Application Note

Audio Signal Detector

AN-CM-319

Abstract

This application note describes the implementation of an audio signal detector with the SLG47512. The design can detect human speech or music and can ignore single tone noise or flat random noise. The audio signal detector can be used in safety services or to save energy in audio decks.

This application note comes complete with design files which can be found in the References section.

Contents

Figures

Tables

1 Terms and Definitions

DFF	D Flip Flop
SHR	Shift Register
ACMP	Analog Comparator

2 References

For related documents and software, please visit:

https://www.dialog-semiconductor.com/configurable-mixed-signal.

Download our free GreenPAK Designer software [1] to open the .gp file [2] and view the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Find out more in a complete library of application notes [4] featuring design examples, as well as explanations of features and blocks within the GreenPAK IC.

[1] GreenPAK Designer Software, Software Download and User Guide
[2] AN-CM-319 Audio Signal Detector.gp, GreenPAK Design File
[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage
[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage

Author: Viacheslav Kolsun

3 Introduction

Sound can be represented with both analog or digital audio signals. Analog audio signals use electrical voltage levels. Different types of transducers convert sound to electrical signals and electrical signals to sound. The audio signal frequency range is roughly 20 to 20,000 Hz. Sources such as microphones and loudspeakers produce or receive audio signals, but it is also possible that the signal is white noise or single tone noise. These can be caused by issues in electrical circuits and have a frequency which falls within the audio frequency range. There may also be no signal at all. These possibilities must be considered when detecting audio signals in order to distinguish noise and no signals from true audio (e.g. human speech, music, natural sound).

4 Principles of Audio Signal Detection

The human ear can hear frequencies in the approximate range of 20 to 20,000 Hz. This range can include single tones such as transformer hum or white noise from radio systems. That's hardly to say that these sounds are desirable in audio systems, and a high level of such sounds can damage hearing. Human speech, music, and natural sounds have different frequencies that vary continuously. Therefore, the audio detector should register the frequency variations and pick useful audio signals based on these variations.

The basic theory behind this audio signal detector is shown in Figure 1. The system design considers three reference frequencies: 100 Hz, 500 Hz and 3 kHz. For a given signal, the system counts the number of times the frequency of the signal crosses the reference frequencies in a certain period of time. Only crosses from low to high frequencies are considered (e.g. 50 Hz to 150 Hz will count for 100 Hz; 150 Hz to 50 Hz will not). The design considers the signal as audio if it crosses any of the two reference frequencies a minimum number of times, specified in Table 1.

There are three sample signals shown in Figure 1:
1) Some noise which crosses 3 kHz three times (shown in black).

2) A single tone hum which doesn't cross any frequencies (shown in red).

3) A signal which varies like speech or music (shown in green). It crosses 100 Hz six times, 500 Hz five times, and 3 kHz one time. This curve crosses all three reference frequencies, though the device doesn't detect 3 kHz because it only crosses 1 time (it must cross 2 or more times for detection, as given in Table 1). The device detects 500 Hz (it crosses 5 times; 2 is the minimum in Table 1) and 100 Hz (crosses 6 times; 4 is the minimum in Table 1). Since it crosses two of the reference signals a sufficient number of times, the signal is detected as audio.

