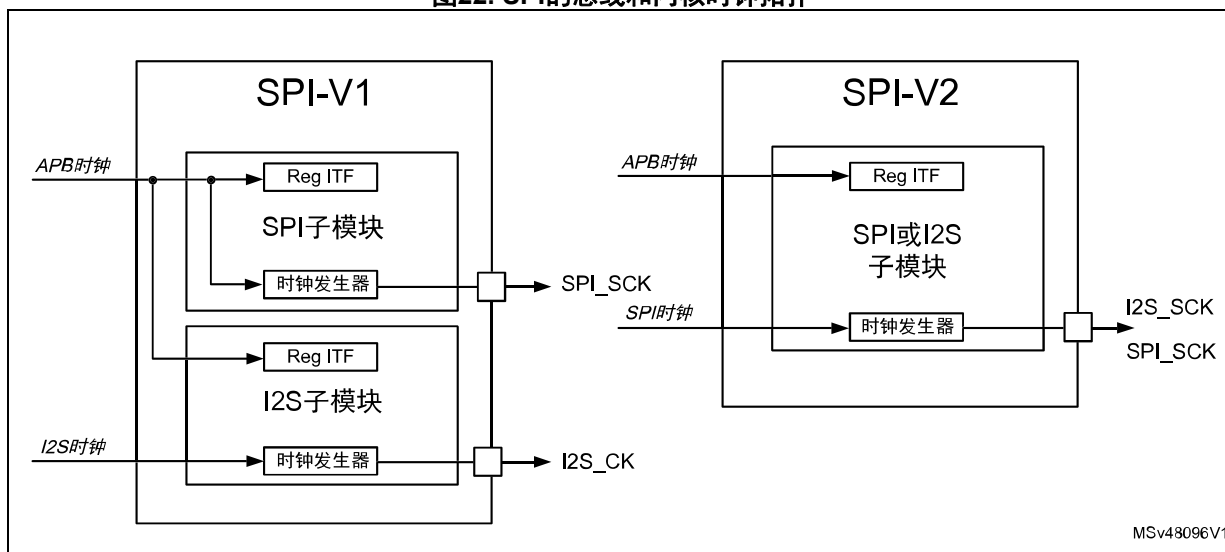
- 一个时钟用于寄存器接口控制（如APB时钟）
- 一个时钟用于串行接口的时序生成，即内核时钟。

根据选择的外设和协议，外设时钟生成器使用总线接口或专用参考时钟。例如，如果使用SPI协议，SPI-V1会使用APB时钟作为参考时钟，而使用I2S协议时，会使用I2S时钟。

控制APB时钟的频率通常不如使用专用时钟那么灵活。原因之一是APB时钟会影响所有连接到该APB总线的外设。但是，如果使用定时器支持立体声麦克风配置，使用APB（如定时器的相同时钟）可以解决[使用定时器作为时钟生成器](#)中列出的一个问题。

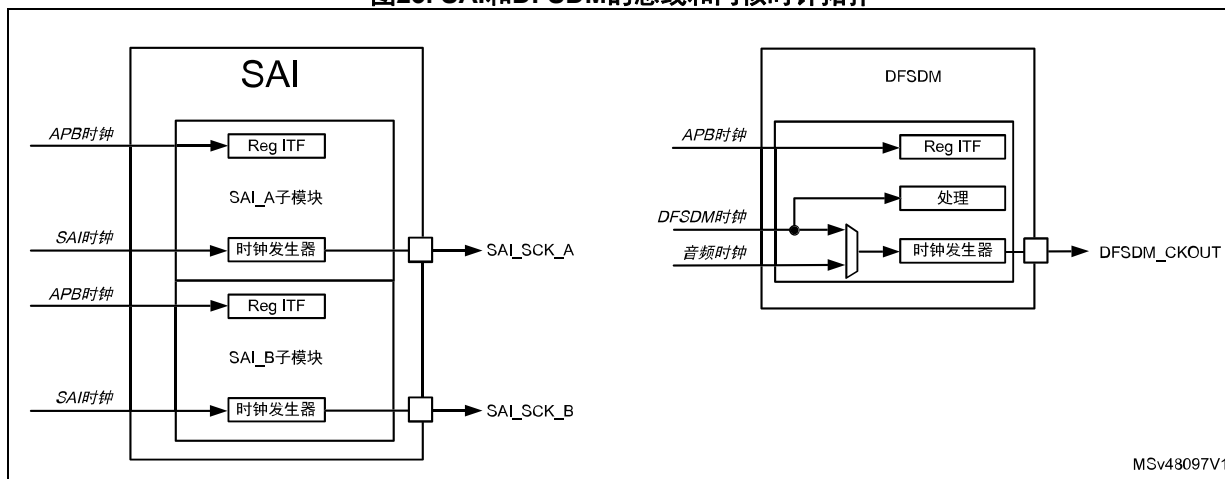
一些微控制器还能提供APB时钟的副本作为内核时钟。

图22. SPI的总线和内核时钟拓扑



其他外设（SPI-V2、SAI和DFSDM）总是提供专用内核时钟用于时钟生成。该选项更为灵活，并使所需时钟频率独立于总线接口频率。

图23. SAI和DFSDM的总线和内核时钟拓扑



注：DFSDM提供了选择独立外设时钟（音频时钟）或选择与APB时钟同步的DFSDM时钟的可能性。

2.5 GPIO数量的考虑

本节帮助用户根据GPIO数选择最适合的STM32外设（SPI2S、SAI和DFSDM），以便连接特定应用中的数字麦克风。

表 7、表 8和表 9分别显示了用于将一个、两个和四个数字麦克风连接到不同音频和串行接口的硬件。

- “GPIO数”一栏显示了将数字麦克风连接到音频接口所需的GPIO数。
- “定时器”一栏显示了音频接口是否需要定时器来将合适的时钟提供给麦克风。

表7. 用于连接一个数字麦克风的硬件

接口	GPIO数量	定时器
SPI	2	无
I2S	2	无
(SAI)	2	无
DFSDM	2	无

表8. 用于连接两个数字麦克风的硬件

接口	GPIO数量	定时器
SPI	4	有
I2S	4	有
(SAI)	4	有
SAI（两个同步子模块）	4	无
具有PDM接口的SAI	2	无
DFSDM	2	无

表9. 用于连接四个数字麦克风的硬件

接口	GPIO数量	定时器
SAI（两个同步子模块）	6	有
具有PDM接口的SAI	3	无
DFSDM	3	无

3 数字信号处理

本章将介绍两种将PDM数据转换为PCM数据的方式：第一种是软件解决方案，即PDM音频软件解码库，第二种是硬件解决方案，使用DFSDM外设滤波器。

3.1 PDM音频软件解码库

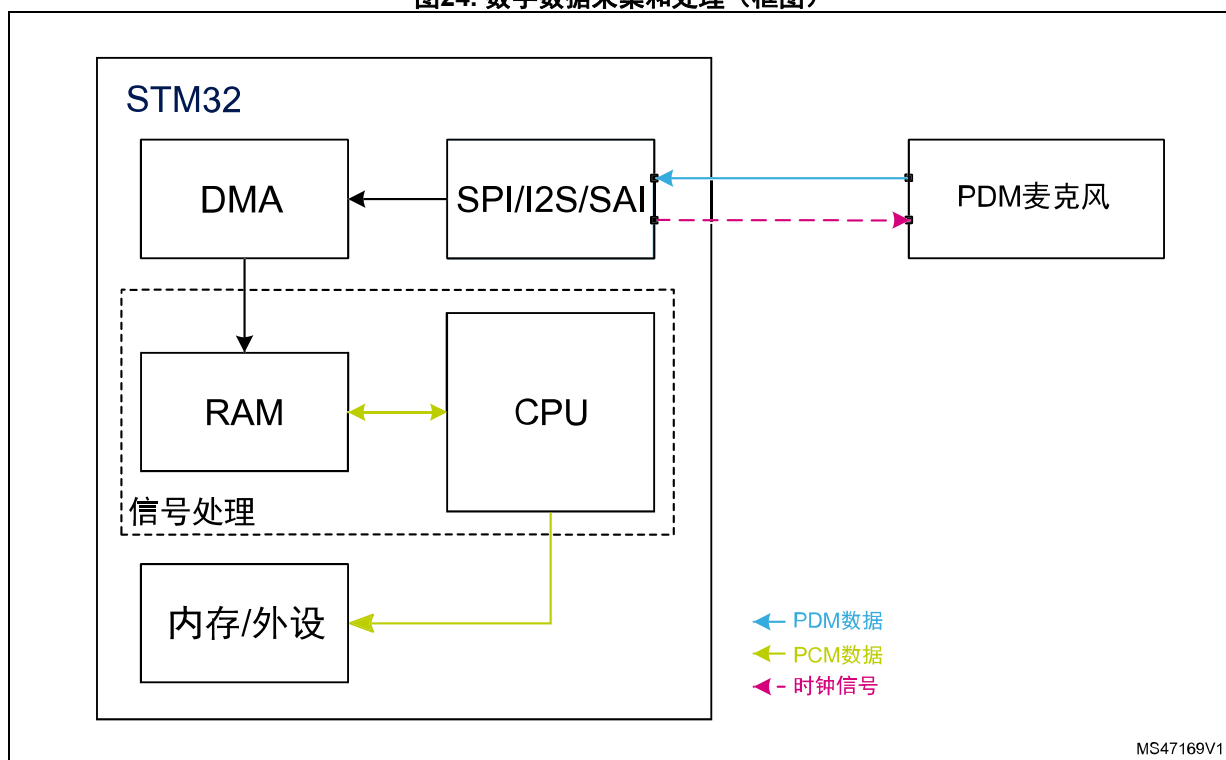
3.1.1 概述

当数字MEMS麦克风与STM32 MCU或MPU相连时，PDM音频软件解码库是用于PDM信号解码和音频信号重建的优化软件实现。此库为来自数字麦克风的1位PDM高频信号输出实现多个滤波器，并将输出转换为具有合适音频频率的16位PCM。

3.1.2 数字数据流

数字MEMS麦克风输出PDM信号，它是1位数字采样的高频（1至3.25 MHz）数据流。通过STM32器件中嵌入的串行接口采集PDM数据。此数据通过DMA（从而减少软件开销）传输至系统RAM缓冲区并等待处理。转换后，可根据应用实现来处理PCM原始数据（作为波形/压缩数据保存在大容量存储介质中，传输到外部音频编解码器DAC等）。

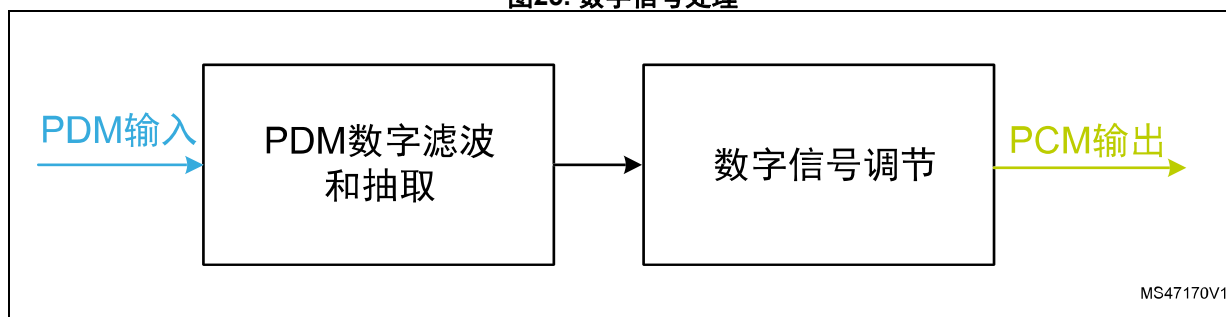
图24. 数字数据采集和处理（框图）



3.1.3 数字信号处理

PDM音频软件解码库提供两步数字信号处理：PDM数字滤波和抽取与数字信号调节。

图25. 数字信号处理



第一步，为了获取具有所需频率和分辨率的声音信号，需要对来自麦克风的PDM信号进行滤波和抽取。

第二步，使用低通滤波器和高通滤波器对先前滤波流水线产生的数字音频信号进行进一步处理，以便进行适当的信号调节。这两个滤波器都可以使用滤波器初始化功能进行使能/禁用和配置（截止频率）。

