

**Information Technology Institute**

**( ITI )**

**Graduation project**

**Title:-**

**- An adaptable digital hearing aid.**

**Team members:-**

**1) Hanna Nabil.**

**2) Bishoy Medhat.**

**Under the supervision of:-**

**1) Eng.\ Ahmed Al-Ashmawy.**

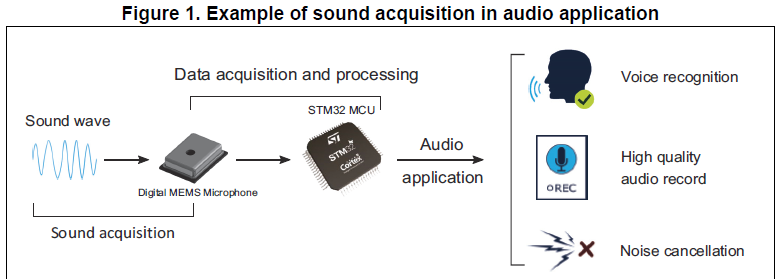
**2) Eng.\ Yousef Nofel.**

**Description:-**

**Block diagram:-**

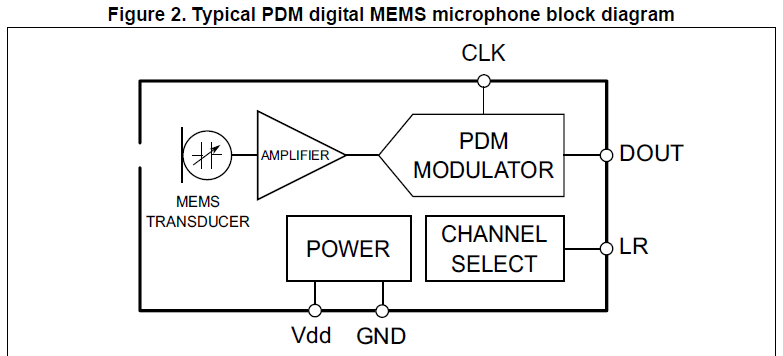
**Mems microphones:-**

- The digital MEMS microphone is a sensor that convert acoustic pressure waves into a digital signal. The STM32 microcontroller acquires digital data from the microphone(s) through particular peripherals to be processed and transformed into data standard for audio. The audio data is then handled by the microcontroller according to the targeted audio application.

****

**PDM digital microphone block diagram:-**

- The main parts in a digital microphone are a **MEMS transducer**, an **amplifier** and a **PDM modulator**.



**MEMS TRANSDUCER:-**

- The MEMS TRANSDUCER is a variable capacitance that converts the change into air pressure caused by sound waves to a voltage.

**AMPLIFIER:-**

- The AMPLIFIER buffers the voltage provided by the MEMS TRASDUCER, and provides a sufficiently strong signal to the PDM MODULATOR.

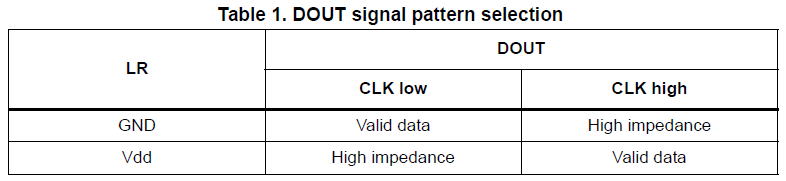
**PDM MODULATOR:-**

- PDM MODULATOR converts the buffered analog signal into a serial pulse density modulated signal. The clock input (CLK) is used to control the PDM modulator. The clock frequency range for ST digital microphones is from 1 MHz to 3.25 MHz. This frequency will define the sampling rate at which the amplifier’s analog output signal is sampled to produce a discrete-time representation (PDM bit stream).

**CHANNEL SELECT:-**

- The microphone’s output is driven to the proper level on a selected clock edge and then goes into a high impedance state for the other half of the clock cycle. The CHANNEL SELECT defines the clock edge on which the digital microphone outputs valid data. The LR pin must be connected to Vdd or GND.

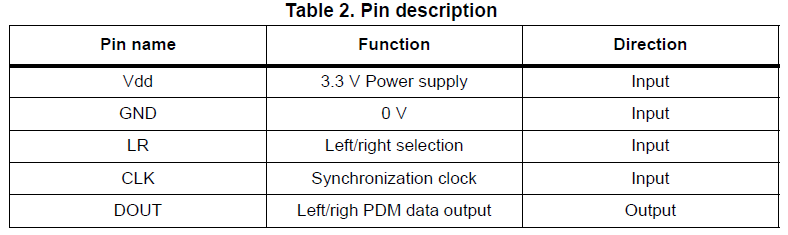
- Table 1 shows how to select the DOUT signal pattern.



