AN5142

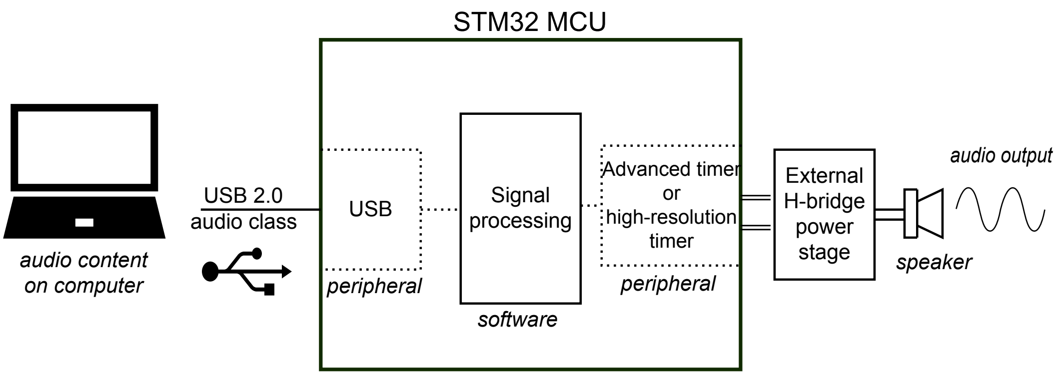
Application note

Class-D audio amplifier implementation on STM32 32-bit Arm® Cortex® MCUs

Introduction

This application note describes a Class-D audio implementation using STM32 MCUs. The amplifier consists of a software modulator, a pulse width modulator (PWM) timer and an external H-bridge power stage. The first section of this document presents the Class-D amplifier principle, and the main parameters that need to be taken into account to achieve good audio listening quality. The subsequent section describes each of the building blocks comprising a PWM modulator, using either an advanced timer or a high-resolution timer peripheral available in the STM32 microcontroller. The application note concludes by presenting the achievable audio performance in relation to the required CPU processing resources and implementation effort.

Figure 1. Overview of the proposed Class-D amplifier implementation



1 General information

This document describes a Class-D audio implementation on STM32 microcontrollers. STM32 products are Arm®-based devices.

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

2 Audio Class-D amplifier

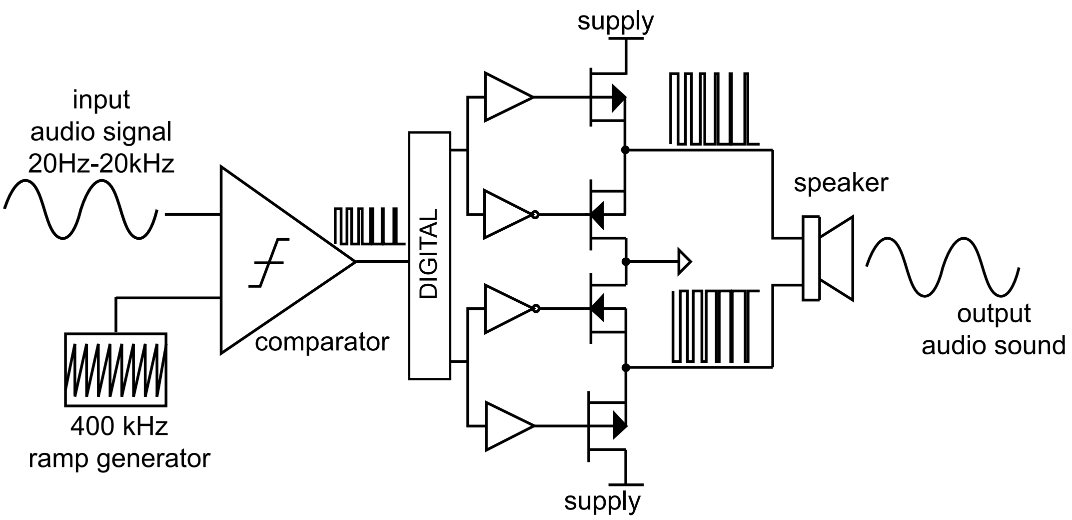
2.1 Class-D architecture overview

The Class-D architecture is designed to simplify the usual audio amplifier structure. Mostly based on digital cells, it is less expensive and easier to integrate into an application. Even the power stage can be defined as a digital cell because it is based on power MOSFETs (metal-oxide-semiconductor field-effect transistor). Depending on the application environment, a Class-D amplifier can be open-loop or closed-loop. Open-loop architecture amplifiers are a good candidate when the application supply voltage is stable and clean; this option is the less expensive solution and it is acceptable for most consumer products. In the other hand, in the automotive market, the variation level and noise of the battery voltage are not negligible and should be compensated continuously with a closed-loop Class-D architecture. The closed-loop Class-D architecture is also used on mobile phones to deal with battery disturbances due to their RF part. This application note presents an open-loop Class-D implementation for consumer applications.

2.2 Pulse width modulation (PWM)

The PWM modulation transforms the audio input signal interface into digital signals which control the gate of the MOSFET in the power stage at a frequency of about 400 kHz for audio application. The PWM can be implemented either using analog or a digital cells. This application note presents only the digital implementation of a PWM modulator. Digital Class-D amplifiers require digital processing such as interpolation filters and sigma-delta modulators to achieve the highest audio performances.

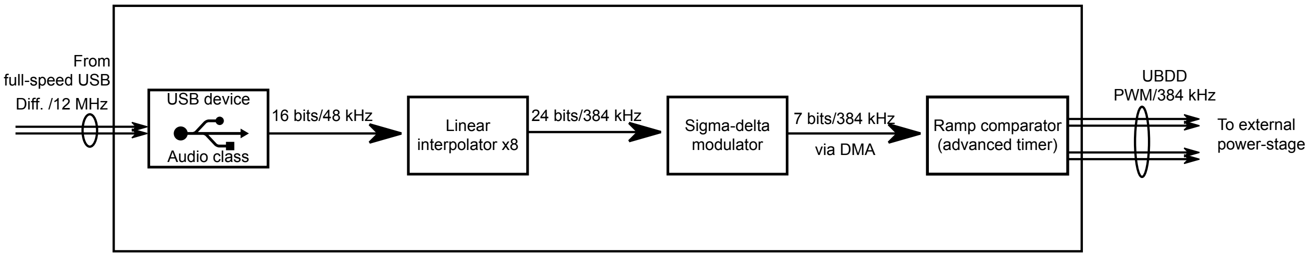
Figure 2. Principle of an analog Class-D audio amplifier (simplified)



2.3 PWM input signal

Generally, the digital audio input use the pulse code modulation (PCM) format. The data width may be either 16 or 24 bits while the sampling rate could be 44.1 kHz or 48 kHz. Because the main processing of the PWM modulator runs at high frequency, it is necessary to interpolate the input signal at a higher rate. This processing can be done easily using a linear interpolation. The following figure shows the complete processing audio path.

Figure 3. PWM digital modulator processing path



2.4 Linear interpolation

The linear interpolation is a signal processing technique which allows to increase the sampling rate of a signal. The audio digital signal received form the USB audio device is sampled at 48 kHz. In order to avoid adapting the rate of the incoming audio stream, the PWM carrier frequency (Fc) must be a multiple of 48 kHz (384 kHz for example). The linear interpolation processing has a main parameter which is the over sampling ratio (OSR), (or multiplication factor) between the input signal sampling rate and the output signal sampling rate. The simplest way to implement and proceed is to use power of 2 factor (2,4,8), which is easy to implement and is not CPU demanding; it just requires basic shifts and add. This is the option chosen for this application note.

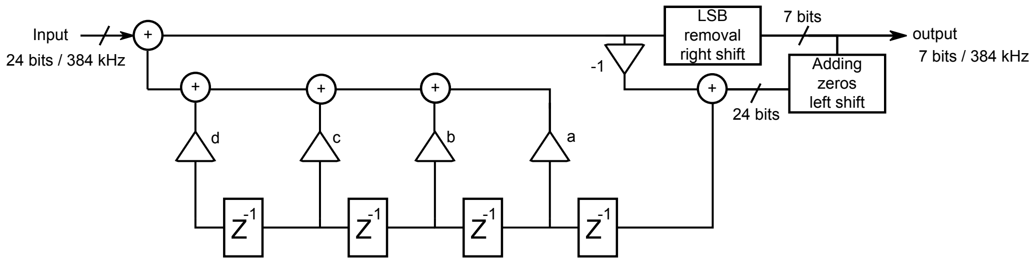
2.5 Sigma-delta modulation or noise shaping

The sigma-delta modulation (or ΣΔ modulation) is used to convert from a low-rate/large-data bit width to a high-rate/low-data width; the noise is shaped by the filter of the transfer function. The main idea of this implementation is the use of a PWM modulator with a limited number of bits in order to lower the frequency of the PWM modulator (FPWM clock).

