Receiving S/PDIF audio stream with the STM32F4/F7/H7 Series

Introduction

The Sony/Philips Digital Interface Format (S/PDIF) is a point-to-point protocol for serial and uni-directional transmission of digital audio through a single transmission line for consumer and professional applications. The transmission of data can be done in several ways, by electrical or optical means.

The S/PDIFRX peripheral embedded in STM32 devices is designed to receive an S/PDIF flow compliant with IEC-60958 and IEC-61937, which define the physical implementation requirements as well as the coding and the protocol. These standards support simple stereo streams up to high sample rates, and compressed multi-channel surround sound, such as those defined by Dolby or DTS.

This application note describes electrical interfaces, to properly connect the S/PDIF stream generated by an external device to an STM32 device embedding the S/PDIFRX interface peripheral, since the voltage level of the S/PDIF line is not the same as that used in STM32 devices.

1 S/PDIF Interface

This document applies to Arm®-based devices.

arm

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

1.1 S/PDIF background

S/PDIF is an audio interface for transmission of digital audio data over reasonably short distances between modules of systems such as home theaters or hi-fi. S/PDIF is a single-wire serial uni-directional, self-clocking interface.

S/PDIF is based on the AES3 interconnect standard. S/PDIF and AES3 are compatible at the protocol level but their electrical characteristics differ, for exampe in terms of voltage levels and impedance. S/PDIF can carry two channels of uncompressed PCM audio, or compressed 5.1/7.1 surround sound (such as DTS audio codec data). However, due to limited bandwidth, it does not support uncompressed audio formats (other than 2-channel LPCM) such as Dolby True HD and DTS master.

S/PDIF doesn't specify any default data transmission rate. The device has to extract the clock from the input signal. This is achieved by use of bi-phase mark code that includes one or two transitions for each bit. The transmission rates typically used are 44.1 kHz for stereo CD audio, and 48 kHz for digital audio tape (DAT). The standards also support simple stereo streams up to a high sample rate (192 kHz).

The format has been standardized in the consumer domain in the form of IEC60958-3 (International Electrotechnical Commission, IEC), and in the professional domain as AES3 and IEC60958-4.

The main purpose of the S/PDIF format is the ability to transfer data between two pieces of digital audio equipment without going through an analog connection, which would imply a loss of quality. As a consequence, it is preferable to keep the signal in the digital domain as far as is as possible, and convert it to analog close to the amplifier. This one-way communication protocol, which allows transmission of digital data from a transmitter to a receiver, can be implemented using:

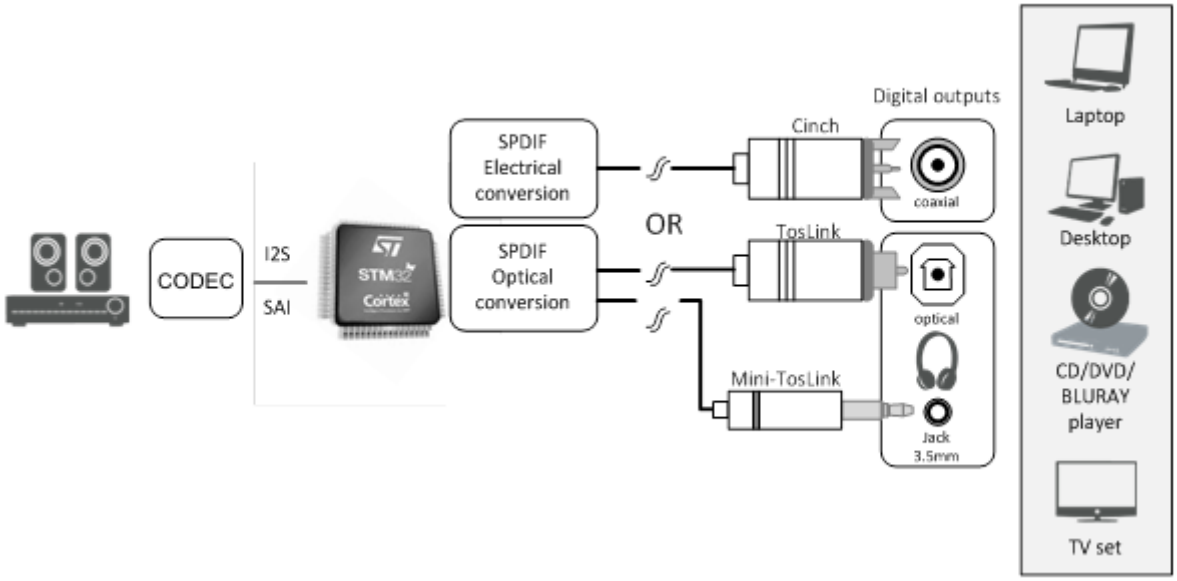
- electrical transmission, using a coaxial cable and cinch connectors
- optical transmission, using optical fiber cable and TOSLINK or mini-TOSLINK connectors

As shown in Figure 1. S/PDIF usage examples, most well-known consumer equipment is able to exploit S/PDIF transmission:

- CD, DVD or BLURAY players
- PC sound cards
- TV sets
- …

In STM32 devices, the S/PDIF can also be used to connect different devices. In such cases there is no need for electrical adapters.

Figure 1. S/PDIF usage examples

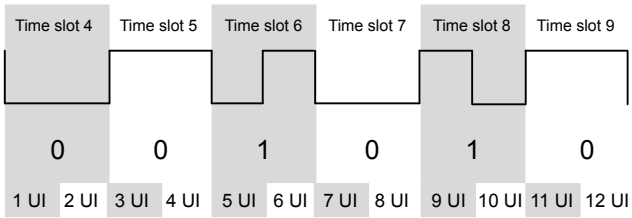


1.2 S/PDIF format

S/PDIF is a serial, bi-phase-mark encoded data stream, where the data is transmitted in frames in order to identify the various components of the bitstream.

The bi-phase-mark encoded data stream has one or two transitions for every transmitted bit, allowing extraction of the clock signal from the data signal itself.