**POWER:-**

- POWER delivers Vdd and GND supplies to the different digital microphone’s components. The power supply should be properly provided to the microphone since any ripple can generate noise on the output.

**Pin description:-**

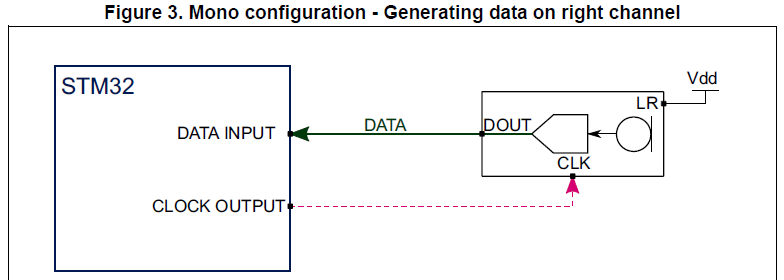


**Basic digital microphones connection:-**

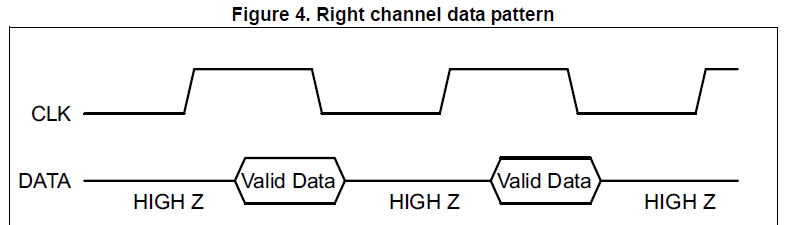
**Mono mode:-**

- In this mode LR pin can be either connected to Vdd or to GND.

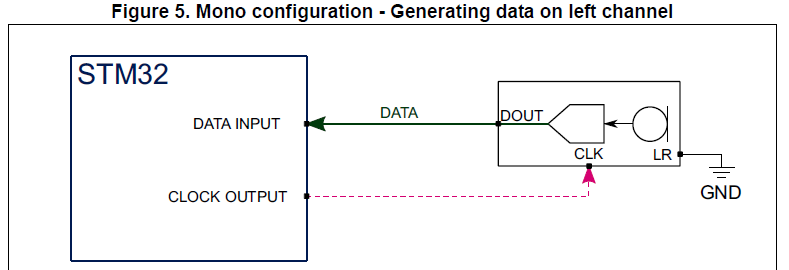
- LR pin is connected to Vdd.



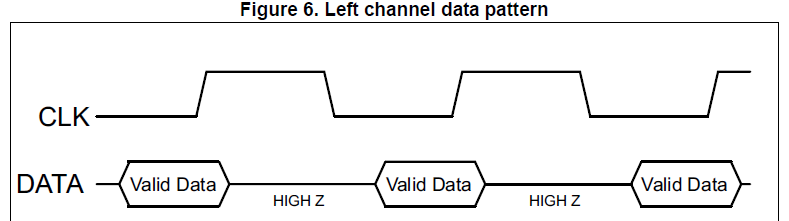
- On the rising edge of the clock, the microphone will generate valid data for half of the clock period, then goes into a high impedance state for the other half.



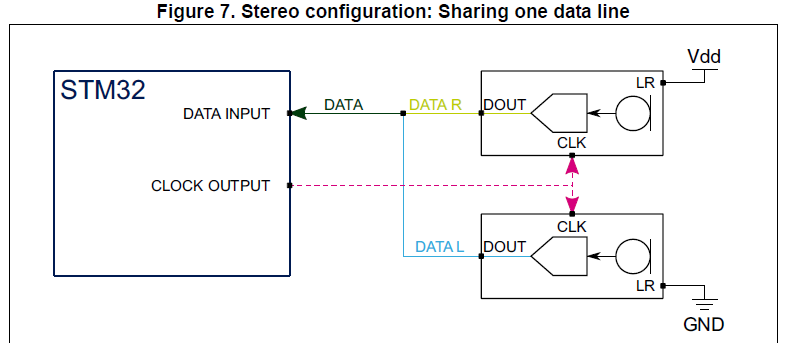
- LR pin is connected to GND.



