**NXP Semiconductors**
Application Note

Document Number: AN5255
Rev. 0, 06/2016

# G.726 Audio Vocoder Solution

## Implementation of the G.726 speech co/decoder on Kinetis L MCUs

## 1. Introduction

The Kinetis family of MCUs includes powerful ARM® Cortex® cores and specialized peripherals that enable effective audio processing with a minimum of external hardware required. In the past, audio processing was a dominant role of DSPs or specialized customer ASICs. Due to increasing CPU performance, low-cost ARM MCUs replace DSPs in basic audio-processing applications.

This document describes the usage of Kinetis MCUs in a simple audio vocoder circuit, such as the audio codec IC, coding and decoding the audio signal (human voice) using the G.726 software codec compression. A simple demonstration application based on the Tower System Modular Development Platform modules and software was created. The basic audio input/output implementation methods and principles of the run-time digital audio processing are described as well.

### Contents

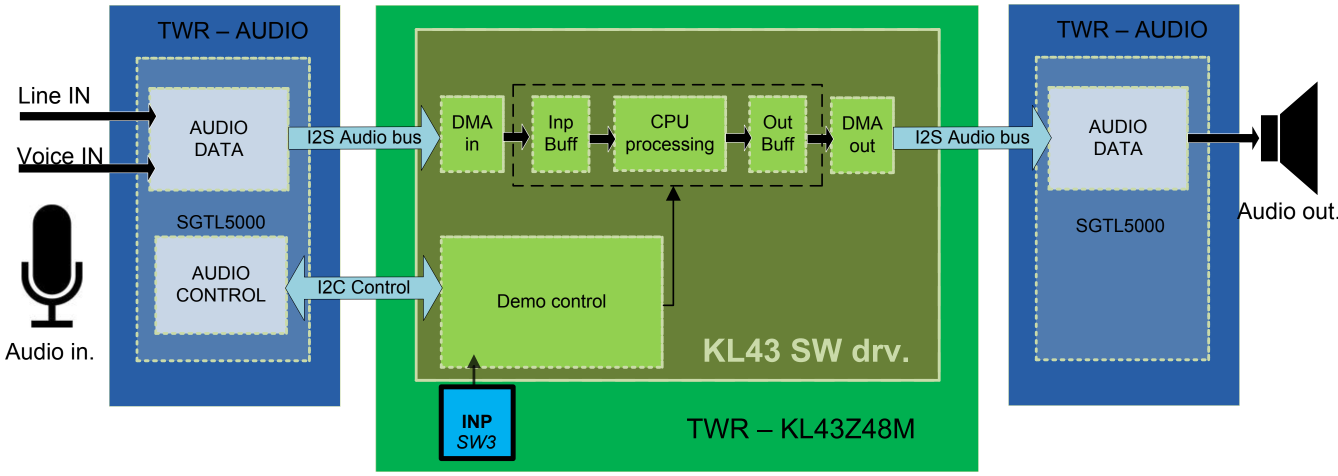


**Figure 1. Vocoder concept**

## 1.1. Abstract

The vocoder is an application used for capturing, encoding, decoding, and reproducing human voice. This application is typically used for voice recording/playback, VoIP, automated announcement systems, and so on.

Several different codec algorithms were developed and used originally for the international telecommunication and then for VoIP calls. The initial purpose was to compress the voice audio data to reduce the communication band and bitrate, while maintaining a satisfactory audio quality.

The G.726 waveform speech codec was chosen for this demonstration application.

## 1.2. G.726 codec

The G.726 is an ITU-T (division of the International Telecommunication Union) speech-compression codec covering the voice communication with a bandwidth of 0.3–3.4 kHz at bit rates of 16, 24, 32, and 40 kbit/s. The codec is based on Adaptive Differential Pulse Code Modulation (ADPCM). The G.726 standard was introduced in 1990 and supersedes both the G.721 and G.723 introduced earlier. The four supported bit rates correspond to the bit size of the compressed audio sample (2, 3, 4, and 5 bits). The most common bitrate is 32 kbit/s. It is a standard codec used in DECT wireless phone systems.

