AN5795

Application note

Sound capture with multifunction digital filter on STM32U5 Series

Introduction

The MDF (multifunction digital filter) is a high-performance peripheral dedicated to sample acquisition, which is available in STM32U5 Series microcontrollers. It is of particular interest for audio and speech capture, or any application that provides a digital signal that needs to be filtered and decimated, such as motor control and metering.

Although the MDF is a pure digital peripheral, it is designed to support a wide range of external analog front-ends, and in particular sigma-delta (∑∆) modulators. By means of the external interface offered by the MDF, the user can choose an analog front-end part and specifications to suit the application. The MDF is low-power oriented, providing a low-speed clock to the modulator.

The MDF processes the digital stream with various configurations to support user requirements, such as output data rate, output data width, and frequency range that can be adjusted as a function of the input data.

The MDF also offers some extra options such as out-off limit detection and offset error compensation, giving more specific control to the user.

1 General information

This document applies to STM32 Arm®-based microcontrollers.

arm

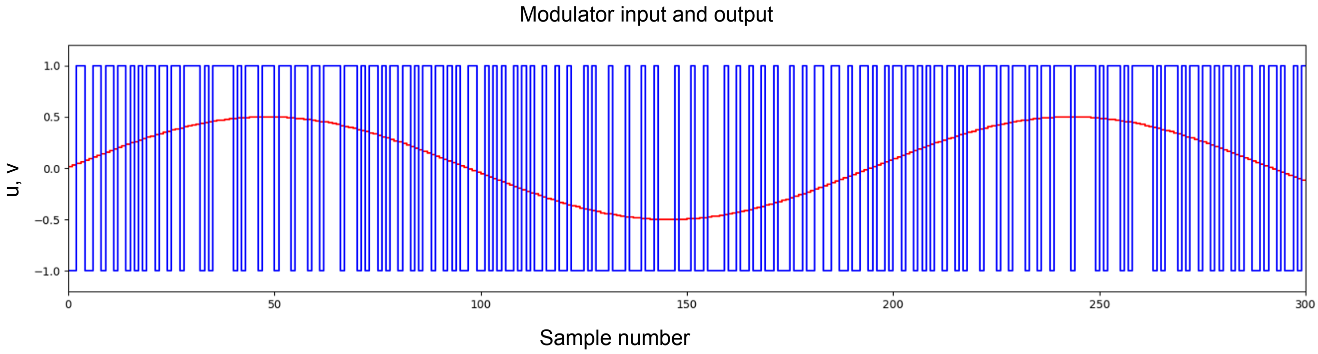
Note: Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Multirate filter basics

2.1 Pulse density modulation (PDM)

The MDF performs a process on digital data provided by an external ∑∆ modulator, which converts the analog signal into a digital 1-bit stream called PDM (pulse density modulation). The PDM stream is a high sampling-rate serial line where the analog signal is converted in a stream of digital ones and zeroes, depending on the amplitude of the analog signal.

Figure 1. PDM of an analog signal



Note: For analysis, the digital stream is converted from binary 0 and 1 weight, into +1 and -1 weights.

The PDM is obtained by a ∑∆ modulator, which consists in a 1-bit analog-to-digital (A/D) converter that digitizes the input analog data into a serial digital data stream. Simply converting the analog signal into a 1-bit stream generates an important uniform quantization noise.

The PDM modulators use mainly the two following tricks to improve the A/D conversion:

- Sample the input signal at a frequency rate significantly higher than requested. For example, if the signal to be converted into digital has a useful band of B, the PDM modulators sample this signal at k.B where k is a number significantly bigger than 2 (for example 32, 64, or 128).
- The PDM modulators reshape the quantization noise to keep the noise level as low as possible in the useful band B, and “push” the noise energy outside the useful band.

Note: The output of ∑∆ modulator can be multibit. This document focuses on a 1-bit A/D converter, which is the most frequent case.

2.2 Interest of PDM filtering

As the PDM stream contains the useful signal with low noise and the out-of-band signal (which is noisier), the filtering is necessary to reduce the noise in out-of-band while keeping a low attenuation in the passband.

The main MDF function is to filter the out-of-band noise, and to reduce the sampling rate (decimate). The MDF processing consists, in a first way, in averaging a fast-rate-input serial stream into a parallel, lower-rate, and higher-resolution data output (usually 12 to 24 bits). The digital filter also removes out-of-band frequency components (for example: quantization noise or unwanted signal), and reduces the data rate by decimation.

The filter implementation has a strong influence on data output resolution and quality. It results of a compromise between filter performance (such as sharpness of filter, filter tuning, or final resolution), and hardware implementation in terms of area, which leads with cost issue and power consumption.

2.2.1 Multirate filter interest

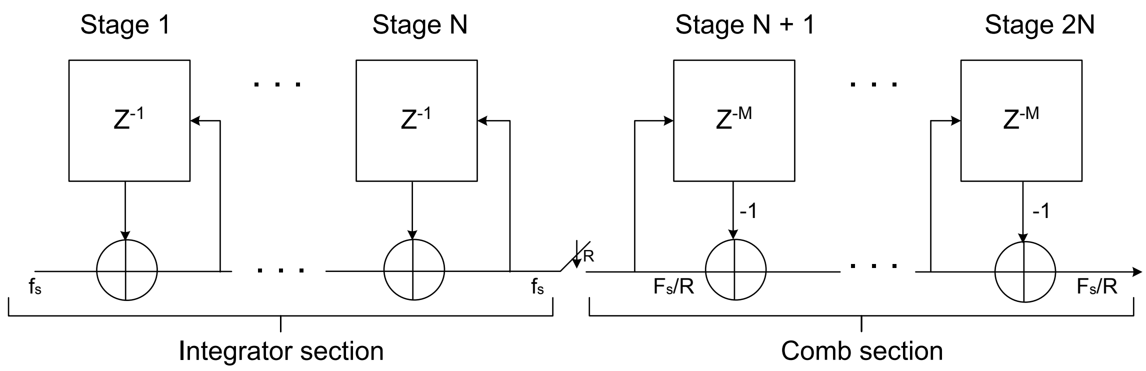
Performing a low-pass filtering operation with decimation, while keeping a very-low noise level in the useful band (with a useful band as flat as possible), and a strong rejection of aliasing, is hardly achievable with a single filter. Performing the decimation in several steps is more efficient and easier to implement. Cascading filters working at different rates is called multirate filters. Multirate stands for having different sampling rates at each stage of filtering operations. In the case of sound capture, the aim of this class of filters is to reduce the sampling rate, which significantly minimizes the number of operations required to perform a classic FIR (finite impulse response). Design constraints, power consumption, and filter latency are then highly improved.

To reach a lower rate, a decimation is performed. Decimating the sampling rate by an integer of M means to discard every M-1 sample or equivalently, keeping every M^{th} sample. The decimation must be performed after a low-pass filtering.

2.2.2 CIC filter characteristics

The CIC (cascaded integrator-comb) filter is one of the most popular classes of decimation filter, with an integrator section, which, consists of N ideal digital integrator stages operating at f_S. This integrator section is followed by a comb section operating at a lower sampling rate f_S / R, where R is the integer rate change factor. This part is designed with an N comb stage with a differential delay of M samples per stage.

Figure 2. **CIC filter structure**



The following equations give the transfer function of an N order:

- for a single integrator

$$H_I(z) = \frac{1}{1 - z^{-1}}$$

- for a single comb stage

$$H_C(z) = 1 - z^{-RM}$$

Then

$$H(z) = H_I^N(z) \times H_C^N(z) = \frac{\left(1 - z^{-RM}\right)^N}{\left(1 - z^{-1}\right)^N} = \left[\sum_{k=0}^{RM-1} z^{-k}\right]$$

Where N is the order of the CIC filter.

The gain G_{CIC} and the output data size DS_{CIC} can be expressed as follows versus CIC parameters:

$$G_{CIC} = R^N$$

$$DS_{CIC} = \frac{N \times \ln R}{\ln 2} + DS_{IN}$$

Where DS_{IN} = 1 for a PDM stream, and ln is the napierian logarithm.

The CIC filter also acts as a moving average filter (low-pass filter equivalent). Once the low-pass filter limited the signal band, the sampling rate can be reduced, avoiding aliasing. The greater the order of the CIC filter, the greater the attenuation outside the useful band. The aliasing rejection increases as the decimation rate increases.

The CIC filter does not have a flat response in the passband, a low rejection between f_S / 2 and f_S.
To compensate these drawbacks, the decimation stage is followed by an IIR (infinite impulse response) filter to improve the attenuation of the stop band, to preserve the flatness of the useful band, and to compensate the ripple of the in-band signal.

The CIC is compared to a FIR filter means (linear response and constant group delay avoiding a signal distortion).

Note: *The CIC output data size must not exceed 26 bits.*

2.3 MDF filter chain

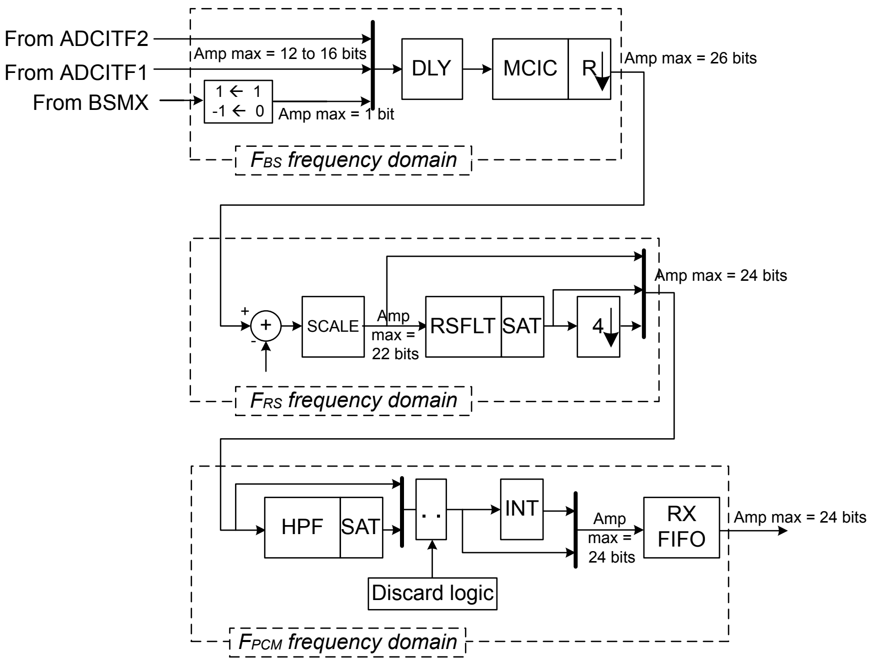
According to its configuration, the MDF embeds several instances of the filter chain (DFLTx). The figure below shows a simplified view of elements potentially used for audio applications. The main components are:

- a symbol remapper (SBR)
- a delay bloc (DLY)
- a fourth or fifth order CIC (MCIC)
- a reshape filter (RSFLT)
- a high-pass filter (HPF)
- a discard block
- an integrator (INT)

Succession of filters (such as MCIC, RSFLT, and HPF) can be used to reach the highest resolution possible.