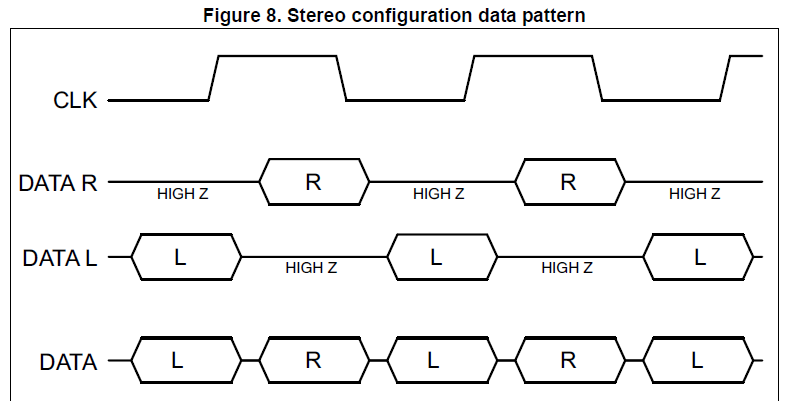
- On the falling edge of the clock, the microphone will generate valid data for half of the clock period, then goes into a high impedance state for the other half.



**Stereo configuration:-**



- Two different digital MEMS microphones are connected on the same data line, configuring the first to generate valid data on the rising edge of the clock by setting the LR pin to Vdd and the other on the falling edge by setting the LR pin to GND.

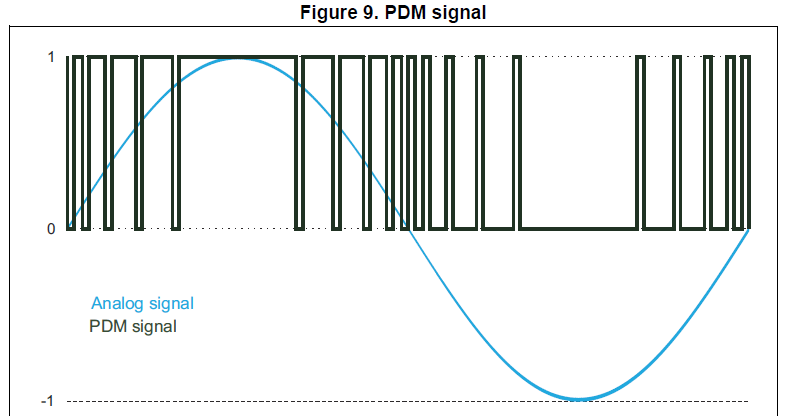


**Pulse density modulation signal (PDM):-**

- PDM is a form of modulation used to represent an analog signal in the digital domain. - It is a high frequency stream of 1-bit digital samples.

- In a PDM signal, the relative density of the pulses corresponds to the analog signal's amplitude.

- A large cluster of 1s correspond to a high (positive) amplitude value, when a large cluster of 0s would correspond to a low (negative) amplitude value, and alternating 1s and 0s would correspond to a zero amplitude value.



**Pulse code modulation signal (PCM):-**

- In the PCM signal, specific amplitude values are encoded into pulses.

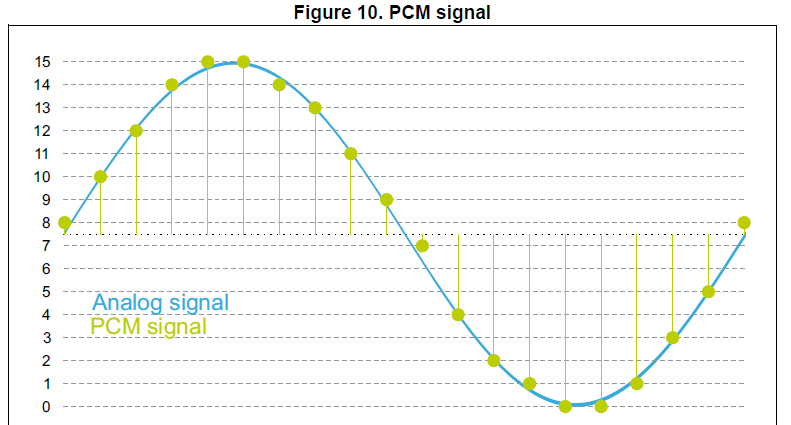
- A PCM stream has two basic properties that determine the stream's fidelity to the original analog signal:

- The sampling rate.