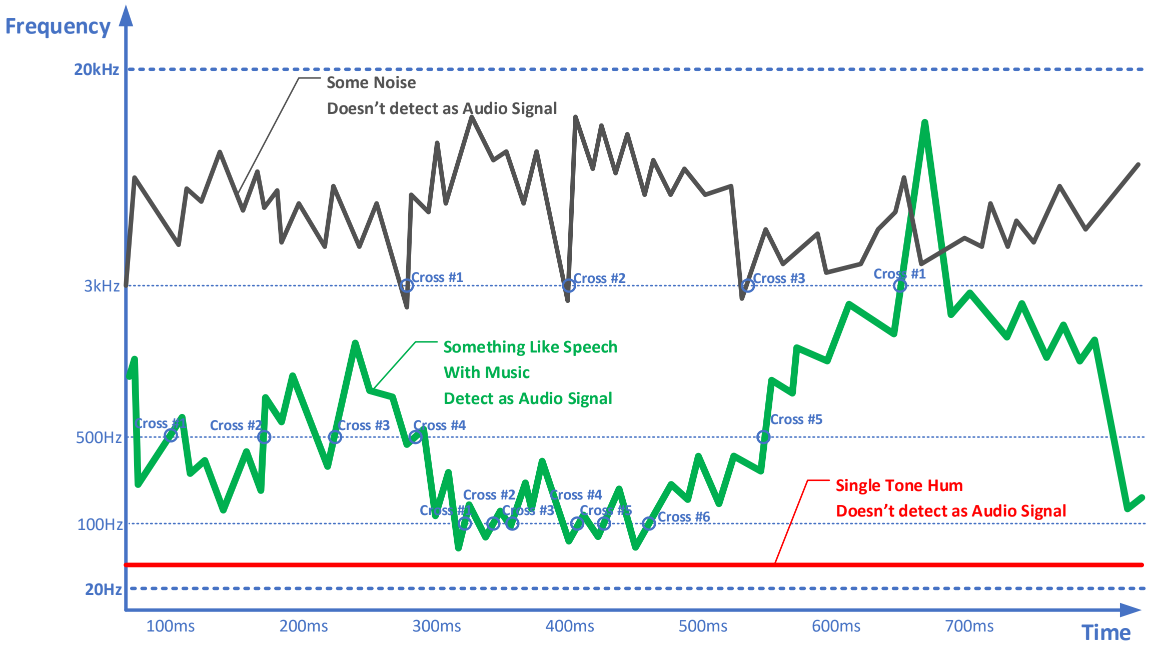


Figure 1: Audio Signal Detection

Table 1: Minimum Frequency Crossings to Detect Audio Signal

Frequency	Number of Crossings	Counting period
100Hz	4	600ms
500Hz	2	600ms
3kHz	2	600ms

Note 1 These numbers can be adjusted according to the needs of the user through I2C.

Note that speech or music can have pauses. There is a famous composition by John Milton Cage Jr. called 4'33" which is performed with the absence of any sound. Naturally, the design shouldn't determine such a long pause as an audio, though a pause less than 5 seconds will be ignored by the detecting algorithm.

Finally, the design should cut inaudible frequencies (less than 20 Hz and more than 20 kHz).

We will use these principles as the basis for designing an audio signal detector with the SLG47512.

5 Device Implementation

5.1 Design Architecture

The architecture of this device is shown in Figure 2 and contains the following blocks:

1 – Quantization of the analog audio signal. This maps the continuous analog values to double values. All that is needed to know after this process is the frequency of the audio signal.

2 – High Cut Filter. This ignores frequencies higher than 20 kHz.

3 – Low Cut Filter. This ignores frequencies lower than 25 Hz.

4 – Frequency Crossing Counter. This counts the number of crossings of signal frequencies and reference frequencies (high frequency, mid frequency, low frequency) in a certain period of time (measuring time) according to Table 1.

5 – Audio Pause. This detects audio pauses and ignores them if less than 5 seconds.

6 – Measuring Time. The given period of time during which calculations are made.

7 – DFF. This stores audio detection during the measuring time and outputs it to PIN12 (AudioDetect).

8 – Five Minutes No Audio Signal. This detects a five minute idle time of the audio signal and sets a high level on PIN11 (FiveMinutesNoAudioSignal).

