Freescale Semiconductor
Application Note

Document Number: AN4955
Rev. 1, 04/2015

Emulating the I2S Bus Master with the FlexIO Module

by: Michael Galda

Contents

1 Overview

This application note shows one of the typical use cases of the FlexIO peripheral module, which was initially introduced on the Freescale Kinetis KL43 MCU family (ARM® CM0+ MCU) in the role of the I^2S audio bus master.

The FlexIO is a highly configurable module capable of emulating a wide range of serial/parallel communication protocols including UART, I^2C, SPI, I^2S, etc.

The purpose of this application note is to show that the FlexIO peripheral is capable of generating all required I^2S bus signals, and can be alternatively used instead of the classical I^2S/SAI peripherals for transferring the audio data stream without any significant restrictions in functionality or CPU resources.

For the I^2S use case validation a simple software driver has been implemented. For this demonstration, the Freescale Tower System has been used. The audio record is stored in the MCU's internal flash memory. The audio record sample rate is 8.000 kHz, single-channel (mono) with 16-bit resolution. I^2S word size is set to 32-bit. The SGTL5000 audio codec IC, placed on the TWR-AUDIO card, is used for audio reproduction.

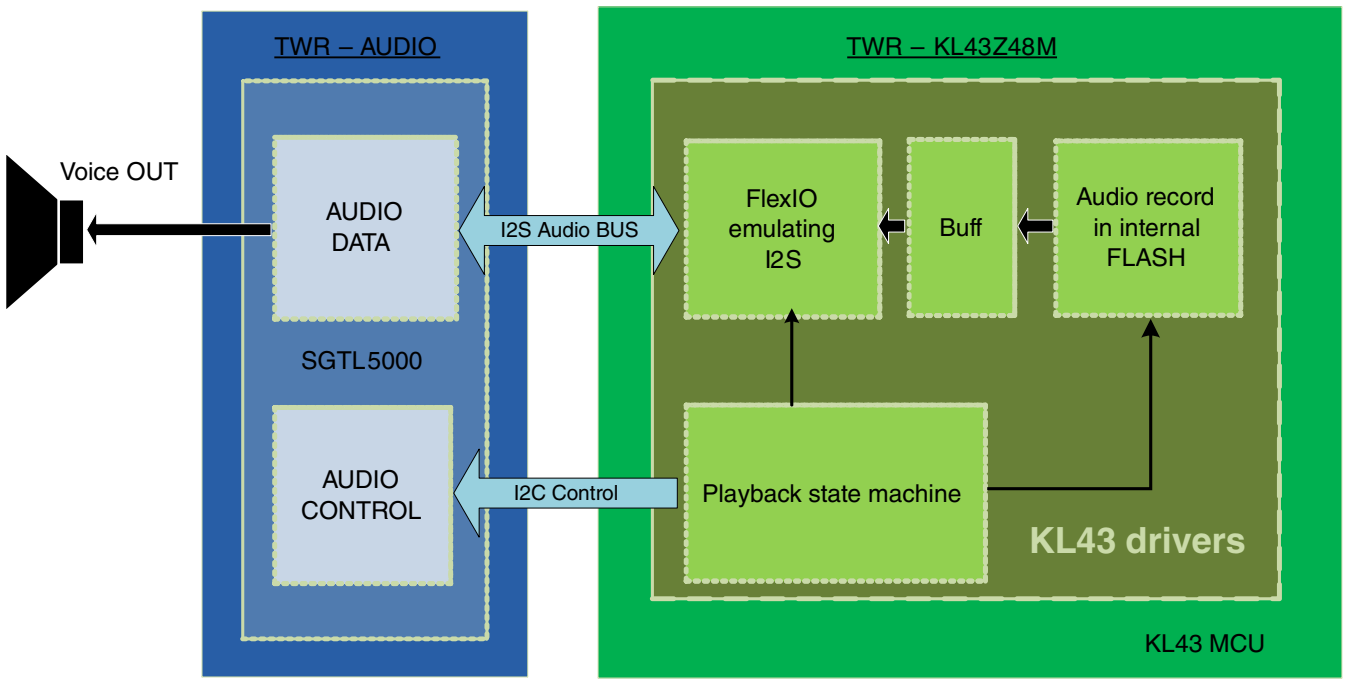

Figure 1. Implementation of the application on the Tower system

2 Required hardware

This document describes the application based on the Freescale Tower System, but the basic concept and idea can be easily reproduced on customized hardware as well.