## 1.3. G.726 features

- It is based on ADPCM.
- It supports a sampling frequency of 8 kHz for all bitrates.
- It supports bitrates of 16, 24, 32, and 40 kbit/s.
- It supports compressed audio sample bit depth of 2, 3, 4, and 5 bits.

This coder operates on a sample-by-sample basis. The input samples are represented in linear PCM or 8-bit G.711 (m-law/A-law) formats (i.e., 64 kbit/s). For the 32 kbit/s operation, each sample is converted into a 4-bit quantized difference signal, resulting in a compression ratio of 2:1 over the G.711 format. For operation from 24 to 40 kbit/s, the quantized difference signal is 3 bits and 5 bits, respectively.

**NOTE**

The BITRATE is given by SAMPLE_RATE × CODE_BITS.
For example, if the compressed sample bit size (code bits) is 4 bits per sample, the formula is 8 kHz × 4, resulting in a bitrate of 32 kbit/s for the compressed audio. Because the compressed audio sample is calculated from the difference of two consecutive 16-bit linear PCM audio samples (2 × 16 bits => 4 bits), the resulting compression rate is 1:4.

## 1.4. Vocoder demo hardware setup

Connect an analog microphone to the TWR-AUDIO-SGTL card. The audio card is built around the SGTL5000 audio codec IC. The codec digitalizes the input microphone analog signal and sends the converted digital audio samples to the MCU via $I^2S$. The CPU processes the data (converts the 16-bit linear PCM data to a G.726 stream and decompresses the G.726 stream back to PCM audio samples).

The $I^2S$ data samples are in a mono 16-bit PCM format. The codec IC includes a digital equalizer and low-pass filter, which you can enable and use for input signal to cut off the higher frequencies and suppress the aliasing effect (anti-aliasing filter). The TWR-AUDIO module interfaces to the TWR-KL43 MCU module. The KL43 is based on ARM® Cortex®-M0+, with the CPU running at 48 MHz.

# 2. Software Solution

Several software and hardware methods and approaches are commonly used in applications that require continuous audio-data flow and processing with a low CPU load.

The aim is to transfer the stream of the input audio data to the MCU memory and to the audio output with a low CPU load and without any signal dropouts or distortion.

One of the recommended methods is to use double buffering and DMA (Direct Memory Access). The CPU reads and processes the input audio data, while the DMA hardware state machine feeds the audio data input and output with audio samples in the background.

## 2.1. Double buffering

In this application, DMA and double buffering are used on both sides of the audio signal loop (two buffers are used for the audio signal input and additional two buffers are used for the audio signal output). One DMA channel is used for the audio input data and the second DMA channel is used for the audio output data (to manage the $I^2S$ bus data exchange).

At any time, one buffer is actively being processed or played (front buffer), while the second (background) buffer is being filled with new audio samples. When the signal processing or playing completes, the two buffers interchange the roles. This is usually accomplished by switching the pointers. For more details, see the following figure.