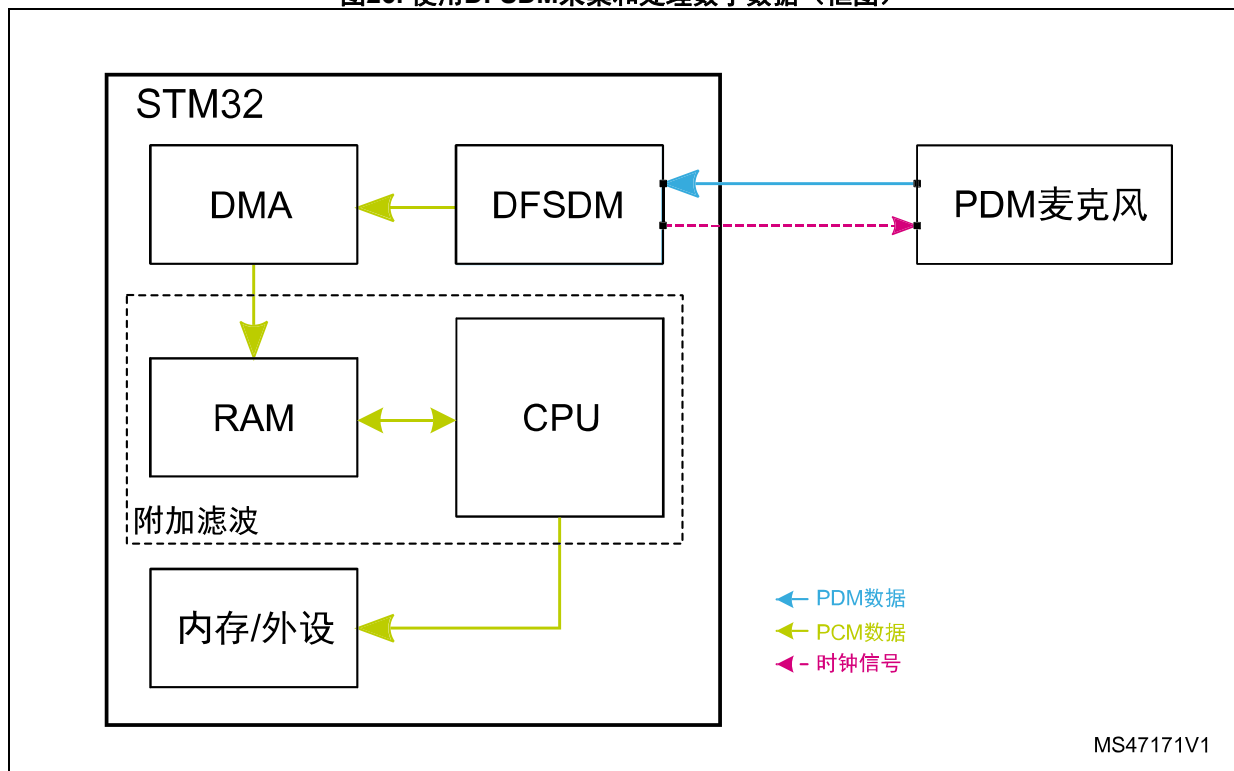
注： 在立体声配置中，如果两个麦克风共享一条主数据线且PDM数据按位交叉，则需要执行软件去交叉步骤以将两个麦克风的信号分开，然后继续进行PDM至PCM的转换。

3.2 数字信号处理用DFSDM滤波器

3.2.1 数字数据流：采集和处理

数字MEMS麦克风输出是一个PDM信号，它是1位数字采样的高频（1至3.25MHz）数据流。数据由DFSDM串行收发器采集，它提供与数字麦克风外部 $\Sigma\Delta$ 调制器的连接。数字滤波器执行无CPU滤波，对来自SD调制器的1位输入数据流进行平均，得到更高分辨率和更低采样率。此数据通过DMA（从而减少软件开销）传输至系统RAM缓冲区并等待进一步的滤波。此后，可根据应用实现来处理PCM原始数据（作为波形/压缩数据保存在大容量存储介质中，传输到外部音频编解码器DAC等）。

图26. 使用DFSDM采集和处理数字数据（框图）



4 基于STM32CubeMX的配置示例

本章指导用户执行必要的步骤，以创建基本的音频应用，即在单声道或立体声模式下从数字麦克风采集PDM数据，然后转换为PCM数据。

主要步骤包括选择合适的硬件配置和使用STM32CubeMX工具生成C初始化代码。第二步，将合适的用户代码添加到生成的项目。

最后，用户可以参考STM32Cube_FW包音频示例和“X-CUBE-MEMSMIC1”包，以完成每个示例所需的用户代码。

本章假设：

- 用户想要获得16或48 kHz的PCM数据流
- 数字麦克风接收到频率比PCM数据流频率高64倍的时钟（64倍过采样）。

4.1 示例1：使用I2S、SPI或者单个SAI模块，连接单声道或者立体声模式下的数字麦克风

本例以NUCLEO-F413ZH板为基础并使用外部数字麦克风，这些麦克风连接到I2S、SPI或SAI。

对于16 KHz采样率，接口生成的位时钟频率必须是1.024 MHz（单声道模式）或2.048MHz（立体声模式）。

对于48 KHz采样率，接口生成的位时钟频率必须是3.072 MHz（单声道模式）或6.144MHz（立体声模式）。

4.1.1 使用STM32CubeMX配置硬件

I2S

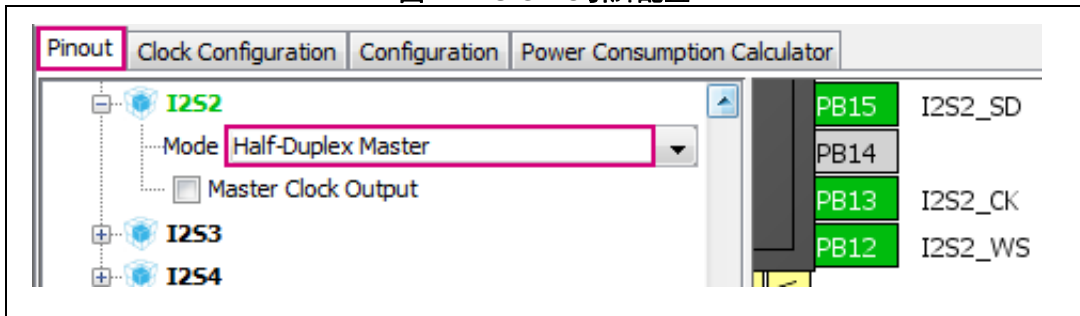
GPIO和引脚配置

从“引脚排列”选项卡列出的硬件中选择I2S2外设，并将其配置为半双工主模式。

[图 27](#)显示了如何以半双工主模式使能I2S2。

在正确配置I2S外设GPIO后，使能的引脚（I2S2_SD、I2S2_CK和I2S2_WS）会以绿色突出显示。

图27. I2S GPIO引脚配置



注： 在本例中，I2S2_WS引脚未使用。在生成项目后，通过对MSP初始化文件代码（stm32f4xx_hal_msp.c）稍作修改，可以释放该引脚并用作GPIO。

时钟配置：

本节描述对于16 kHz和48 kHz数据流，在单声道和立体声模式下I2S的不同时钟配置。使用HSE = 8 MHz作为时钟源。

在立体声模式下，定时器和I2S必须使用来自相同参考的时钟，因此，选择PLL2作为I2S的时钟源。

注： 精度是预期音频频率与实际频率之间的误差。

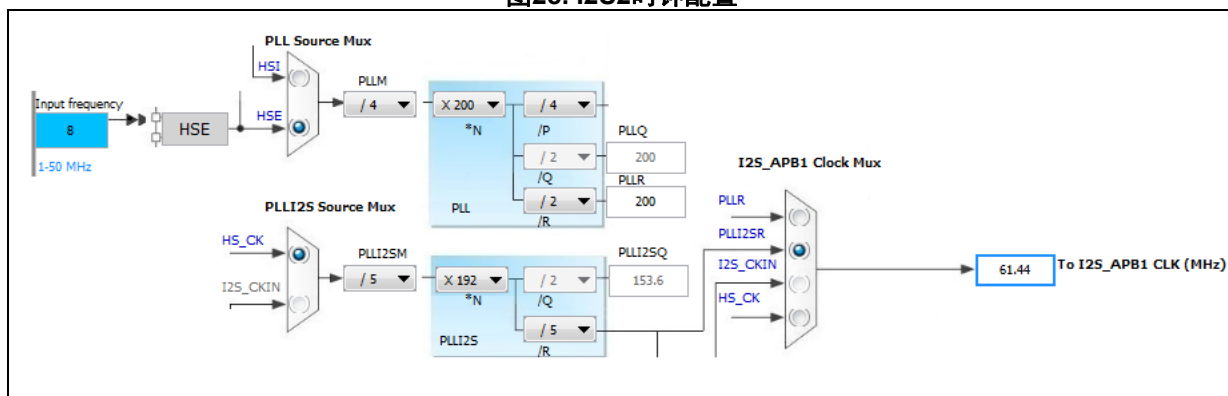
表10. I2S2时钟配置和精度

目标音频频率 (kHz)	麦克风模式	I2S_APB1 时钟Mux 源	DivM	PLLN	DivR	DivP	I2S时钟 (MHz) ⁽¹⁾	精度 (ppm)
16	单声道	PLLI2SR	5	192	2	-	153.60	0
		PLLI2SR	5	192	5	-	61.44	0
	立体声 (有定时器)	PLL2	7	344	6	4	65.52	-186
48	单声道	PLLI2SR	5	192	2	-	153.60	0
		PLLI2SR	5	192	5	-	61.44	0
	立体声 (有定时器)	PLL2	7	344	4	4	98.29	-186

1. 为了正常工作，I2S时钟频率必须高于其APB时钟频率。

图 28所示为单声道模式下I2S时钟配置的示例。

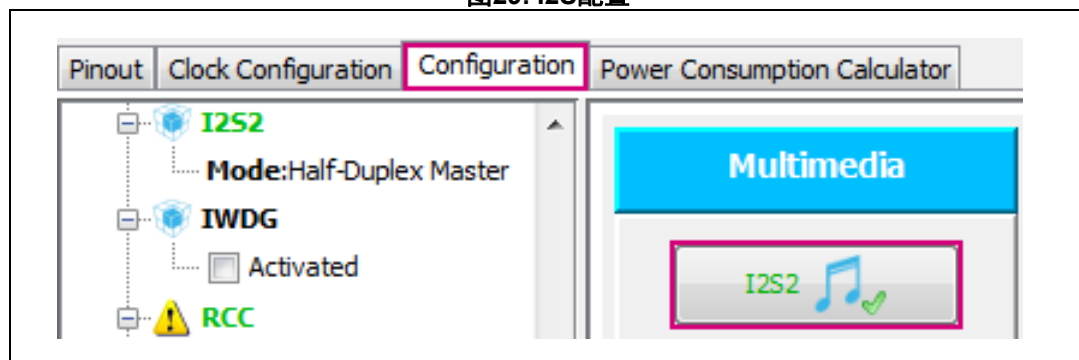
图28. I2S2时钟配置



I2S配置

选择“配置”选项卡，然后点击“I2S2”按钮，如图 29所示。

图29. I2S配置



a) I2S参数设置:

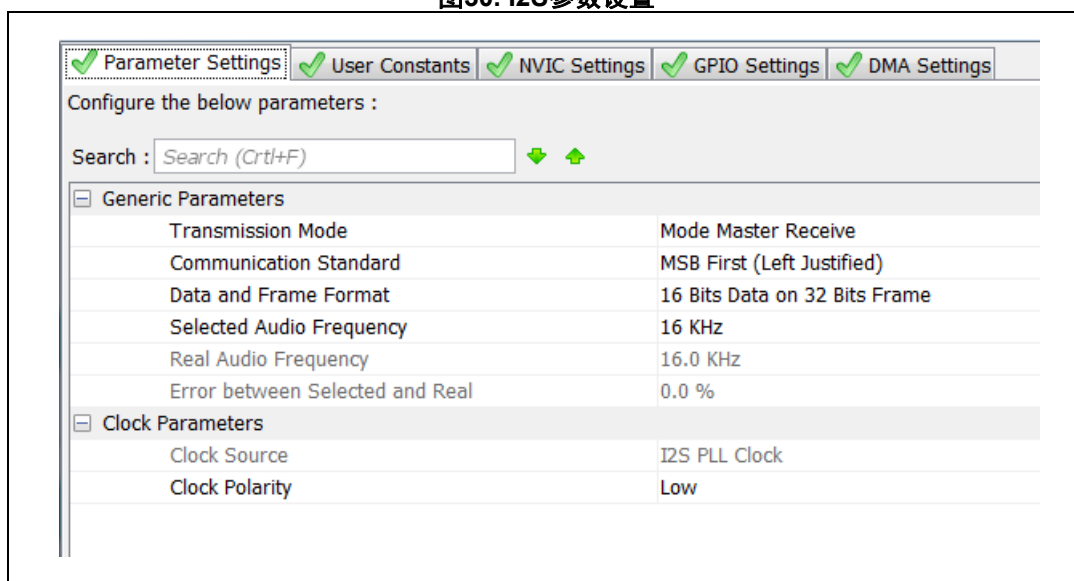
在I2S配置窗口中，选择“参数设置”选项卡，然后配置参数。

根据以下条件配置I2S:

- 选择的音频频率 = AUDIO_SAMPLING_FREQUENCY
- 通信标准=MSB优先（左对齐）：I2S在时钟下降沿读取数据。在单声道配置下，麦克风的LR引脚必须连接到GND。