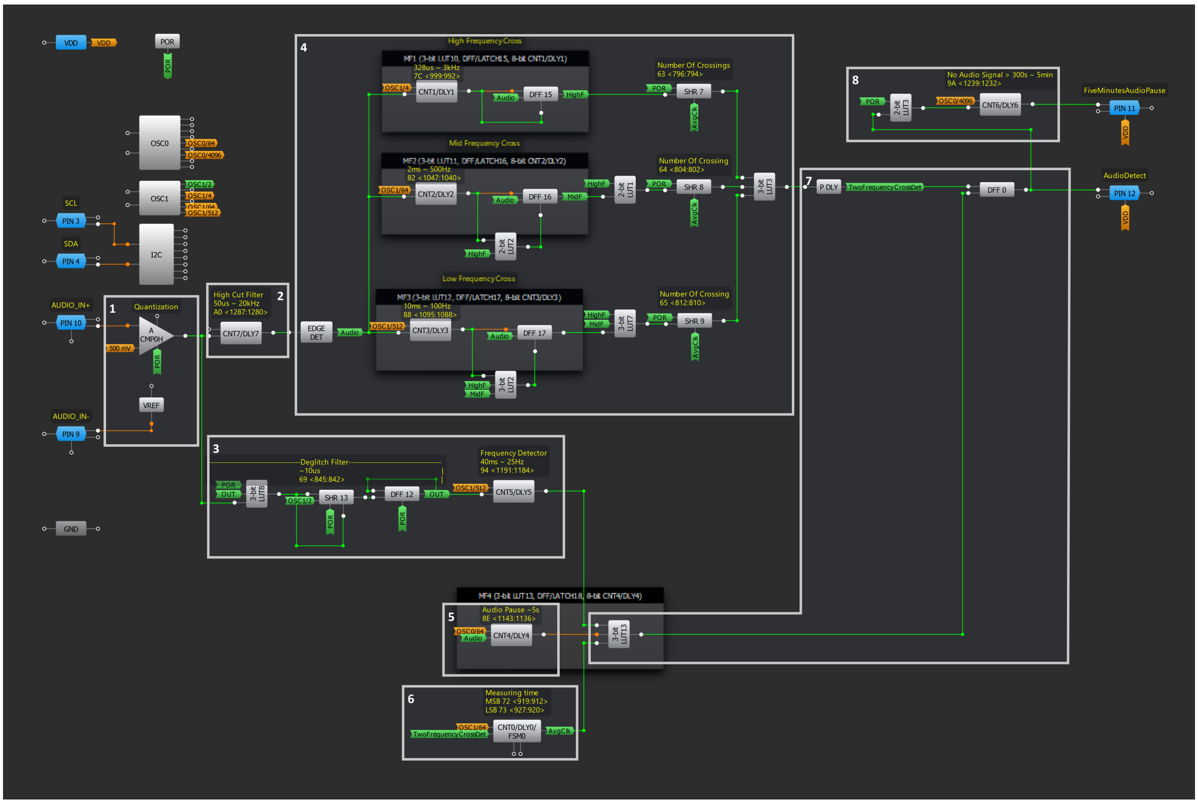


Figure 2: Design Architecture

5.2 Block Configuration

5.2.1 Analog Part

The source of the audio signal should be connected to PIN9 (AUDIO_IN-) and PIN10 (AUDIO_IN+). PIN10 (AUDIO_IN+) is an input of the analog comparator (ACMP). PIN9 (AUDIO_IN-) is a reference voltage (500mV). Taking into account the fact that the audio signal is an alternating signal, and the IC is single voltage-supplied, the design biases the input audio signal by 500mV to avoid negative voltage. Afterward, the input audio signal goes to ACMP0H (Figure 3). ACMP0H quantizes the audio signal, which is handled with the remaining part of the design.

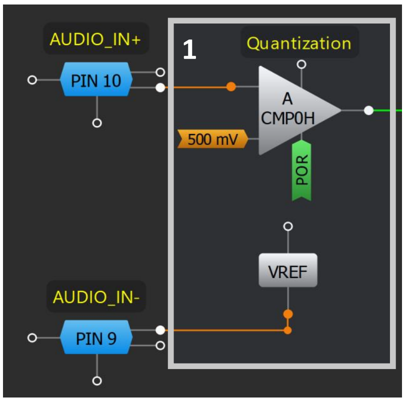


Figure 3: Analog Part

5.2.2 High Cut Filter

A Delay (8-bit CNT7/DLY7 (MF7)) is used to filter out frequencies higher than 20kHz (Figure 4). The customer can adjust the period of the frequency by writing Counter Data to 0xA0 <1287:1280> through I2C.