The application can be easily set up using the following Tower System boards:

- TWR-KL43Z48M
- TWR-AUDIO-SGTL5000
- TWR-PROTO (with additional wiring connections — see Table 1)
- TWR-ELEV (primary & secondary)

The FlexIO module acts as the I^2S bus master producing all required signals:

- Master Clock (MCLK = 12 MHz)
- Word Select (WS/FSYNC/LRCLK = 8 kHz)
- Continuous Serial Clock (SCK/SCLK = 256 kHz)
- Serial Data (SD/DOUT)
- FlexIO input frequency is the bus clock = 48.000 MHz

The KL43 MCU can set up and control the SGTL5000 codec settings via I^2C bus.

3 I^2S bus theory

I^2S is an audio bus using a three-wire connection for synchronous serial data communication. The fourth wire can be additionally used for MCLK output of the synchronous master clock for the I^2S slave peripheral.

Data are transmitted on the SD (DOUT) line (MSB first) in little-endian format. Data length is 16-, 24-, or 32-bit. Transmitter data is synchronized on the rising edge of the SCK and receiver data on the falling edge of the SCK. The two-channel audio signal is represented by two data words, the right and the left channel samples, transmitted and multiplexed on the same wire. The FSYNC control signal determines if the word is for the right or left channel. This signal also denotes the data length (the beginning and end of the word). The FSYNC can be synchronized to either the rising or falling SCK edge, and precedes the MSB by one SCK period in order to have enough time to store the data in the receiver.