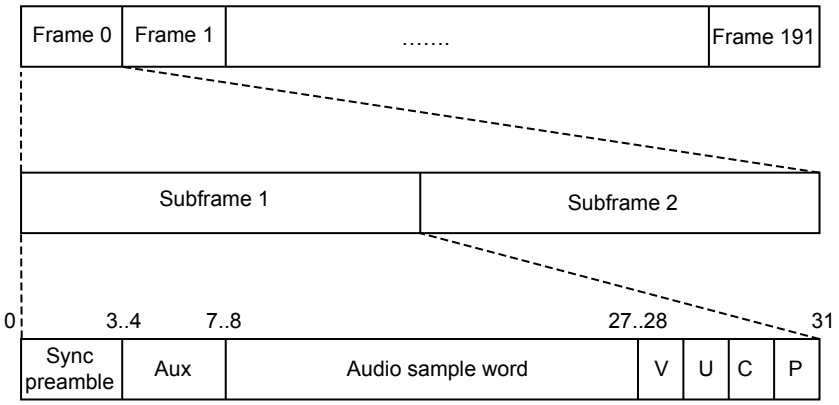
Figure 2. Bi-phase encoding in an S/PDIF stream



Each S/PDIF block is made up of 192 frames. The S/PDIF frame consists of two sub-frames. The sub-frame is a concatenation of:

- preamble – a synchronization pattern used to identify the start of a 192-frame block or sub-frame
- 4-bit auxiliary data (AUX)
- 20-bit audio data (24-bit when combined with AUX)
- validity bit - indicates if the data is valid
- user bit – over 192 frames, this forms a user data block
- channel bit – over 192-frames, this forms a channel status block
- parity bit – used to maintain even parity over the sub-frame (except the preamble)

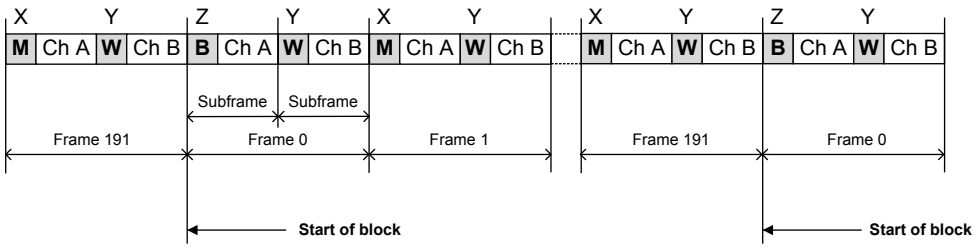
Figure 3. **S/PDIF block**



1.2.1 Synchronization preambles

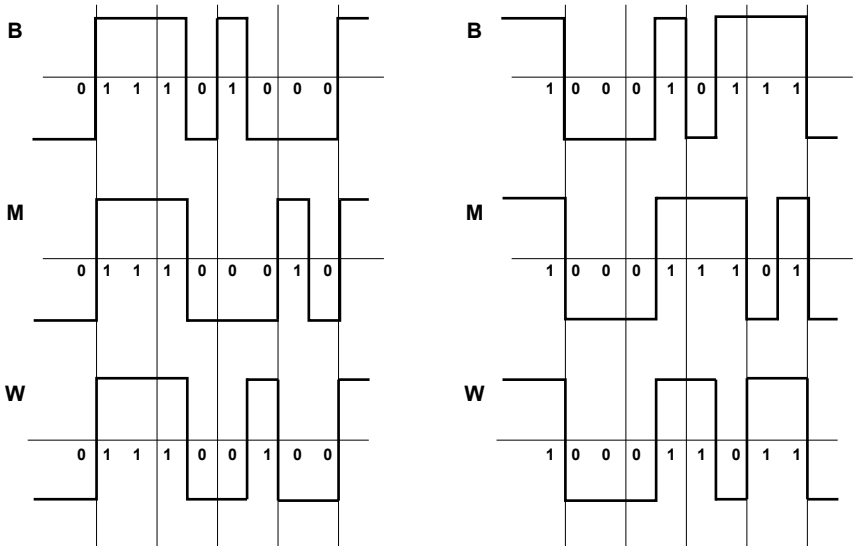
There are three kinds of preamble, the order of which defines the frame. In the consumer domain they are called “B”, “M” and “W”, while in the professional domain they are called “Z”, “X” and “Y”. Preamble “B” is only used in the first sub-frame of the very first frame of the block. All the other frames start with preamble “M”. In the second sub-frame of each frame the “W” preamble is used.

Figure 4. **S/PDIF block format**



1. For historical reasons preambles “B”, “M” and “W” are, for use in professional applecations, referred to as “Z”, “X” and “Y” respectively.

Figure 5. **Preambles**



1.2.2 Audio data

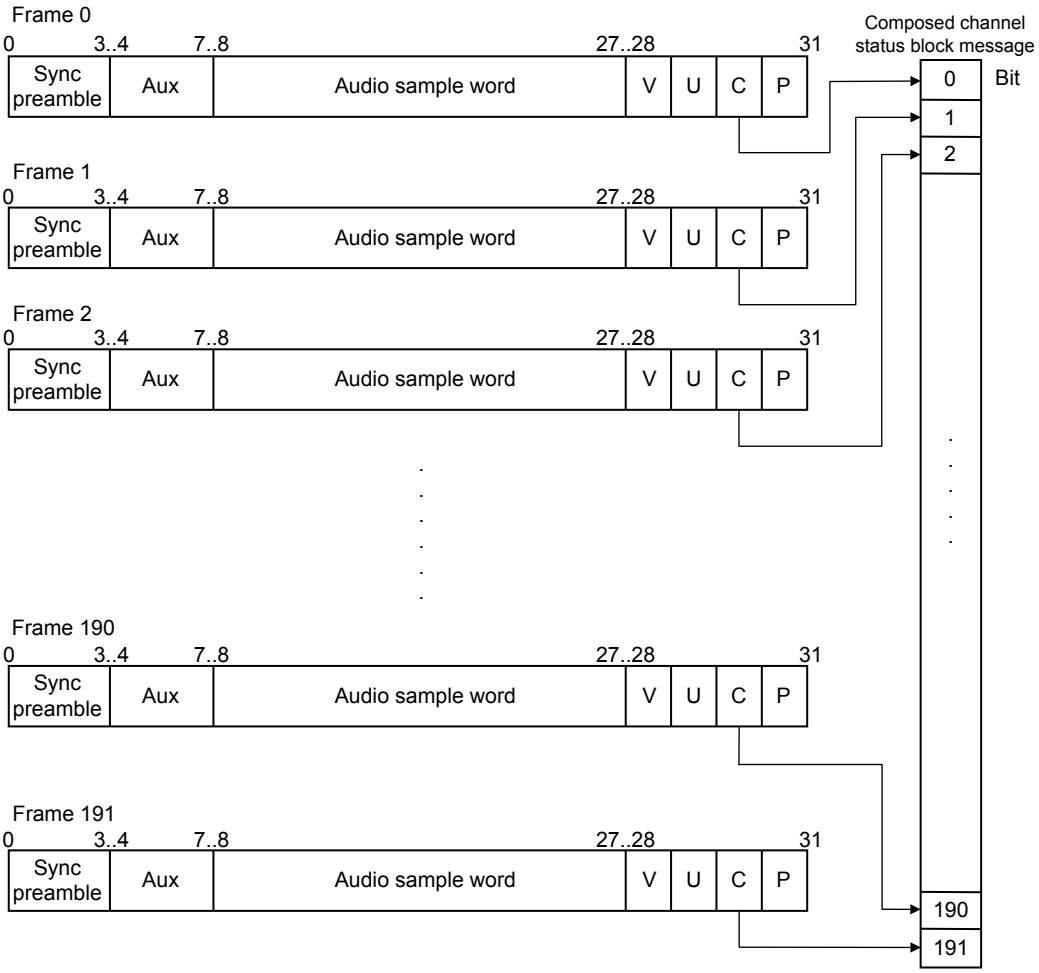
The audio sample has a maximum length of 24 bits/sample. If the full size of the audio data is not used, the unused bits are equal to zero. The auxiliary bits can be used to extend the audio sample. If unused, they are also equal to 0. For example, a CD player uses only 16 bits.