- The bit depth.

- The sampling rate is the number of samples of a signal that are taken per second to represent it digitally.

- The bit depth determines the number of bits of information in each sample.



**PDM to PCM conversion:-**

- In order to convert the PDM stream into PCM samples, the PDM stream needs to be filtered and decimated.

- In the decimation stage, the sampling rate of the PDM signal is reduced to the targeted audio sampling rate (16 kHz for example).

- By selecting 1 of each M samples, the sample rate is reduced by a factor of M. Therefore, the PDM data frequency (which is the frequency of the microphone clock) is M times the target audio sampling frequency needed in an application, where M is the decimation factor.

**- PDM Frequency = Audio Sampling Frequency \* Decimation Factor**

- The decimation factor is generally in the range of 48 to 128.

- The decimation stage is preceded by a low-pass filter to avoid distortion from aliasing.

**Connecting PDM digital microphones to STM32 MCUs:-**

- This section describes how to connect digital MEMS microphones to the SPI/ I2S, SAI and DFSDM peripherals embedded in STM32 microcontrollers in both mono and stereo configurations.

**Serial peripheral interface/Inter-IC sound (I2S):-**

- The STM32 microcontrollers offer a Serial Peripheral Interface block named SPI. Some of these SPI blocks also offer the possibility to use the Inter-IC Sound audio protocol (I2S).

- As we will in the next section, it is possible to connect one or two digital microphones to a SPI block by either using the SPI or I2S protocol.

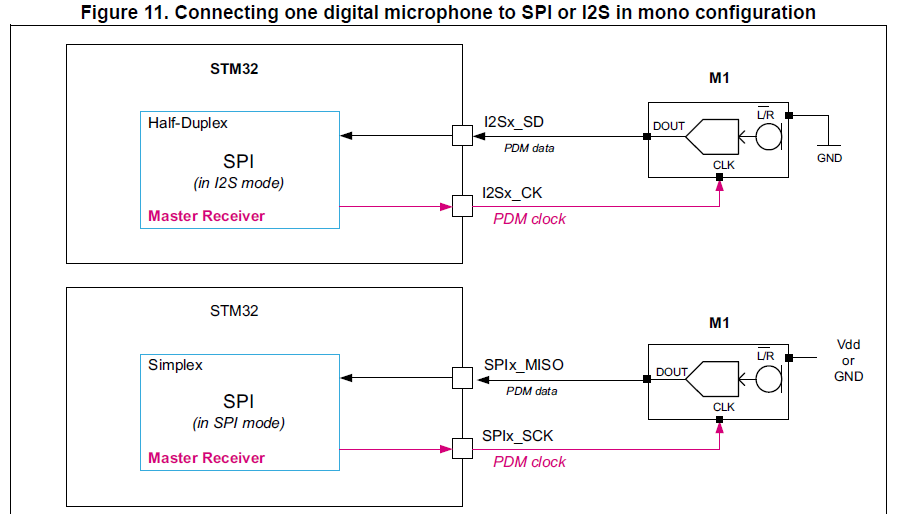
- The SPI protocol provides simple communication interface allowing the microcontrollers to communicate with external devices.

- The I2S protocol is widely used to transfer audio data from a microcontroller/DSP (Digital Signal Processor) to an audio codec, in order to play melodies or to capture sound from a microphone.

**Mono configuration:-**

- A single digital microphone is connected to the SPI block. The SPI block can be configured either in SPI or in I2S mode.

- In both cases, the SPI block is configured in master receiver mode. In this mode, the periph­eral provides the clock to the digital microphone. The audio samples are acquired through the serial data pin.



- If the SPI protocol is used, the L/R channel selection (LR) pin of the microphone can be con­nected either to Vdd or to GND. The SPI clock polarity shall be aligned with the configura­tion of L/R input.

- If L/R = GND, then the SPI shall sample the incoming data using the rising edge of SPIx\_SCK.

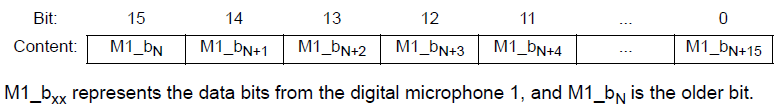
- If L/R = Vdd, then the SPI shall sample the incoming data using the falling edge of SPIx\_SCK,

- If the I2S protocol is used, it is recommended to set the L/R channel selection (LR) pin of the microphone to GND. By default the I2S protocol will sample the incoming data using the ris­ing edge of I2Sx\_CK. Note that the SPI-V2 block also offers the possibility to configure the sampling edge for the I2S protocol.