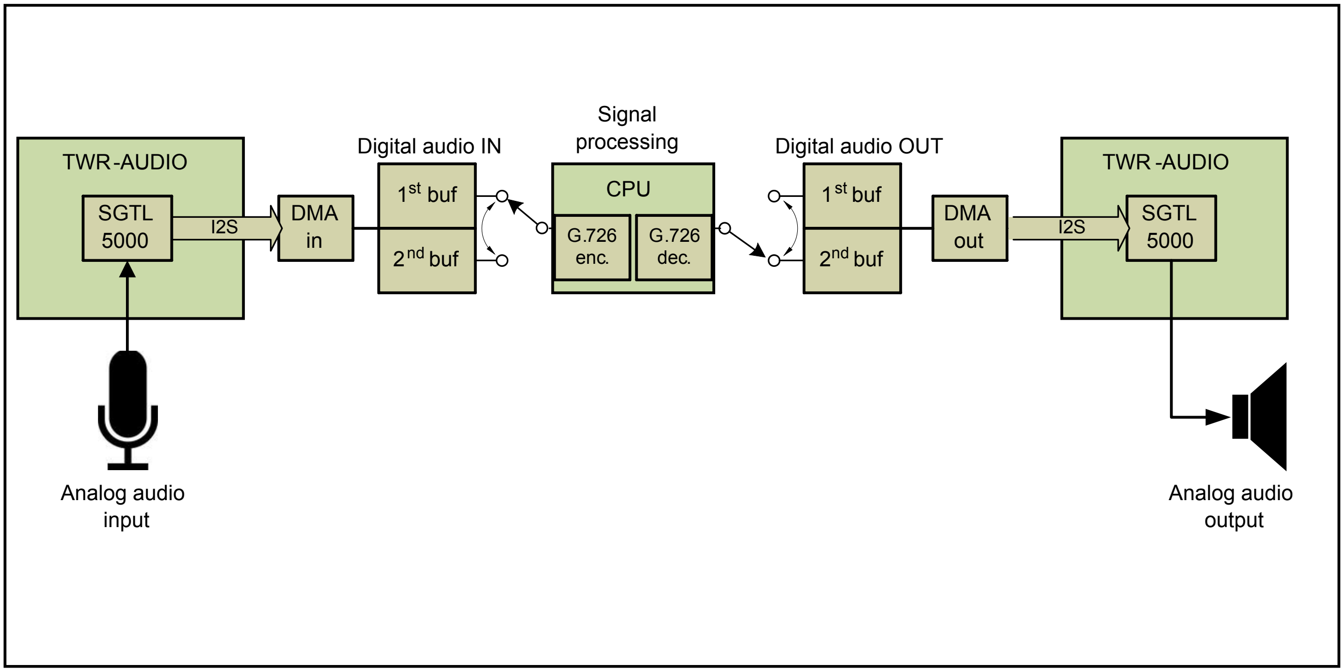


**Figure 2. Audio input-output processing**

## 2.2. DMA

DMA (Direct Memory Access) is ensured by the DMA controller, which is a special MCU peripheral capable of transferring large blocks of data between memory locations without CPU intervention.

The software initializes a DMA channel with the DMA transfer attributes, source and destination memory address, number of bytes transferred in the minor loop (single memory access), and the number of the minor loops inside the major loop, which defines the total size of the transferred data block. The source/destination address is automatically incremented after the completion of each minor loop. When a complete block is transferred (major loop is done), the DMA channel generates an IRQ request. Two separated (or adjacent) physical RAM memory arrays are used to create two “ping-pong” buffers.

When a “ping-pong” audio buffer is completely transferred by the DMA channel, the “DMA transfer complete” IRQ is generated. The DMA source address is switched to the beginning of the second buffer and the first buffer is ready to be loaded with new audio samples.

## 2.3. I$^2$S data format

The I$^2$S data is sent from MSB to LSB, starting from the second clock cycle after the word select clock transition. The transmitting MSB first enables both the transmitting and the receiving devices to not care about the audio precision of the remote device. For example, if the transmitter sends 32 bits per channel to a device with only 24-bit or 16-bit resolution, the receiving device can simply ignore the extra bits. If the transmitter sends 16-bit samples per channel to a receiving device with a 24-bit resolution, the receiver simply fills the missing bits with zeros. The data words are transferred at a frequency equal to the audio sample rate.

# 3. Application Software Example

The example software project targeted for Kinetis KL43 MCU is created to demonstrate the audio-conversion performance.

The software is based on bare-metal Kinetis-L software drivers, including the drivers for the SGTL5000 audio codec IC. The G.7xx codec software sources are adopted from Sun Microsystems, Inc. These files are provided for unrestricted usage.

The TWR-KL43Z48 module is used to develop and test the demonstration software. However, the software application can be easily ported to any other Kinetis MCU.