1.2.3 Channel status block

The channel status bits across the block together compose information related the audio channel. The channel status is identical for both subframes (except for the channel number).

The first bit is always transmitted in the frame with preamble “B” (consumer domain) or “Z” (professional domain).

Figure 6. Channel status block composition



1. For both subframes within one frame the channel status information is identical

The most important bit is bit 0, which carries information about whether the data are dedicated for consumer or professional use. Then, the channel status block contains information about the transmitted audio data, such as: data format, sampling frequency, number of channels, copyright information and so on.

1.2.4 User data block

The user data block is composed in the same way as the channel status block, and can carry any information of maximun size 192 bits. It can contain auxiliary data for end users such as running time, song name, and so on.

As the content is not defined in the S/PDIF standard, it might be ignored by some devices.

1.2.5 Hardware specification

Two types of transmission lines are defined by the S/PDIF standard: 75 Ω coaxial or optical fiber lines.

Table 1. S/PDIF characteristics

-	S/PDIF coaxial	S/PDIF optical
Interconnecting cables	75 Ω ± 5% (l < 10 m) or 75 Ω ± 35% (l > 10 m)	
Line driver V_{out}	0.4 V_{PP} . . 0.6 V_{PP}, < 0.05 V_{DC}	
Line receiver Vin	0.2 V_{PP} .. 0.6 V_{PP}	
Rise and fall times	< 0,4 UI (unit interval)	
Connector	RCA	TosLink/mini TosLink
Max. distance	15 m	10 m

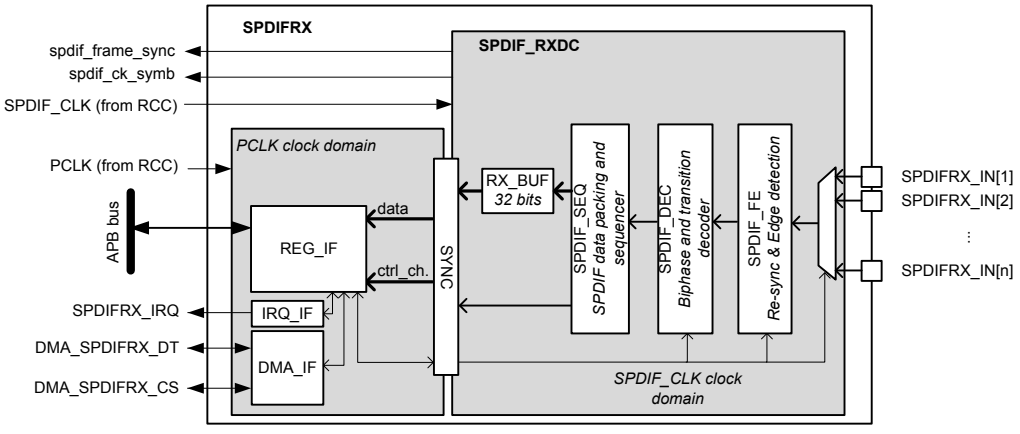
1.3 SPDIFRX overview

The SPDIFRX peripheral embedded in the STM32 (STM32F4, STM32F7 and STM32H7 Series, see the applicabe device datasheet) is designed to receive an S/PDIF flow compliant with the IEC-60958 and IEC-61937 specifications. This receiver provides all the necessary features to detect the symbol rate, and decode the incoming data stream received from one of the SPDIFRX_IN inputs. In addition the SPDIFRX separates the data stream from the channel status (C) and the user (U) information.

The received data stream can be stored in the MCU memory using either a dedicated DMA interface (DMA_SPDIFRX_DT) or interrupt services. In the same way, the channel status (C) and user (U) data information can be stored in the MCU memory either by using a dedicated DMA interface (DMA_SPDIFRX_CS) or interrupt services.

In most cases, once processed, this data is sent to one or several audio codecs using, for example, an SAI peripheral. The codec adapts the signal in order to provide a sound through loudspeakers or a headset.

Figure 7. Block diagram



The SPDIFRX peripheral also provides two kinds of signals: the symbol clock (spdifrx_ck_symb) and the sampling clock (spdifrx_frame_sync) extracted from the incoming stream. These signals can be used to synchronize the received stream with other devices. It is important to note that these signals might have appreciable jitter due to the resampling of the incoming stream with the S/PDIF internal clock.

1.3.1 SPDIFRX synchronization and decoding

The received signal on the selected SPDIFRX_IN[x] input, is first re-sampled using the SPDIF_CLK clock (acquisition clock). Then filtering is applied in order to cancel spurs, and an edge detector detects the level transitions.

The resulting signal goes to the decoder block, which performs the synchronization and the biphase decoding.

Once enabled, the SPDIFRX tries to synchronize to the selected input stream. If the SPDIFRX is not properly synchronized, it is not able to properly decode the incoming data. The synchronization and decoding is based on measurements of the time intervals between consecutive edges.

The synchronization is performed in two steps:

- The COARSE synchronization phase:

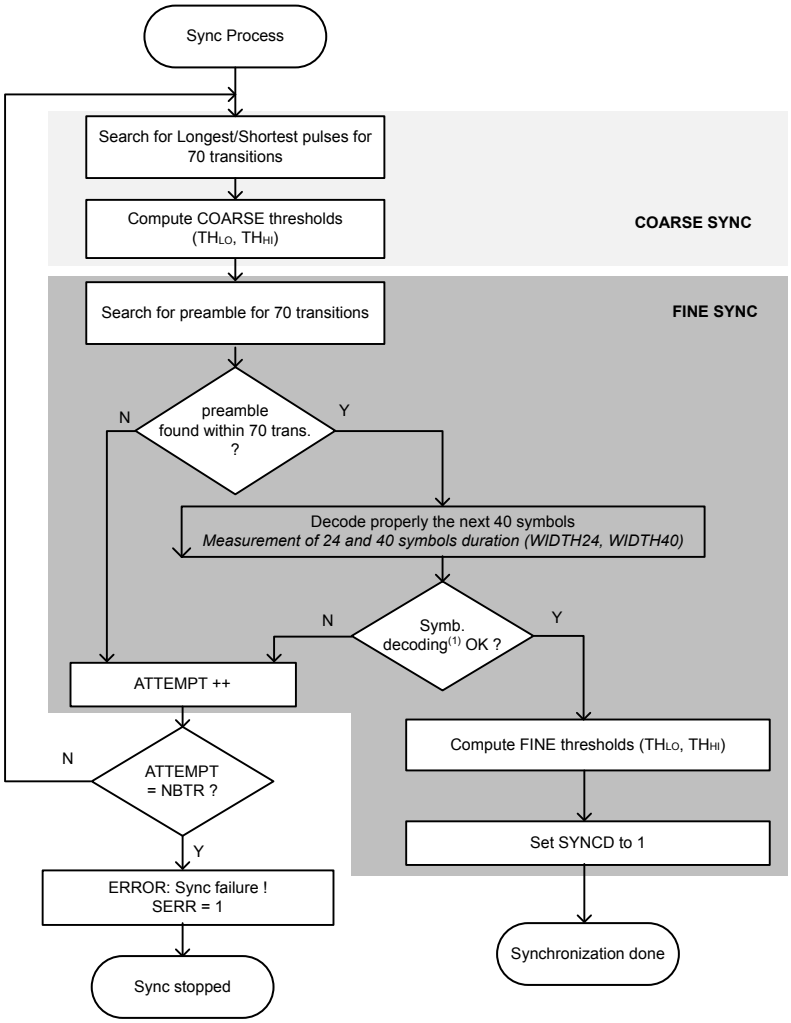
In this phase the preambles are used to estimate the symbol length and find the preamble boundary. This synchronization is not accurate and is sensitive to the quality of the received signal, because the time interval measured in this phase is short and it is not averaged. As shown in the figure below, the COARSE synchronization searches for the shortest and longest time interval between two transitions. This COARSE synchronization is normally performed once, but several attempts can be programmed if the signal is noisy.

- The FINE synchronization phase:

In this phase the symbol length is estimated with a longer time window, making this synchronization less sensitive to signal quality. This synchronization is performed on every frame, in order to remain locked.

The synchronization process is shown in the Figure 8. Sequencer.

Figure 8. Sequencer



1. The decoding is considered correct when the symbols are properly decoded, and the preamble occurs at the expected position.

Once the SPDIFRX is synchronized, the data is decoded by measuring the time-interval between consecutive transitions.

Please refer to the SPDIFRX specification for additional details.

2 Interfacing a S/PDIF coaxial to SPDIFRX

As shown in Table 1. S/PDIF characteristics, the received S/PDIF signal needs to be adapted to levels accepted by the microcontroller input pads. An electrical adapter is needed in order to increase the signal amplitude and adjust the DC value to match the V_{IL} and V_{IH} PAD characteristics.

2.1 Challenges for S/PDIF signal conversion

The electrical adapter must avoid degrading the S/PDIF signal. Particular care must be taken to avoid jitter, and duty cycle degradation.

Note that as the symbol decoding of an S/PDIF steam is based on the measurement of the time-interval between transitions, the decoding quality is by construction sensitive to duty-cycle and cycle-to-cycle jitter degradation. SPDIFRX is not sensitive to the signal polarity, and only transitions are taken into account.

2.2 Electrical signal adapter

S/PDIF devices provide signals between 0.2 V_{PP} and 0.6 V_{PP} into a 75 Ω load over an unbalanced circuit. The electrical adapter must amplify the received S/PDIF signal to a logic level accepted by microcontrollers. This signal conversion is the purpose of the electrical interface which must be developed.

The minimum gain required for the signal conversion mainly depends on the STM32 application power supply (V_{DD}) assuming that the incoming S/PDIF signal could be 0.2 V_{PP} worst case.

The required amplification gain in dB is:

$$G\left(dB\right) = 20 \cdot \log \frac{V_{DD}}{V_i}$$

with V_i = 0.2 V

Table 2. Amplification gain versus application power supply shows the amplification gain required for a few typical (noticeable) MCU supply voltages (V_{DD}).

Table 2. Amplification gain versus application power supply

V_{DD}	Amplification gain required
1.7 V	x8.5 (18.5 dB)
2.5 V	x12.5 (22 dB)
3 V	x15 (23.5 dB)
3.3 V	x16.5 (24.5 dB)
3.6 V	x18 (25 dB)

This bandwidth is critical for S/PDIF applications where the S/PDIF sample rate can reach 192 kHz, which corresponds to a symbol rate of 12.288 MHz.

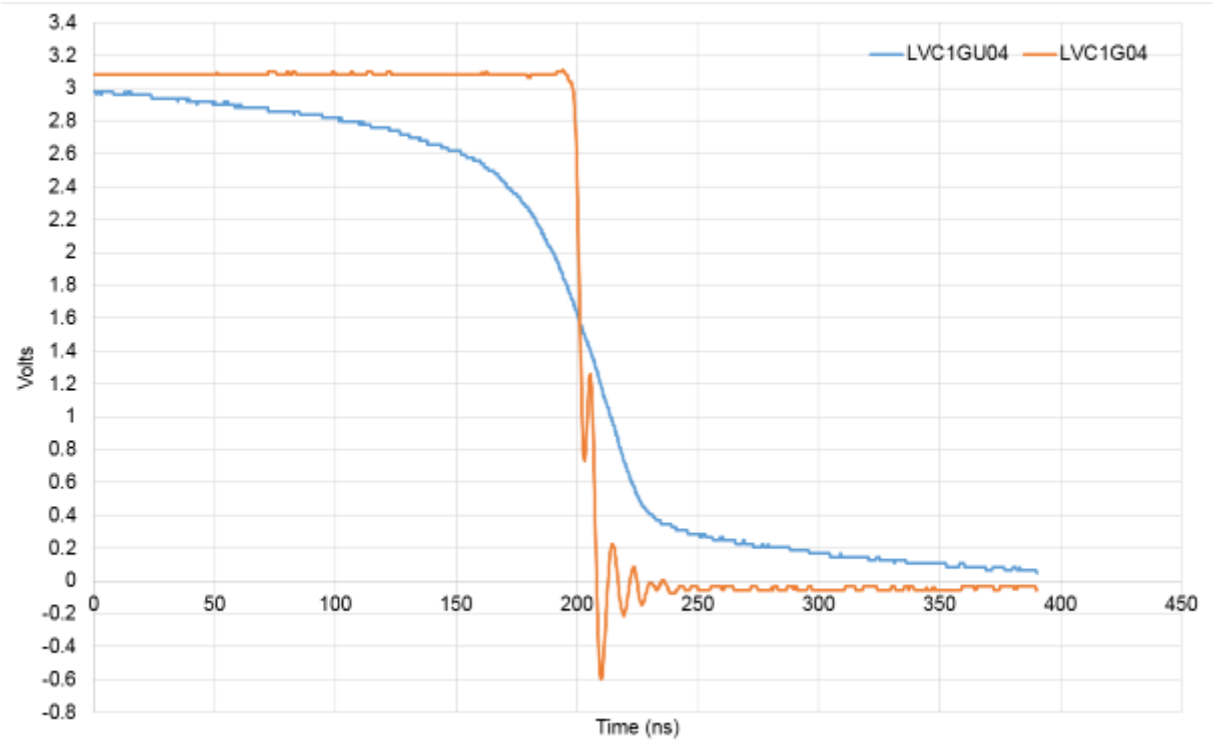
2.2.1 Using simple inverters

Using logical inverters is a simple and cost effective way to adapt the S/PDIF signals. In this case the input signal is considered as an analog signal.