图 30所示为单声道模式下音频采样频率 = 16 kHz时的I2S配置示例。

图30. I2S参数设置

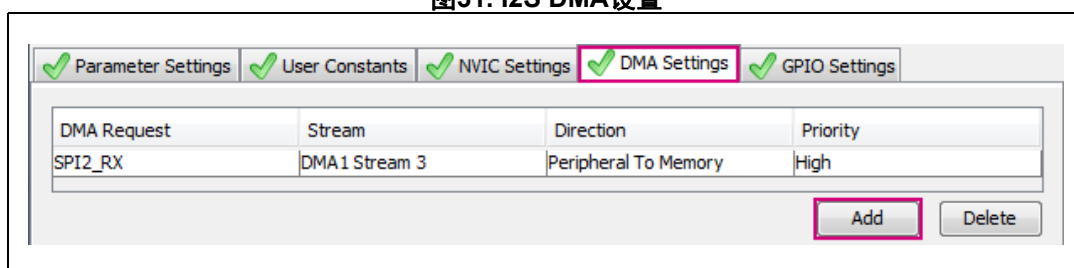


b) I2S DMA配置

在本例中，DMA处理从I2S传输到存储器的PDM数据。

在I2S配置窗口中，选择“DMA设置”选项卡并添加DMA请求。[图 31](#)显示了如何使能DMA。

图31. I2S DMA设置



点击新建的DMA请求，并参考DMA请求设置章节完成DMA配置。

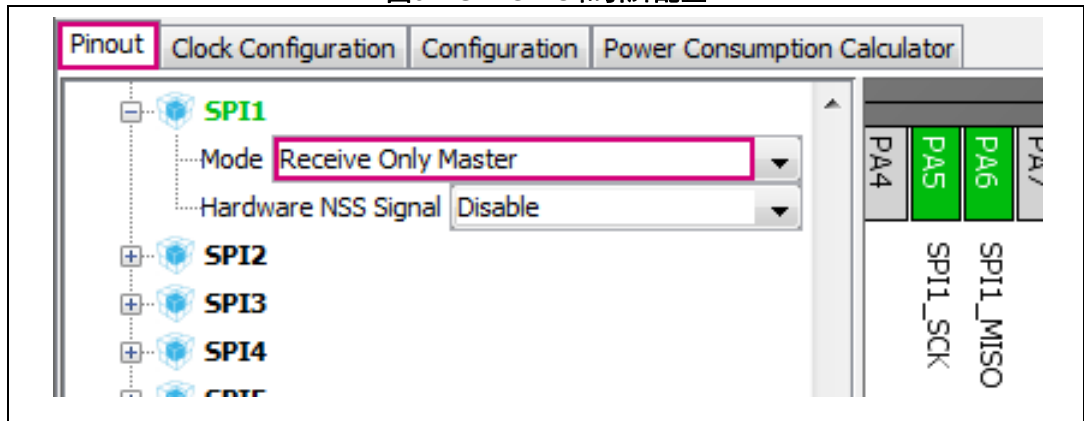
SPI

GPIO和引脚配置

从“引脚排列”选项卡列出的硬件中选择SPI1外设，并将其配置为仅接收主模式。

[图 32](#)显示了如何以仅接收主模式使能SPI1。在正确配置SPI1外设GPIO后，使能的引脚（SPI1_SCK和SPI1_MISO）会以绿色突出显示。

图32. SPI GPIO和引脚配置



SPI时钟配置

本节描述对于16 KHz和48 KHz数据流，在单声道和立体声模式下SPI的不同时钟配置。使用HSE = 8MHz作为时钟源。

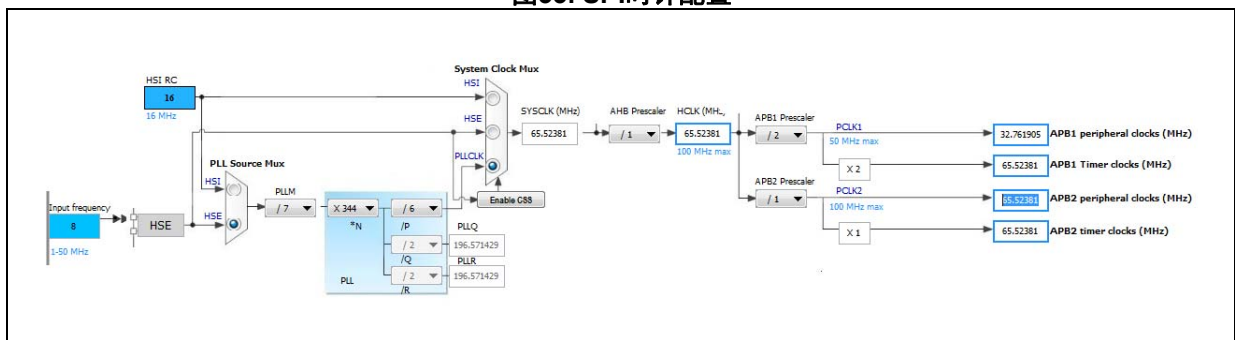
使用APB2时钟作为SPI的参考时钟。

表11. SPI时钟配置和精度

目标音频频率 (kHz)	麦克风模式	DivM	PLLN	DivR	DivP	SPI时钟 (MHz)	精度 (ppm)
16	单声道	7	344	-	6	65.52	-186
	立体声 (有定时器)	7	344	-	6	65.52	-186
48	单声道	7	344	-	4	98.29	-186
	立体声 (有定时器)	7	344	-	4	98.29	-186

图 33所示为单声道模式下SPI时钟配置的示例。

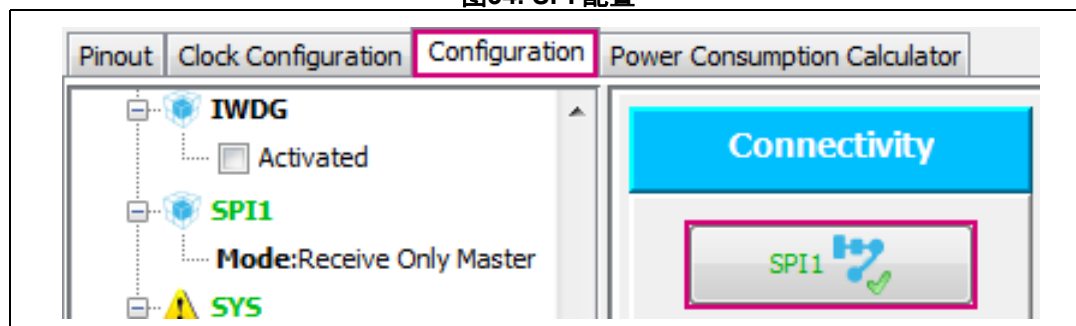
图33. SPI时钟配置



SPI 配置

选择“配置”选项卡，然后点击“SPI”按钮，如 [图 34](#)所示。

图34. SPI 配置



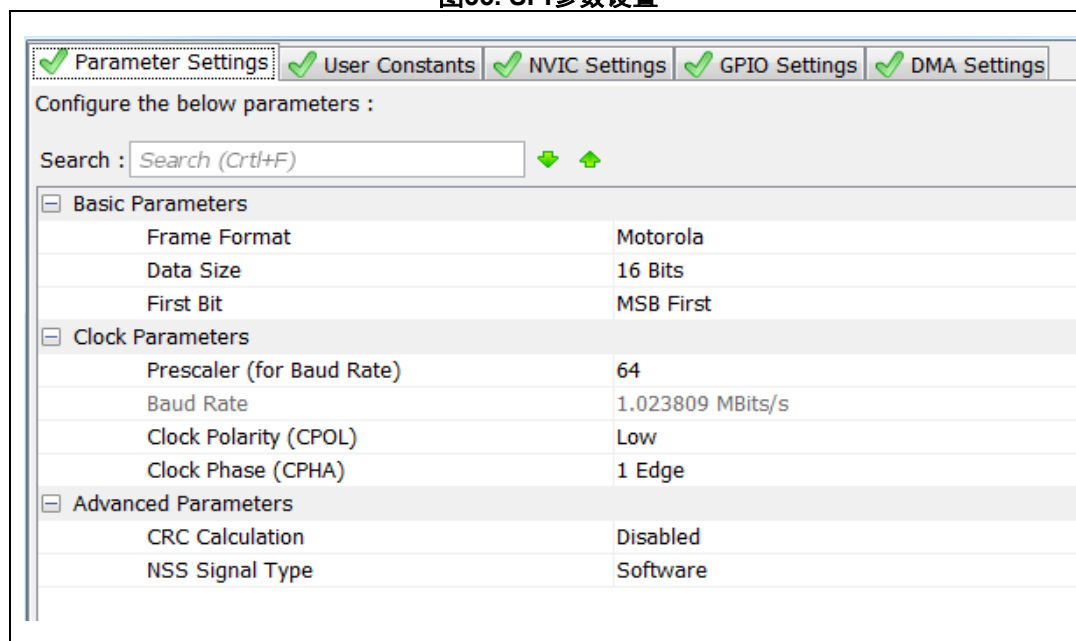
a) SPI参数设置

在SPI配置窗口中，选择“参数设置”选项卡，然后配置参数。

在单声道模式下，按照麦克风L/R引脚的状态选择时钟极性和时钟相位。

[图 35](#)所示为单声道模式下音频频率 = 16 kHz时的SPI配置示例。

图35. SPI参数设置

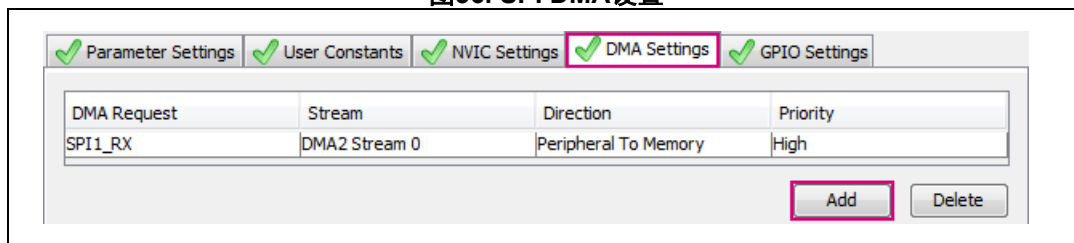


b) SPI DMA设置

在本例中，DMA处理从SPI传输到存储器的PDM数据。

在SPI配置窗口中，选择“DMA设置”选项卡并添加DMA请求。图 36显示了如何使能DMA。

图36. SPI DMA设置



点击DMA请求，并参考DMA请求设置章节完成DMA配置。

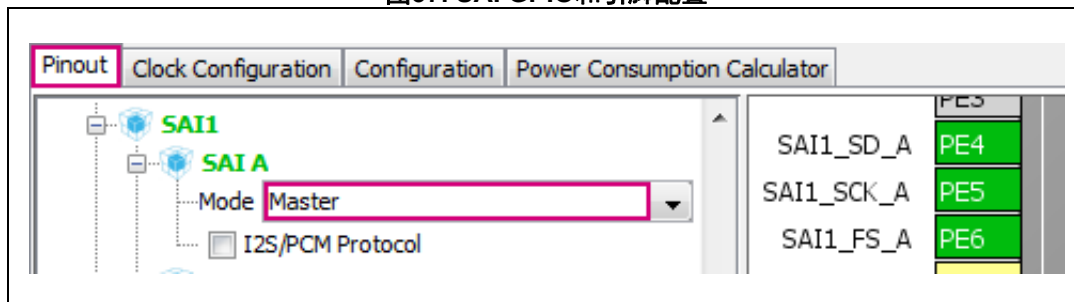
(SAI)

GPIO和引脚配置

从“引脚排列”选项卡列出的硬件中选择SAI1外设，并以主模式使能其子模块A。

图 37显示了如何以主模式使能SAI的子模块A。在正确配置SAI接口GPIO后，使能的引脚（SAI1_SD_A、SAI1_SCK_A和SAI1_FS_A）会以绿色突出显示。

图37. SAI GPIO和引脚配置



注：在本例中，不能使用SAI1FS_A引脚。在生成项目后，通过对MSP初始化文件代码（stm32f4xx_hal_msp.c）稍作修改，可以释放该引脚并用作GPIO。