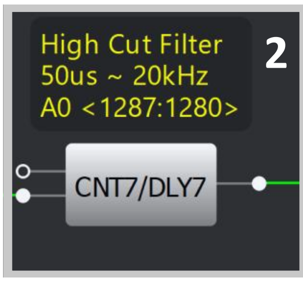


Figure 4: High Cut Filter

5.2.3 Low Cut Filter

The low cut filter (Figure 5) consists of two parts:

1 – Deglitch Filter. Taking into account the fact that there are no CNT/DLY blocks to filter random glitches, a decision was made to implement a deglitch filter with a look-up table (3-bit LUT8), shift register (SHR 13), and DFF (DFF12). The customer can adjust the time of random pulses writing Counter Data to 0x69 <845:842> through I2C.

2 – Low Cut Filter. This is implemented with a frequency detector (CNT5/DLY5) which cuts off frequencies lower than 25Hz. The customer can adjust the cutting period of frequency writing Counter Data to 0x94 <1191:1184> through I2C.

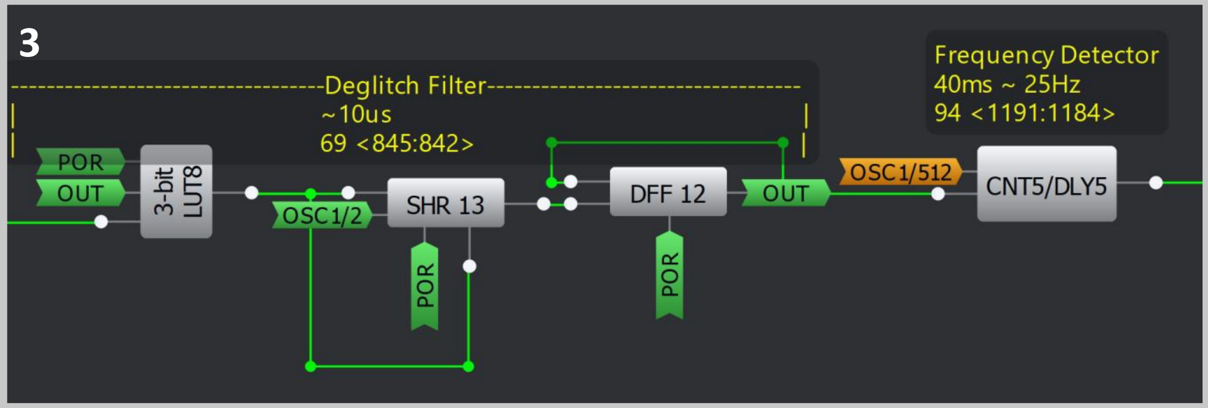


Figure 5: Low Cut Filter

5.2.4 Frequency Crossing Counter

This block consists of several parts.

The first part is EDGE DET (Figure 6). It converts a double-level audio signal to a series of short pulses which save the frequency of the current audio signal.

The next step is detecting the crossing of the current frequency of the audio signal with the reference frequencies (Table 2, Figure 7).