**Data format:-**

- The samples acquired by the SPI block in I2S or SPI mode can be stored into the memory using either DMA or interrupt signaling.

- The receive data register (SPIx\_DR) will provide contiguous bits from the microphone like shown in the example hereafter for a 16-bit format:



**Stereo configuration:-**

- Two digital microphones can be connected to the SPI block using a timer.

- The SPI block can be configured either in SPI or in I2S mode.

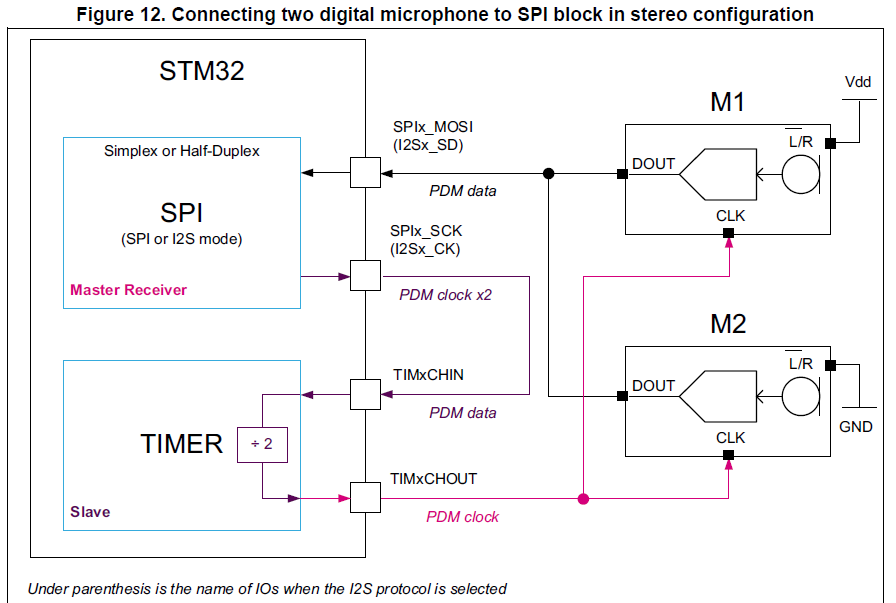
- In both cases, the SPI block is configured in master receiver mode. In this configuration, the SPI peripheral operates at twice the microphone frequency in order to read the data pro­vided by both microphones, on the falling edge of its clock.

- This allows the two microphones to share a single data line.

- The SPI block provides the clock to an embedded timer which divides the serial interface clock (SPIx\_SCK or I2Sx\_CK) by 2.

- The divided clock is delivered to the digital microphone.

- The audio samples are acquired by the I2S peripheral from the digital microphones data output pins.



**Using the timer as clock generator**

- When the timer is used to generate the clock for the digital microphones, two points have to be taken into account:

- The application must insure that the delay introduced by the clock division performed by the timer still insures a margin in the setup time (TS) of the samples provided by the microphones.

- For that purpose, the timer shall use a clock as fast as possible.

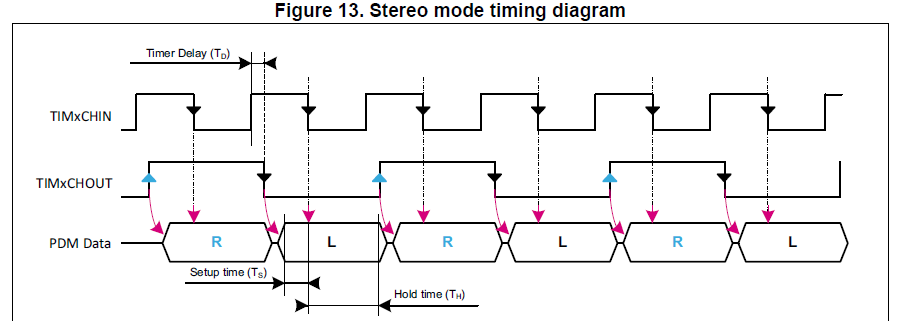
- The maximum delay (TD) introduced by the timer between the input (TIMxCHIN) and the output clock (TIMxCHOUT) will be 5 clock cycles of the timer reference clock.