The first thing to consider is the choice of inverter among two main types: buffered or unbuffered. Compared to buffered inverters, unbuffered inverters have a single inverting stage, and a significantly lower open loop gain.

Figure 9. Output responses shows the output response of LVC1GU04 (unbuffered) and LVC1G04 (buffered) inverters to a ramp signal, in open loop mode. The transition region is quite linear for the LVC1GU04, and very steep for the LVC1G04. For the LVC1G04, some instability in the transition region can be seen, due to the slow slope of the ramp signal and the strong open loop gain, making it very sensitive to noise in the transition region.

Figure 9. Output responses



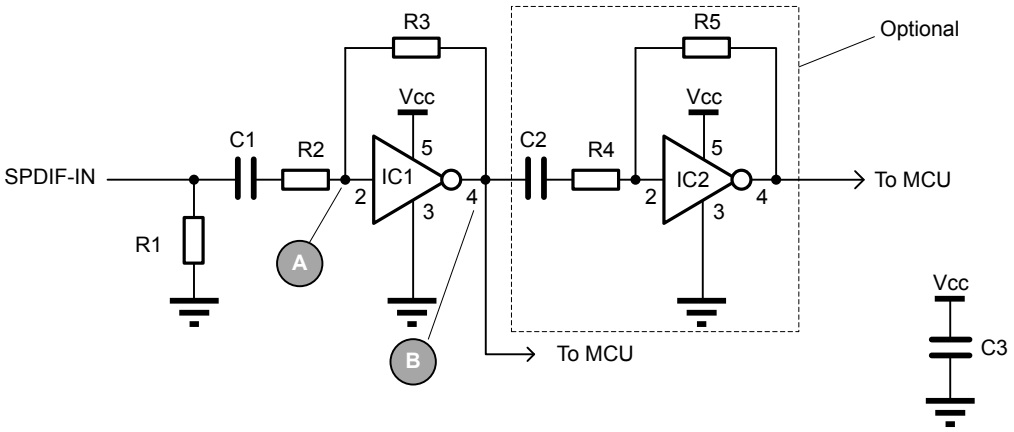
For our application, both buffered and unbuffered types can do the job, however, unbuffered inverters are less sensitive to external component values. Hence the proposed implementation is based on unbuffered inverters.

A possible implementation, built around one or two inverters, is shown in the figure below. The second stage can be omitted if the amplitude provided by the first stage is sufficient. This depends on the targeted S/PDIF frequencies, the capability of the inverter used and the VDD voltage used for the STM32.

For example, if the application targets support of S/PDIF streams up to 96 kHz (that is, symbol rate = 6.144 MHz), a single stage is generally enough. If the application needs to support up to 192 kHz (that is, symbol rate = 12.288 MHz), a dual stage is safer.

Proposed implementation

Figure 10. Variant A – implemented in STM32F769I Discovery board



The components can have the following values:

Table 3. Proposed component values

Component	Value	Unit
IC1, IC2	SN74LVC1GU04	-
R1	75	Ω
R2, R4	220	Ω
R3, R5	100	kΩ
C1, C2	4.7	nF
C3	100	nF

Resistor R1 terminates the S/PDIF cable providing the signal.

Capacitor C1 (C2) blocks the DC value of the incoming stream. Its value is not critical, though a signal in the range of 500 kHz to 12.288 MHz must be transmitted without attenuation. Note that if the value of this capacitor is too big, the DC setting time at the IC1 input increases.

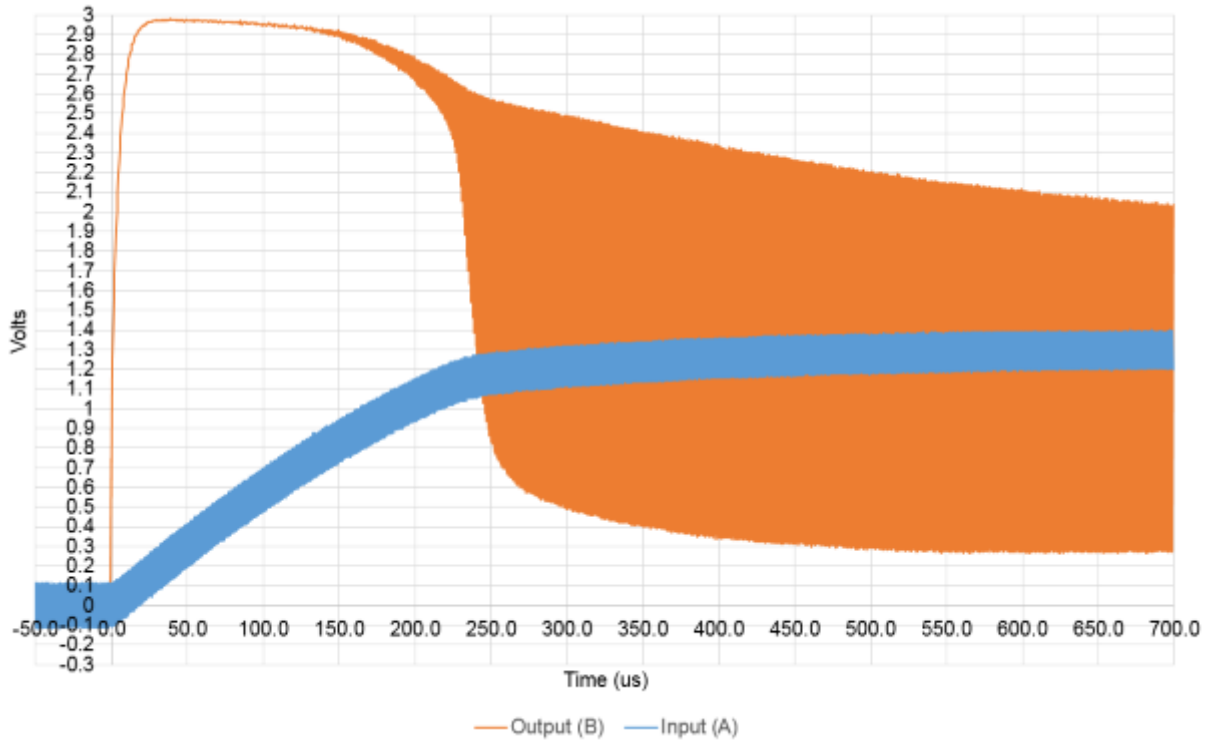
Figure 11. DC setting time shows the DC setting time at A. After power-up, the signal at the inverter input is close to 0 Volts, and the inverter output goes to a high level following the settling time of the supply. The inverter charges capacitor C1 through R1, R2 and R3. In our example, R1 and R2 can be neglected compared to the value of R3. When the DC voltage at the inverter input closely approaches the transition point, the inverter starts to amplify the AC component of the input signal.