时钟配置

在“时钟配置”选项卡中设置SAI时钟。

表 12提供了时钟配置的音频采样频率精度值。

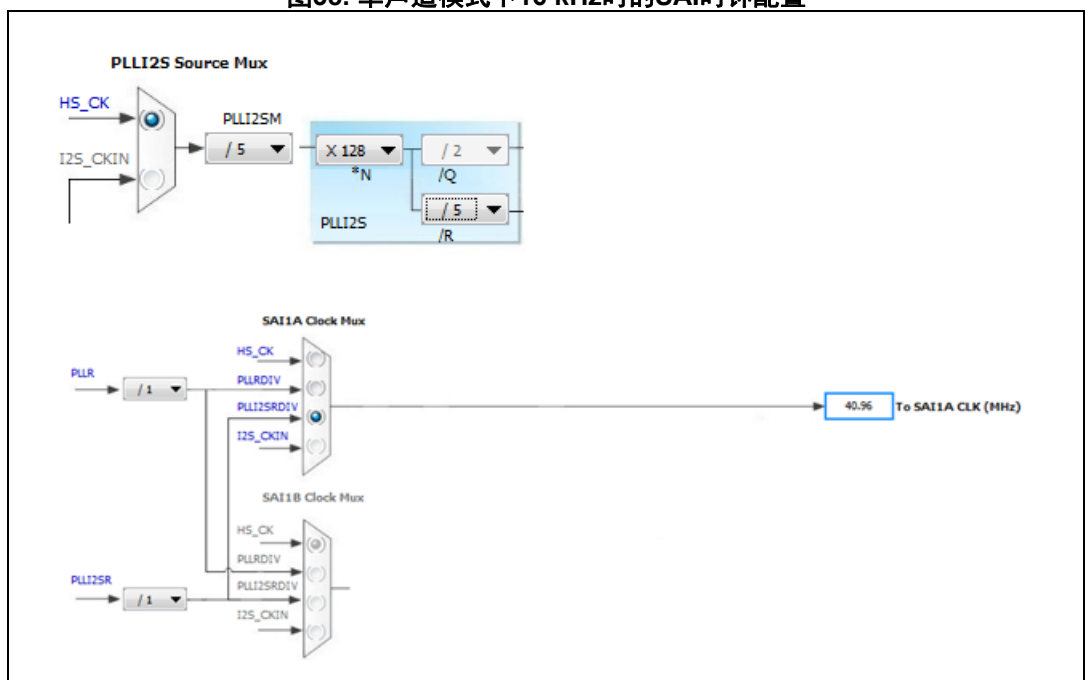
使用PLLI2SR作为时钟源。

表12. 时钟配置和精度

目标音频频率 (kHz)	麦克风模式	DivM	PLLN	DivR	SAI时钟 (MHz)	精度 (ppm)
16	单声道	5	128	5	40.96	0
	立体声 (有定时器)	5	128	5	40.96	0
48	单声道	5	192	25	12.29	0
	立体声 (有定时器)	5	192	25	12.29	0

图 38显示了SAI时钟配置的示例。

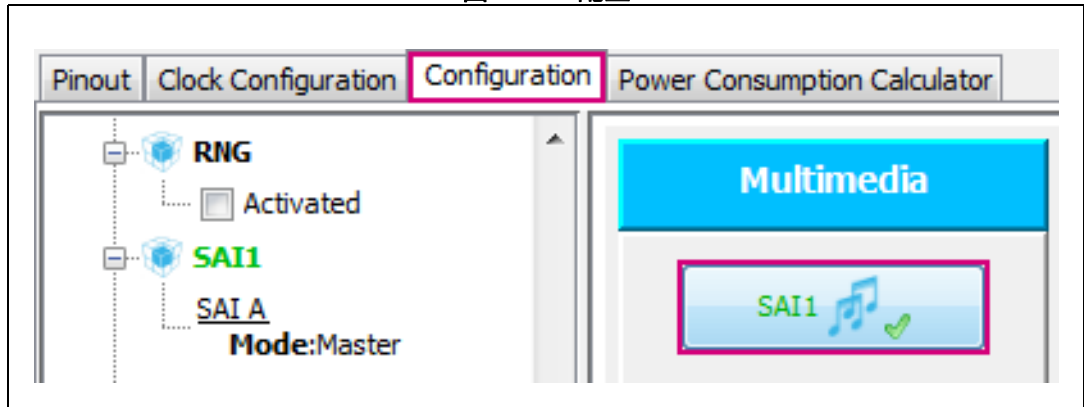
图38. 单声道模式下16 kHz时的SAI时钟配置



SAI配置

选择“配置”选项卡，然后点击“SAI1”按钮，如 [图 39](#)所示。

图39. SAI配置



a) SAI参数设置

在SAI配置窗口中，选择“参数设置”选项卡，然后配置参数。根据以下条件配置使能的SAI子模块：

- 数据大小 = 16位
- 输出模式 = 立体声（无论是一个还是两个麦克风连接到SAI都不要更改该参数）
- 压扩模式 = 无压扩模式
- 帧同步定义 = 通道识别
- 插槽数 = 4
- 激活插槽 = 全部
- 帧同步激活长度 = 帧长度 / 2
- 主时钟分频器 = 使能（其他可能的解决方案禁用主时钟分频器）
- 音频频率 = 音频采样频率
- 在单声道模式下，按照麦克风L/R引脚的状态选择有效时钟边沿。

[图 40](#)所示为单声道模式下音频频率 = 16 kHz时的SAI配置示例。

图40. SAI参数设置

Parameter Settings
 User Constants
 NVIC Settings
 GPIO Settings
 DMA Settings

Configure the below parameters :

Search :

SAI A

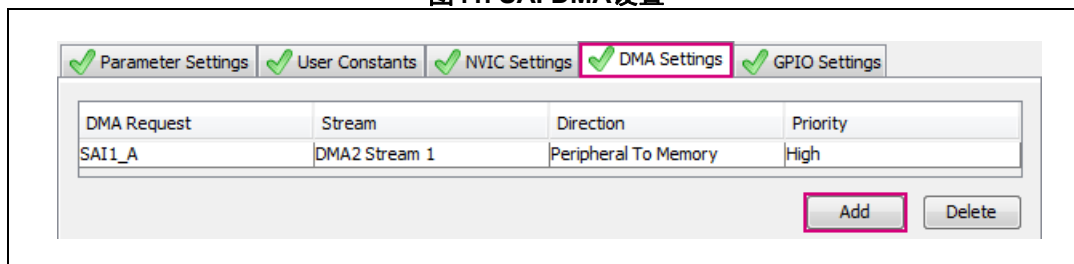
Basic Parameters	
Protocol	Free
Audio Mode	Master Transmit
Frame Length	64 bits
Data Size	16 Bits
Slot Size	DataSize
Output Mode	Stereo
Companding Mode	No companding mode
SAI SD Line Output Mode	Driven
Frame Parameters	
First Bit	MSB First
Frame Synchro Active Level Length	32
Frame Synchro Definition	Channel Identification
Frame Synchro Polarity	Active Low
Frame Synchro Offset	First Bit
Slot Parameters	
First Bit Offset	0
* Number of Slots (only Even Values)	4
Slot Active Final Value	0x0000FFFF
Slot Active	All
Clock Parameters	
Clock Source	I2S PLL Clock
Master Clock Divider	Enabled
Audio Frequency	16 KHz
Real Audio Frequency	16.0 KHz
Error between Selected	0.0 %
Clock Strobing	Falling Edge
Advanced Parameters	
Fifo Threshold	Empty
Output Drive	Disabled
Synchronization External	Disabled

b) SAI DMA设置

在本例中，DMA处理从SAI传输到存储器的PDM数据。

在SAI配置窗口中，选择“DMA设置”选项卡并添加DMA请求。[图 41](#)显示了如何使能DMA。

图41. SAI DMA设置

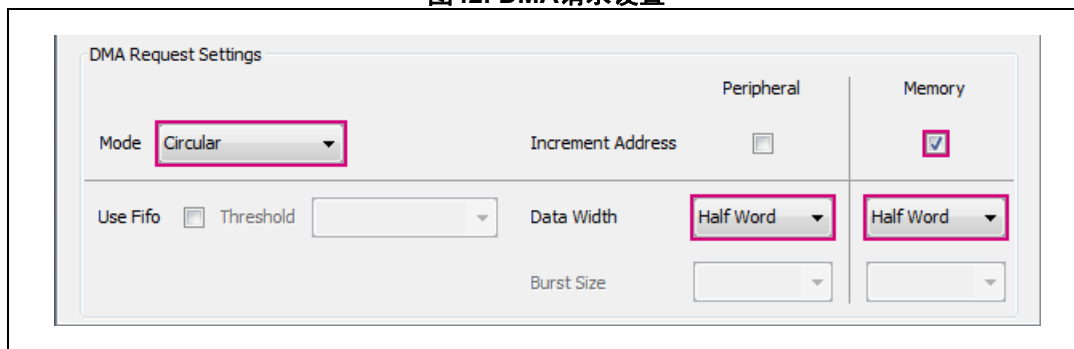


点击新建的DMA请求，并参考DMA请求设置章节完成DMA配置。

DMA请求设置

[图 39](#)所示为每个音频接口的DMA配置请求。

图42. DMA请求设置



定时器

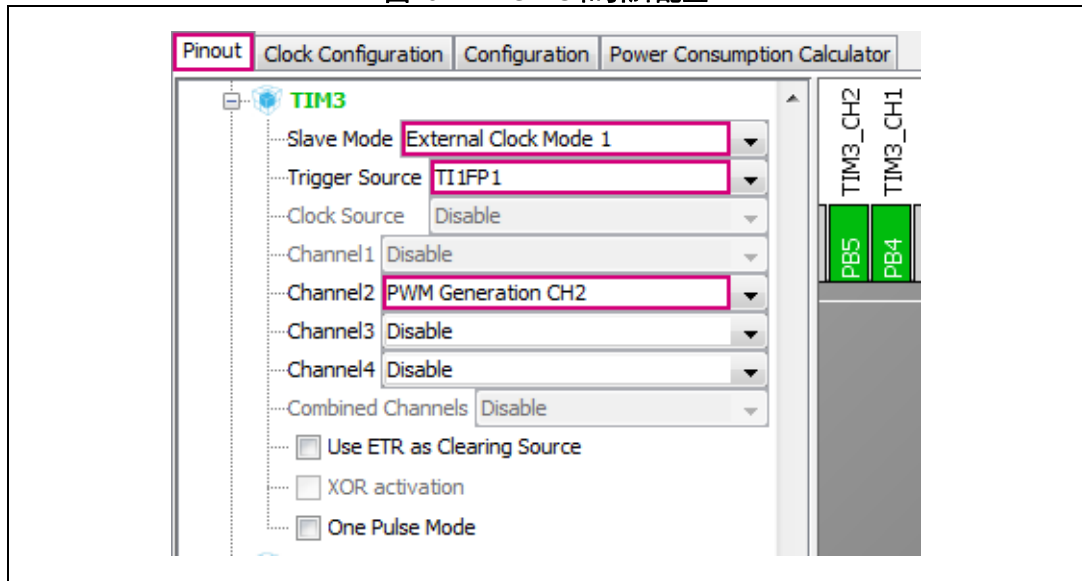
在立体声模式下，使用定时器对时钟接口生成的时钟频率进行分频，并将分频后的时钟提供给数字麦克风。

GPIO和引脚配置

在“引脚排列”选项卡中，从列出的硬件中选择“定时器3”，并通过选择“外部时钟模式1”使能从模式。选择TI1FP1作为触发源，并以“PWM生成CH2”使能通道2。

[图 39](#)显示了如何使能TIM3。在正确配置TIM3 GPIO后，使能的引脚（TIM3_CH1和TIM3_CH2）会以绿色突出显示。

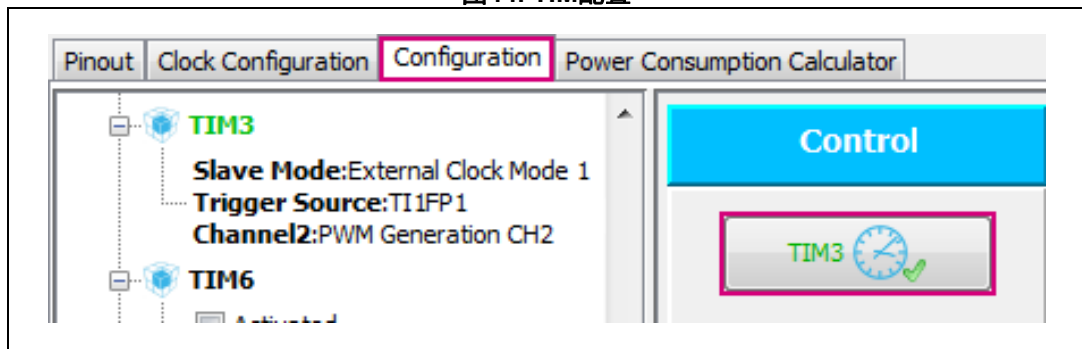
图43. TIM GPIO和引脚配置



定时器配置

选择“配置”选项卡，然后点击“TIM3”按钮，如 图 44所示。

图44. TIM配置

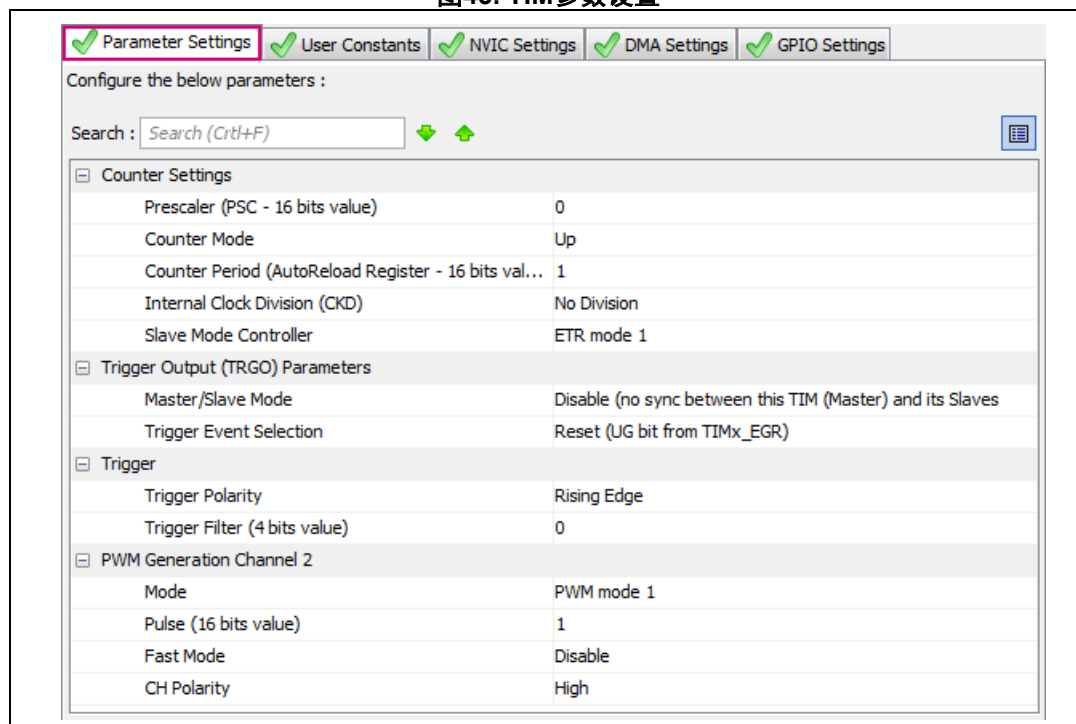


TIM参数设置

在TIM3配置窗口中，选择“参数设置”选项卡。

图 44显示了如何配置TIM3以将触发源时钟频率除以2。

图45. TIM参数设置



4.1.2 添加PDM软件解码库中间件文件

选择常用的工具链，使用STM32CubeMX生成项目并打开生成的项目。

对于STM32F413xx MCU，PDM音频软件解码库包含一个头文件pdm_filter.h和以下平台的二进制/对象代码：

- libPDMFilter_CM4F_IAR.a：适用于IAR编译器
- libPDMFilter_CM4F_Keil.lib：适用于ARM编译器
- libPDMFilter_CM4F_GCC.a：适用于GNU编译器