- The timers generally use their APB clock or a multiple of their APB clock as reference. See Figure 13.

- The application must insure that the peripheral providing the clock to TIMxCHIN input and the timer used for the division, are working with the same reference clock.

- If this rule is not respected, then from time to time the digital microphone will receive a clock having a longer or shorter period.

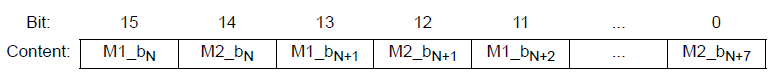
- This jitter may degrade the quality of the analog to digital conversion of the microphone.



**Data format:-**

- The samples acquired by the SPI block in I2S or SPI mode can be stored into the memory using either DMA or interrupt signaling.

- In this configuration, the data read from the microphones are interleaved bit per bit. The data stored into the SPIx\_DR register will be interleaved as shown in the example hereafter for 16-bit format:



- M1\_bxx represents the data bits from the digital microphone 1, and M1\_bN is the older bit.

- M2\_bxx represents the data bits from the digital microphone 2, and M2\_bN is the older bit.

**Digital signal processing:-**

- This section presents two ways to convert PDM data into PCM data:

- The first is a software solution which is the PDM audio software decoding library and the second is hardware solution using the DFSDM peripheral filters.

**PDM audio software decoding Library:-**

**Overview:-**

- PDM audio software decoding Library is an optimized software implementation for PDM signal decoding and audio signal reconstruction when connecting digital MEMS microphones with an STM32 microcontroller.

- This library implements several filters for the 1-bit PDM high frequency signal output from a digital microphone and transforms it into a 16-bit PCM at a proper audio frequency.

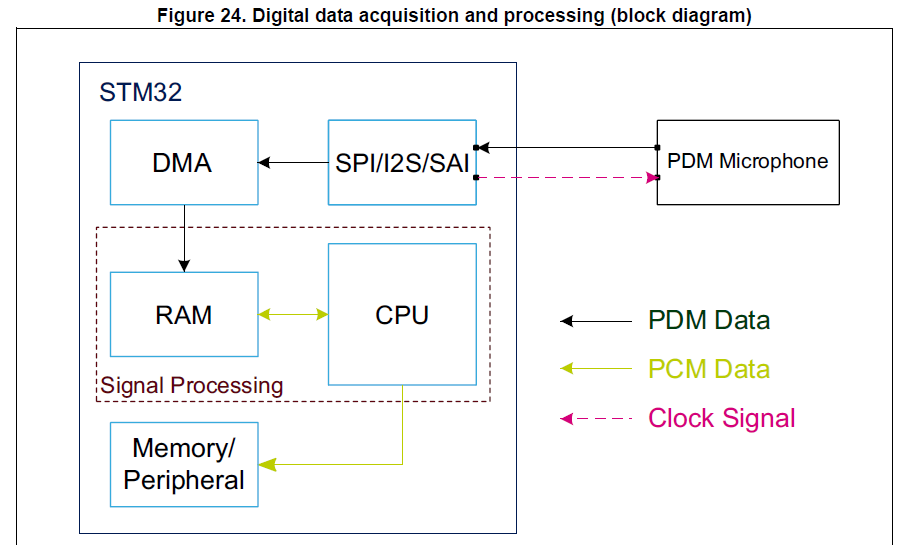
**Digital data flow:-**

- The digital MEMS microphone outputs a PDM signal, which is a high frequency (1 to 3.25 MHz) stream of 1-bit digital samples.

- The PDM data is acquired by a serial interface embedded in the STM32.

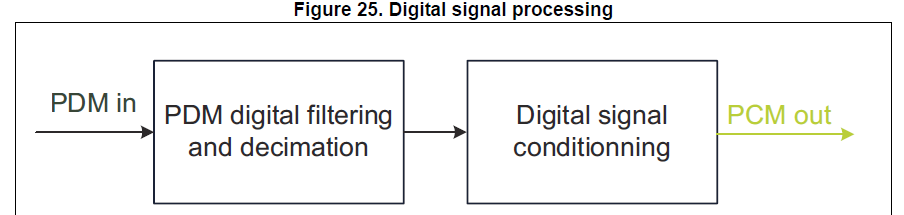
- This data is transferred through DMA (thus reducing the software overhead) to a system RAM buffer to be processed.