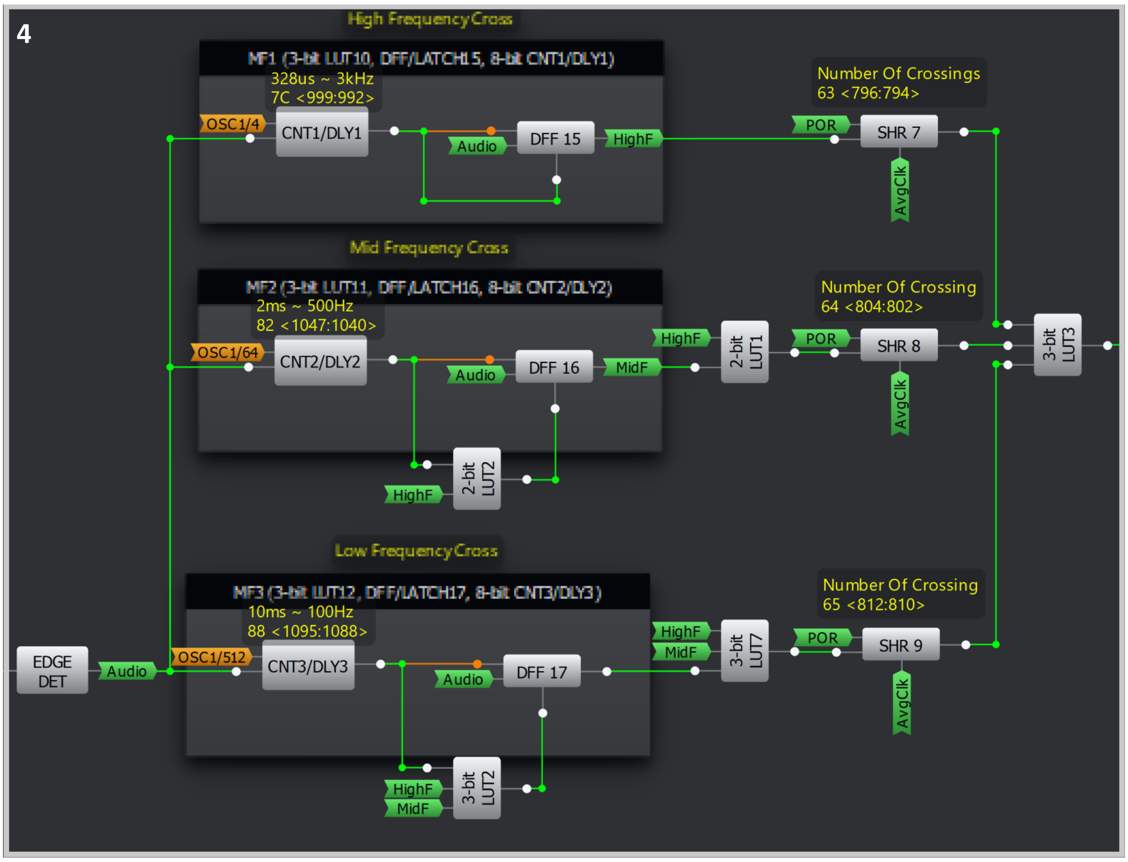


Figure 6: Frequency Crossing Counter

Table 2: CNTDLYs for Frequency Detecting

Parameter	Frequency	# of CNTDLY	Access Address	Registers
High Frequency Cross	~3kHz	CNT1/DLY1	0x7C	<999:992>
Mid Frequency Cross	~500Hz	CNT2/DLY2	0x82	<1047:1040>
Low Frequency Cross	~100Hz	CNT3/DLY3	0x88	<1095:1088>

Note 2 The crossing frequencies can be updated through I2C.