Important: *The ADF embeds the same chain filter as the MDF. All elements presented in this document are also applicable to the ADF, including the configuration proposals.*

Figure 3. DFLTx filter chain



The first CIC stage has already been presented is the previous section, and acts as the first decimation filter stage where the data output size is known. To maximize the efficiency of the RSFLT and to avoid a saturated output, the input size must be 22 bits. As CIC output is between 3 and 26 bits, the scale block performs an amplitude adjustment by decreasing the input signal up to 8 bits (-48.2 dB), or increasing it up to 12 bits (90 dB) by 3 dB steps (±0.5 dB).

The gain adjustment policy is defined as below:

1. Check that the data size from the CIC filter does not exceed 26 bits. This can be checked using this formula:

$$\frac{\ln\left(SIN_{pp} \times R^N\right)}{\ln 2} < 26$$

   where N represents the CIC order, R the decimation ratio, and SIN_{pp} the maximum peak-to-peak amplitude of the input signal. For a PDM stream, the maximum peak-to-peak amplitude is equal of two (±1).
   If the peak-to-peak amplitude never exceeds 0.5, then SIN_{pp} can be one (±0.5).

2. Adjust the scale value in line with the RSFLT use by this formula:

$$SCALE\ (dB) < 20 \times \log_{10}\left(\frac{2^{NB}}{SIN_{pp} \times R^N}\right)$$

   where NB is equal to 22 if the RSFLT is enabled, or 24 if the RSFLT is bypassed.

Note: After the scale block, the signal is saturated at a maximum of 24 bits. When the RSFLT is used, the SCALE value must not exceed 22 bits. Otherwise, it can be up to 24 bits.

The RSFLT is highly recommended for audio application. It is designed as an IIR filter of seventh order. The RSFLT is used to improve the attenuation of the stop band, and to preserve the ripple in an inband signal. The RSFLT cutoff frequency F_C is equal to 0.111 × F_{RS}, where FRS is the RSFLT input sampling rate defined by $F_{RS} = F_{BS} / R$ at CIC output. The F_{PCM} (RSFLT output frequency) can be decimated by four.

The computation ended with an HPF operating at F_{PCM} suppresses the DC component introduced by a parasitic low-frequency noise in the input data source in continuous conversion mode. The HPF is a first order IIR where the cutoff frequency can take the following values:

- 0.000625 x F_{PCM}
- 0.00125 x F_{PCM}
- 0.00250 x F_{PCM}
- 0.00950 x F_{PCM}

The HPF output is saturated at 24 bits.

The HPF activation is highly recommended: audio is conveyed electrically as an alternating current signal. The signal is not necessarily symmetrical about the 0 V line. The preamplifier and the microphone A/D converter induce a voltage offset (called DC offset). This HPF can directly follow the CIC output even if the RSFLT is bypassed.

2.4 CIC frequency response and noise aliasing

The following formula gives the CIC frequency response:

$$|H(f)| = \left|\frac{\sin(\pi \times M \times f)}{\sin\left(\frac{\pi \times f}{R}\right)}\right|^N$$

Where

- N is the filter order.
- R is the decimation ratio.
- M is the delay.
- f is the frequency relative to the low-sampling rate f_S / R.

This response highlights that the output spectrum has nulls at multiples of f = 1 / RM.

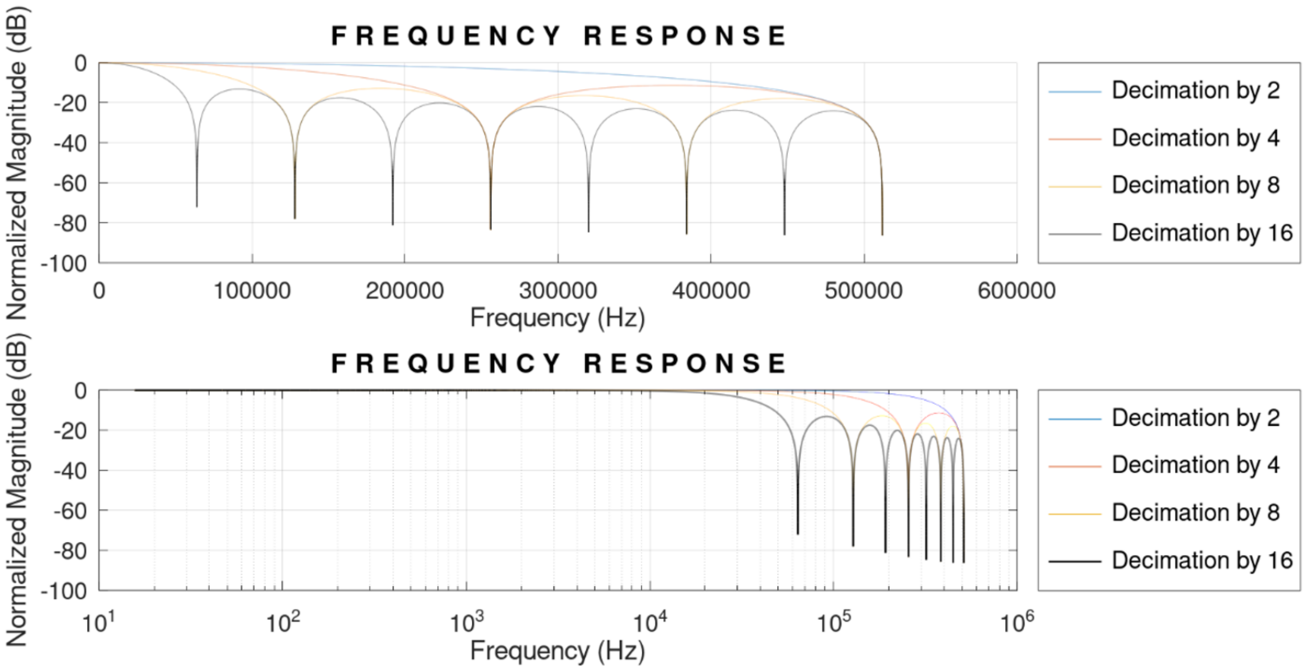
Note: In this document, the analysis focuses on N and R (with M = 1).

2.4.1 CIC transfer function

The figure below shows the transfer function of a first-order CIC filter, for different decimation ratios, in the case of an incoming stream sampled at 1.024 MHz. The attenuation of the high-frequency part increases with the decimation ratio.

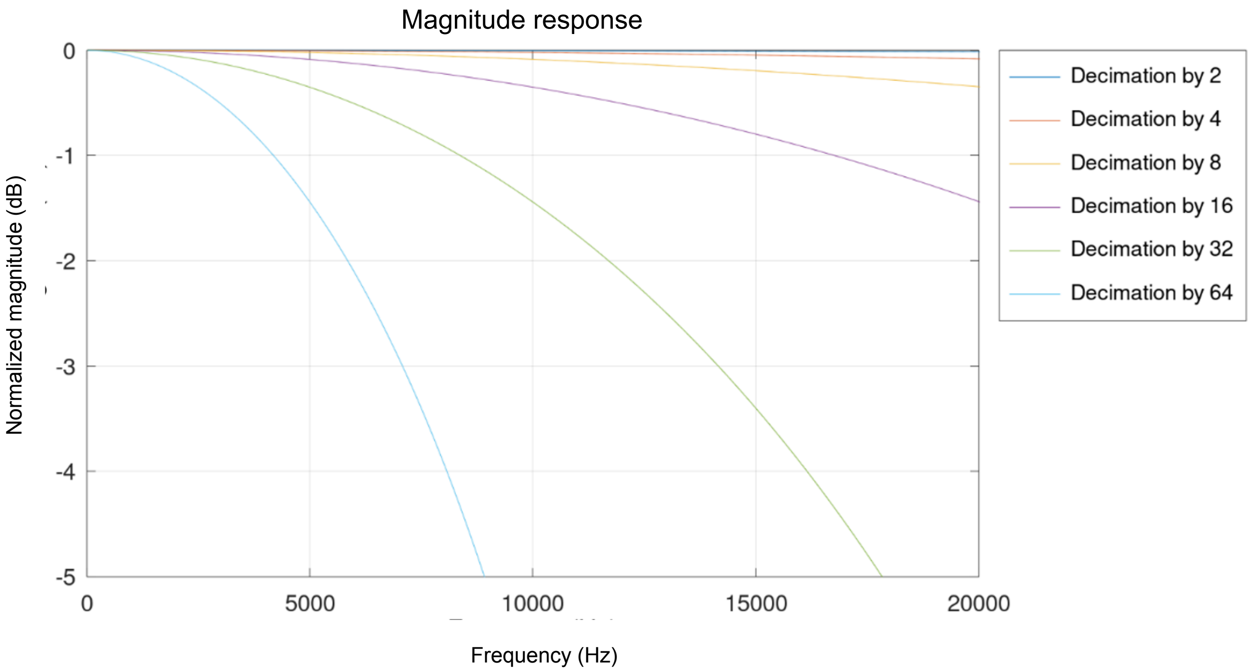
If the CIC is used to convert a signal provided by a ∑Δ modulator, increasing the decimation ratio R reduces the amount of out-of-band noise folded into the useful bandwidth. An increased R also decreases the useful band. The number of zeroes depends on R, following the law previously mentioned.

Figure 4. **CIC frequency response and position of nulls vs the decimation ratio**



The figure below shows the frequency response of the 20 kHz bandwidth. The decimation process is not flat on the inband signal. Higher the decimation ratio is, the more important the passband droop is. For instance, at 8 kHz, the attenuation is about -5 dB for a decimation by 64, against 1.2 dB for a ratio of 32.

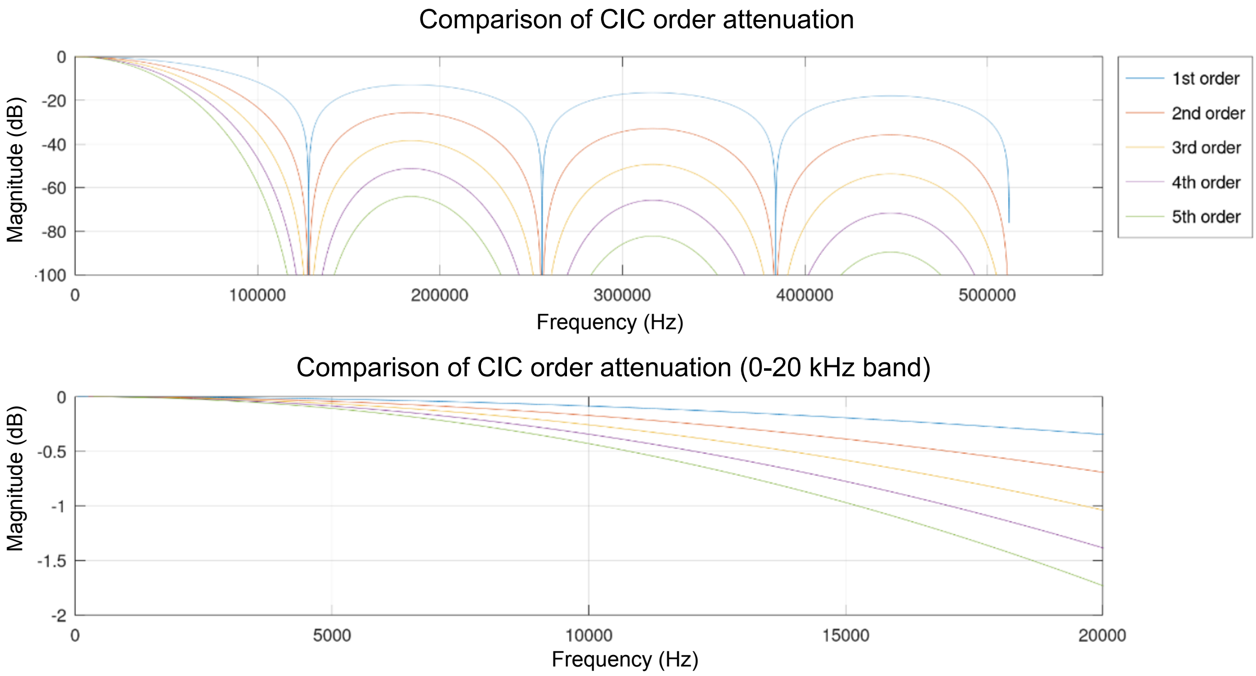
Figure 5. Droop gain in the passband for multiple decimation ratios



2.4.2 CIC order effect on frequency spectrum

The CIC frequency response depends on the decimation ratio R and the filter order N. The figure below gives the frequency response at R = 8 and N = 1 to 5. By incrementing one order, the side lobes are attenuated about an additional 13 dB. The flat part of the main lobe is also reduced when the order increases (see the last part of the figure).

Figure 6. Inband and outband attenuation vs CIC order



2.4.3 Aliasing and folding of CIC decimation stage

Figure 7 explains the way that the decimation works on CIC filters. To simplify the drawing, the CIC filter performs a decimation by four, the input data are sampled at F_S, and the CIC output delivers samples at a rate of F_S / 4.

On this figure:

- Light-grey boxes represent the useful bandwidth that the application wants to preserve (±B).
- Plot (I) shows the CIC transfer function.
- Plot (II) shows a simplified view of the spectrum of a digital microphone signal injected into the CIC input.
- MF(f) in Plot III shows the microphone signal filtered by the CIC (without considering the decimation).

  The CIC reduces the out-off band noise and preserves the useful band.
- On plot (IV), the F_D spectrum is ideally a Dirac comb. All Dirac peaks are spaced by the new sampling rate (F_S / 4)

  Performing the decimation can be seen as resampling the resulting spectrum MF(f) by the new sampling rate F_D.

$$F_D(f) = \sum_{k=-\infty}^{\infty} \delta\left(f - k\frac{F_S}{4}\right)$$

- Plot V shows the result of the convolution of the MF(f) spectrum with the F_D(f) spectrum.

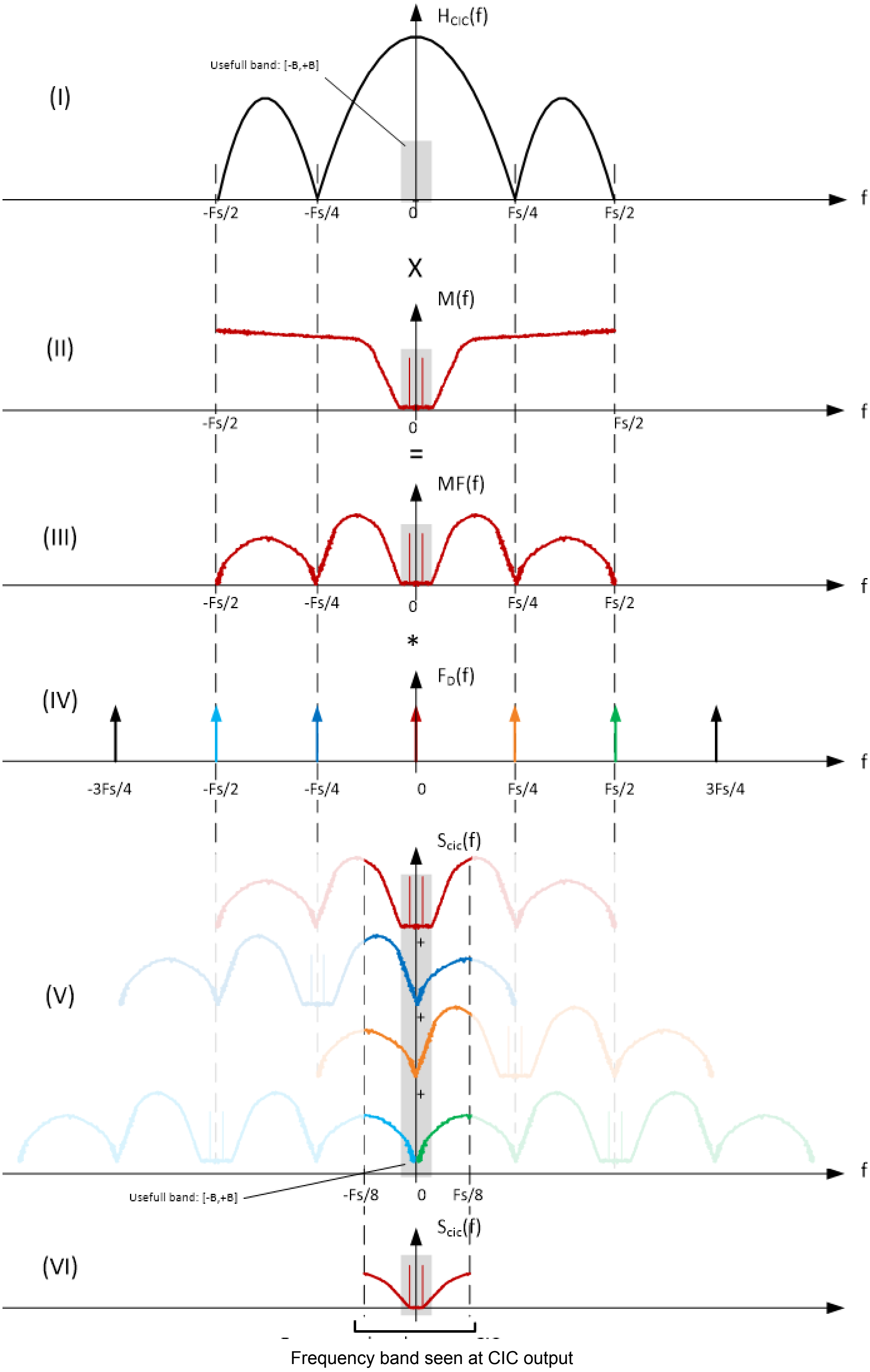
  Sampling a signal in the time domain means to convolve the MF(f) spectrum with the sampling rate (F_D(f)) spectrum. Convolving a signal with a Dirac is equivalent to perform a frequency translation, as given in the following formula:

$$MF(f) \times \delta(f-a) = MF \times (f-a)$$

  The convolution result is a superposition of MF(f), translated around 0, ±F_S / 4, ±F_S / 2. A part of the out-off band noise is ‘folded’ into the useful band (for example±F_S / 4 or ±B).
- Plot VI: the amount of out-off band noise ‘folded’ into the useful band is minimized as the folded parts are the regions where the CIC has zeroes (maximum noise attenuation).

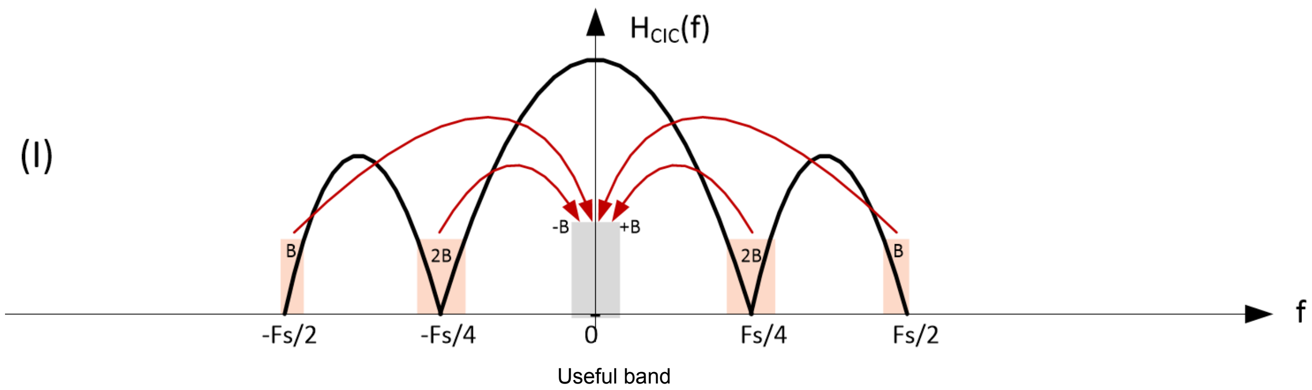
  The narrower B is (versus F_S / 8), the less noise is folded.

Figure 7. Folding of replicas into the passband



The figure below illustrates in a simpler way, the decimation performed by the CIC: red and orange boxes represent which part of the CIC transfer function is folded into the useful band.

Figure 8. Summary of CIC folding principle



2.5 RSFLT frequency response

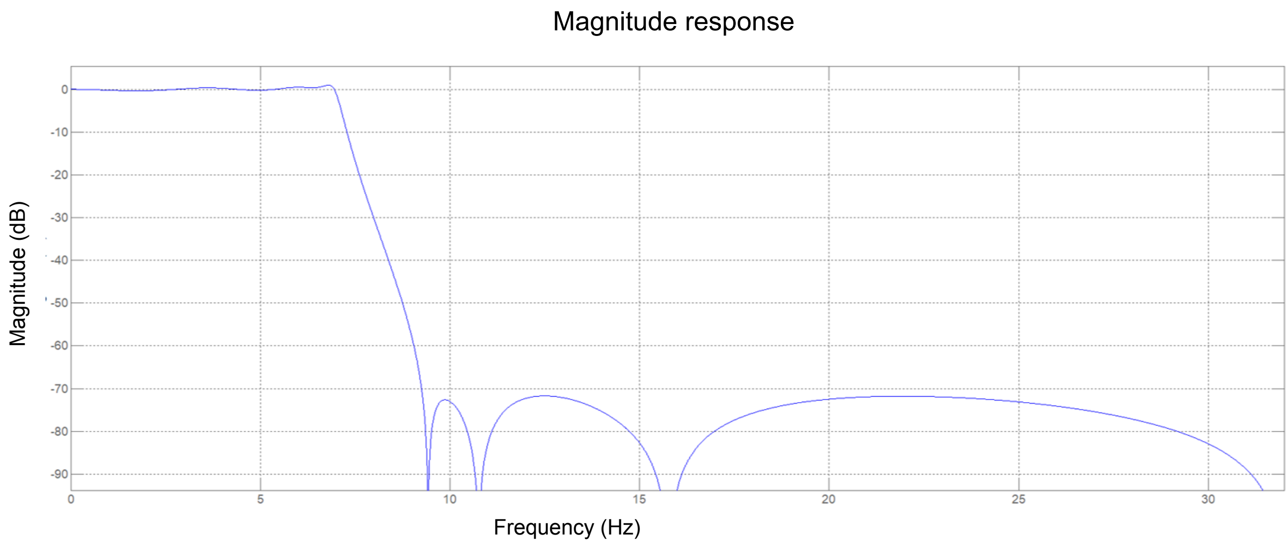
The figure below shows the normalized RSFLT frequency response for a sample rate at 64 kHz. This response can be extended to any sample rate.

Only the cutoff frequency moves with the sample rate at its input according to the table below.

Table 1. Passband versus sample rate

Sample rate (kHz) at RSFLT (F_{RS})	Passband (kHz)	Decimation ratio	PCM sampling rate (kHz)
32	3.55	4	8
64	7.1		16
128	14.2		32
192	21.3		48

Figure 9. RSFLT frequency response at F_{RS} = 64 kHz



The RSFLT has a steep transition band and an out-of-band rejection of 72 dB to significantly attenuate the remaining quantization noise, not suppressed by the CIC, while maintaining an acceptable in-band ripple of ±0.65 dB. By cascading the CIC filter and this one, the inband ripple is reduced to ±0.42 dB.

The implemented RSFLT has about 9.3 dB gain introduced to avoid some SNR degradation.

As an optional decimation by four follows this filter, the application can perform extra processing at F_{RS} rate, if needed.

3 MDF configuration examples

3.1 Low-power and performance use case

The MDF efficiency depends primarily on the microphone input signal, and the targeted power performance. The following use cases focus on some classical configurations, which can be used for very-low-power and high-performance applications. The frequency response and some performance measurements are presented for each use case.

Even if the application can decrease the MDF kernel clock and bypass some chain filter parts, the microphone sampling frequency is the key point for low-power. Some digital microphones have a specific range of sampling frequency or working mode linked to their power consumption.

The following modes are usually provided:

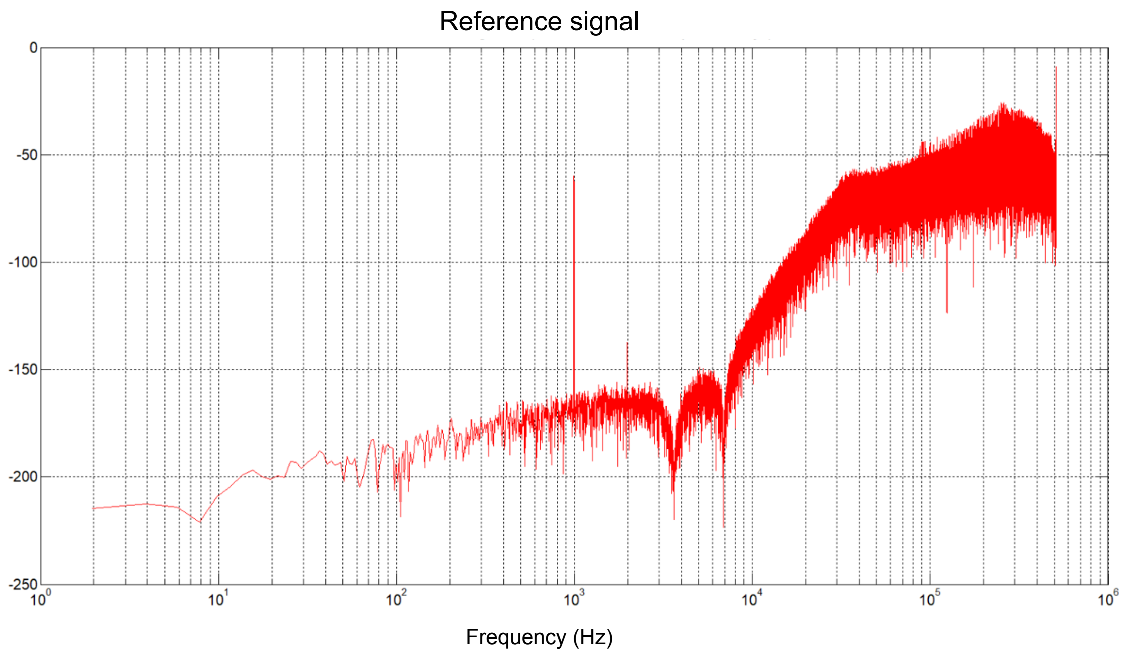
- standby mode: no clock or very-low frequency clock provided to the microphone
  The microphone does not work. Its consumption is reduced to few µA.
- low-power mode: sampling frequency between 350 kHz and 800 kHz, power consumption from 180 µA to 330 µA
- normal or performance mode: sampling frequency from 1 MHz to 3.3 MHz, power consumption from 400 µA to 1 mA

According to the sampling frequency, the MDF must be set to fit with the desired PCM frequency and the best reachable signal-to-noise ratio (SNR).

Four use cases are detailed in the next section with the recommended setup. Measurements have been performed on STM32U5 device, using a five-order ∑∆ modulator, with the microphone noise emulated by a noise shaper (see the frequency response of this model in the figure below).

Note: This noise shaper contains properly the noise in the useful band, for decimation ratios higher or equal to 64. For lower decimation ratio, the SNR in the useful band is degraded. The SNR measurements performed at MDF outputs are then also mechanically degraded.

Figure 10. Input signal spectrum from the emulated sigma-delta modulator



The goal is to show the MDF filter performances, and to avoid external elements interfering with these measurements. For the next measurements, a 1 kHz sine wave is used.

3.2 Full hardware configuration

3.2.1 Configuration 1: audio and voice detection

This very-low-power configuration is recommended for a sound-detection application, focusing on power consumption whatever the SNR. The bitstream frequency range of the microphone is between 350 kHz and 800 kHz. To reach a high SNR even if the microphone runs in low-power mode, the decimation ratio R needs to be as high as possible.

The maximum R for a CIC5 is 32 for a full-scale input signal. This R can be increased if the amplitude of the input signal is smaller. Using a CIC4 with a higher R offers a better aliasing rejection, but increases the attenuation in the higher part of the useful band.

Selecting CIC5, CIC4, and R is a trade-off between the useful band integrity (attenuation of high-frequency components), and the noise rejection.

The R can also be increased (work on a reduced PCM band) depending on the required band needed by the application. For example, key-word spotting may request a signal quality not requested by a simple-voice activity detection, or sound-activity detection.

Note: In low-power mode, microphones deliver a signal with a limited SNR.

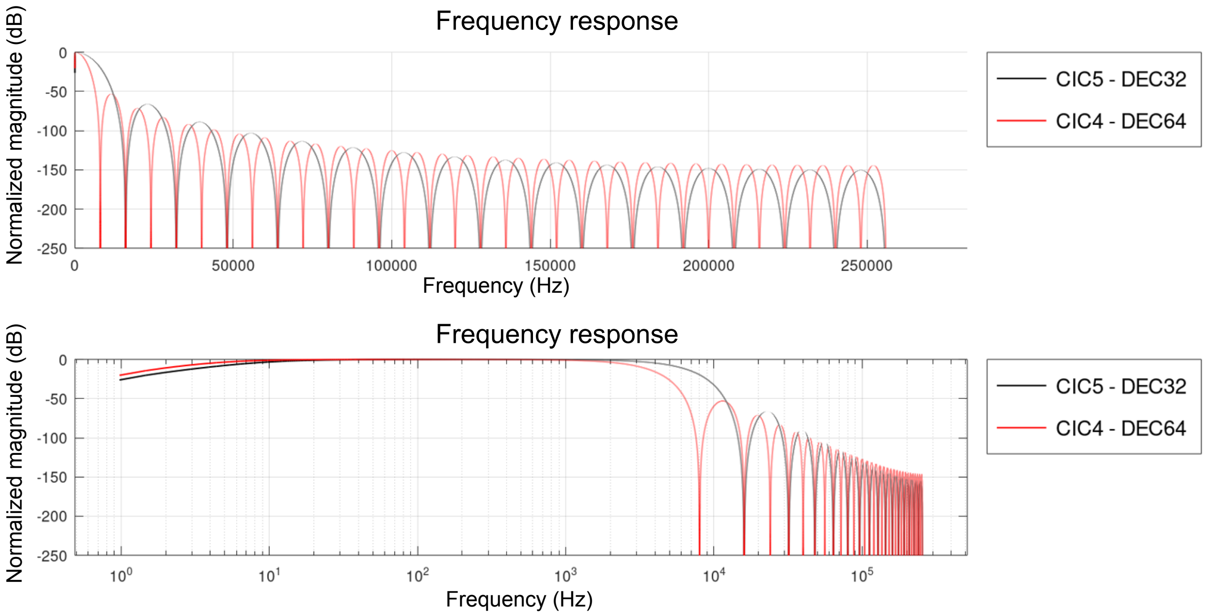
There is no RSFLT used to reduce as much as possible the frequency of the kernel clock.

Table 2. Configuration 1 recommended settings

Parameter	CIC5 settings	CIC4 settings
CIC mode	$Sinc^5$	$Sinc^4$
CIC decimation mode	32	64
Gain adjustment	0x2B (-14.5 dB)	0x2F (-2.5 dB)
RSFLT state	Disable	
HPF state	Enable	
Total decimation ratio	32	64
PCM rate	16 kHz	8 kHz
Input sample rate	512 kHz	

The theoretical frequency responses of these setups are depicted in the figure below.

Figure 11. Theoretical frequency response for Sinc^5 and Sinc^4 configuration 1



Due to the high CIC order and decimation rate, the attenuation at 8kHz is about -18.9 dB with a decimation ratio of 64. In this specific use case, the noise rejection can be improved by switching to a CIC4 and a higher decimation ratio (here 64). The PCM sampling rate is half of the CIC5 setup since the PDM frequency remains the same. The in-band attenuation is a bit higher due to this decimation rate. The -3 dB cutoff frequency gain is reached at 3400 Hz for CIC5 against 2066 Hz in CIC4 mode. This concession improves the SNR by 13 dB on their respective band. The noise folds differently between these two configurations: comparing them withing a single-frequency band is not appropriate.

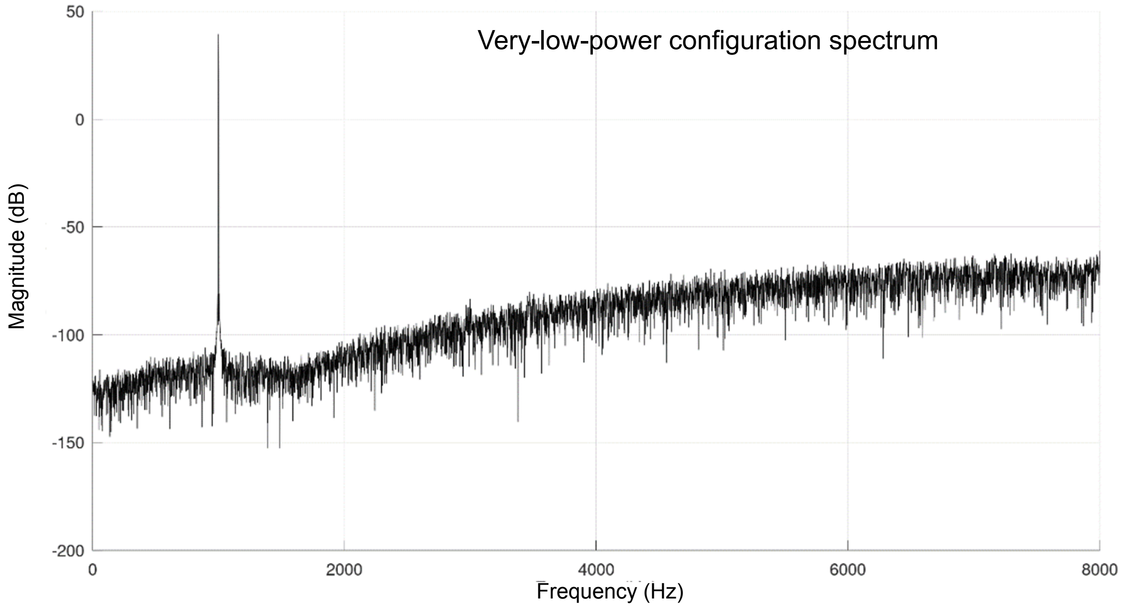
Important: *With these configurations, the rejection of frequencies between F_{PCM} / 2 and F_{PCM} is poor.*

The SNR obtained for CIC5-DEC32 depends a lot on the noise-shaper model. The current noise-shaper model gives an optimal SNR when the decimation ratio is bigger or equal to 64, else there is a degradation of SNR.

Table 3. Configuration 1 measurements

Parameter	CIC5-DEC32	CIC4-DEC64
PCM frequency	16 kHz	8 kHz
SNR	78.787 dB	91.855 dB
THD + N	78.733 dB	91.845 dB
-3 dB cutoff frequency	3400 Hz	2066 Hz

Figure 12. **Output spectrum of configuration 1 with Sinc^5 mode**



3.2.2 Configuration 2: very-low-power configuration

This configuration is made for an audio capture for low-frequency PDM stream (PDM frequency ≤ 512 kHz). The filter chain is composed of the CIC decimator, and the scale block to adjust the input level for the RSFLT (which is enabled in this configuration). At the end, the high-pass filter is always enabled to remove the DC component. The PCM stream at 16 kHz is obtained through two decimation stages to reach a total decimation of 32. The CIC first decimates by eight, followed by the integrated decimation ratio by four of the reshape filter.

Table 4. **Configuration 2 recommended settings**

Parameter	Settings
CIC mode	Sinc^5
CIC decimation mode	8
Gain adjustment	0x0C (36.1 dB)
RSFLT state	Enable
HPF state	Enable
Total decimation ratio	32
PCM rate	16 kHz
Input sample rate	512 kHz

Figure 13. **Theoretical frequency response of configuration 2**

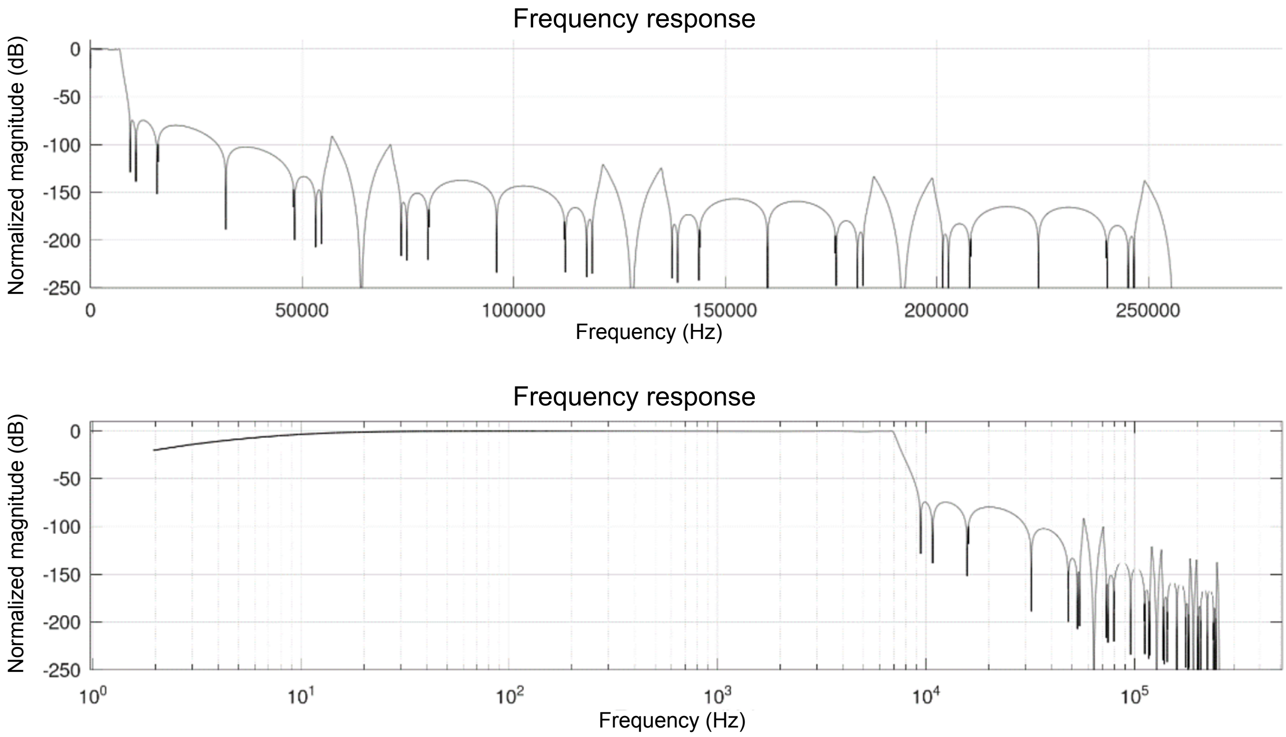


Table 5. **Configuration 2 measurements**

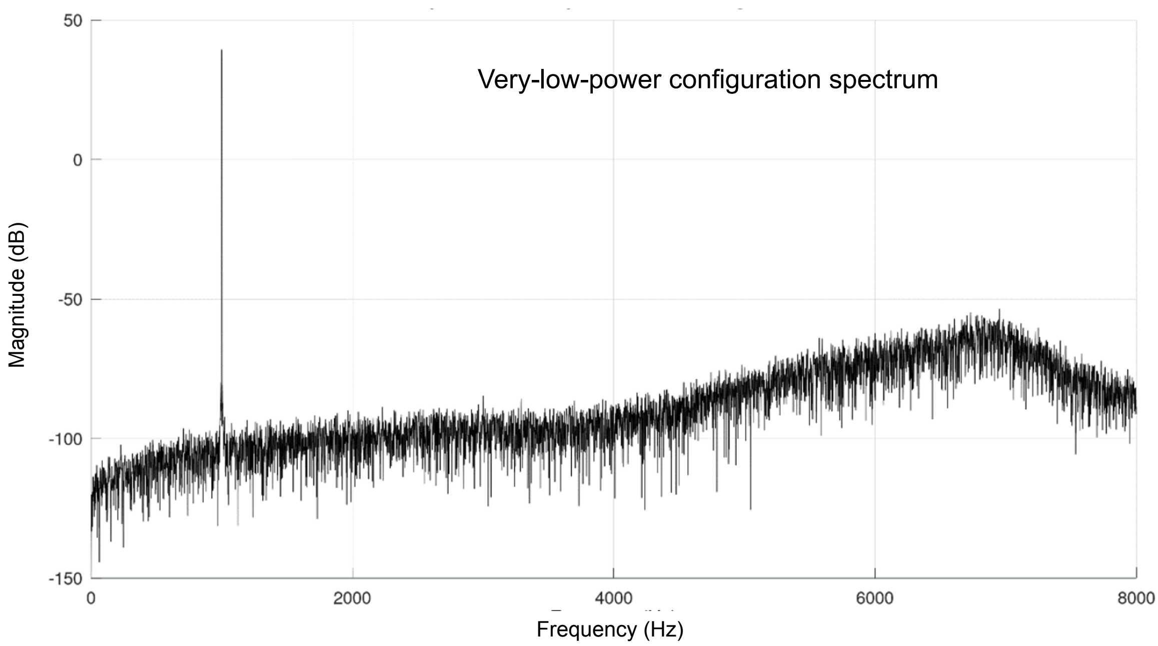
Parameter	Results
PCM frequency	16 kHz
SNR	75.337 dB
THD + N	75.282 dB
Inband ripple	0.42 dB

As for the CIC5- DEC32, the SNR value obtained for this use case is also because the digital microphone emulator performances are degraded when the decimation ratio is lower than 64. Setting the CIC decimation ratio to eight strongly reduces the attenuation of high-frequency components of the useful signal. Unfortunately the SNR degradation observed in this configuration is because the CIC decimation did not reject enough the high-frequency noise.

The reshape filter gives the following advantages:

- less ripple
- useful band extended to up to 7.1 kHz
- better rejection of out-off band signals

Figure 14. **Output spectrum of configuration 2**



3.2.3 Configuration 3: low-power balanced performance

By increasing the input signal from 512 kHz to 768 kHz, the application is set at the margin of low-power application. The CIC decimation ratio can be set to higher value (for example 12), giving a better rejection of the out off band noise.

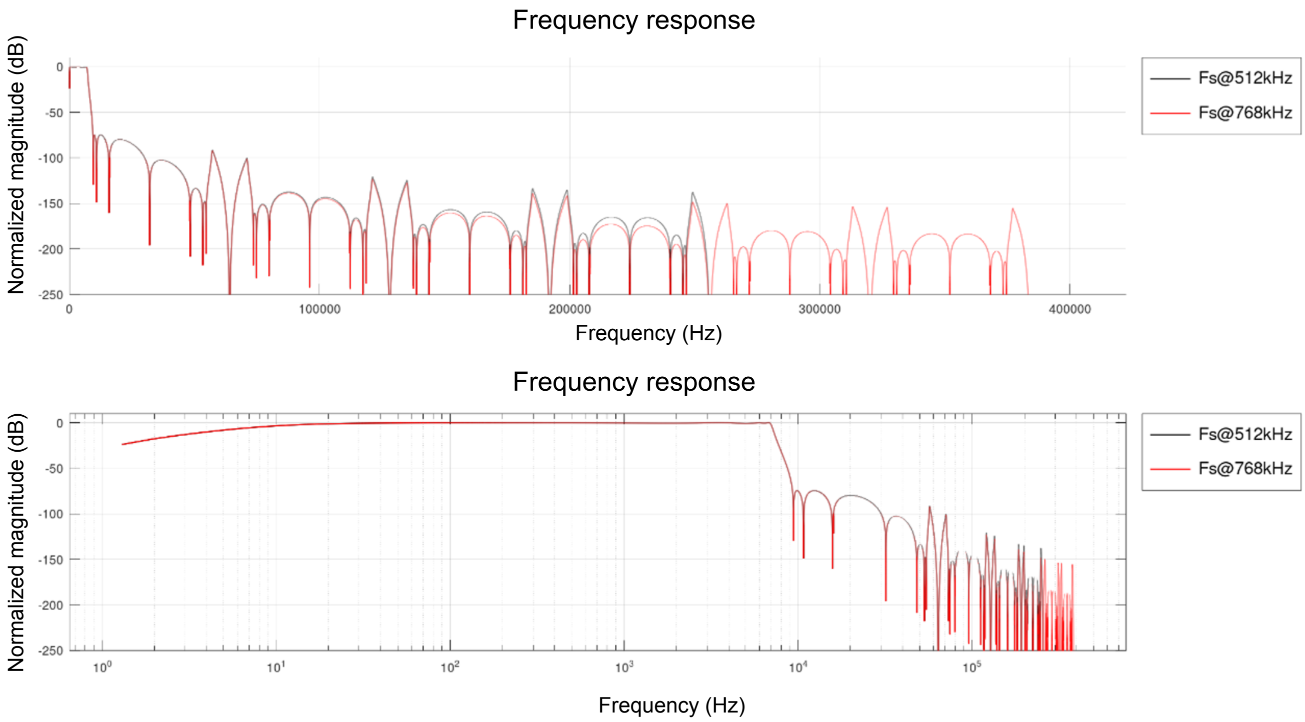
The following points highlight how the sampling frequency and the decimation alter both the noise distribution and aliasing, to make them suitable for audio capture:

- The noise generated by the sigma-delta modulator is more shifted to high frequencies.
- Out off band noise is more attenuated by the MDF filters, and more effective above as shown by Figure 15.

Table 6. **Configuration 3 recommended settings**

Parameter	Settings
CIC mode	$Sinc^5$
CIC decimation mode	12
Gain adjustment	0x06 (18.1 dB)
RSFLT state	Enable
HPF state	Enable
Total decimation ratio	48
PCM rate	16 kHz
Input sample rate	768 kHz

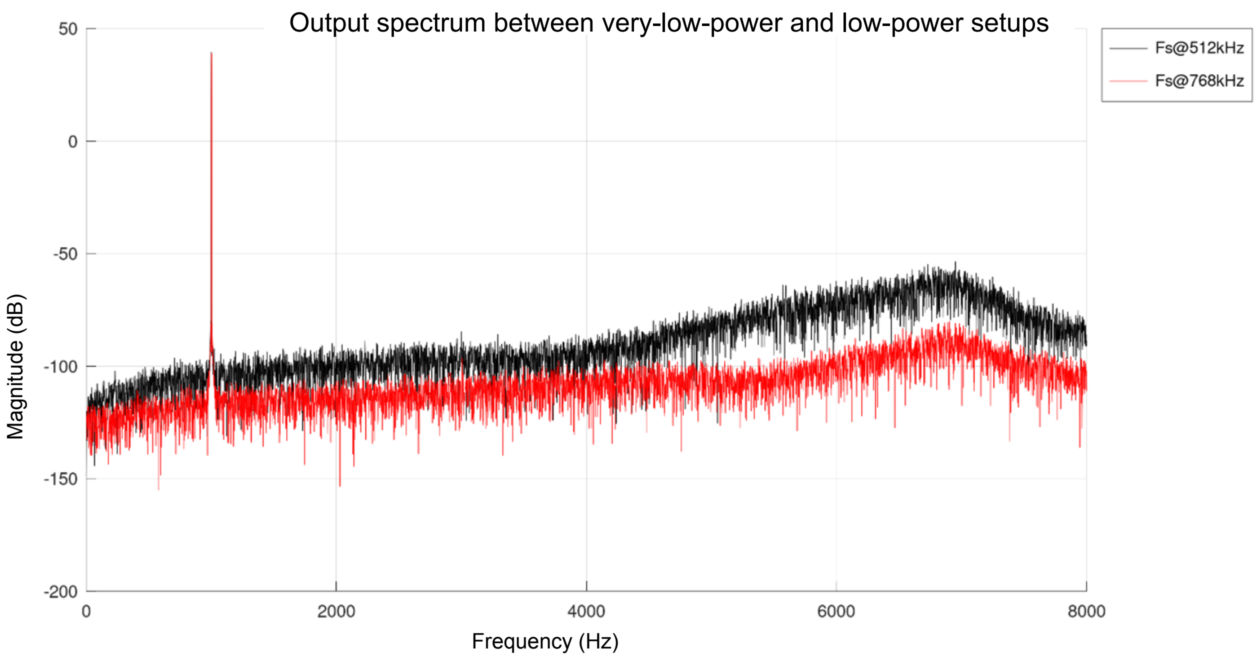
Figure 15. **Theoretical frequency response for an input sample rate equal to 512 kHz and 768 kHz**



The output spectrum difference is mainly related to the ∑∆ modulator. As the sample rate has been increased, the noise is shifted to higher frequencies. The decimation ratio has also been enlarged to 12, which means the high-frequency noise is attenuated in a better way. The contrast between the two sample rates is observed after 5 kHz, with a gap about 25 dB (see Figure 16). The SNR is then improved for the higher sample frequency.

This emphasizes the compromise between the sample rate and the achievable SNR.

Figure 16. **Output spectrum for an input sample rate equal to 512 kHz and 768 kHz**



The table below gives dynamic parameters of this configuration at F_S = 768 kHz and shows a significant improvement of the aliasing rejection.

Table 7. **Configuration 3 measurements**

Parameter	Results
PCM frequency	16 kHz
SNR	100.3 dB
THD + N	100.2 dB
-3 dB cutoff frequency	7100 Hz
Inband ripple	0.42 dB

3.2.4 Configuration 4: most efficient 16 kHz PCM

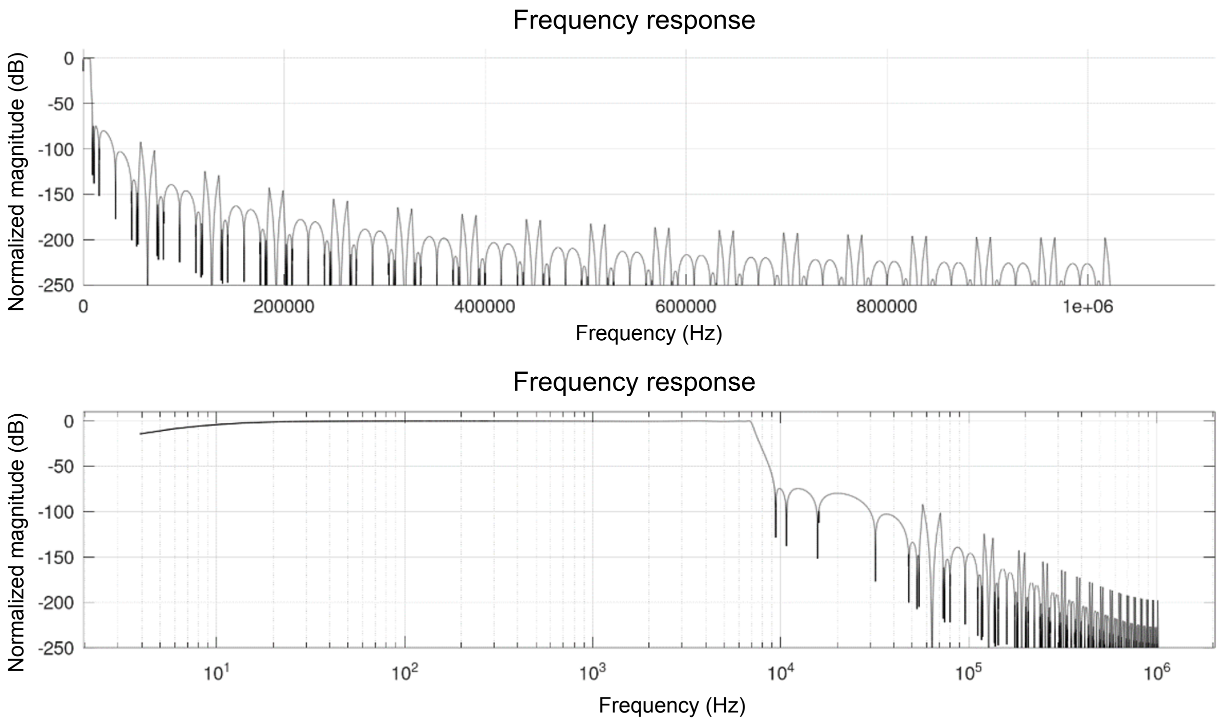
Maximum performances are reached when the MDF operates with the microphone input signal in normal or performance mode. The target of configurations listed below is to reach the best audio performances. A typical setting using a 1.024 MHz sampling frequency, and a configuration working at 2.048 MHz for optimal performances are described.

Table 8. **Configuration 4 recommended settings**

Parameter	Settings for 2.048 MHz	Settings for 1.024 MHz
CIC mode	$Sinc^5$	
CIC decimation mode	32	16
Gain adjustment	0x27 (-26.6 dB)	0x02 (6 dB)
RSFLT state	Enable	
HPF state		
Total decimation ratio	128	64
PCM rate	16 kHz	
Input sample rate	2.048 MHz	1.024 MH

A configuration using a decimation by 64 can also be used for 48 kHz audio. In this case, the input sample rate is 3.072 MHz, the SNR remains the same as well as the ripple, and the -3 dB cutoff frequency is 21.3 kHz. The MDF gets similar performances at 48 kHz.

Figure 17. **Theoretical frequency response of configuration 4 with 2.048 MHz input sample rate**



The frequency response shows a significant number of zeros introduced by the CIC and the RSFLT, together with a greater aliasing rejection. These two settings explain the output spectrum shape presented in Figure 18.

The noise floor remains flat in the whole passband. The RSFLT has an effect from 7.1 kHz, where the attenuation is stronger. The measured dynamic parameters are aligned with the output spectrum and justify the 120 dB on SNR.

Figure 18. **Output spectrum of configuration 4 with 2.048MHz input sample rate**

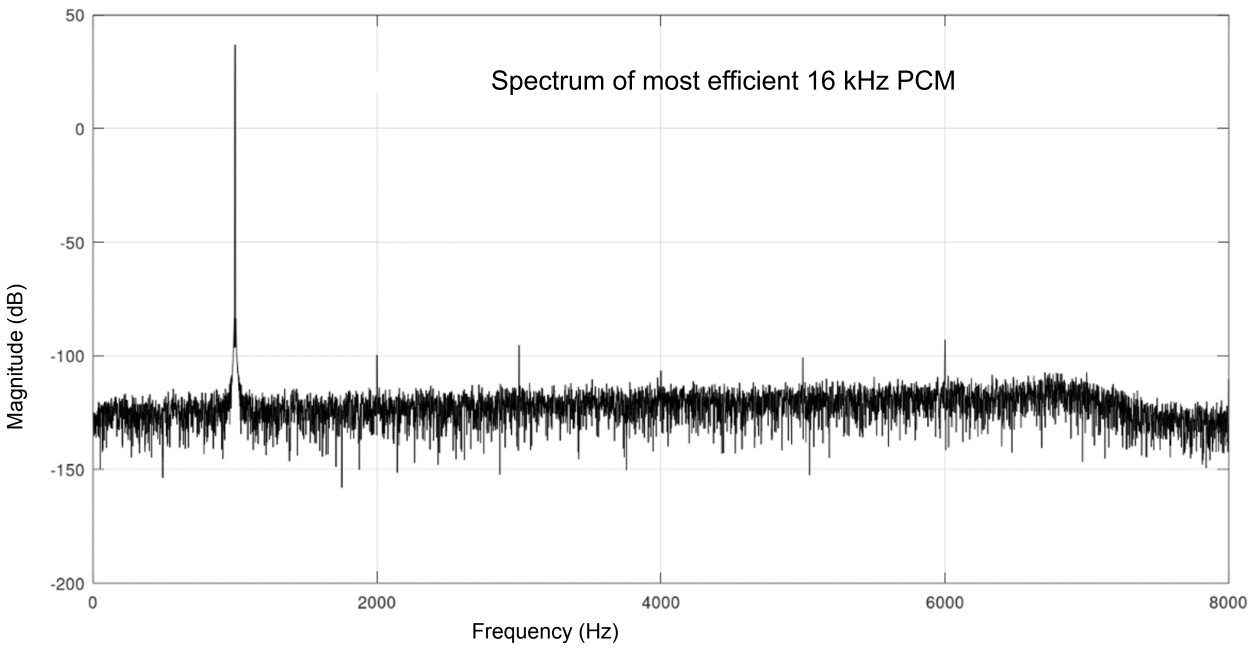


Table 9. Configuration 4 measurements

Parameter	Results for 2.048 MHz	Results for 1.024 MHz
PCM frequency	16 kHz	
SNR	119 dB	115 dB
THD + N	118 dB	114 dB
-3 dB cutoff frequency	7100 Hz	
Inband ripple	0.42 dB	

3.3 Mix hardware/software filter

3.3.1 Linear time-invariant filter

In digital signal processing, there are two types of LTI (linear time-invariant) filter: FIR (finite impulse response) and IIR (infinite impulse response). Both operate on a digital input but not in the same way, as detailed below:

- latency
  A FIR filter has usually more TAPs than an IIR filter. The input-to-output delay between is higher on a FIR filter. A FIR filter is then less suitable than an IIR one for applications requesting very-small latency.
- computation and memory requirements
  With its high number of coefficients, a FIR filter requires much more memory (coefficient storage, intermediate filter state for computation), and possibly more computing steps.
- group delay
  For a signal composed of multiple-frequency components, the group delay means that, in frequency domain, each component is not delayed by the same amount of time, which causes a shape distortion of the input signal.

A FIR filter is a linear-phase filter: each frequency component is equally delayed. The RSFLT is an IIR filter. Its largest group delay is around the cutoff frequency due to transition from passband to stopband.

3.3.2 FIR filter based on Arm CMSIS DSP library

Filtering is one part of an audio reproduction chain that introduces a group delay, even if the RSFLT guarantees an acceptable group delay. This group delay is not acceptable for some specific applications.

The RSFLT can be bypassed (including the decimation by 4). A software implementation can then substitute the RSFLT by a FIR filter. The input signal is partially processes by the MDF. The hardware part includes the signal decimation with the CIC before carrying out the HPF. The designed software FIR filter computes then the output data buffer.

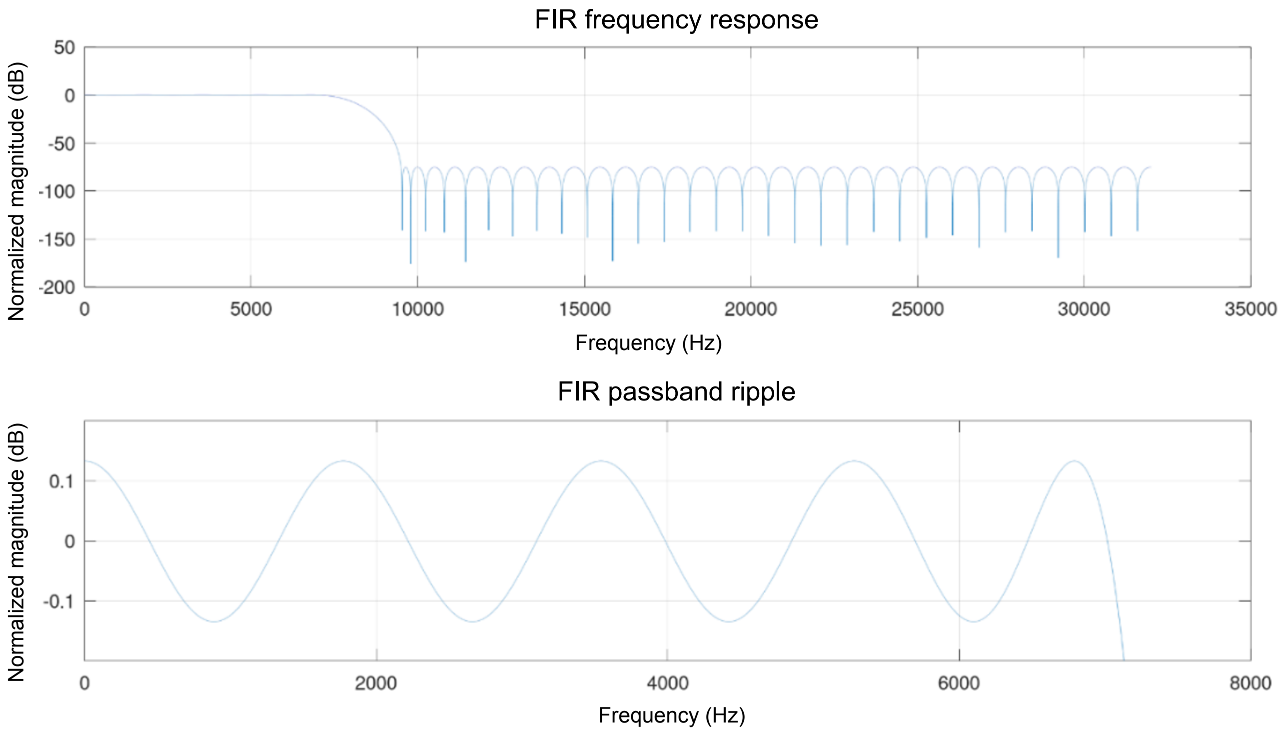
This alternative leads more flexibility to compensate inband attenuation, stopband rejection, and passband ripple. It also avoids a nonlinear group delay.

The CPU is used to compute data through a FIR filter. The proposal is to implement an equivalent FIR filter to the RSFLT: this improves inband ripple and stopband attenuation, with a moderate filter length. The characteristics of this designed filter are given in the table below.

Table 10. Characteristics of the software FIR filter

Parameter	Characteristics
Order	80
Filter length	81
Cutoff frequency	0.120 × FRS (7680 Hz FRS = 64 kHz)
Out-of-band rejection	75 dB
Passband ripple	±0.13 dB

Figure 19. **Frequency response of software FIR**



This software implementation is based on Arm CMSIS DSP Software Library Version 1.7.0.

The STM32U5 Series device embeds an Arm Cortex-M33 core including a single precision FPU (floating point unit) version FPv5-SP-D16. The proper library used for this FPU is `libarm_ARMv8MMLldfsp_math`. MDF output data are stored in a buffer. The FIR function computes them per block of 256 samples. The RSFLT usually decimates by four. The FIR decimator is selected in the DSP library to properly follow the hardware implementation. The filter coefficients are stored in single-float precision.

The table below compares a hardware IIR and the implemented software FIR for various configurations. The FIR coefficients used are detailed in Section Appendix A FIR coefficients for hardware/software filter.

Table 11. **Comparison between hardware and software filtering characteristics**

Parameter	Configuration 2		Configuration 3		Configuration 4	
	Hardware	Software	Hardware	Software	Hardware	Software
SNR (dB)	75.3	70.3	100.3	94.7	119	119.8
SINAD (dB)	75.3	70.3	100.2	94.8	118.3	119.2
THD	$1.49\ e^{-3}$	$1.83\ e^{-3}$	$7.82\ e^{-5}$	$8.98\ e^{-5}$	$4.59\ e^{-5}$	$3.88\ e^{-5}$
MIPS load	0(1)	5.8(2)	0(1)	5.8(2)	0(1)	5.8(2)
Inband ripple (dB)	0.42	0.26	0.42	0.26	0.42	0.26

1. *Input data are only computed by the hardware filter (MDF). The memory transfer is done by the GPDMA. The CPU is never used in pure hardware mode.*
2. *MIPS is based on a FIR sampling rate at 64 kHz, and before the decimation by four. MIPS value is given for one channel.*

The proposed FIR filter has a wider inband compared to the hardware IIR. Due to the digital microphone emulator noise with a low decimation rate, the noise on the inband is higher than with the hardware IIR filter. This explains the 5 dB difference of SNR loss.

This proposal does not fit with all applications. It must be adapted depending on needs (especially considering the microphone used).

Note: Improving the transition and inband ripples (mostly for efficient filtering) can increase the filter length, and consequently the latency and MIPS load.

3.4 MDF clock generator for audio application

3.4.1 Clock generator overview

The clock generator (CKGEN) embedded into the MDF has two main goals:

- Generate the processing clock (mdf_proc_ck) used to run signals processing, and to resample the incoming serial and parallel stream.
- Generate output clock to MDF_CCK0 and MDF_CCK1 pins.

Both clocks are derived from the kernel clock (mdf_ker_ck). Multiple sources can drive the kernel clock through the MDF1 clock mux.

The MSIK is well adapted to generate common frequency for sound and voice activity detection. The kernel clock is then driven by this input source. In some applications, the use of a PLL and an accurate reference generated from a crystal are needed

From the mdf_ker_ck, two dividers are available to generate processing and output clocks:

- PROCDIV[6:0] used to adapt the kernel clock frequency to the constraints of the parallel and serial interfaces, and to the processing blocks
- CCKDIV[3:0] used to adapt the frequency from mdf_proc_ck to MDF_CCK0 and MDF_CCK1 clocks

When the RFLST is used, the MDF processing block requires 24 mdf_proc_ck cycles to process a sample. This process timing constraints the clock setting. For an audio-capture application using digital microphones, the special mode LF_MASTER_SPI is highly recommended: in this mode, the mdf_proc_ck frequency can be only two times higher than the sensor clock. The LF_MASTER_SPI mode is recommended for applications that capture a signal from digital microphones.

The table below details the minimal clock setting in this mode.

Table 12. MDF clock constraints in LF_MASTER_SPI mode (F_{MDF_CCKy} max frequency limited to 5 MHz)

F_{MDF_CCKy} represents the clock frequency of MDF_CCK0 and MDF_CCK1 pins.

RSFLT disabled	RSFLT enabled
$F_{mdf\_proc\_ck} > 2 \times F_{MDF\_CCKy}$ and $F_{mdf\_hclk} > F_{mdf\_proc\_ck}$	$F_{mdf\_proc\_ck} > 24 \times \frac{F_{MDF\_CCKy}}{MCICD + 1}$ and $F_{mdf\_proc\_ck} > 2 \times F_{MDF\_CCKy}$ and $F_{mdf\_hclk} > F_{mdf\_proc\_ck}$

3.4.2 Clock setup for proposed configuration

The table below details the minimal frequency clock requirements for various configurations.

Note: The configuration 4 is excluded since it is clocked by a PLL to get the 2.048 MHz sample-rate frequency. A combination of PROCDIV and CCKDIV to get 2.048 MHz does not exist.

Table 13. MDF clock settings using MSIK as clock source

<table>
<tr><th>Parameter</th><th>Config 1</th><th>Config 2</th><th>Config 3</th></tr>
<tr><td>MSIK frequency (MHz)</td><td rowspan="2">1.024</td><td rowspan="2">1.536</td><td rowspan="2">3.072</td></tr>
<tr><td>mdf_ker_ck</td></tr>
<tr><td>PROCDIV(1)</td><td colspan="3">0</td></tr>
<tr><td>mdf_proc_ck (MHz)</td><td>1.024</td><td>1.536</td><td>3.072</td></tr>
<tr><td>CCKDIV(2)</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>MDF_CCKy frequency (kHz)</td><td>512</td><td>512</td><td>768</td></tr>
</table>

1. *PROCDIV[6:0] value.*
2. *CCKDIV[3:0] value.*

The MDF clock source depends on the application (limited frequency choice and peripherals shared with the selected source). For setups detailed in the tables below, HCLK (PLL1CLK output) clocks the MDF. These settings can also be applied on all PLLs, which can drive mdf_ker_ck. When sharing a PLL output with many peripherals, it is recommended to use a high frequency and to divide it across peripherals.

Note: The minimum decimation ratio of 32 is required to apply these clock settings.

Table 14. PLL integer mode and MDF clock setting for 2.048 MHz MDF_CCKy frequency

<table>
<tr><th>PLL source</th><th>DVM + 1</th><th>DVN + 1</th><th>DVR + 1</th><th>mdf_ker_ck (MHz)</th><th>PROCDIV + 1</th><th>mdf_proc_ck (MHz)</th><th>CCKDIV + 1</th><th>MDF_CCKy frequency (MHz)</th></tr>
<tr><td>HSI RC 16 MHz(1)</td><td rowspan="4">1</td><td>16</td><td rowspan="2">5</td><td rowspan="2">51.20</td><td rowspan="4">1</td><td rowspan="2">51.20</td><td rowspan="2">25</td><td rowspan="4">2.048</td></tr>
<tr><td>HSE 8 MHz</td><td rowspan="3">32</td></tr>
<tr><td rowspan="2">HSI RC 16 MHz</td><td>125</td><td>4.096</td><td>4.096</td><td>2</td></tr>
<tr><td>25</td><td>20.48</td><td>20.48</td><td>10</td></tr>
</table>

1. *HSE 16 MHz can replace HSI RC as PLL input clock source.*

Table 15. PLL fractional mode and MDF clock setting for 2.048 MHz MDF_CCKy frequency

<table>
<tr><th>PLL source</th><th>DVM + 1</th><th>DVN + 1</th><th>DVR + 1</th><th>FRACV + 1</th><th>mdf_ker_ck (MHz)</th><th>PROCDIV + 1</th><th>mdf_proc_ck (MHz)</th><th>CCKDIV + 1</th><th>MDF_CCKy frequency (MHz)</th></tr>
<tr><td>HSI RC 16 MHz</td><td rowspan="4">1</td><td rowspan="2">9</td><td rowspan="2">1</td><td rowspan="2">8064</td><td rowspan="2">159.75</td><td>6</td><td>26.625</td><td>13</td><td rowspan="4">2.048</td></tr>
<tr><td>HSE 8 MHz</td><td>26</td><td>6.14</td><td>3</td></tr>
<tr><td rowspan="2">HSI RC 16 MHz</td><td>30</td><td>94</td><td>5888</td><td>98.3</td><td>4</td><td>24.575</td><td>12</td></tr>
<tr><td>9</td><td>3</td><td>8064</td><td>53.25</td><td>2</td><td>26.625</td><td>13</td></tr>
</table>

4 Conclusion

The MDF peripheral embeds features designed for various applications (from motor control to audio capture) based on a digitized analog signal. This document introduces the fundamentals of multirate filters applied to audio processing. The filter characteristics and the configuration examples highlight key parameters to fit with application requirements. The proposed methodology can be employed in the same manner for any analog signal process. The MDF modularity makes easier the move from hardware to mix hardware/software processing to cover additional features.

Appendix A FIR coefficients for hardware/software filter

```
#define NUM_TAPS    81

const float FIR_Coeffs[NUM_TAPS] = {
-0.0001331688982,2.479605973e-05, 0.0004620492691,
0.001279119053, 0.002269938355,0.002963012317,
0.002809111495, 0.001525794738, -0.0005894111819,
-0.002620612038, -0.003405907191, -0.002208728343,
0.0006755130016, 0.003814224154, 0.00525664771,
0.0036663434, -0.000653038267, -0.00555244647,
-0.007996876724, -0.005838687997, 0.0005731916171,
0.008046226576, 0.01197536848, 0.009016320109,
-0.0004628565221, -0.01178676635, -0.01801933348,
-0.01391186565,0.0003455153201, 0.01797029935,
0.02830379643, 0.02255534753, -0.0002430360764,
-0.03052200936, -0.05086934194, -0.04359084368,
0.0001734420803, 0.07435979694,   0.1584162265,
0.2247111946,      0.249851197,   0.2247111946,
0.1584162265, 0.07435979694,0.0001734420803,
-0.04359084368, -0.05086934194, -0.03052200936,
-0.0002430360764,0.02255534753, 0.02830379643,
0.01797029935,0.0003455153201, -0.01391186565,
-0.01801933348, -0.01178676635, -0.0004628565221,
0.009016320109, 0.01197536848, 0.008046226576,
0.0005731916171, -0.005838687997, -0.007996876724,
-0.00555244647, -0.000653038267, 0.0036663434,
0.00525664771, 0.003814224154,0.0006755130016,
-0.002208728343, -0.003405907191, -0.002620612038,
-0.0005894111819, 0.001525794738, 0.002809111495,
0.002963012317, 0.002269938355, 0.001279119053,
0.0004620492691,2.479605973e-05, -0.0001331688982};
```

Revision history

Table 16. Document revision history

Date	Version	Changes
2-Aug-2022	1	Initial release.

Contents

List of tables

List of figures