The demonstration hardware consists of these Tower System modules:

- TWR-KL43Z48M
- TWR-ELEV (primary and secondary)
- TWR-AUDIO-SGTL

For more information, visit www.nxp.com/tower and www.towergeeks.org.

## 3.1. Demonstration application functionality

Connect an external analog microphone to the MIC-IN input on the TWR-AUDIO module. You can also connect an external audio source to the LINE-IN input, such as a smart phone or an audio player. See Figure 3 for details. Select the audio input using SW3 on the TWR-KL43 module.

Use SW3 to switch between the TWR-AUDIO card inputs. The LEDs on the MCU module signalize the status. The red and green LEDs signalize the MIC or LINE-IN input selection, respectively. Connect headphones to the HEADPHONE output.

The SGTL5000 codec IC on the TWR-AUDIO module acts as the A/D and D/A converter for the input and output audio signals. The audio data is transferred via the $I^2S$ bus in a form of 16-bit signed PCM samples. See the following figure for details about the external audio interface.

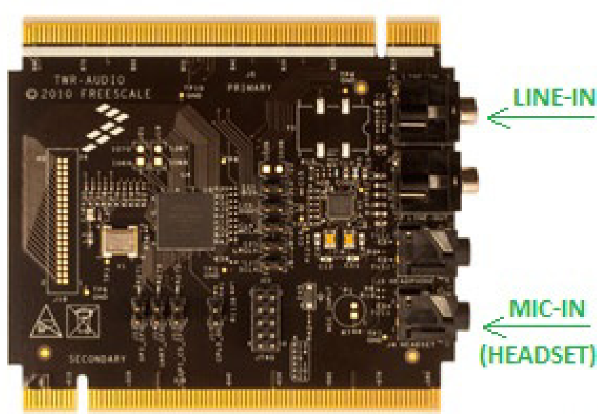


**Figure 3. TWR-AUDIO card interface**

# 4. Testing and Validation

The CPU compresses and decompresses the digital audio data in the loop using the G.726 software compression described earlier. The G.726 data is processed in 10-ms time slots (typ. used) corresponding to the small data packets of 80-PCM sample size. The digital audio samples are transferred via the DMA without CPU intervention.

The loop audio data processing is intended for demonstration purposes only. In the real-world application, the compressed data would be exchanged with a remote station (two or more devices exchanging the audio remotely).

The G.726 (32 kbit/s, 8 KHz, 4 bits per sample) encoding and decoding tasks take approximately 6 ms from the 10-ms time slot; i.e, the total CPU load for processing the G.726 (audio in/out) is about 60 % for the KL43 device, with the CPU running at 48 MHz. The complete Vocoder application (including the KL43 MCU drivers and audio card support) uses approximately 9.4 kB of flash and 1.2 kB of RAM. See this table for more details:

**Table 1. Vocoder performance**

| Parameter | G.726 Vocoder performance (linear PCM, 32 kbit/s) | |
|---|---|---|
| | Encoding | Decoding |
| CPU load [MHz] | 15.8 | 12.9 |
| Flash [kB] | 9.4 | |
| RAM [kB] | 1.2 | |

# 5. Conclusion

This document describes the basics of real-time audio processing capability of Kinetis MCUs. It is demonstrated using the G.726 software speech codec and Tower System hardware.

The main advantage of using Kinetis MCUs is that you can run the audio conversion task as one of several tasks performed simultaneously on a single Kinetis MCU. The audio conversion task can become a small part of a larger software application. You can reuse and adjust the software solution easily, and implement it to other existing applications and further designs.

# 6. References


- Wikipedia (en.wikipedia.org/wiki/G.726)
- International Telecommunication Union web page (www.itu.int/rec/T-REC-G.726/en)


# 7. Revision History

This table summarizes the changes done to this document since the initial release:

**Table 2. Revision history**

| Revision number | Date | Substantive changes |
|---|---|---|
| 0 | 06/2016 | Initial release. |

***How to Reach Us:***

**Home Page:**
nxp.com

**Web Support:**
nxp.com/support



Document Number: AN5255
Rev. 0
06/2016