- After the conversion, the PCM raw data can be handled depending on the application implementation (stored as wave/compressed data in a mass storage media, transferred to an external audio codec DAC...).



**Digital signal processing:-**

- The PDM audio software decoding Library offers a two steps digital signal processing: - PDM digital filtering and decimation and Digital signal conditioning.



- On the first step, the PDM signal from the microphone is filtered and decimated in order to obtain a sound signal at the required frequency and resolution.

- On the second step, the digital audio signal resulting from the previous filter pipeline is fur­ther processed for proper signal conditioning implementing a low pass filter and a high pass filter.

- Both these filters can be enabled/disabled and configured (cut-off frequencies) by using the filter initialization function.

**DFSDM filters for digital signal processing:-**

**Digital data flow: acquisition and processing:-**

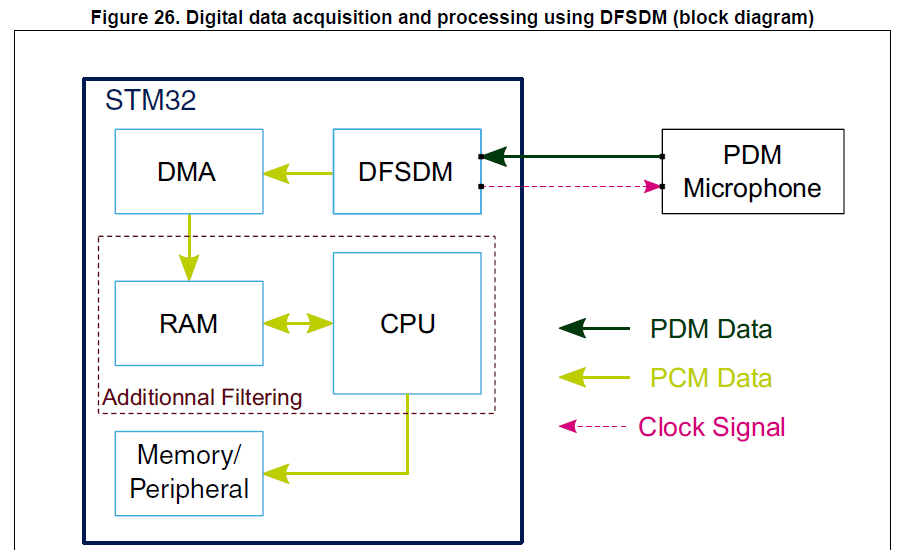
- The digital MEMS microphone outputs a PDM signal, which is a high frequency (1 to 3.25 MHz) stream of 1-bit digital samples.

- The data is acquired by the DFSDM serial transceiver that provides connection to the external Sigma-Delta modulator of the digital microphone.

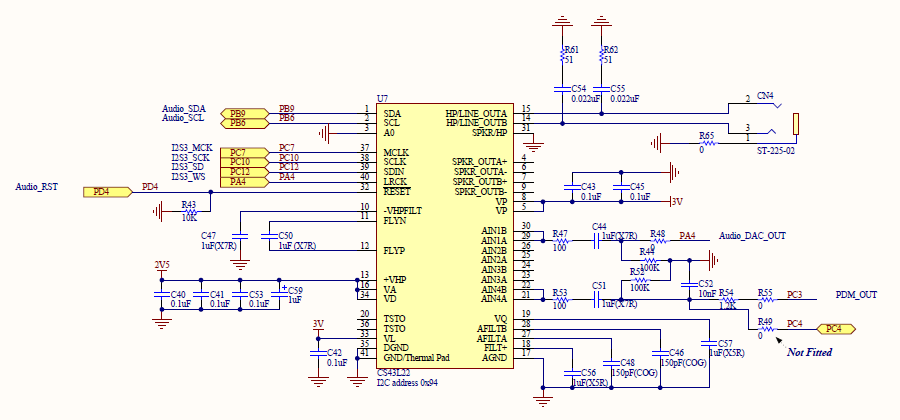
- The digital filters perform CPU-free filtering that averages the 1-bit input data stream from the SD modulator into a higher resolution and a lower sample rate.

- This data is transferred through DMA (thus reducing the software overhead) to a system RAM buffer to be further fil­tered.

- After that, the PCM raw data can be handled depending on the application implemen­tation (stored as wave/compressed data in a mass storage media, transferred to an external audio codec DAC...).



**Audio codec:-**



**Audio Bluetooth module:-**