The current analysis takes the assumption that the PWM modulator uses a clock of 49.152 MHz for a 48 kHz stream. This clock value is calculated with below values:

- Input audio stream at 48 kHz
- Oversampling the input audio stream by 8: 48 kHz × 8 = 384 kHz
- The noise shaper reduces the data size from 24 bits to 7 bits
- The PWM uses this 7-bit signal as input
- 7 bits correspond to 128 possible levels for the PWM
- The PWM must update the level at 384 kHz, so it requires a clock 128 times faster, which results in a 49.152 MHz clock.

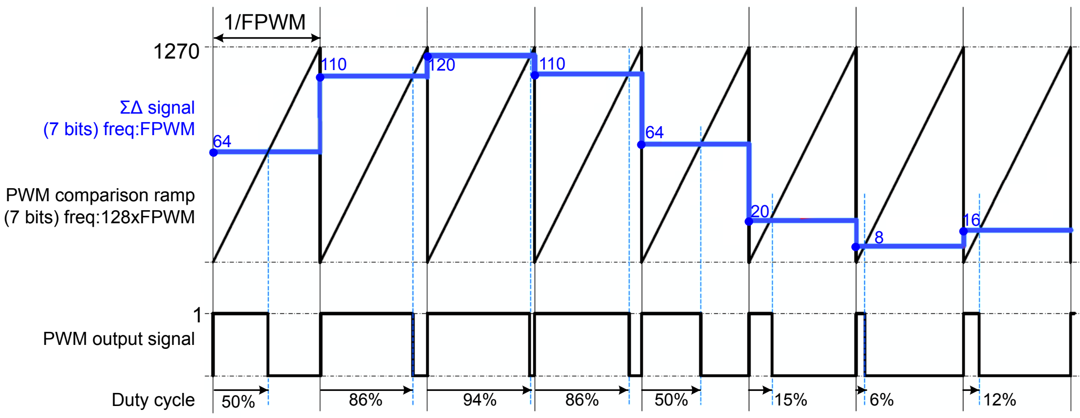
Figure 4. Sigma-delta modulator architecture proposal



2.6 PWM modulation with a ramp comparison

The output values of the sigma-delta modulator are compared to a ramp. The frequency of the ramp is equal to interpolated audio signal, which is 384 kHz in the example used in this application note. Each audio sample is compared to the ramp amplitude. The result of this comparison gives output pulses with a width proportional to the signal amplitude: this is the principle of the pulse width modulation (PWM) .

Figure 5. Principle of the PWM modulation



3 STM32 MCUs timer peripheral

The timer peripheral is part of the essential set of peripherals embedded in all the STM32 microcontrollers. The STM32 MCUs timer peripheral was conceived to be the keystone peripheral for a large number of applications: from motor-control applications to periodic-events generation applications. The specifications on the timer peripheral available in all STM32 reference manuals are very wide due to their versatility.

3.1 Using the timer peripheral for PWM modulation

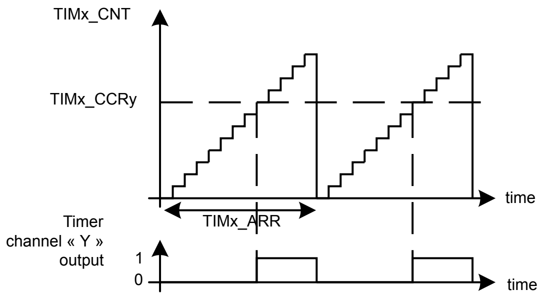
The PWM modulation is based on a signal comparison with a ramp which can be easily performed with a timer peripheral function. The timer embeds a counter clocked from the internal clock tree. This counter is configurable and managed with the following main registers:

- TIMx_CNT timer register is used to read and write the content of the timer counter
- TIMx_ARR timer register contains the reload value of the timer counter
- TIMx_CCRy is the compare register to the content of the timer counter

Once the timer is enabled, the counter increment starts. The ramp amplitude is defined by TIMx_ARR value. The duty cycle of the PWM is adjusted by the value of TIMx_CCRy register. The associated output channel "y" switch from low level to high level.

The following figure present the basic timer mechanism:

Figure 6. Simplified PWM generation with an STM32 timer



3.2 PWM modulation with an advanced timer

The advanced-control timers (TIM1/TIM8) can each be seen as a three-phase PWM multiplexed on six channels. They have complementary PWM outputs with programmable inserted dead-times.

3.2.1 Timer peripheral clock frequency

The advanced timer main clock is connected to the internal clock of the microcontroller. The calculation of the amplitude of ramp and associated resolution depends on the maximum operating frequency and the PWM frequency.

See below an example for the STM32L4 Series timer peripheral running at their maximum frequency (80 MHz) to achieve the highest ramp resolution:

- Timer clock = 80 MHz (resolution = 1/80 MHz = 12.5 ns)
- PWM frequency = 400 kHz
- Timer reload value (ramp size) = 80 MHz/400 kHz = 200 levels
- Equivalent number of bits (ENOB) = log2(Ramp_size) = log2(200) = 7.64 bits
- Dynamic range from ENOB calculation = (ENOB x 6.02) + 1.76 = 47.8 dB

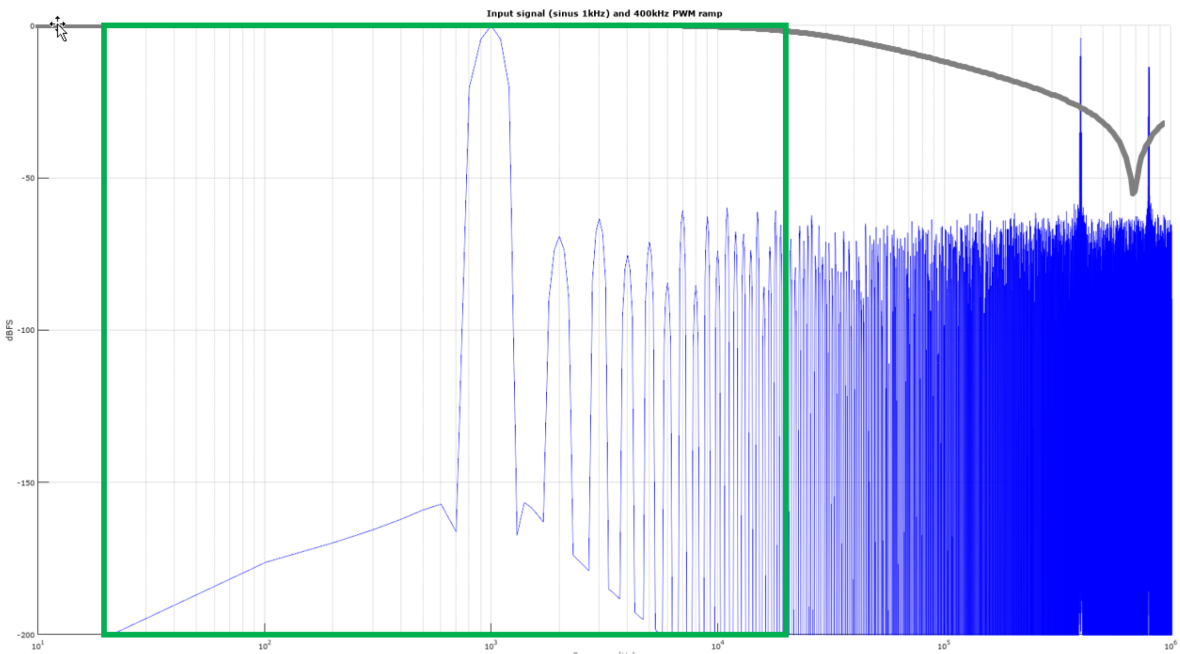
This approximation of the dynamic range using the ENOB formula shows that equivalent resolution of the timer is not enough for a good quality audio application. The implementation of the sigma-delta modulator is mandatory to move the noise energy on the high frequency part, then the low frequency benefits of a good dynamic range.

Table 1. Example of some STM32 PWM dynamic range without Sigma-delta modulator

Series	Maximum SYSCLK frequency	Ramp size	ENOB	Dynamic range
STM32F1 STM32F3	72 MHz	180	7.5	46.9 dB
STM32L4	80 MHz	200	7.64	47.8 dB
STM32F2 STM32L4+	120 MHz	300	8.22	51.3 dB
STM32F4	180 MHz	450	8.81	54.8 dB
STM32H7	200 MHz	500	8.96	55.7 dB
STM32F7	216 MHz	540	9.07	56.4 dB

The following figure shows the full spectrum of a PWM modulated signal (with the noise shaper activated) where the bandwidth for the application is the audio range 20 Hz-20 kHz (green area) while the speaker frequency response is plotted with a grey line (speaker modelization is 8 Ω + 47 uH coil).