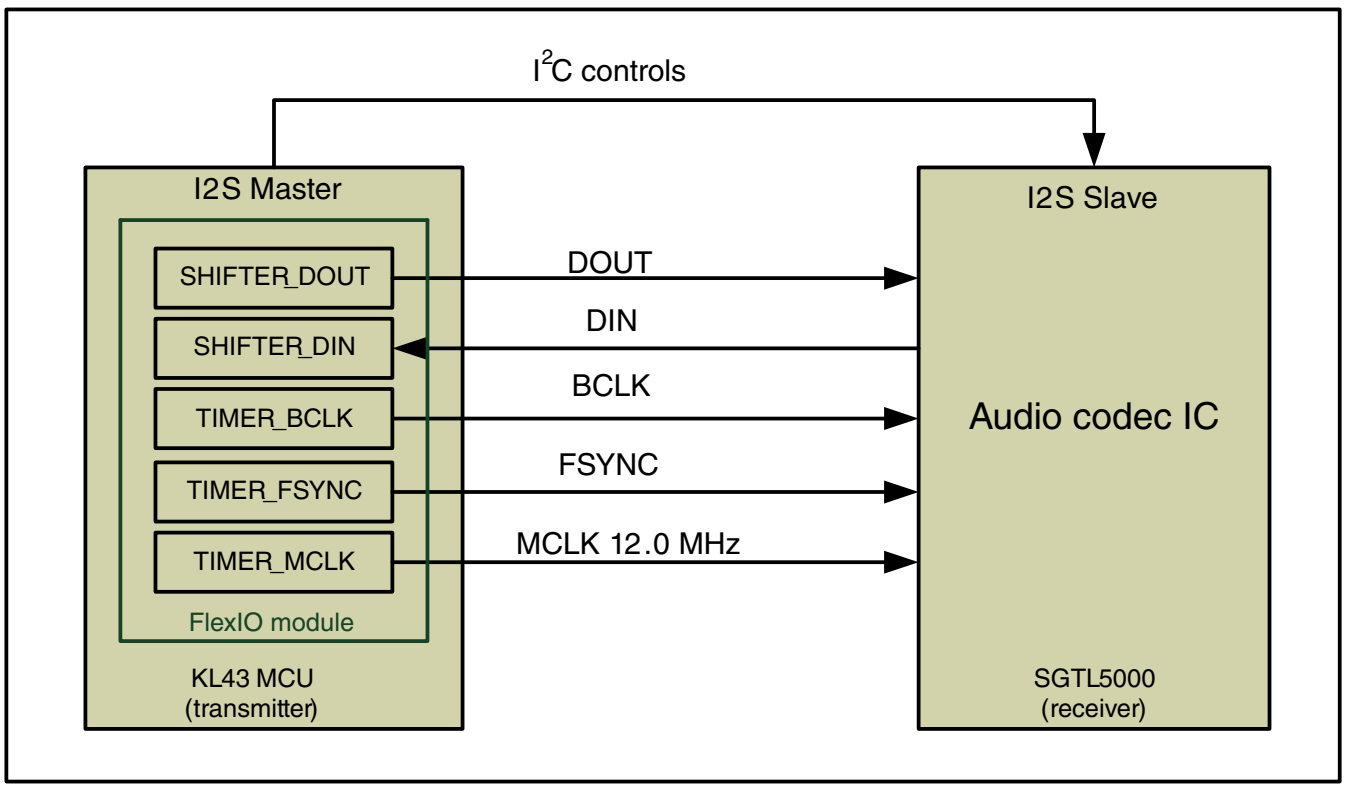

Figure 2. Master—slave with emulated I²S

4 I²S master emulation by FlexIO

I²S master mode can be supported using two timers, two shifters, and four pins. One timer is used to generate the bit clock and control the shifters, while the second timer is used to generate the frame sync. The FlexIO module waits for the first write to the transmit data buffer before enabling the bit clock and FSYNC generation. Data transfers can be supported using the DMA controller, and the shifter error flag will be set if there is a transmit underrun or receive overflow. One timer and one pin are additionally used to generate the MCLK output.

The bit clock frequency is an even integer quotient of the FlexIO clock frequency, and the initial frame sync assertion occurs at the same time as the first bit clock edge. The timer uses the start bit to ensure that the FSYNC is generated one clock cycle before the first output data.

Due to synchronization delays, the setup time for the receiver input is 1.5 FlexIO clock cycles. Therefore the maximum baud rate is the FlexIO clock frequency divided by four.

The audio samples are reproduced by the SGTL5000 audio codec IC on the TWR-AUDIO-SGTL card. The SGTL5000 works in I²S slave mode. Any other I²S audio codec IC can be used for reconstruction of the audio signal on the customized hardware.

Table 1 summarizes the required wired connections, which must be additionally created on the TWR-PROTO board. TWR-PROTO signals noted in bold must be interconnected.

Table 1. TWR-PROTO signal connections for I²S emulation

I²S Functionality	TWR-KL43Z48	TWR-AUDIO	TWR-PROTO (connected)	
I²S Tx data	PTD0	ELEV_I2S0_DOUT	**A24**	**B46**
I²S Rx data	PTD1	ELEV_I2S0_DIN	**A25**	**B48**
I²S Bit clock	PTD2	ELEV_I2S0_SCLK	**A23**	**B45**
I²S FSYNC/WS	PTD3	ELEV_I2S0_LRCLK	**A22**	**B44**
I²S Master clock	PTD4	ELEV_I2S0_MCLK	**A21**	**B59**

5 Software description

The software application is based on the KL43 peripheral bare metal drivers.

NOTE

The software has been written to work on the TWR-TWR-KL43Z48M, with no additional hardware modifications needed on the MCU Tower card. There is a FlexIO clock precision limitation caused by general use cases oriented to Tower card design. On the Tower card the core clock is derived from the high-frequency IRC 48 MHz giving 48.0 MHz for the system and FlexIO module. With this input frequency it is difficult to achieve standard audio sample rates such as 22.050 kHz or 44.100 kHz using the integer dividers available in the FlexIO module.