“STM32Cube_FW_F4”固件包（V1.16及之后的版本）中提供了此库，路径为“Middlewares\ST\STM32_Audio\Addons\PDM”

务必将头文件pdm_filter.h的路径添加到项目的Include Paths，并将相应的二进制/对象文件添加到项目源文件。

4.2 示例2：使用两个同步SAI子模块，连接立体声模式下的数字麦克风

本例以NUCLEO-F413ZH板为基础并使用两个外部数字麦克风，每个麦克风连接到一个SAI子模块。它适用于 [第 2.2.2 节：使用两个同步SAI子模块](#) 所述的硬件连接。

4.2.1 使用STM32CubeMX配置SAI

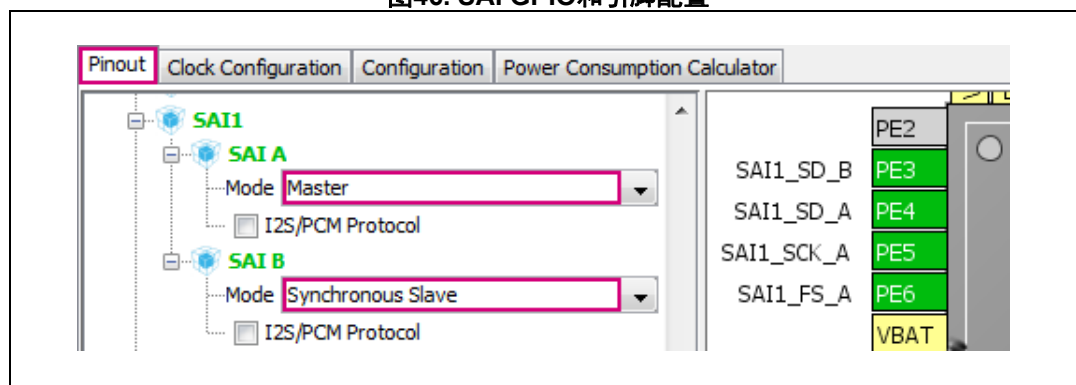
GPIO和引脚配置

在创建STM32CubeMX项目后，从“引脚排列”选项卡上列出的硬件中选择SAI1外设，将其子模块A配置为主模式，并将子模块B配置为同步从模式。

[图 46](#)显示了如何以主模式使能SAI1的子模块A并以同步从模式使能子模块B。在这种情况下，两个子模块共享位时钟和帧同步信号，以减少通信时占用外部引脚的数量。

在正确配置SAI1外设GPIO后，使能的引脚（SAI1_SD_A、SAI1_SD_B、SAI1_SCK_A和SAI1_FS_A）会以绿色突出显示。

图46. SAI GPIO和引脚配置



使用STM32CubeMX配置时钟

本节描述对于16 KHz和48 KHz数据流，使用两个同步子模块时的SAI时钟配置。

对于16KHz采样率，SAI生成的位时钟频率必须是1.024MHz。

对于48KHz采样率，SAI生成的位时钟频率必须是3.072MHz。

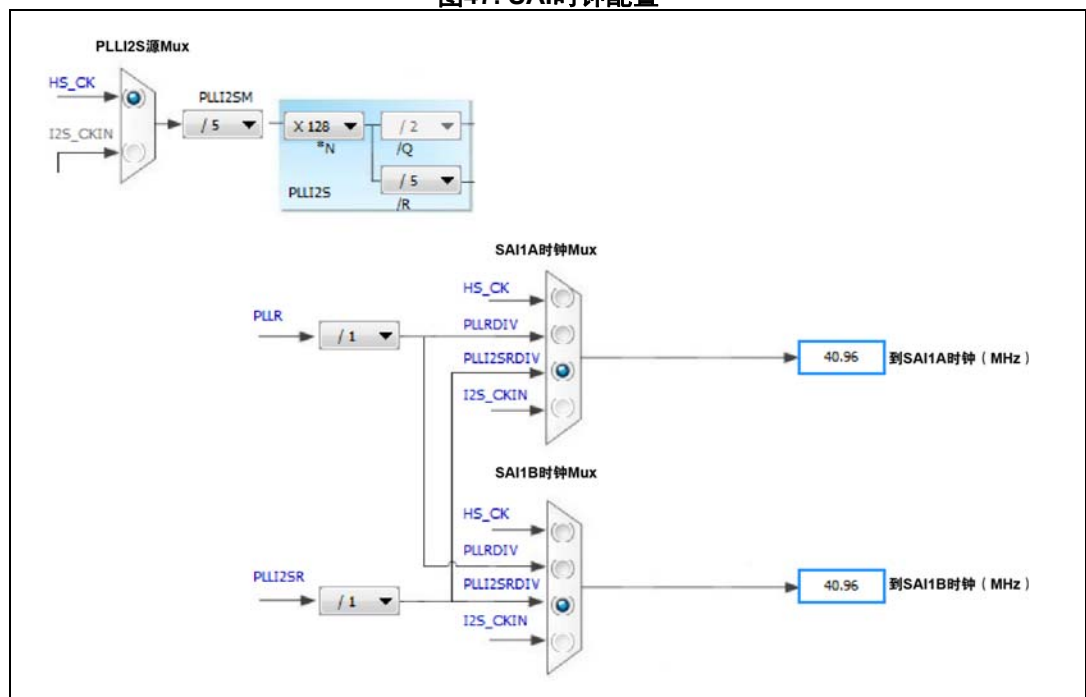
使用PLLI2SR作为时钟源。为了获得准确的位时钟频率，用户必须配置合适的MCKIDV。

表13. SAI时钟配置和精度

目标音频频率 (kHz)	DivM	PLL N	DivR	SAI时钟 (MHz)	精度 (ppm)
16	5	128	5	40.96	0
48	5	192	25	12.29	0

图 47提供了本例中使用的SAI时钟配置。

图47. SAI时钟配置



参数设定

在SAI配置窗口中，选择“参数设置”选项卡，然后配置参数。应根据以下条件配置SAI子模块：

- 数据大小 = 16位
- 帧长度 = 4 x 数据大小
- 输出模式 = 立体声
- 压扩模式 = 无压扩模式
- 帧同步定义 = 通道识别
- 插槽数 = 4
- 激活插槽 = 全部
- 帧同步激活长度 = 帧长度 / 2
- 主时钟分频器 = 使能
- 音频频率 = AUDIO_SAMPLING_FREQUENCY

根据连接到每个子模块的麦克风的L/R引脚状态选择有效时钟边沿。



 48 和  49 分别显示了在音频采样频率 = 16 kHz时如何配置SAIA和SAIB。

图48. SAI参数设置

Parameter Settings
 User Constants
 NVIC Settings
 GPIO Settings
 DMA Settings

Configure the below parameters :

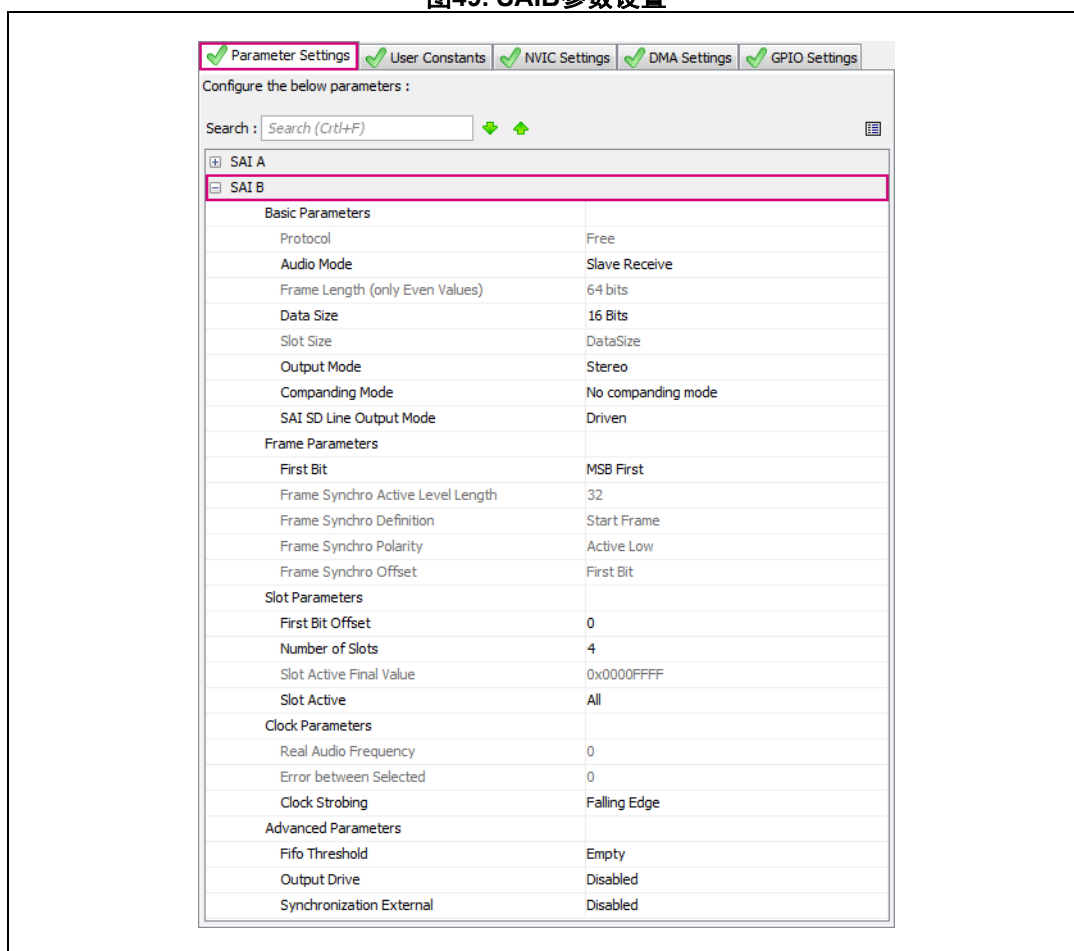
Search :

SAI A

Basic Parameters	
Protocol	Free
Audio Mode	Master Receive
Frame Length	64 bits
Data Size	16 Bits
Slot Size	DataSize
Output Mode	Stereo
Companding Mode	No companding mode
Frame Parameters	
First Bit	MSB First
Frame Synchro Active Level Length	32
Frame Synchro Definition	Channel Identification
Frame Synchro Polarity	Active Low
Frame Synchro Offset	First Bit
Slot Parameters	
First Bit Offset	0
Number of Slots (only Even Values)	4
Slot Active Final Value	0x0000FFFF
Slot Active	All
Clock Parameters	
Clock Source	I2S PLL Clock
Master Clock Divider	Enabled
Audio Frequency	16 KHz
Real Audio Frequency	16.0 KHz
Error between Selected	0.0 %
Clock Strobing	Rising Edge
Advanced Parameters	
Fifo Threshold	Empty
Output Drive	Disabled
Synchronization External	Disabled