The time constant managing the DC offset settling time can be approximated by the following equation:

$$V_{dc}(t) = V_f \cdot \left(1 - e\left(\frac{-t}{R3 \cdot C1}\right)\right)$$

Where V_f is the final value, which can be approximated to 3 V in Figure 11. DC setting time. This formula is not completely true because in realty the DC component at the inverter output is decreasing when the DC value at the inverter input is approaching the transition level.

Figure 11. **DC setting time**



Resistors R2 (R4) and R3 (R5) are also used to limit the gain of the inverter. The open loop gain (G_{OL}) measured on the SN74LVC1GU04 sample used for this document is quite good at about 36 (that is, 31 dB).

The formula below gives the inverter gain:

$$G = -\frac{G_{OL}}{1 + \frac{R2}{R3} \cdot (1 + G_{OL})}$$

The current gain is limited to 33 (that is 30.4 dB). The measured value is 32.8.

It is important to notice that when no signal is connected to the electrical adapter, oscillations or instabilities may appear at the output under certain conditions.

If the inverter output is high, capacitor C1 is charged, and the inverter input voltage increases until it is sufficient to toggle the output. At this moment the input voltage decreases until the output again toggles to a high level.

Controlling the gain avoids spurious oscillations when no signal is connected.

In the Figure 12. Inverter output, the upper plot is obtained with an LVC1G04, with R3 = 3 MΩ, and the lower plot is obtained with an LVC1GU04 and R3 = 100 kΩ.

The strong gain of the LVC1G04 brings about instability on the inverter output.

Figure 12. **Inverter output**

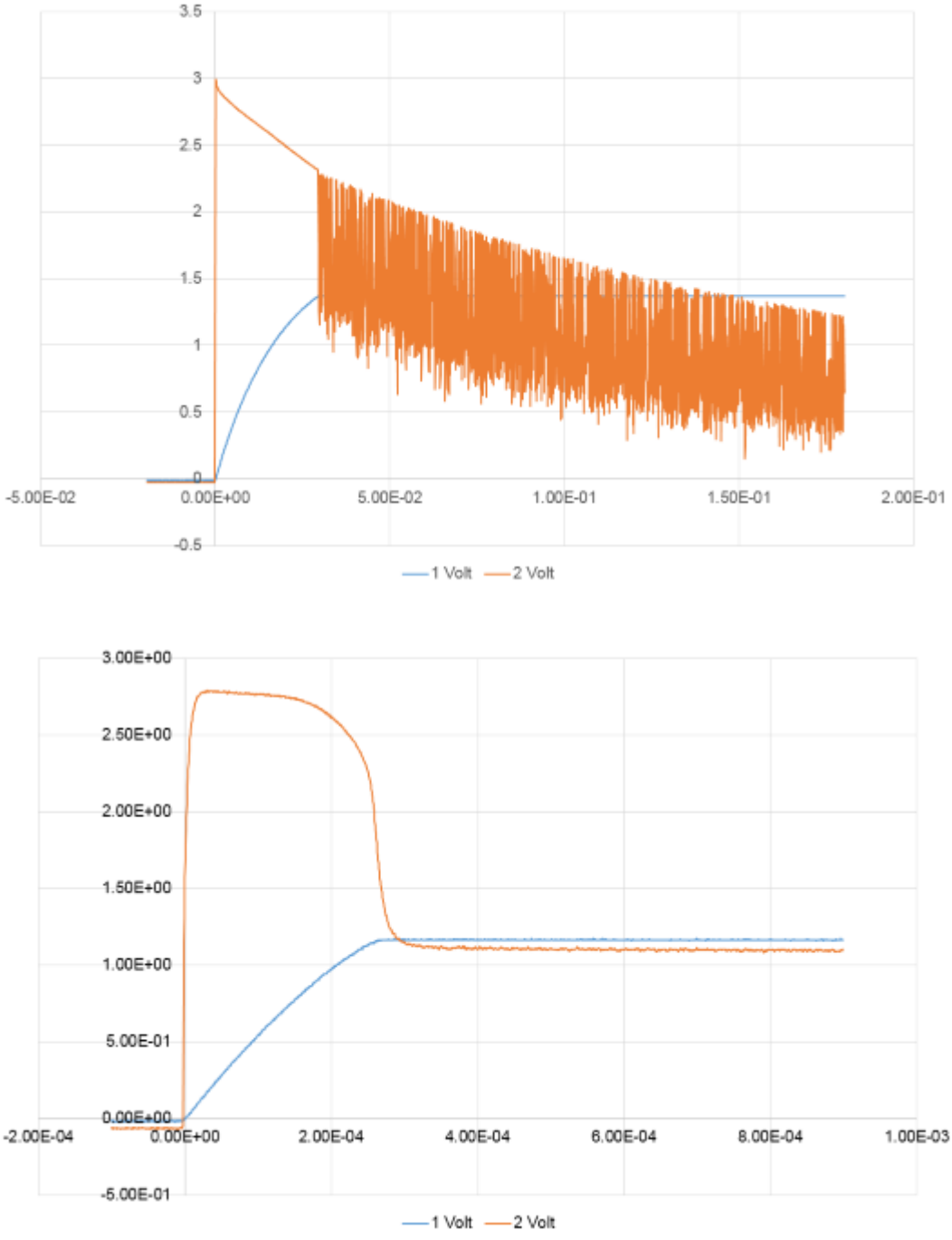
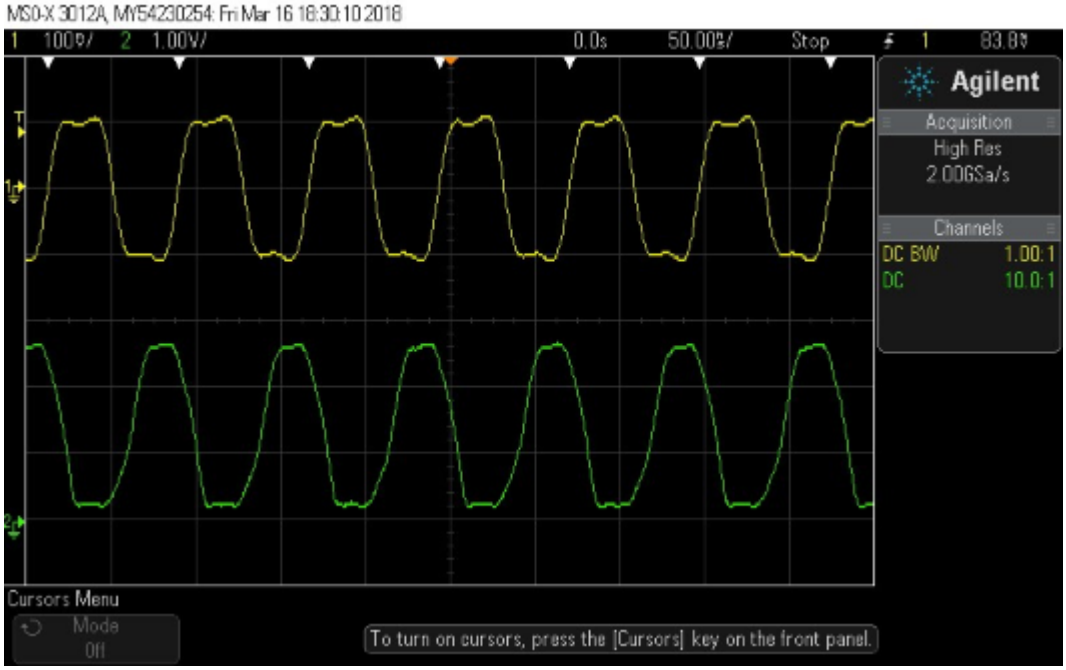