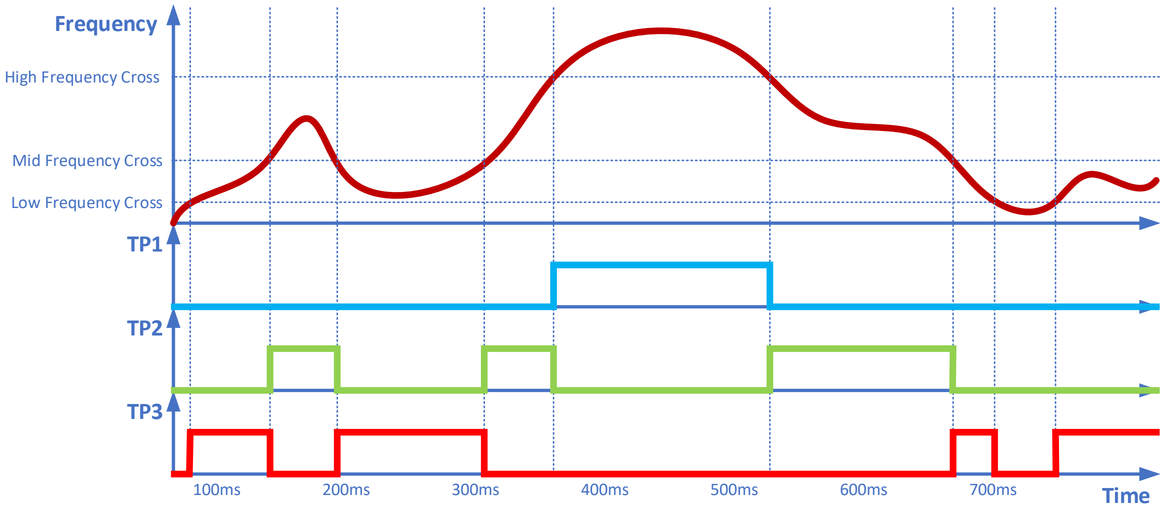


Figure 7: Frequency Detecting

Counting the number of frequency crossings with the reference frequencies is carried out by the shift registers (SHR7, SHR8, SHR9).

Table 3: Shift Registers for Counting the Number of Crossings

Parameter	# of Shift Register	Access Address	Registers
High Frequency Cross	SHR 7	0x63	<796:794>
Mid Frequency Cross	SHR 8	0x64	<804:802>
Low Frequency Cross	SHR 9	0x65	<812:810>

Note 3 The crossing frequencies can be updated through I2C.

Audio signal identification is defined by a LUT (3-bit LUT3).

5.2.5 Audio Pause

The audio pause block is implemented with the frequency detector (Figure 8, Table 4). The pause of the audio signal is detected with this block and ignored if it is less than 5 seconds. The audio signal is considered continuous. If the pause is more than 5 seconds, the design detects this as no audio signal at all.

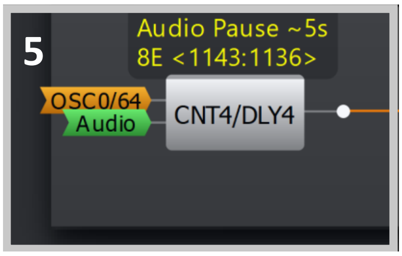


Figure 8: Audio Pause Block

Table 4: CNTDLY for Frequency Detecting

Parameter	Pause time	# of CNTDLY	Access Address	Registers
Audio Pause	~5s	CNT4/DLY4	0x8E	<1143:1136>

Note 4 The crossing frequencies can be updated through I2C.

5.2.6 Measuring Time

The design counts the number of crossings of reference frequencies at a specific time which is controlled by a counter (Figure 9, Table 5). If the frequency crossing counter doesn't detect an audio signal (including audio pause) during the measuring time, the design identifies it as no signal.

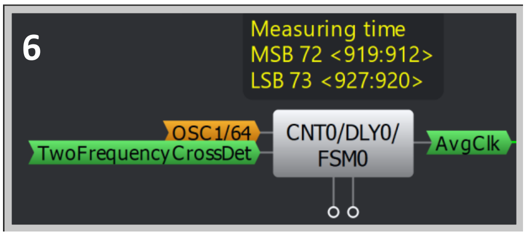


Figure 9: Measuring Time Block

Table 5: CNTDLY for Measuring Time

Parameter	Measuring time	# of CNTDLY	Access Address	Registers
Measuring Time	~600ms	CNT0/DLY0	MSB 0x72	<919:912>
			LSB 0x73	<927:920>

Note 5 The crossing frequencies can be updated through I2C.

5.2.7 Audio Signal Presence Storage

Audio signal presence storage is carried out by DFF0 (Figure 2). The signal is set using P DLY (Mode is Both edge delay) and LUT (3-bit LUT13).

5.2.8 No Audio Signal

If the design doesn't detect any audio signal during ~5 minutes, then it sets a high level on PIN11 (FiveMinutesAudioPause). Counting this time is carried out with an LUT (3-bit LUT3) and a delay (CNT6/DLY6). This time is set according to Table 6.

Table 6: CNTDLY for no Audio Signal Detecting

Parameter	Pause time	# of CNTDLY	Access Address	Registers
No Audio Signal	~5min	CNT6/DLY6	0x9A	<1239:1232>

Note 6 The crossing frequencies can be updated through I2C.

5.3 Typical Application Circuit

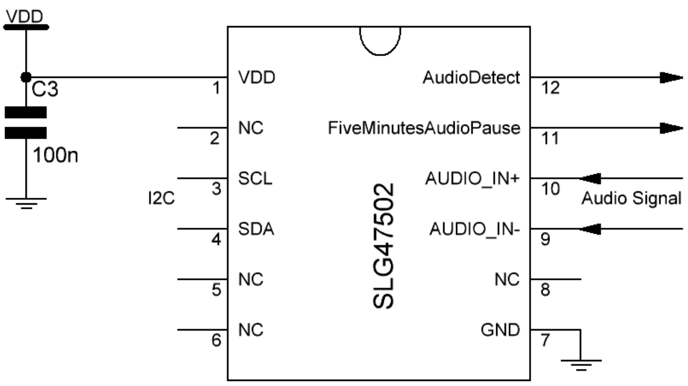


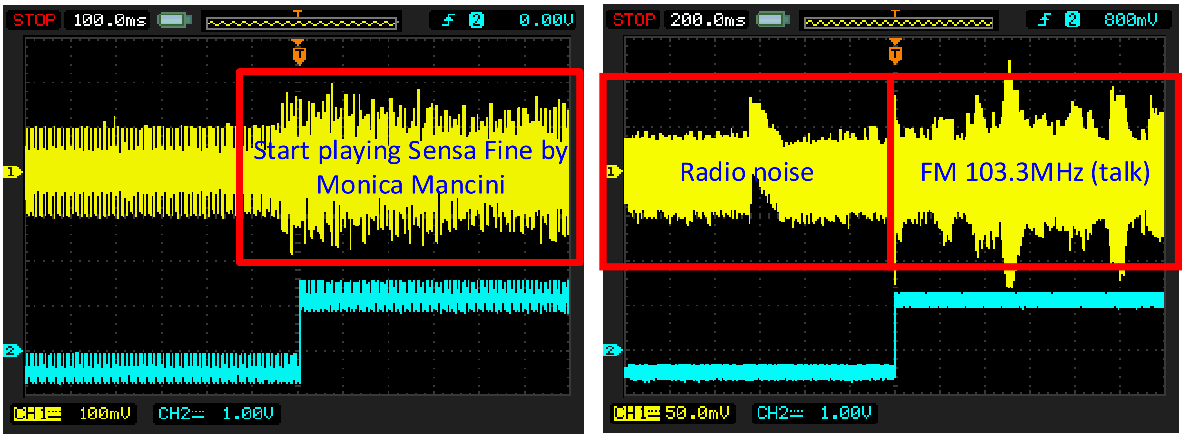
Figure 10: Typical Application Circuit

5.4 Hardware Testing

Channel 1 (yellow, top) – PIN#10 (AUDIO_IN+)

Channel 2 (blue, bottom) – PIN#12 (AudioDetect)

Ground of oscilloscope is connected to PIN9 (AUDIO_IN-)



a) b)

Figure 11: Waveforms (a - testing with a record playing; b - testing with FM-radio tuning)

6 Conclusion

The application note describes the design of an audio detector with SLG47512. The proposed method is based on the changing frequency of an audio signal. If the frequency of the input signal changes a certain number of times, then the device identifies this signal as audio. The design makes allowances for pauses in audio. If no audio signal is identified within five minutes, then the device sets a high level on PIN11 (FiveMinutesAudioPause). If the level of the input signal is relatively low then this design cannot identify audio.

Revision History

Revision	Date	Description
1.0	25-Aug-2021	Initial version

IMPORTANT NOTICE AND DISCLAIMER

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.