For the best performance and compatibility, the MCU clock can be supplied by an external crystal or oscillator at 12.288 MHz, 24.576 MHz, or 49.152 MHz (typical audio application crystal frequencies).

5.1 Initial software settings

During the initialization, all required peripheral module clocks are enabled in SIM, and the 48 MHz HIRC is selected as the clock source. The clock ratio between System: Bus: Flash is then set to 1:2:2. PortD multiplexers are switched to support FlexIO functionality.

In the FLEXIO_I²S_Init function the FlexIO shifters and timers are initialized. The configuration values are given from the file appconfig.h. These values denote which timers, shifters, and pins are used for emulating the I²S.

The I²C bus is used for initial configuration of the SGTL5000 audio codec IC placed on the TWR-AUDIO card, which is used for this application's demonstration and testing.

The audio record is stored in the internal flash memory. At the beginning the empty RAM buffers must be initialized with the audio data. After that the data are read from flash memory to a double-buffer in RAM. The audio samples are written to the SHIFTER inside the FlexIO IRQ callback function. Every time the free (unused) RAM buffer is selected and fed by fresh audio samples, the filled buffer is used for shifter output.

5.2 I²S bus emulation in software

I^2S bus functionality is emulated by the following mechanism, which ensures a smooth, continuous stream of audio data on the FlexIO output:

1. SHIFTER0 is used for audio data output in 32-bit frames. Transmit data are loaded on the first shift. The stop bit is disabled. Data transmit is driven by Timer0. Data are shifted out on the rising clock edge on Pin0.
2. SHIFTBUF0: Transmit data can be written to SHIFTBUFBBS. The Shifter Status Flag is used to indicate when the data can be written using an interrupt or DMA request. LSB at first data format can be supported by writing to SHIFTBUF register instead.
3. SHIFTER1 can be used for audio data input. The shifter is configured for receiving, using Timer0 on the falling clock edge with input data on Pin1. The SHIFTER1 start/stop bit is disabled (unused).
4. SHIFTBUF1: Received data can be read from SHIFTBUFBBS. The Shifter Status Flag is used to indicate when data can be read using an interrupt or DMA request. LSB at first data format can be supported by writing to SHIFTBUF register instead.
5. TIMER0 is configured as a dual 8-bit counter using Pin2 output (BCLK), with the SHIFTER0 flag as the inverted trigger. PINPOL is set to invert the output shift clock. The start bit is enabled and the timer is enabled on trigger high. The initial clock state is 1. The TIMER0 compare (TIMCMP) register is configured for 32-bit transfer with a baud rate equal to the FlexIO clock divided by four. Set TIMCMP[15:8] = (number of bits × 2) – 1. Set TIMCMP[7:0] = (baud rate divider ÷ 2) – 1.
6. TIMER1 is configured as a 16-bit counter using the inverted Pin3 output (as the FSYNC signal). TIMER1 is enabled when TIMER0 is enabled (and never disabled).
7. TIMER2 is configured to generate MCLK (Master Clock) output for the external codec IC.

6 Conclusion

This application shows the FlexIO peripheral, available on the Freescale KL43 MCU, emulating the I^2S audio bus in the role of I^2S master transmitter. An audio record is stored in the MCU's internal flash memory and reproduced by an I^2S slave device SGTL5000 audio codec. The application is demonstrated using the Freescale Tower system. The I^2S bus functionality can be successfully emulated using the described method. An application software example is available on the Freescale website for free download.

7 References

1. Freescale Tower System Modular Development Platform: www.freescale.com/tower
2. TWR-KL43Z48M: TWR board for L4KS: www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-KL43Z48M

8 Revision history

This section documents the changes done to this document.

Table 2. Revision history

Revision	Substantial changes
0	Initial release
1	Removed external references

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All other product or service names are the property of their respective owners.

Document Number AN4955
Revision 1, 04/2015

ARM® POWERED

freescale™