SAI B

图49. SAIB参数设置

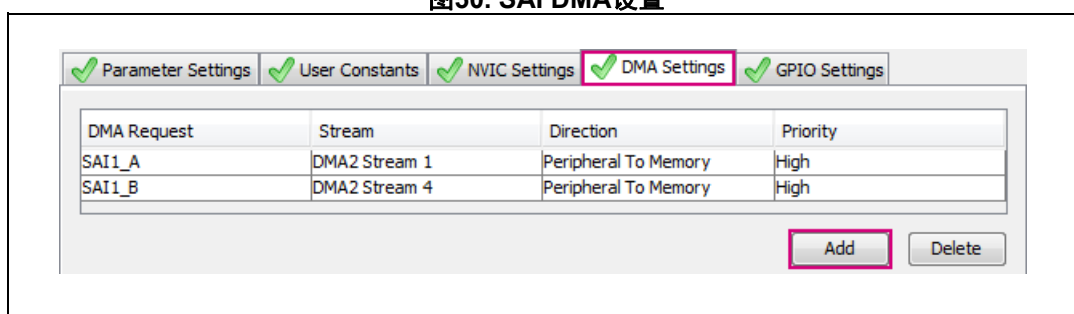


SAI DMA设置

在本例中，DMA处理从每个SAI子模块传输到存储器的PDM数据。

在SAI配置窗口中，选择“DMA设置”选项卡，并为每个子模块添加DMA请求。图 50显示了如何使能DMA。

图50. SAI DMA设置



点击新建的DMA请求，并参考DMA请求设置章节完成DMA配置。

4.2.2 添加PDM软件解码库中间件文件

请参见 [第 4.1.2 节：添加PDM软件解码库中间件文件](#)。

4.3 示例3：使用SAI的PDM接口，连接立体声模式下的数字麦克风

本例以NUCLEO-H743ZI板为基础，使用两个外部数字麦克风，在立体声模式下，麦克风通过PDM接口连接到SAI。

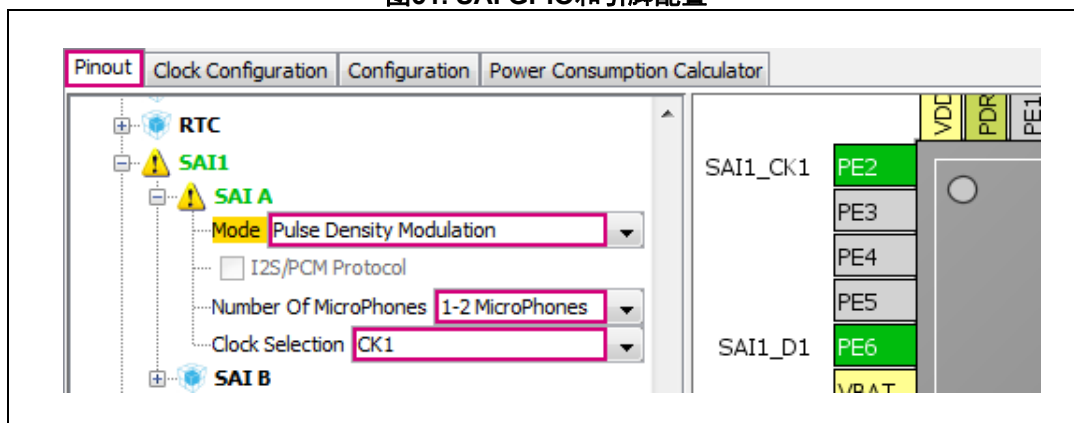
4.3.1 使用STM32CubeMX配置SAI

GPIO和引脚配置

在创建STM32CubeMX项目后，从“引脚排列”选项卡上列出的硬件中选择SAI1外设，并将其配置为脉冲密度调制模式。选择麦克风数量和输出时钟。

[图 51](#)显示了如何以脉冲密度调制（PDM）模式使能SAI1的子模块A，支持两个麦克风并选择CK1作为输出时钟。在正确配置SAI外设GPIO后，使能的引脚（SAI1_CK1和SAI1_D1）会以绿色突出显示。

图51. SAI GPIO和引脚配置

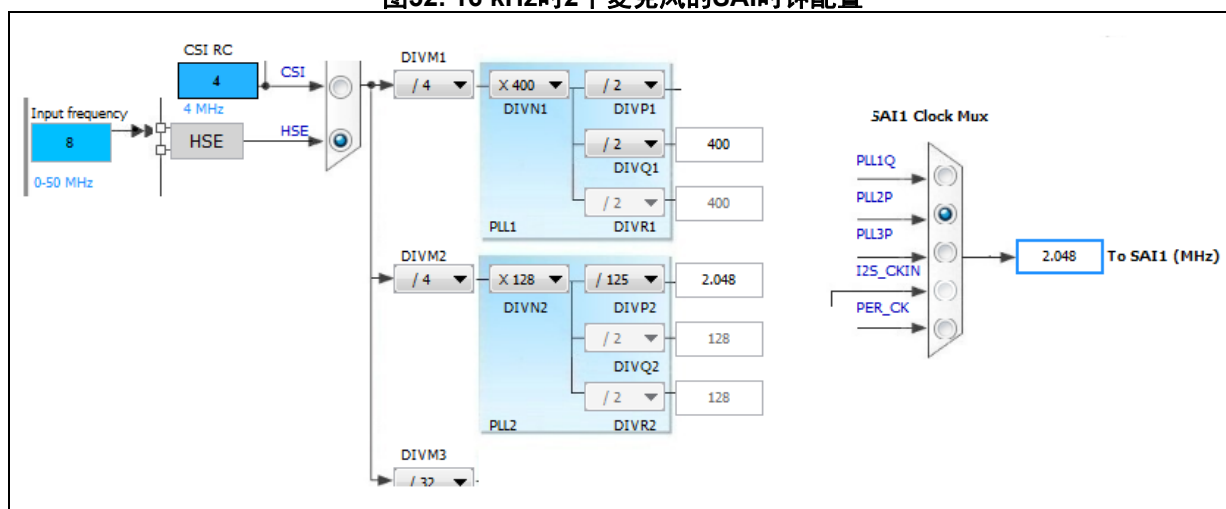


时钟配置

[图 52](#)提供了本例中使用的SAI时钟配置。

有关时钟配置的更多详细信息，请参阅参考手册RM0433中的表格“允许的TDM帧配置”。

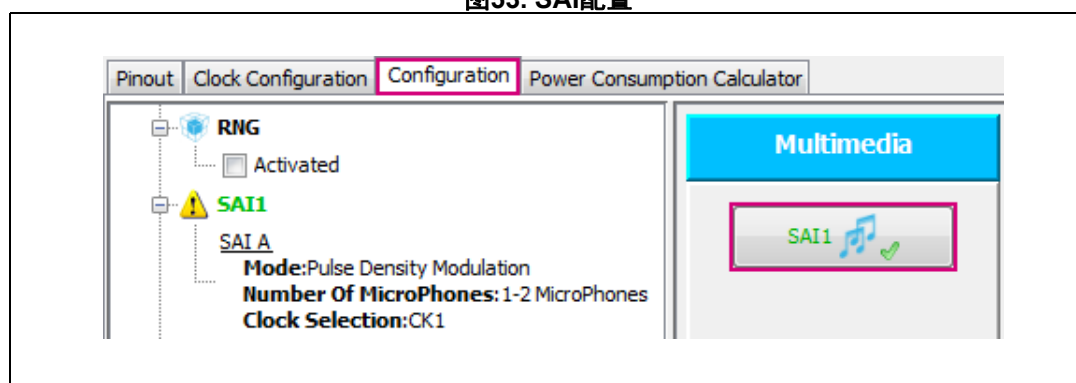
图52. 16 kHz时2个麦克风的SAI时钟配置



SAI配置

选择“配置”选项卡，然后点击“SAI1”按钮，如图 53所示。

图53. SAI配置



参数设定

在SAI配置窗口中，选择“参数设置”选项卡，然后配置参数。应根据以下条件配置SAI子模块A:

- 数据大小 = 16位
- 帧长度 = 16
- 输出模式 = 立体声
- 压扩模式 = 无压扩模式
- 帧同步定义 = 起始帧
- 插槽数 = 1
- 激活插槽 = 全部
- 帧同步激活长度 = 1
- 帧同步极性 = 高电平有效
- 主时钟分频器 = 禁用

图 54显示了在音频采样频率 = 16 kHz时如何配置SAIA。

图54. SAI参数设置

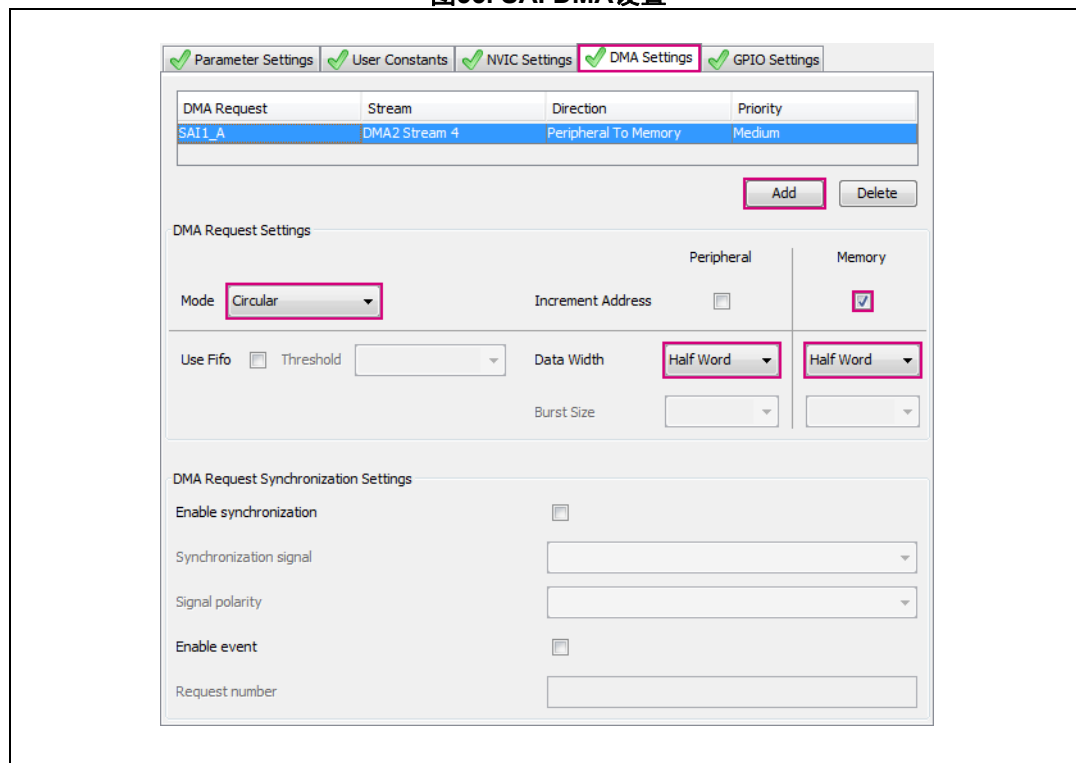
Parameter Settings	User Constants	NVIC Settings	GPIO Settings	DMA Settings																																														
Configure the below parameters :																																																		
Search : <input type="text" value="Search (Ctrl+F)"/>																																																		
<div style="border: 1px solid black; padding: 2px;"> <div style="background-color: #f0f0f0; padding: 2px;"> [-] Basic Parameters </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Protocol</td><td>Free</td></tr> <tr><td>Audio Mode</td><td>Master Receive</td></tr> <tr><td>Data Size</td><td>16 Bits</td></tr> <tr><td>Output Mode</td><td>Stereo</td></tr> <tr><td>Companding Mode</td><td>No companding mode</td></tr> <tr><td>First Bit</td><td>LSB First</td></tr> </table> <div style="background-color: #f0f0f0; padding: 2px;"> [-] Data Frame Parameters </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Frame Length (only Even Values)</td><td>16 bits</td></tr> <tr><td>Frame Synchro Active Level Length</td><td>1</td></tr> <tr><td>Frame Synchro Definition</td><td>Start Frame</td></tr> <tr><td>Frame Synchro Polarity</td><td>Active High</td></tr> <tr><td>Frame Synchro Offset</td><td>First Bit</td></tr> </table> <div style="background-color: #f0f0f0; padding: 2px;"> [-] Block Slot Parameters </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Slot Size</td><td>DataSize</td></tr> <tr><td>First Bit Offset</td><td>0</td></tr> <tr><td>Number of Slots</td><td>1</td></tr> <tr><td>Slot Active Final Value</td><td>0x0000FFFF</td></tr> <tr><td>Slot Active</td><td>All</td></tr> </table> <div style="background-color: #f0f0f0; padding: 2px;"> [-] Clock Parameters </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Clock Source</td><td>SAI PLL Clock</td></tr> <tr><td>Master Clock Divider</td><td>Disabled</td></tr> <tr><td>Real Audio Frequency</td><td>128.0 KHz</td></tr> <tr><td>Clock Strobing</td><td>Falling Edge</td></tr> </table> <div style="background-color: #f0f0f0; padding: 2px;"> [-] Advanced Parameters </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Fifo Threshold</td><td>Empty</td></tr> <tr><td>Output Drive</td><td>Disabled</td></tr> <tr><td>Synchronization External</td><td>Disabled</td></tr> </table> </div>					Protocol	Free	Audio Mode	Master Receive	Data Size	16 Bits	Output Mode	Stereo	Companding Mode	No companding mode	First Bit	LSB First	Frame Length (only Even Values)	16 bits	Frame Synchro Active Level Length	1	Frame Synchro Definition	Start Frame	Frame Synchro Polarity	Active High	Frame Synchro Offset	First Bit	Slot Size	DataSize	First Bit Offset	0	Number of Slots	1	Slot Active Final Value	0x0000FFFF	Slot Active	All	Clock Source	SAI PLL Clock	Master Clock Divider	Disabled	Real Audio Frequency	128.0 KHz	Clock Strobing	Falling Edge	Fifo Threshold	Empty	Output Drive	Disabled	Synchronization External	Disabled
Protocol	Free																																																	
Audio Mode	Master Receive																																																	
Data Size	16 Bits																																																	
Output Mode	Stereo																																																	
Companding Mode	No companding mode																																																	
First Bit	LSB First																																																	
Frame Length (only Even Values)	16 bits																																																	
Frame Synchro Active Level Length	1																																																	
Frame Synchro Definition	Start Frame																																																	
Frame Synchro Polarity	Active High																																																	
Frame Synchro Offset	First Bit																																																	
Slot Size	DataSize																																																	
First Bit Offset	0																																																	
Number of Slots	1																																																	
Slot Active Final Value	0x0000FFFF																																																	
Slot Active	All																																																	
Clock Source	SAI PLL Clock																																																	
Master Clock Divider	Disabled																																																	
Real Audio Frequency	128.0 KHz																																																	
Clock Strobing	Falling Edge																																																	
Fifo Threshold	Empty																																																	
Output Drive	Disabled																																																	
Synchronization External	Disabled																																																	