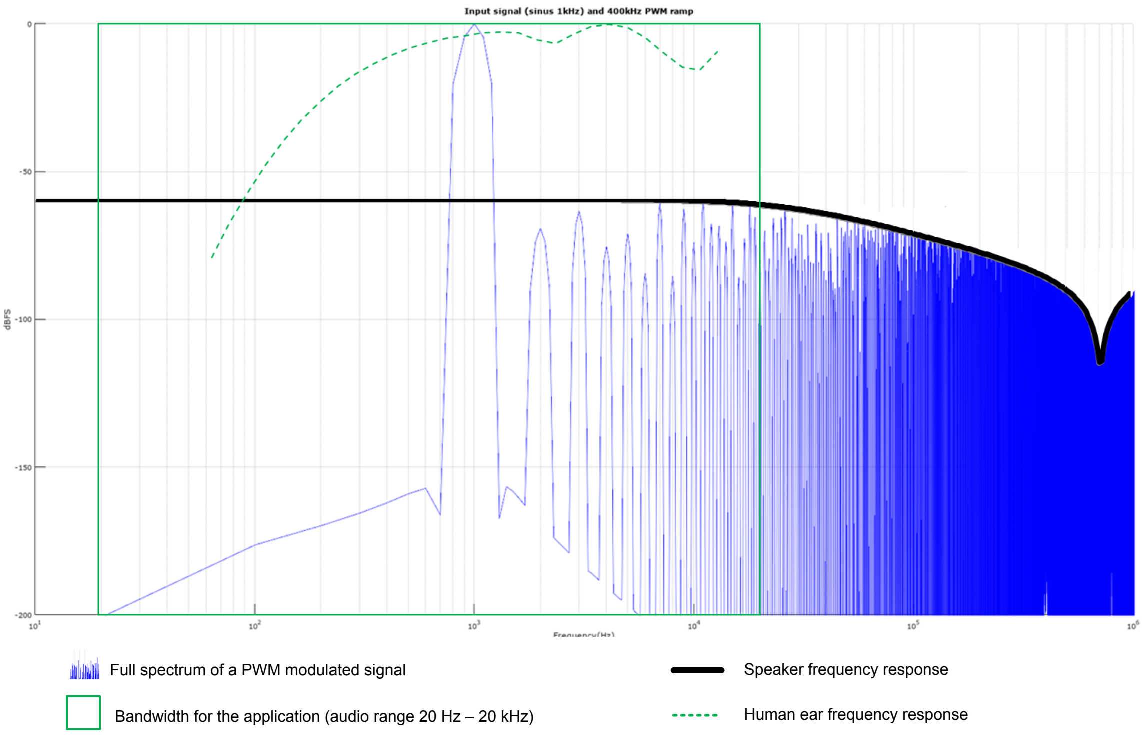
Figure 7. PWM spectrum using advanced timer and speaker current frequency response (simulation)



Full spectrum of a PWM modulated signal | Bandwidth for the application (audio range 20 Hz – 20 kHz) | Speaker frequency response

If the speaker frequency response is applied to the PWM spectrum and the human ear frequency response, the result is a filtered spectrum which is filtered where the PWM fundamental at 400 kHz is attenuated (refer to the figure below).

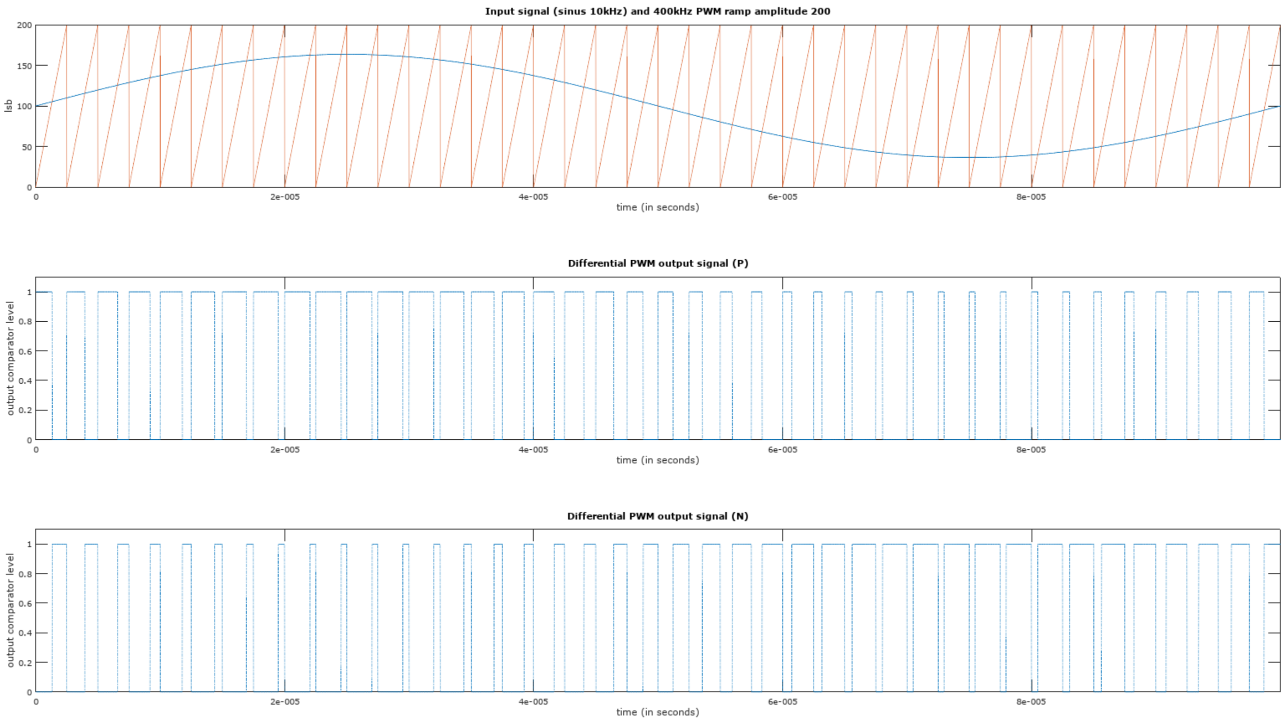
Figure 8. **Speaker filtered PWM spectrum response with human ear response**



3.2.2 PWM modulator gain error

The sigma-delta modulator feeds the timer peripheral input through the direct memory access (DMA) peripheral . The original audio content is defined over 16-bit words from USB and the output of the sigma-delta is 7-bit words corresponding to the maximum signal amplitude at the timer operating range. CMP value is between 0 and 127. The ratio factor between the timer internal clock and the PWM frequency must be correctly configured to fit 100% of the timer operating range, otherwise a gain error occurs.

Figure 9. PWM modulator with gain error example



Example of a gain error using the following parameters (see Figure 9. PWM modulator with gain error example):

- Timer clock = 80 MHz, PWM frequency = 400 kHz, ramp amplitude = 200
- Gain error = 20 x LOG10(input_amp/ramp_amp) = 20 x LOG10(127/199) = -3.9 dB

The frequency of the timer peripheral or the PWM frequency should be adjusted to fit with the maximum PWM modulator (see Figure 10. PWM modulator with correct gain).

- Timer clock = 51.2 MHz, PWM frequency = 400 kHz, ramp amplitude = 128
- Gain error =20 x LOG10(input_amp/ramp_amp) = 20 x LOG10(127/127) = 0 dB

or

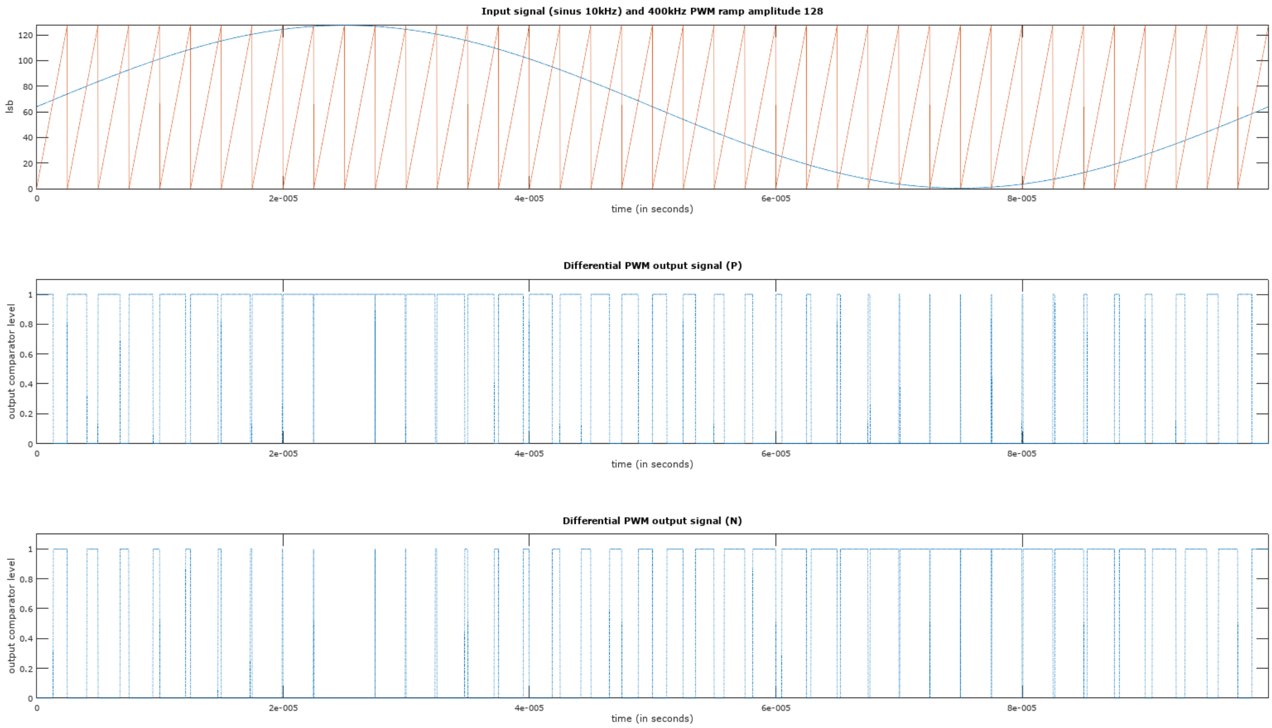
- Timer clock = 48 MHz, PWM frequency = 375 kHz, ramp amplitude = 128
- Gain error =20 x LOG10(input_amp/ramp_amp) = 20 x LOG10(127/127) = 0 dB