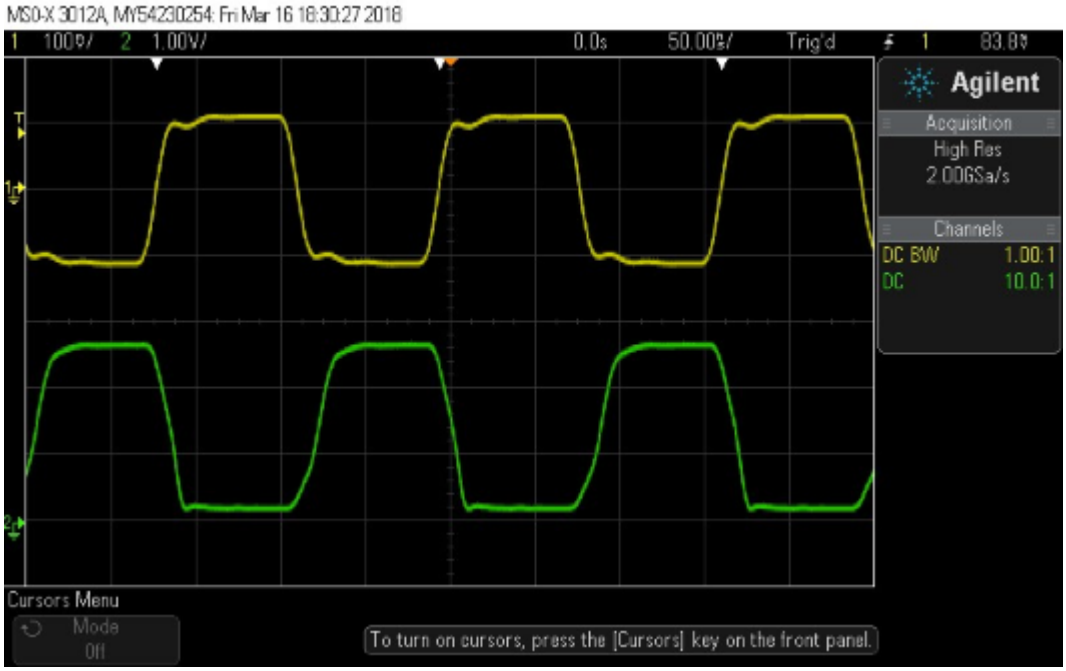
Figure 13. Single inverter stage at 13 MHz and Figure 14. Single inverter stage at 6 MHz show the resulting output signal of a single inverter stage at 13 MHz and 6 MHz.

Figure 13. **Single inverter stage at 13 MHz**



1. The yellow waveform is the input signal and the green waveform is th output signal.

Figure 14. **Single inverter stage at 6 MHz**



1. The yellow waveform is the input signal and the green waveform is th output signal.

Power supply sensitivity

Finally, capacitor C3 filters the supply voltage as much as possible . It should be placed as close as possible to IC1 and IC2.

If the supply voltage is improperly filtered, ripples in the supply voltage can translate into jitter, for two main reasons:

- the propagation delay of the inverter is sensitive to ripple in the supply voltage
- the open loop gain is sensitive to ripple in the supply voltage

A variant of the adapter proposed in Figure 10. Variant A – implemented in STM32F769I Discovery board keeps the same first input stage and just adds a second inverter stage if the signal provided by the first stage is too weak.

Figure 15. Variant B – simplified version

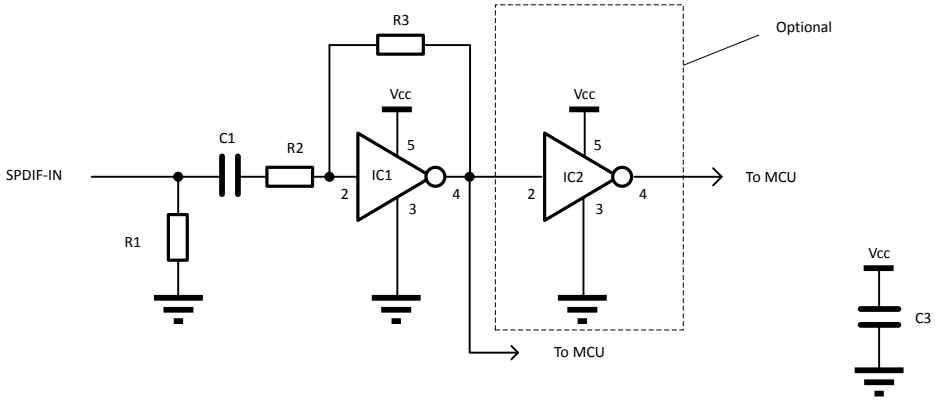
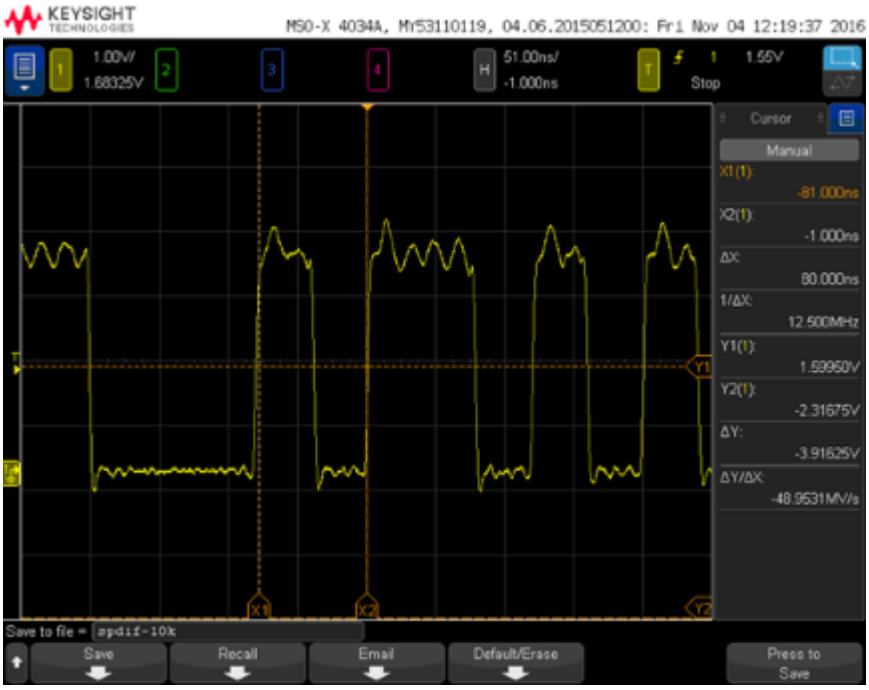


Figure 16. Output of the second stage shows the signal obtained at the second stage output, for an audio bitstream at 192 kHz.

Figure 16. Output of the second stage



1. The yellow waveform is the output signal.

2.2.2 High-speed differential line transceiver

This option uses a high-speed differential line driver and receiver such as an SN65LVDM175. The device is more costly than two inverters, but this is compensated by its simplicity of use. The device is designed for signaling rates of 400 Mbit/s.

Figure 17. Schematic for transceiver receiver mode

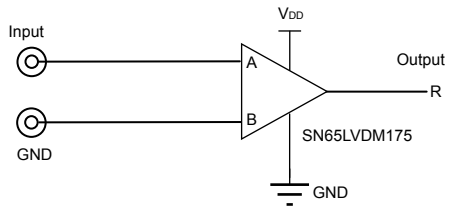


Figure 18. Receiver input and output waveforms shows the signal obtained at the second stage output, for an audio bitstream at 192kHz.

Figure 18. Receiver input and output waveforms

2.2.3 Fast comparator

This option uses a fast comparator such as the LT1713.

In this implementation, the DC component of the input signal is extracted by means of R4 and C3, and connected to the minus input as the threshold signal.

Figure 19. Schematic for fast comparator mode

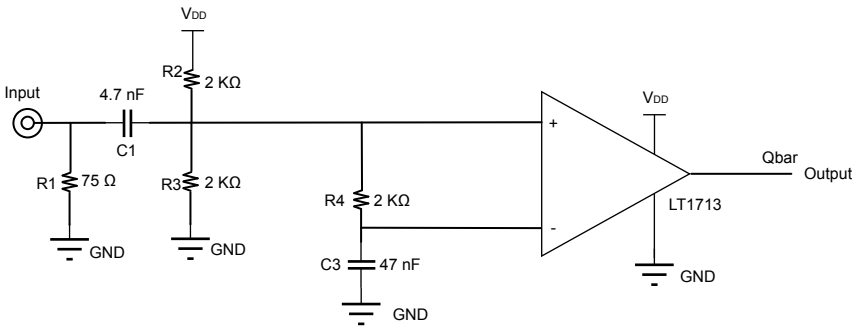


Figure 20. Comparator input and output waveforms

1. The yellow waveform is the input signal and the green waveform is the output signal.

In this case hysteresis (implemented, for example, by a positive feedback loop) is not recommended as it could influence the distance between the signal edges.

To achieve sufficient output-signal quality, a clean and properly decoupled supply is recommended.

2.3 Optical connection

An optical cable is another means of digital audio signal transfer. The optical system for transfer of S/PDIF signals was created by the Toshiba Corporation, which named the system TOSLINK.

The optical fiber in TOSLINK cables can be implemented in different materials. While plastic optical fiber is cheaper, glass or silica optical fibers have lower losses. Because of attenuation of the light, if the distance between devices is more than approximately 7 to 10 meters, the use of coaxial cable is recommended instead. Also, the fiber core of a TOSLINK cable may be permanently damaged if tightly bent.

One advantage of optical fiber over coaxial cable is its immunity to ground loops and RF interference. The drawback however is that it inherently adds jitter. TOSLINK is quite slow, and the slower the rise time of the data link, the more jitter is added.

2.3.1 Proposed implementation

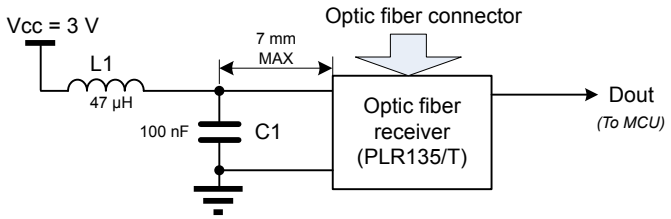
To connect the cable to the S/PDIF-RX, first, the user must transform the optical signal to a digital electrical signal. Converters are available on the market that can do the required conversion. For example:

- Cliff Electronics - FC684205R
- Toshiba - TORX177L
- Everlight - PLR135/T

These products provide data rates up 15 MHz and typical application rise times of 10 ns. All these products require an external supply.

Figure 21. Schematic used for optical connection shows a sample circuit of a TOSLINK device.

Figure 21. Schematic used for optical connection



Of the three converters above, only the PLR135/T provides a 3-Volt output. The other two devices provide 5-Volt outputs. When connecting to STM32 devices, care must taken to choose a 5 V tolerant pin.

3 Conclusion

This document shows several ways to implement an electrical interface for S/PDIF signals, based on the use of inverters, or on the use of more application-specific components such as comparators.

For inverter-based implementations, the choice of an unbuffered version is recommended, as they are less sensitive to external components. Buffered inverters can also work, provided that the gain is properly controlled.

Whichever solution is adopted, this electrical interface must be implemented with care in order to avoid duty-cycle degradation, and jitter. For example, the supply of the electrical or optical interfaces must be properly decoupled, and attention must be paid to the routing of the signals to the microcontroller.

4 List of references

For further details, refer to following documents:

- IEC 60958-3: Digital audio interface Part 3: Consumer applications.
- IEC 60958-4: Digital audio interface Part 4: Professional applications.

Revision history

Table 4. **Document revision history**

Date	Revision	Changes
05-Jun-2018	1	Initial release.
26-Jun-2018	2	Changed confidentiality classification

Contents

List of tables

List of figures