DMA 设置

在本例中，DMA处理从SAI传输到存储器的PDM数据。

在SAI配置窗口中，选择“DMA设置”选项卡并添加DMA请求。[图 55](#)显示了如何配置DMA。

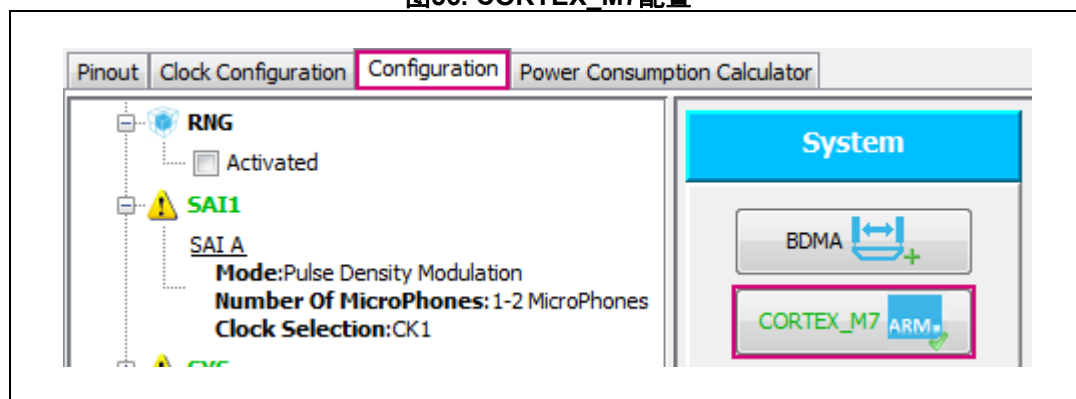
图55. SAI DMA设置



Cortex_M7配置

选择“配置”选项卡，然后点击“CORTEX_M7”按钮，如[图 56](#)所示。

图56. CORTEX_M7配置



参数设定

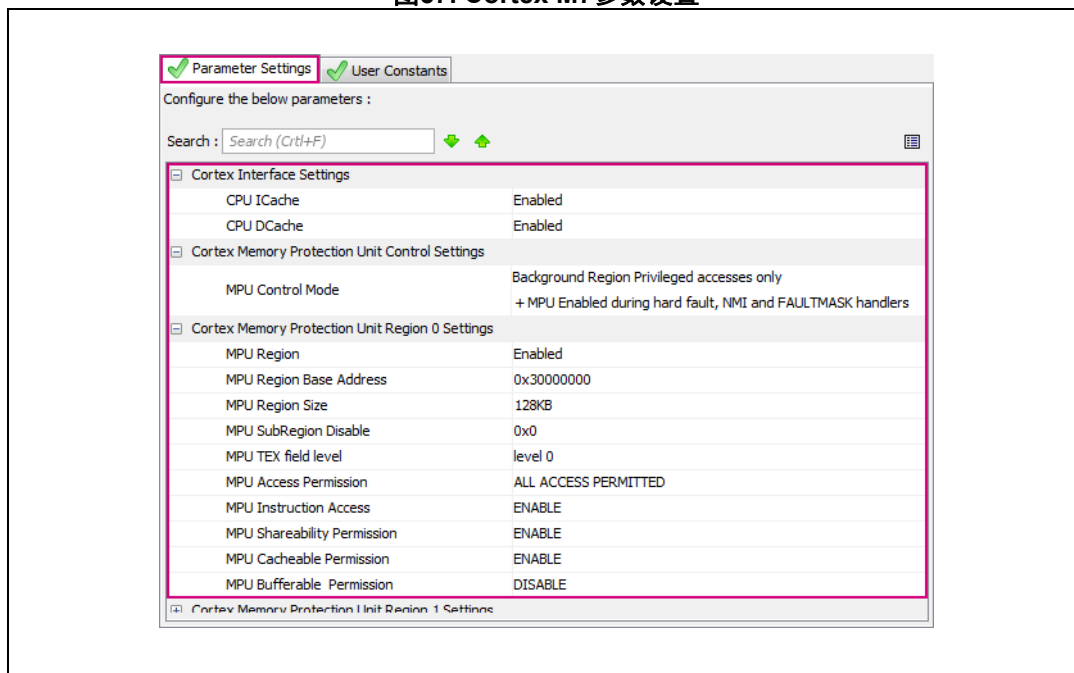
在CORTEX_M7配置窗口中，选择“参数设置”选项卡，然后配置缓存和内存保护单元。当有多个主器件使用AXI接口时，这样可以增强性能。

有关典型MPU配置的详细信息，请参考STM32Cube_FW_H7_V1.0.0固件包（V1.0.0及之后的版本）中的模板项目，路径为“Projects\STM32H743ZI-Nucleo\Templates”。

有关内存保护单元的配置和使用的详细信息，请参考AN4838。

图 57显示了就本例而言如何配置缓存和内存保护单元。

图57. Cortex M7参数设置



4.3.2 添加PDM软件解码库中间件文件

选择常用的工具链，使用STM32CubeMX生成项目并打开生成的项目。

对于STM32H7系列，PDM音频软件解码库包含一个头文件“pdm2pcm_glo.h”和以下平台的二进制/对象代码：

- libPDMFilter_CM7_IAR.a: 适用于IAR编译器
- libPDMFilter_CM7_Keil.lib: 适用于ARM编译器
- libPDMFilter_CM7_GCC.a: 适用于GNU编译器

“STM32Cube_FW_H7”固件包（V1.0.0及之后的版本）中提供了此库，路径为“Middlewares\ST\STM32_Audio\Addons\PDM”

务必将头文件pdm2pcm_glo.h的路径添加到项目的Include Paths，并将相应的二进制/对象文件添加到项目源文件。

4.4 示例4：使用DFSDM连接数字麦克风

本例以NUCLEO-F413ZH板为基础，使用外部数字麦克风，在立体声模式下，麦克风连接到DFSDM。它适用于 [第 2.3 节：数字滤波器，用于ΣΔ调制器 \(DFSDM\)](#) 所述的硬件连接。

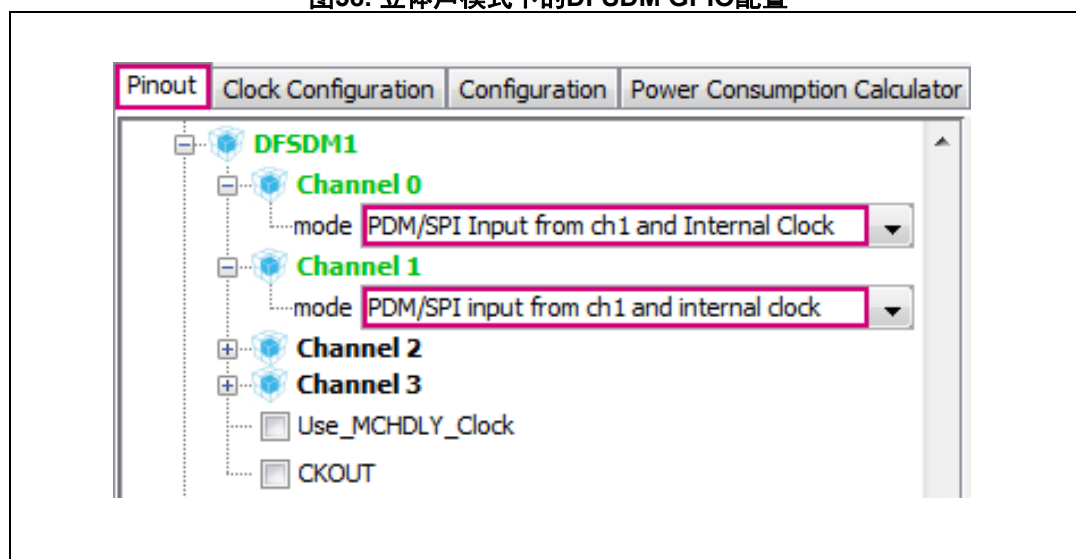
4.4.1 使用STM32CubeMX配置DFSDM

GPIO和引脚配置

在创建STM32CubeMX项目后，从“引脚排列”选项卡上列出的硬件中选择DFSDM1外设，并使能需要的DFSDM通道，如下图所示。

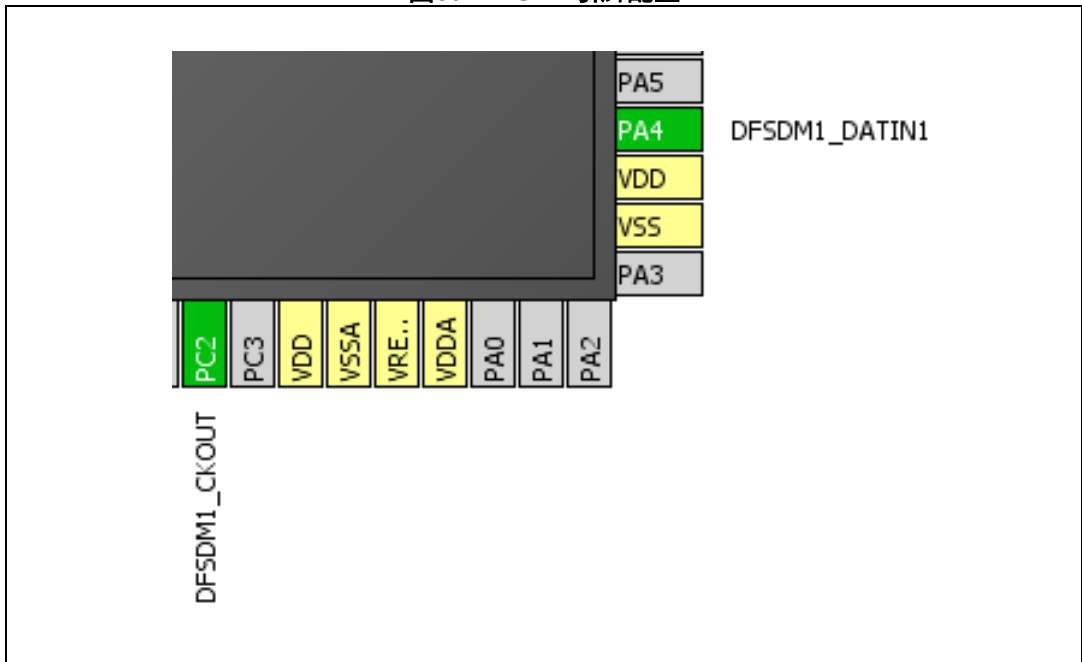
[图 58](#)显示了如何以立体声模式使能DFSDM通道。在该模式下，使能的通道必须是连续的。