Listening quality can be easily degraded if the PWM ramp and maximum input signal amplitude do not match. Depending on the ratio, different phenomena may occur:

- If the maximum input signal amplitude is lower than the PWM ramp amplitude: the output PWM signal is attenuated and loudness is reduced.
- If the maximum input signal amplitude is higher than the PWM ramp amplitude: output PWM signal is saturated and distortion is audible.

In the final application, the input signal sample rate is 8×48 kHz = 384 kHz, the PWM ramp amplitude is 128 so the timer clock should be equal to 0.384 MHz × 128 = 49.152 MHz.

Figure 10. PWM modulator with correct gain

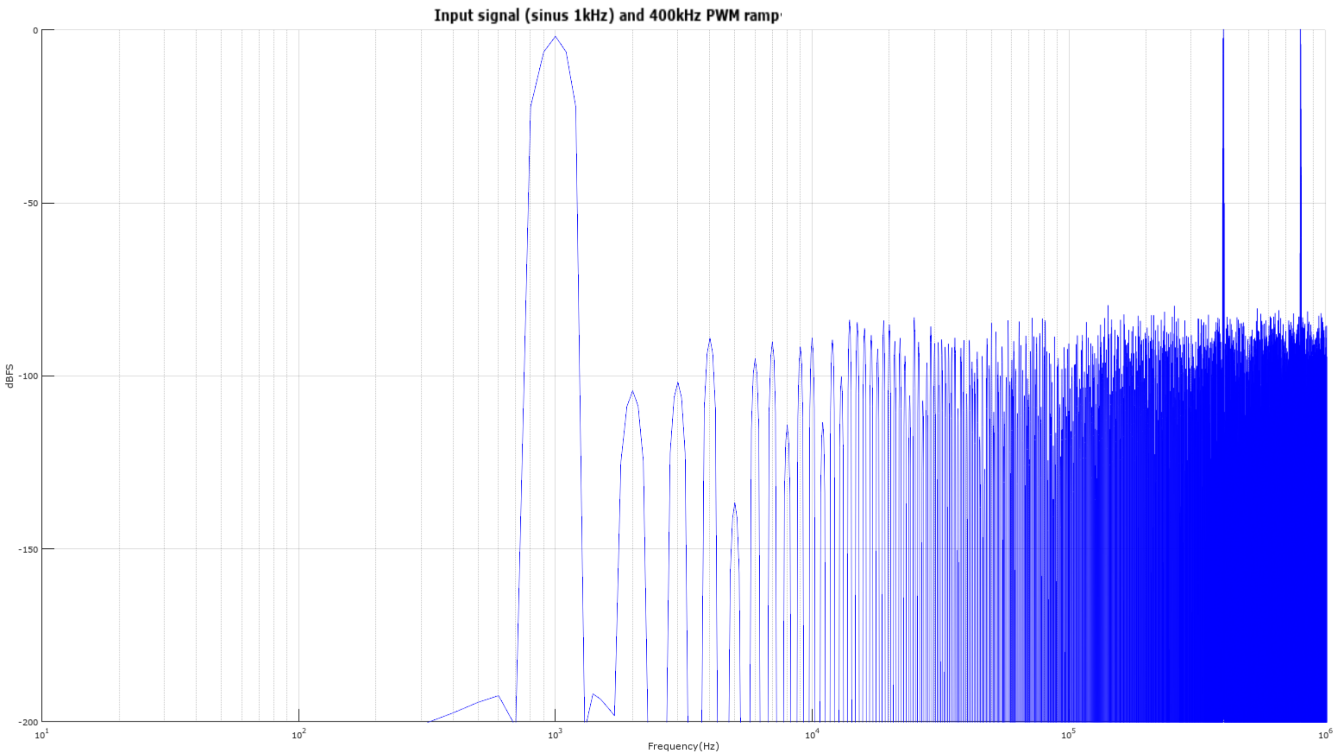


3.3 High-resolution timer

An internal DLL allows the high-resolution timer to generate PWM signals with a resolution for edge positioning 32 times better than one of an advance timer. The equivalent number of bits is calculated as stated below:

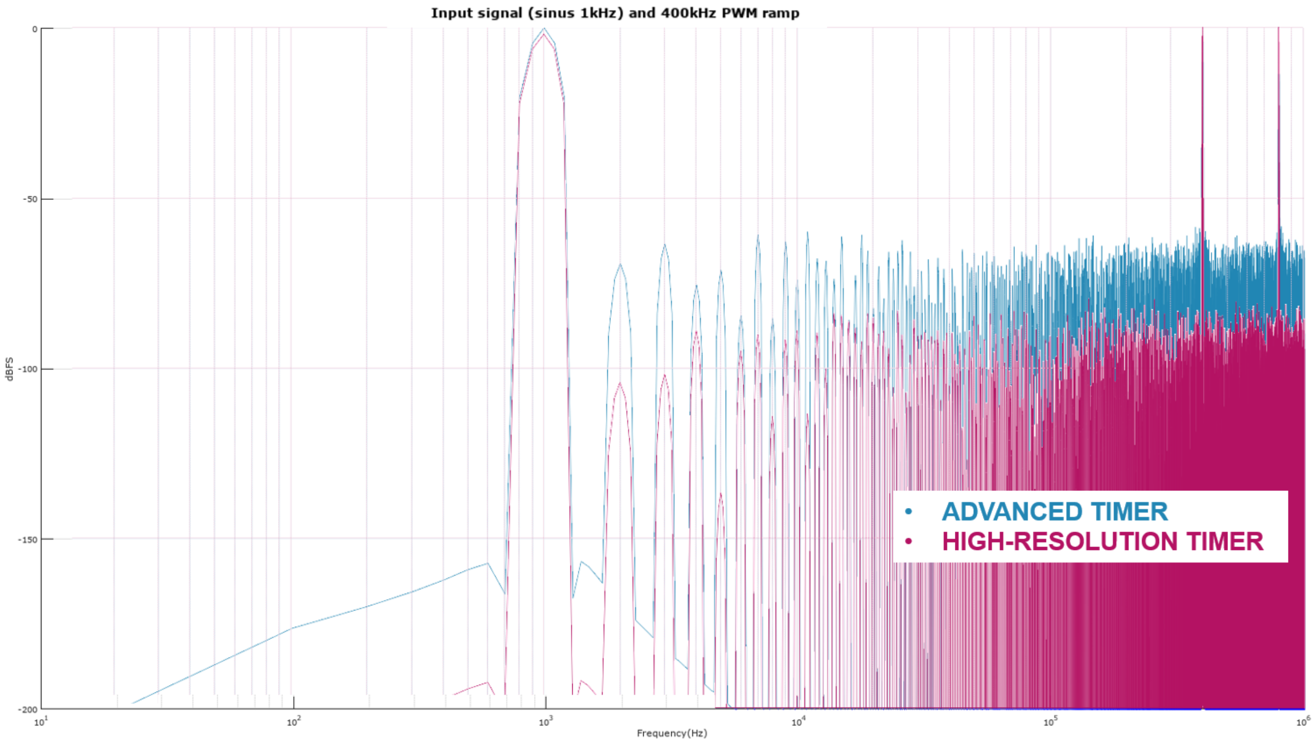
- Timer frequency = 150 MHz × 32 = 4.8 GHz
- PWM frequency = 400 kHz
- Timer reload value (ramp size) = 4.8 GHz ÷ 400 kHz = 12000 samples
- Equivalent number of bits (ENOB) = log2(Ramp_size) = log2(12000) = 13.55 bits
- Dynamic range from ENOB calculation = (ENOB × 6.02) + 1.76 = 83.33 dB

Figure 11. Spectrum - simulated PWM modulation using HRTIMER



The following figure represents the comparison of the PWM spectrum generated with an advanced timer and with a high-resolution timer peripheral. It appears that the noise level on the PWM full bandwidth is higher when using the advanced timer. The reason behind is that the implementation of the advanced timer includes a sigma-delta modulator which is shaping the noise in the high-frequency range.

Figure 12. Spectrum comparison between advanced and high-resolution timers



4 Class-D full-bridge power stage

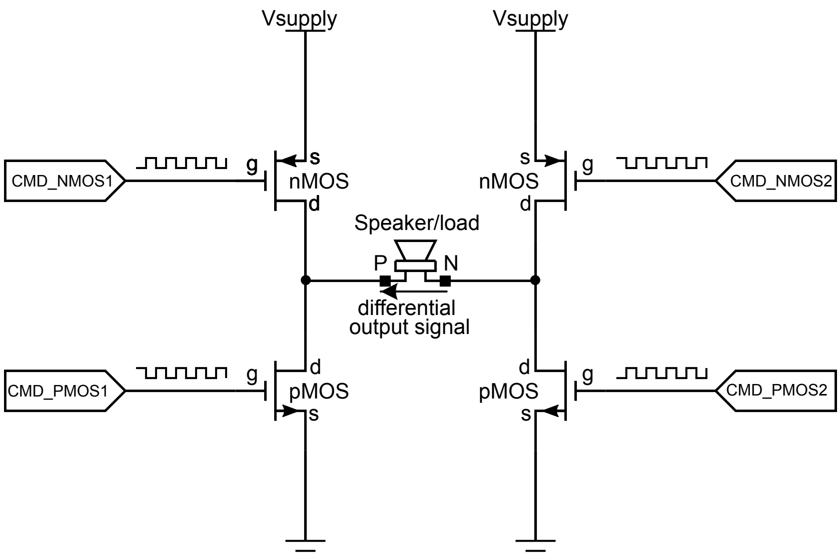
The structure of a Class-D power stage is close to a full-bridge non-inverting buck-boost converter (a type of switched-mode power supply (SMPS)). Whereas buck converters usually function as voltage regulators, delivering a constant DC voltage into a variable load and can only source current (one-quadrant operation), a Class-D amplifier delivers a constantly changing voltage into a fixed load, where current and voltage can independently change sign (four-quadrant operation). A switching amplifier must not be confused with linear amplifiers that use an SMPS as their source of DC power. A switching amplifier may use any type of power supply (battery supply or internal regulated supply).

4.1 Class-D amplifiers

Theoretical power efficiency of Class-D amplifiers is 100%. It means that 100% of the power supplied to the class-D amplifier is delivered to the load and none is turned to heat. The reason for this 100% theoretical efficiency is because an ideal switch in its *ON* state may conduct all the current and may not have any voltage drop across it, hence no heat dissipates. When this ideal switch is off it has the full supply-voltage across it but no leak current flowing through it, and again no heat dissipates.

Real-world power MOSFETs are not ideal switches, but they still commonly perform practical efficiencies over 90% for Class-D amplifiers. In contrast, linear Class-AB amplifiers are always operated with both current flowing through and voltage standing across the power devices. An ideal Class-B amplifier has a theoretical maximum efficiency of 78%. Class-A amplifiers (purely linear, with the devices always "on") have a theoretical maximum efficiency of 50%.

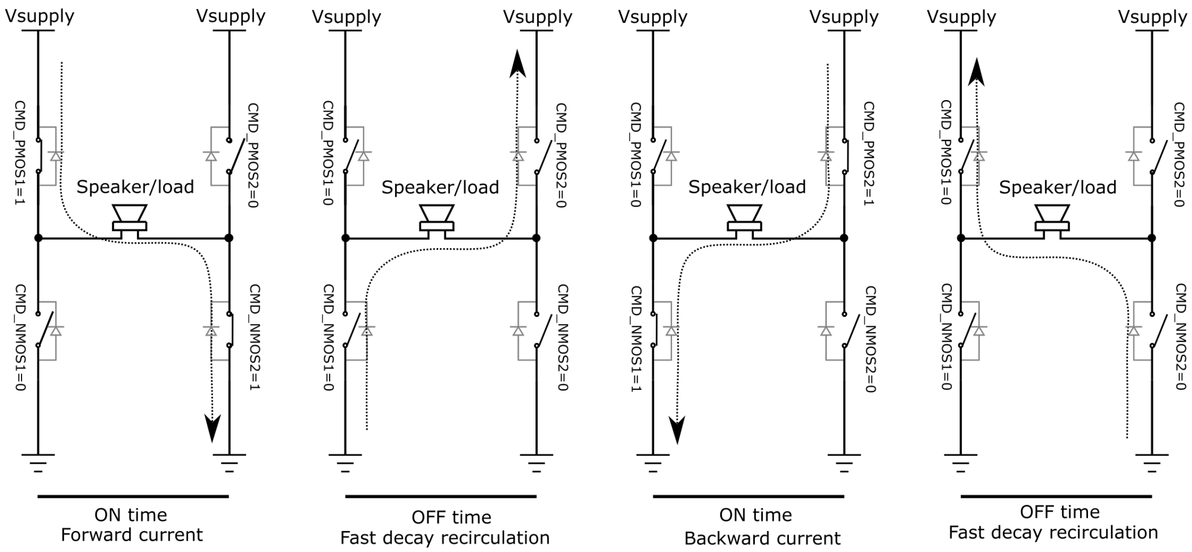
Figure 13. Simple view of an H-bridge power stage



4.2 H-bridge current phases and dead time

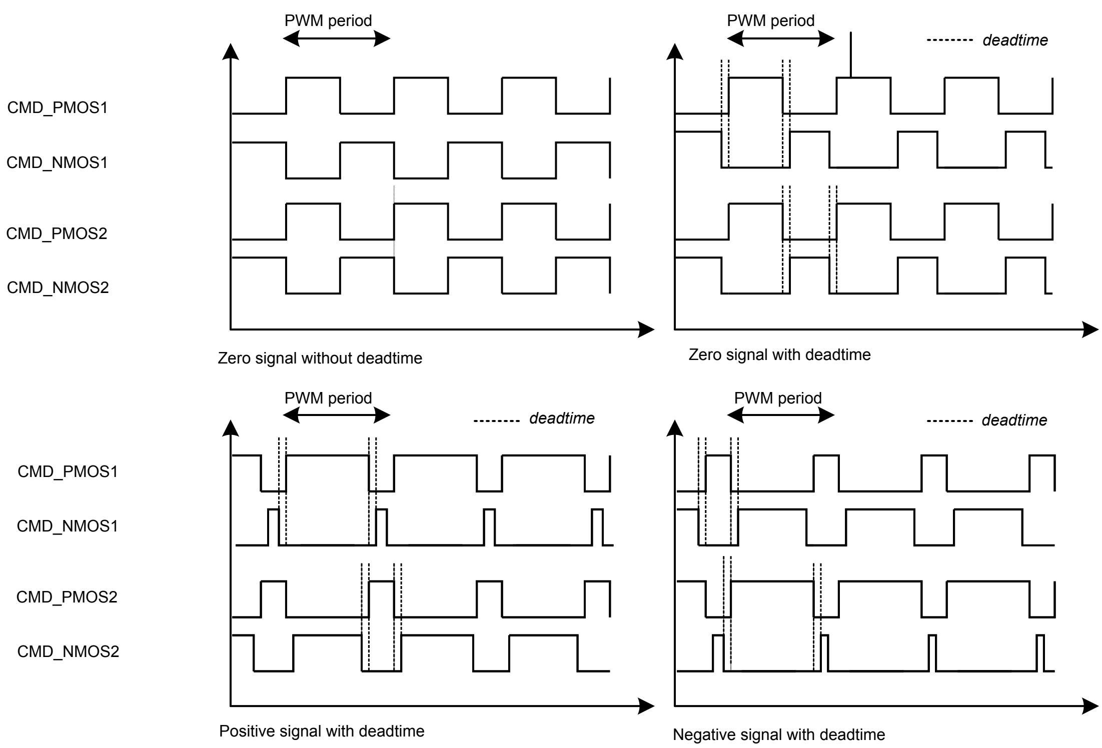
The following figure shows the four different phases where the current direction is controlled through the output load. The system must insert dead time (*OFF* time) to avoid direct shortcut between the bridge supply and the reference ground.

Figure 14. Power stage 4 phase current circulation



The deadtime insertion and duration is directly managed by the timer peripheral itself. The following diagram shows the difference between PWM with and without deadtime for various signal examples.

Figure 15. MOS commands examples with zero/positive/negative signals



4.3 H-bridge commands and output measurements

A 1 kHz sinusoidal waveform is stored in memory, processed by the firmware and sent to the timer peripheral output. The output signal is filtered with a simple low-pass frequency filter (frequency cut off is 30 kHz). The signals on the top window are the digital commands of the PMOS and NMOS of the bridge and the signals on the bottom window are the differential output signal (P and N signals).

Figure 16. Temporal acquisition with digital commands and analog output

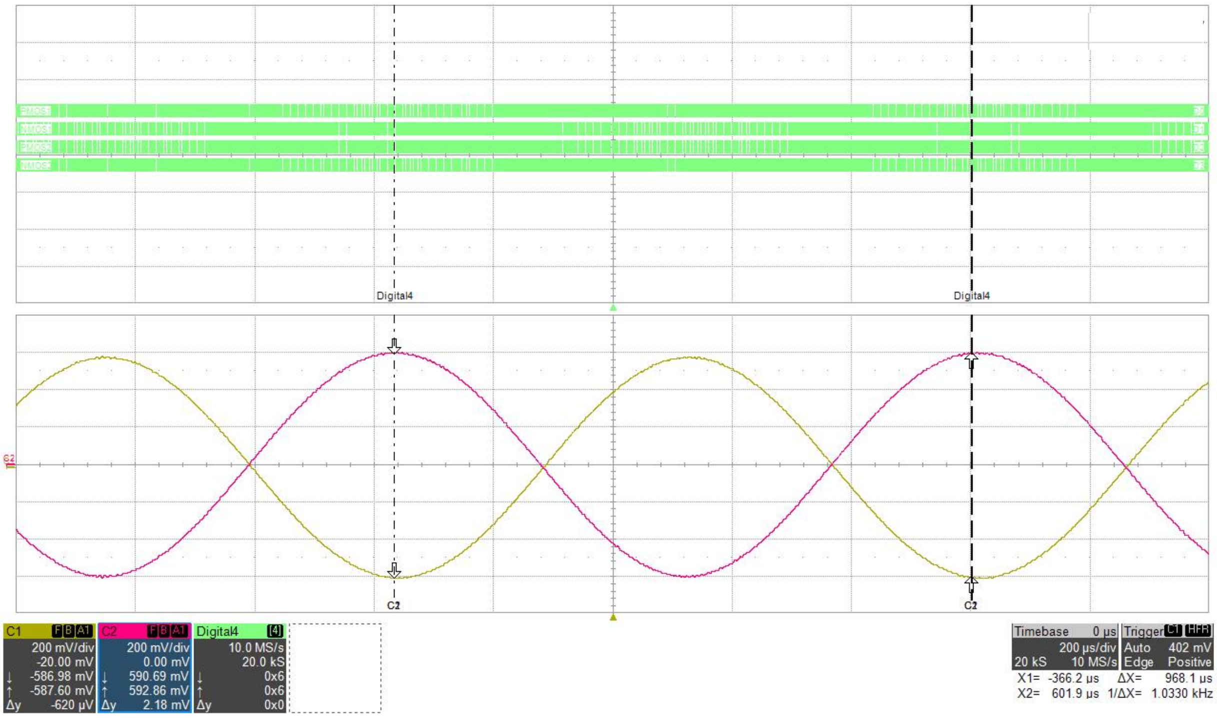
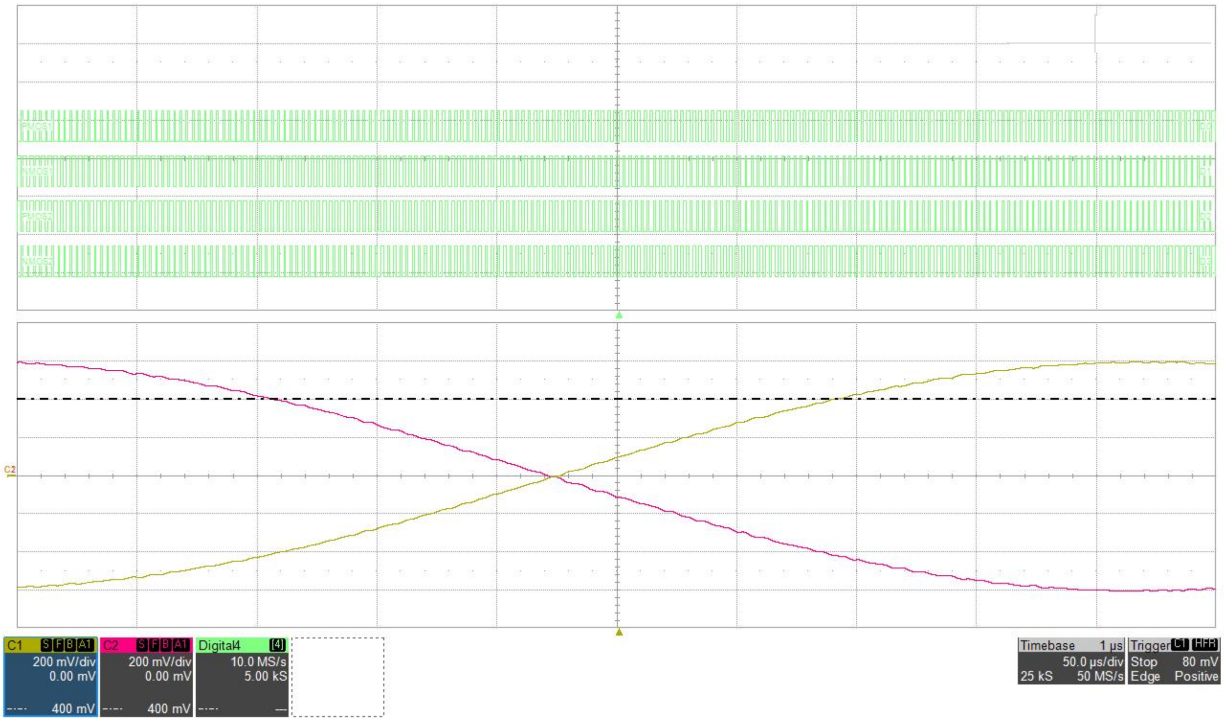


Figure 17. Temporal acquisition with digital commands and analog output (zoom)



5 Audio and processing performances

This section describes the audio and processing performances measurements.

5.1 Setup presentation

A comparison is made between the advanced and high-resolution timer using a ROM-embedded 1 kHz sinus to observe the main audio parameters (noise, distortion). Also the software processing duration using a GPIO to measure the required computation time is observed and compared.

5.2 Audio measurements

The MCU firmware integrates a simple method to reduce the amplitude of the audio signal. The user button is monitored to detect a press condition which applies a -6 dB attenuation on the output signal. The audio measurements are performed at the output of the H-bridge power stage using a dedicated audio instrument.

5.2.1 Conditions

Audio measurements have been performed at room temperature using a passive load (R = 8 Ω, L = 44 µH,C = 2 nF) to model a standard 8 Ω speaker. Interpolation factor = 8 (48 kHz to 384 kHz). PWM frequency = 384 kHz. PWM ramp amplitude is 128 for advanced timer and 12500 for HR timer. Power stage supply = 5 V.

5.2.2 Results

This section presents the results (values and graphs) of the measurements made.

Table 2. Advanced-timer performances (1 kHz frequency, H-bridge supply = 5 V)

Output level (dBR) 0 dBr = 9.2 dBV	THD+N (dB)	Noise (dB)	Dynamic range (dB)
0	-24	-46	46
-7	-69	-68	75
-13	-67	-67	80
-19	-64	-64	83
-25	-60	-60	85
-31	-54	-54	85
-37	-48	-48	85
-43	-42	-42	85
-49	-36	-36	85

Figure 18. Advanced-timer noise floor measurement

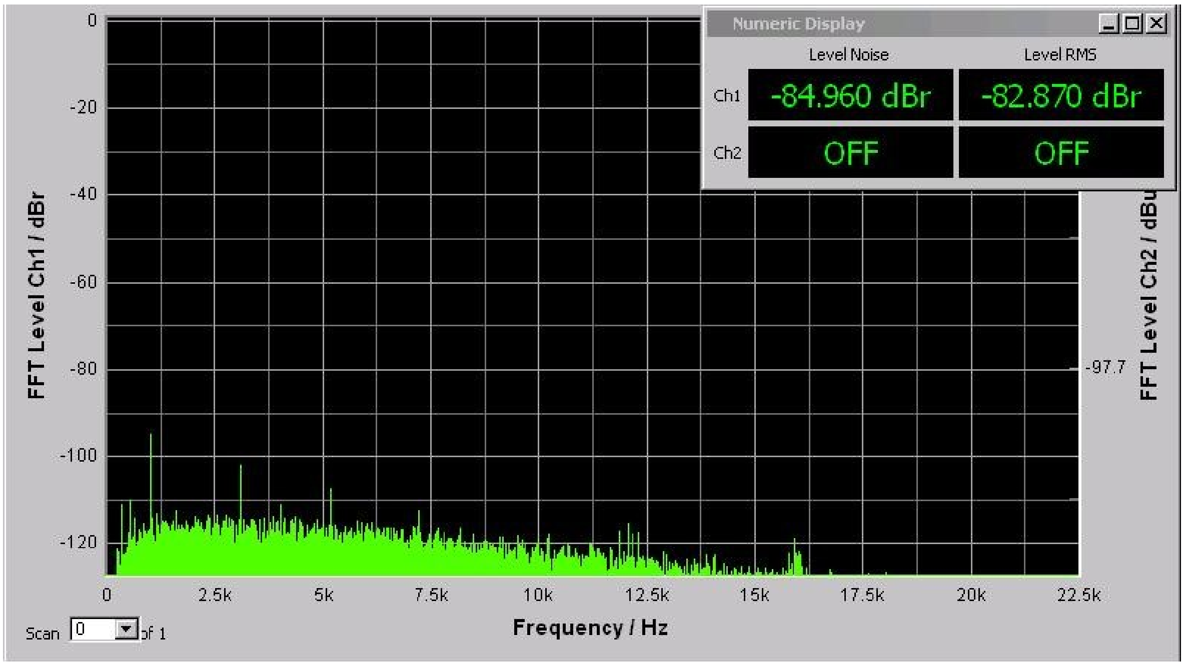
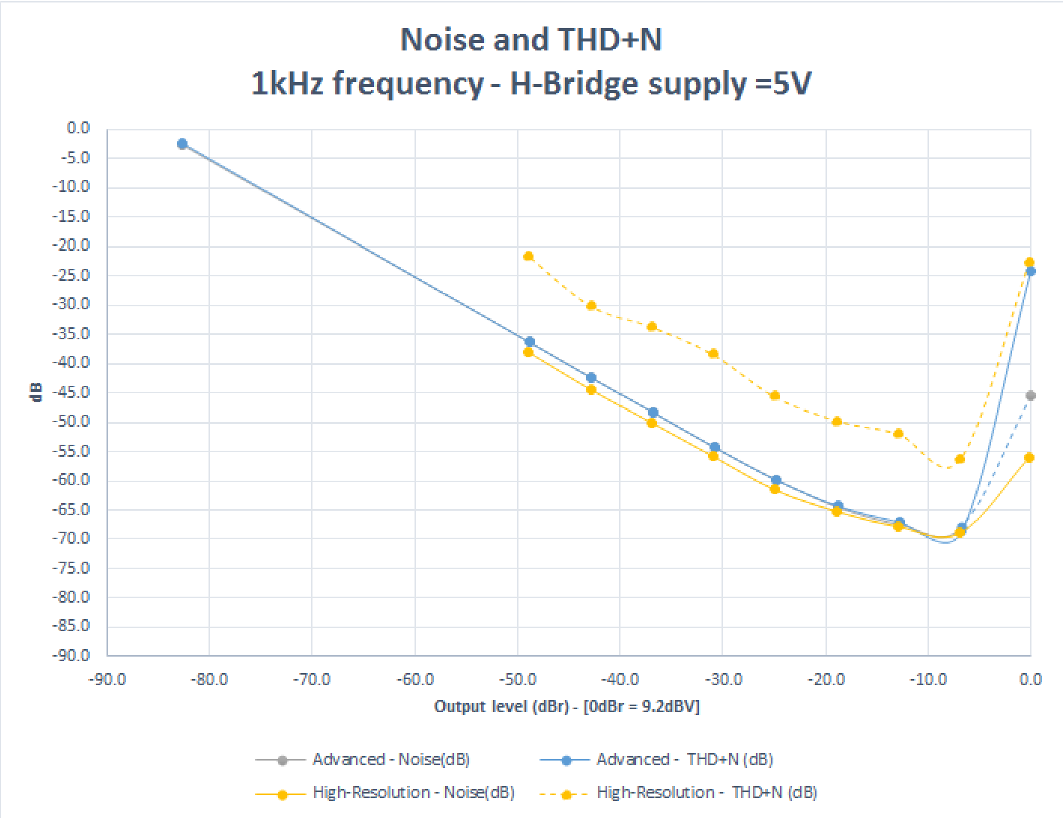


Table 3. High-resolution timer performances (1 kHz frequency, H-bridge supply = 5 V)

Output level (dBR) 0 dBr = 9.2 dBV	THD+N (dB)	Noise (dB)	Dynamic range (dB)
0	-23	-56	56
-7	-56	-69	76
-13	-52	-68	81
-19	-50	-65	84
-25	-46	-62	87
-31	-38	-56	87
-37	-34	-50	87
-43	-30	-44	87
-49	-22	-38	87

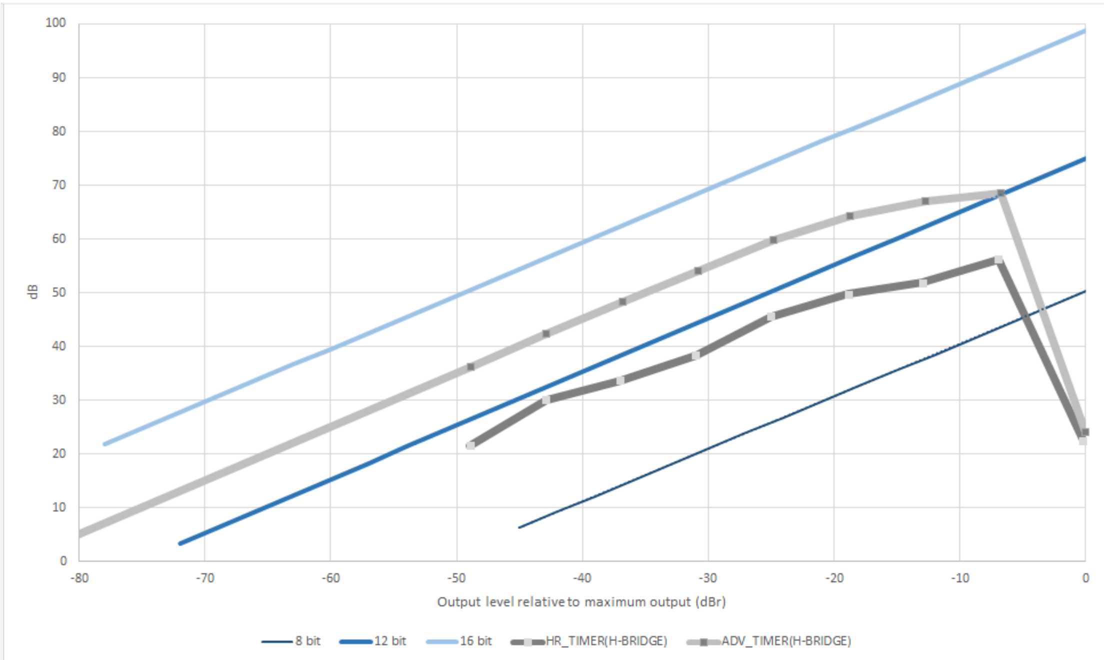
Figure 19. Noise and THD+N vs output level (high-impedance load)



The following figure explains the difference of the SINAD performances between the advanced timer (audio signal source is 16-bit width / reduced to 7-bit at the sigma-delta modulator output) and the high-resolution timer (audio signal source is truncated to 13.6-bit to achieve a PWM ramp frequency equal to 384 kHz using a PWM ramp amplitude equal to 12500; refer to Section 3.3 High-resolution timer). The truncation of the data width creates distortion due to the signal approximation. This phenomena is easily visible when comparing theoretical sinus plotted with various resolution widths and the Class-D audio performances at the H-bridge output. The additional noise and distortion present at external power stage output are due to several items such as:

- Timer programmable dead-time to avoid short circuit in the power stage circuitry
- Mismatch between power MOS in the power stage and current decay recirculation
- Non linearity of the speaker load (RLC model)

Figure 20. SINAD comparison between theoretical sinus and H-bridge output



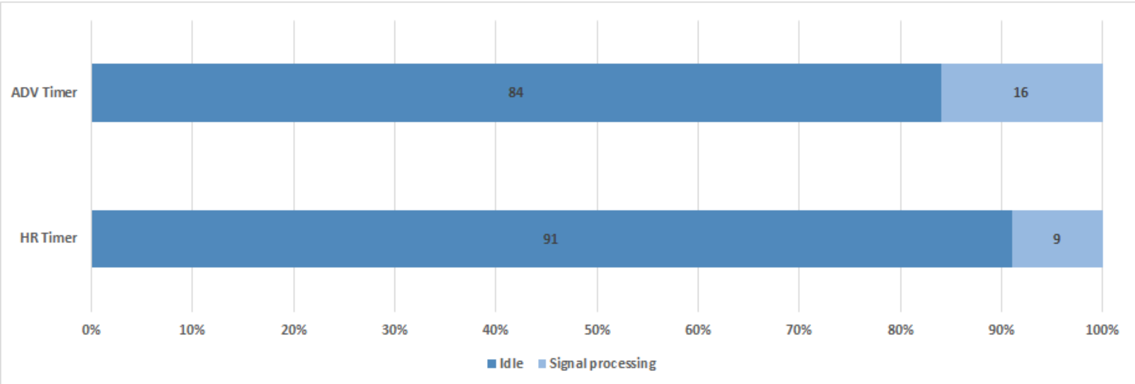
5.3 Computation time comparison

The following table shows the processing duration time required either using advanced timer or a high-resolution timer.

Table 4. Processing time comparison per timer peripheral type

Timer	Processing time (µs)	CPU (MHz)	Number of MCU cycles
High resolution	360	150	54 k
Advanced	623	150	93.4 k

Figure 21. Idle time comparison between high-resolution and advanced timer



The high-resolution timer required less MCU cycles to perform the embedded signal processing and also required a smaller software execution duration compared to the advanced timer implementation. The gain factor is slightly below 2 which allows more idle time for any other CPU activities such as display or monitoring process.

Performing a PWM modulator using a STM32 MCU device with a high-resolution timer peripheral is simpler, and costs less processing time. The advanced timer gives slightly better SINAD performances but with the drawback of requiring additional CPU resources due to the noise shaper.

This application proposal does not include standard signal processing algorithms such as dynamic range compressor or equalization which are usually implemented in consumer audio applications to achieve a maximum loudness and a good sound reproduction.

6 USB audio device application example

With the STM32 peripherals and libraries it is easy to connect the audio Class-D amplifier described in this application note (STM32 MCU + external H-bridge) to an external audio digital source. The amplifier source could be from various peripheral sources such as I2S, SPI, USB, Bluetooth® or ethernet. This section describes how to stream an audio content from a computer using the USB protocol with the audio device class in order to create a simple amplified USB sound card.

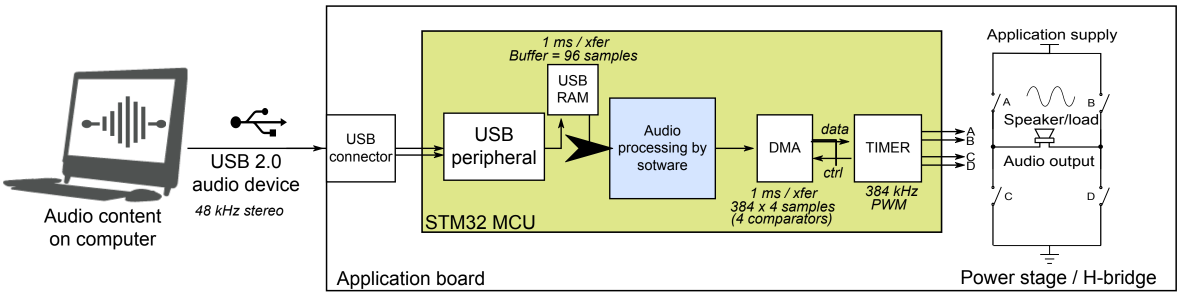
6.1 USB audio device bridge to Class-D audio amplifier

The audio format used is 48 kHz sampling rate with 16-bit sample width and stereo channel. The audio content is streamed from the USB audio source (computer) to the MCU using the USB audio device class and the USB peripheral. When the MCU is connected to the computer, the USB service is started and the audio USB device appears in the computer's operating system.

6.2 System presentation

This section describes a mono application using a single speaker so stereo-to-mono conversion is required to down-mix left and right channels in order to keep the entire audio content. The embedded audio processing software access to the USB RAM every 1 ms, audio samples are buffered and processed before the data transfer occurs to DMA and its associated timer peripheral.

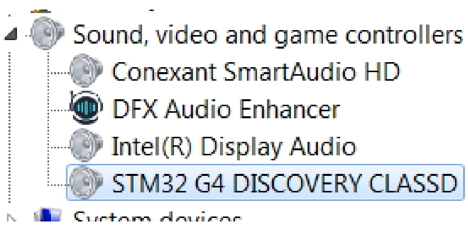
Figure 22. Application note complete system overview



6.2.1 USB audio device connection

Once the audio USB device is connected to the USB host, the USB enumeration is started and the new audio device appears in the host operating system.

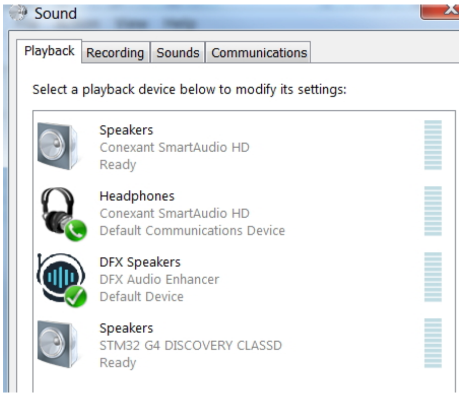
Figure 23. Audio USB device display in Windows® device manager panel



6.2.2 USB audio device playback

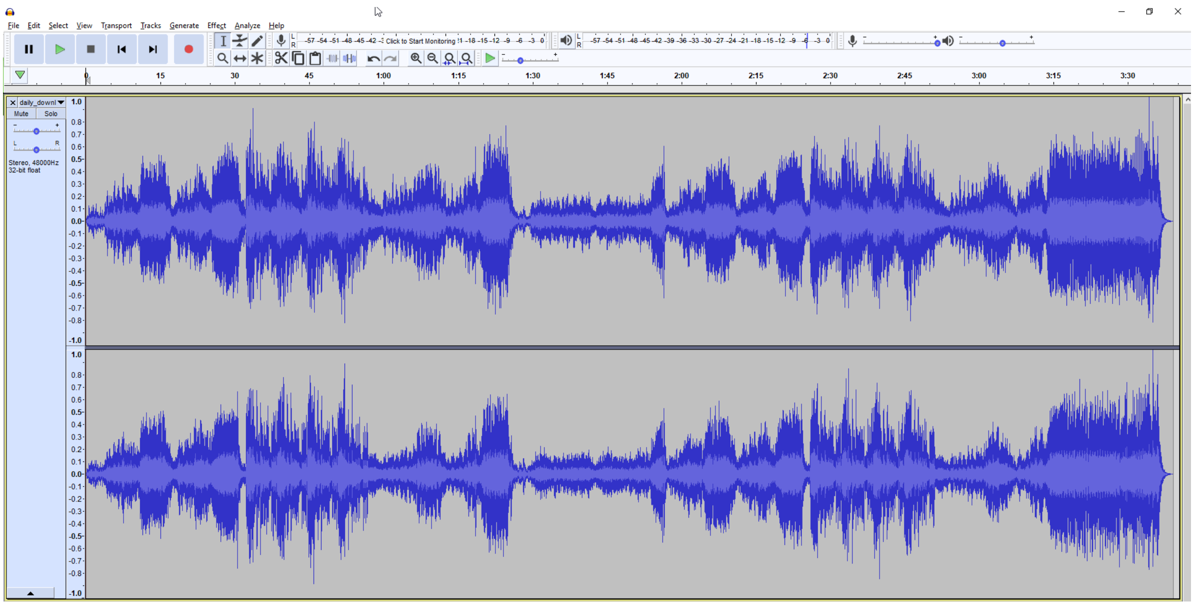
When the new playback audio is device-enabled in the USB host system, it can be chosen to initiate the audio stream to the STM32 MCU / Class-D amplifier.

Figure 24. **Class-D playback device in the Windows® Sound configuration panel**



And finally, the user can select any audio software application to start the audio playback to the STM32 MCU/ Class-D amplifier.

Figure 25. **Audacity-audio playback software screen-shot under Windows® operating system**



7 Conclusion

This application note demonstrates that STM32 peripherals may be combined to create new or advanced features such as a Class-D amplifier using an external H-bridge power stage using the USB audio device class. STM32 MCUs are available with various types of timer peripherals; the two most interesting types for an audio Class-D amplifier application are the advanced and high- resolution timers.

The advanced timer's advantage is its availability in most of the STM32 products while its drawback is the extra processing software cost due to the implementation of a sigma-delta modulator required to increase its low resolution (around 8-bit with 400 kHz comparison frequency).

The high-resolution timer is dedicated to high frequency comparison which is equivalent to 13.55-bit resolution with 400 kHz comparison frequency. It is available in a reduced list of STM32 products but implementing high-resolution comparison is easier.

Finally, the high-resolution timer can give acceptable performances without noise shaper, it makes this option easy to program and saves CPU resources. Those resources can be used to implement additional features such as DRC and equalization in order to improve the customer's audio experience.

Using the advanced timer with a noise shaper gives even better results but the implementation is more complex and requires more CPU resources. It leaves fewer resources to run additional signal processing, if needed.

Note as well that additional signal processing tasks may be performed with the new peripheral filter math accelerator (FMAC). This peripheral helps to offload the processor performing arithmetic operations on vectors, including a MAC unit and address generation logic.

Revision history

Table 5. **Document revision history**

Date	Version	Changes
20-Feb-2019	1	Initial release.
03-Mar-2020	2	Updated Section 3.2 PWM modulation with an advanced timer Updated Section 3.2.1 Timer peripheral clock frequency

Contents

List of tables

List of figures