图58. 立体声模式下的DFSDM GPIO配置



在正确配置DFSDM外设GPIO后，使能的引脚（DFSDM1_DATIN1和DFSDM1_CKOUT）会以绿色突出显示，如 [图 59](#)所示。

图59. DFSDM引脚配置



通道配置

在DFSDM配置窗口中，选择“参数设置”选项卡，然后配置参数。

根据连接到DFSDM通道的麦克风的L/R引脚状态选择“类型”字段的值。在立体声模式下，每个通道必须在不同的时钟边沿读取数据，以允许麦克风共享一条主数据线。

图 60和图 61分别显示了在音频频率 = 16 kHz时如何配置通道1和通道0。

图60. DFSDM通道1配置

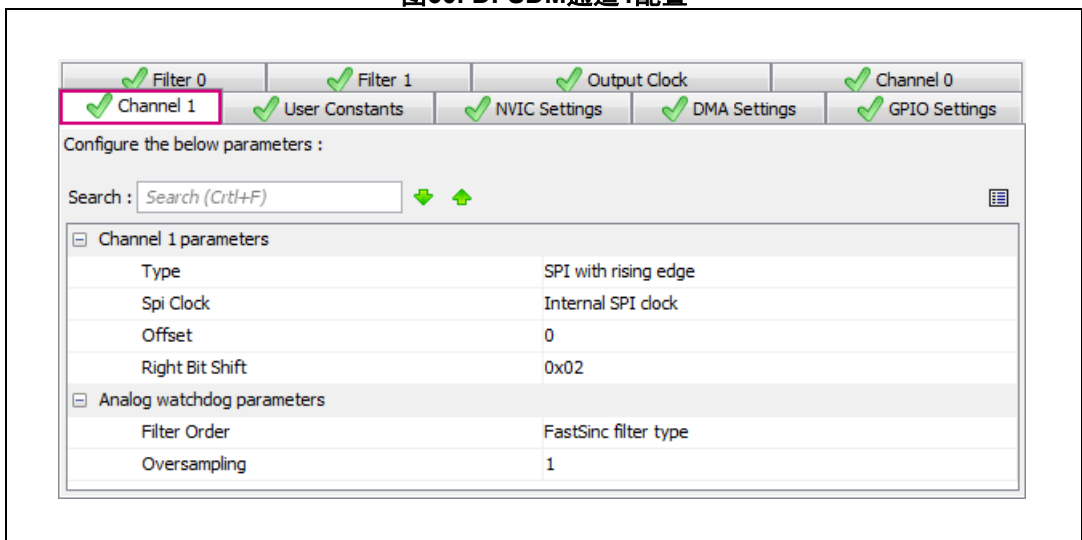
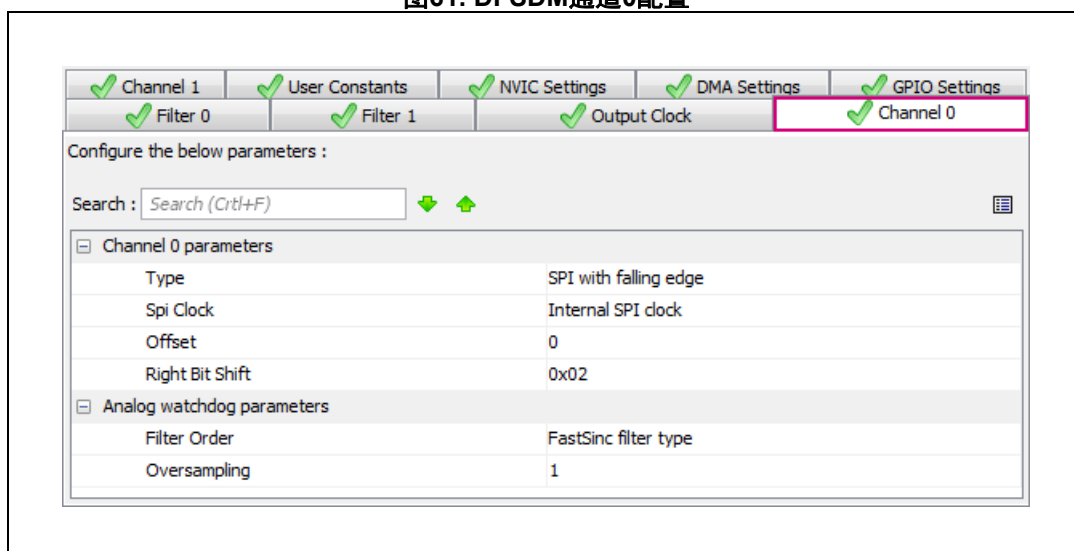


图61. DFSDM通道0配置



滤波器配置

在本例中，过滤器0和过滤器1分别连接到作为常规通道的通道1和通道0。

根据音频采样频率配置辛格阶数(Sinc Order)。表11所示为可能的值。

表14. DFSDM滤波阶数

音频采样频率 (kHz)	滤波阶数
16	Sinc 4或Sinc 5滤波
48	Sinc 4或Sinc 5滤波

“Fosr”字段包含抽取因子的值。

图 62和图 63所示为音频采样频率 = 16 kHz时滤波器1和滤波器0的配置示例。

图62. DFSDM滤波器0配置

Channel 0	Channel 1	User Constants	NVIC Settings	GPIO Settings	DMA Settings
Filter 0	Filter 1			Output Clock	
Configure the below parameters :					
Search : Search (Ctrl+F) <input type="text"/>					
regular channel selection					
regular channel selection	Channel 1				
Continuous Mode	Continuous Mode				
Trigger to start regular conversion	Software trigger				
Fast Mode	Enable				
Dma Mode	Enable				
injected channel selection					
Channel0 as injected channel	Disable				
Channel1 as injected channel	Disable				
Channel2 as injected channel	Disable				
Channel3 as injected channel	Disable				
Filter parameters					
Sinc Order	Sinc 4 filter type				
Fosr	64				
Iosr	1				

图63. DFSDM滤波器1配置

Channel 0	Channel 1	User Constants	NVIC Settings	GPIO Settings	DMA Settings
Filter 0	Filter 1			Output Clock	
Configure the below parameters :					
Search : Search (Ctrl+F) <input type="text"/>					
regular channel selection					
regular channel selection	Channel 0				
Continuous Mode	Continuous Mode				
Trigger to start regular conversion	Software trigger				
Fast Mode	Enable				
Dma Mode	Enable				
injected channel selection					
Channel0 as injected channel	Disable				
Channel1 as injected channel	Disable				
Channel2 as injected channel	Disable				
Channel3 as injected channel	Disable				
Filter parameters					
Sinc Order	Sinc 4 filter type				
* Fosr	64				
Iosr	1				

输出时钟配置

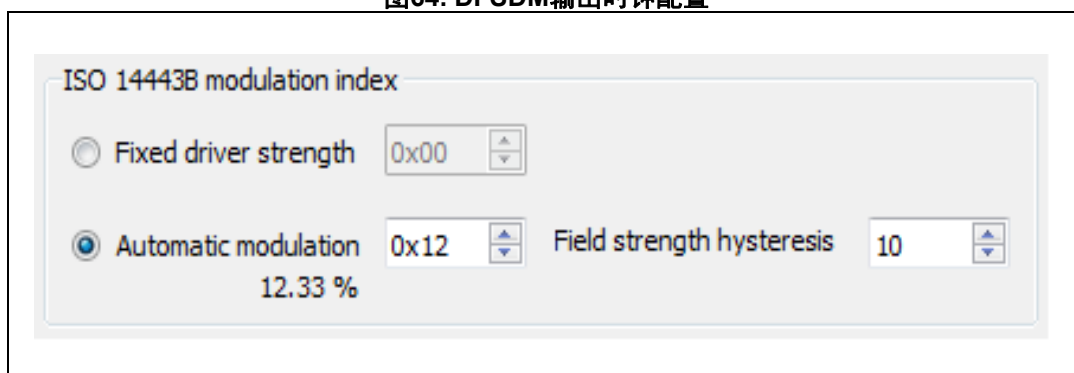
在本例中，选择音频时钟作为DFSDM输出时钟源，在配置时钟时，它会带给用户更大的灵活性。

必须按照以下公式计算时钟分频器值：

$$\text{分频器} = \text{DFSDM时钟源} / (\text{AUDIO_SAMPLING_FREQUENCY} \times \text{DECIAMTION_FACTOR})$$

图 64所示为DFSDM时钟源 = 48 MHz、音频采样频率 = 16 kHz且抽取因子 = 64时的配置示例。

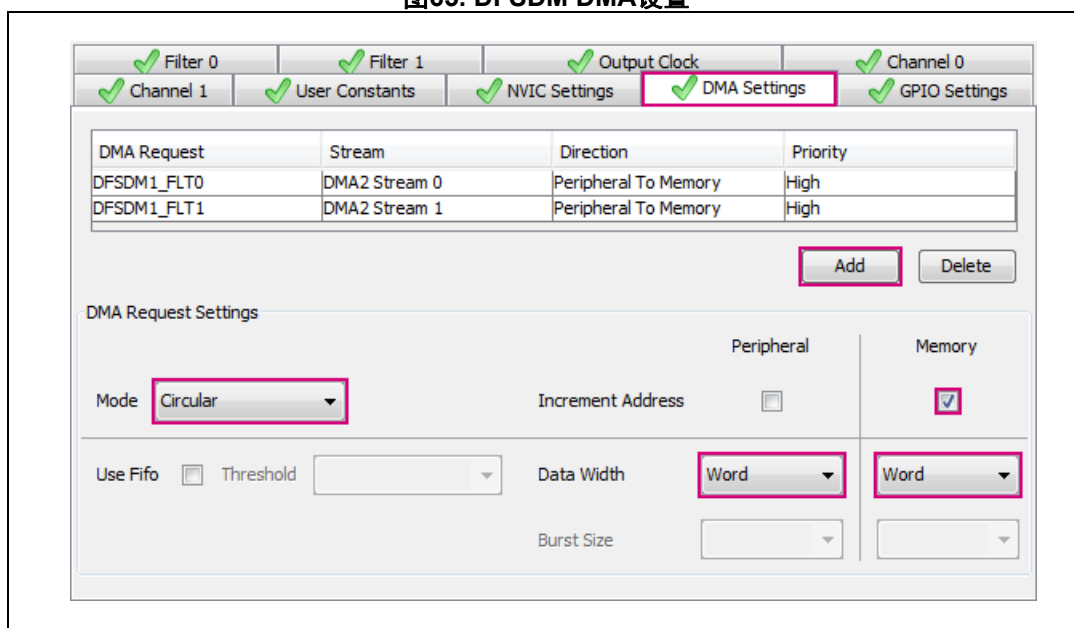
图64. DFSDM输出时钟配置



DFSDM DMA设置

在本例中，由DMA处理从DFSDM通道到存储器的数据传输，以便减少软件开销。在DFSDM配置窗口中，选择“DMA设置”选项卡，并为每个滤波器添加DMA请求。点击新建的DMA请求并配置设置，如 [图 65](#)所示。

图65. DFSDM DMA设置



DFSDM时钟配置

在本例中，DFSDM输出时钟源自音频时钟，它向DFSDM提供61.44kHz时钟，如 [图 66](#)所示。

图66. DFSDM时钟配置

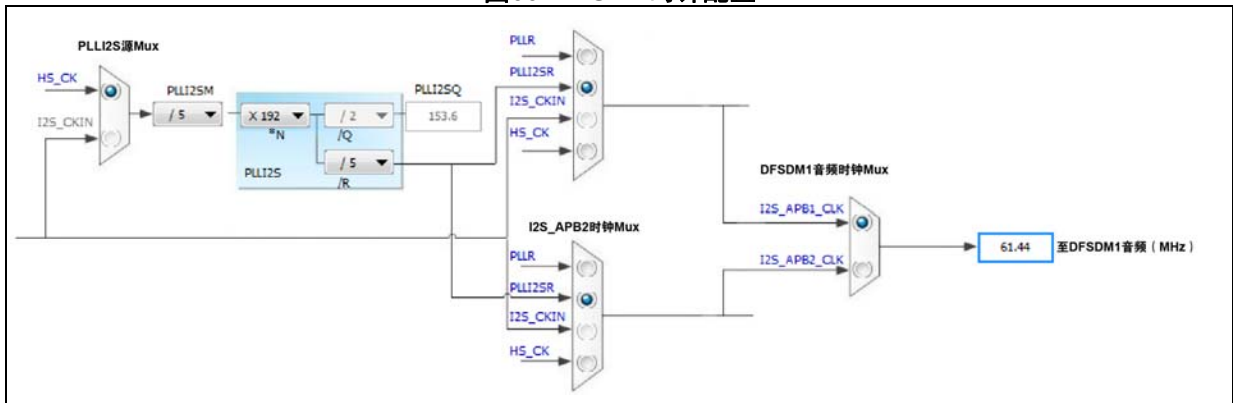


表 15提供了这种时钟配置的音频采样频率精度值。

- 对于16 kHz数据流，CKOUT频率必须是1.024 MHz。
- 对于48 kHz数据流，CKOUT频率必须是3.072 MHz。
- 过采样率为64。

表15. DFSDM时钟配置精度值

目标音频频率 (kHz)	DivM	DivN	DivR	输出时钟分频器	DFSDM 时钟 (MHz)	精度 (ppm)
16	5	192	5	60	61.44	0
48	5	192	5	20	61.44	0

5 结论

STM32 MCU和MPU提供各种用于连接数字麦克风的音频和串行接口，方便用户为每个应用灵活地选择最适合的解决方案。

本应用笔记通过描述将数字MEMS麦克风连接到STM32 MCU和MPU外设的不同方式，以及提供使用和正确配置这些外设以从麦克风采集原始数据并处理的指南，展示了STM32器件连接数字MEMS麦克风的音频功能。

6 版本历史

表16. 文档版本历史

日期	版本	变更
2017年12月14日	1	初始版本。
2019年7月17日	2	更新所有文档，将范围扩大到所有STM32器件（MCU和MPU）

表17. 中文文档版本历史

日期	版本	变更
2021年8月3日	1	中文初始版本。

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。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。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。

© 2021 STMicroelectronics - 保留所有权利

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved