

1 Overview

This application note shows one of the typical use cases of the FlexIO peripheral module, which was initially introduced on the NXP K32L2B MCU in the role of the I²S audio bus master.

The FlexIO is a highly configurable module capable of emulating a wide range of serial/parallel communication protocols including UART, I²C, SPI, I²S, etc.

The application note introduces that the FlexIO peripheral is capable of generating all required I²S bus signals, and is alternatively used instead of the classical I²S/SAI peripherals for transferring the audio data stream without any significant restrictions in functionality or CPU resources.

For the I²S use case validation, a simple software driver has been implemented. For this demonstration, the FRDM-K32L2B Board has been used. The audio record is stored in the MCU's internal flash memory. The audio record sample rate is 48.000 kHz, single-channel (mono) with 16-bit resolution. I²S word size is set to 32 bit. The WM8904 audio codec IC, placed on the NXP Mic/Audio/Oled Shield (MAO), is used for audio reproduction.

2 Required hardware

This document describes the application based on the FRDM-K32L2B Board. The basic concept and idea can be easily reproduced on customized hardware.

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

1. FRDM-K32L2B Board
2. NXP Mic/Audio/Oled Shield (MAO)

The K32L2B MCU can set up and control the WM8904 codec settings via the I²C bus, transfer audio data via emulating I²S bus.

Table 1. Board connection

Peripheral	PIN	K32L2B	MAO
I ² S	FS	J2-12 PTD5	J1-18 (FS)
	BCLK	J2-6 PTD4	J1-16 (BCLK)
	MCLK	J2-4 PTD2	J4-9 (MCLK)
	TX	J1-6 PTD3	J1-20 (RX)
	RX	J2-8 PTD6	J1-10 (TX)
I ² C	SCL	J4-2 (I2C0_SCL) PTB0	J1-1 (SCL)
	SDA	J4-4 (I2C0_SDA)PTB1	J1-3 (SDA)

The hardware resources in FlexIO module contains shifter, timer and pin. The amount of these resources for a given microcontroller is read from the FLEXIO_PARAM register. For example, there are four shifters, four timers, and eight pins in K32L2B.

Contents

1 Overview	1
2 Required hardware	1
3 I²S master emulation by FlexIO	2
4 Software description	8
5 Conclusions	9
6 References	9



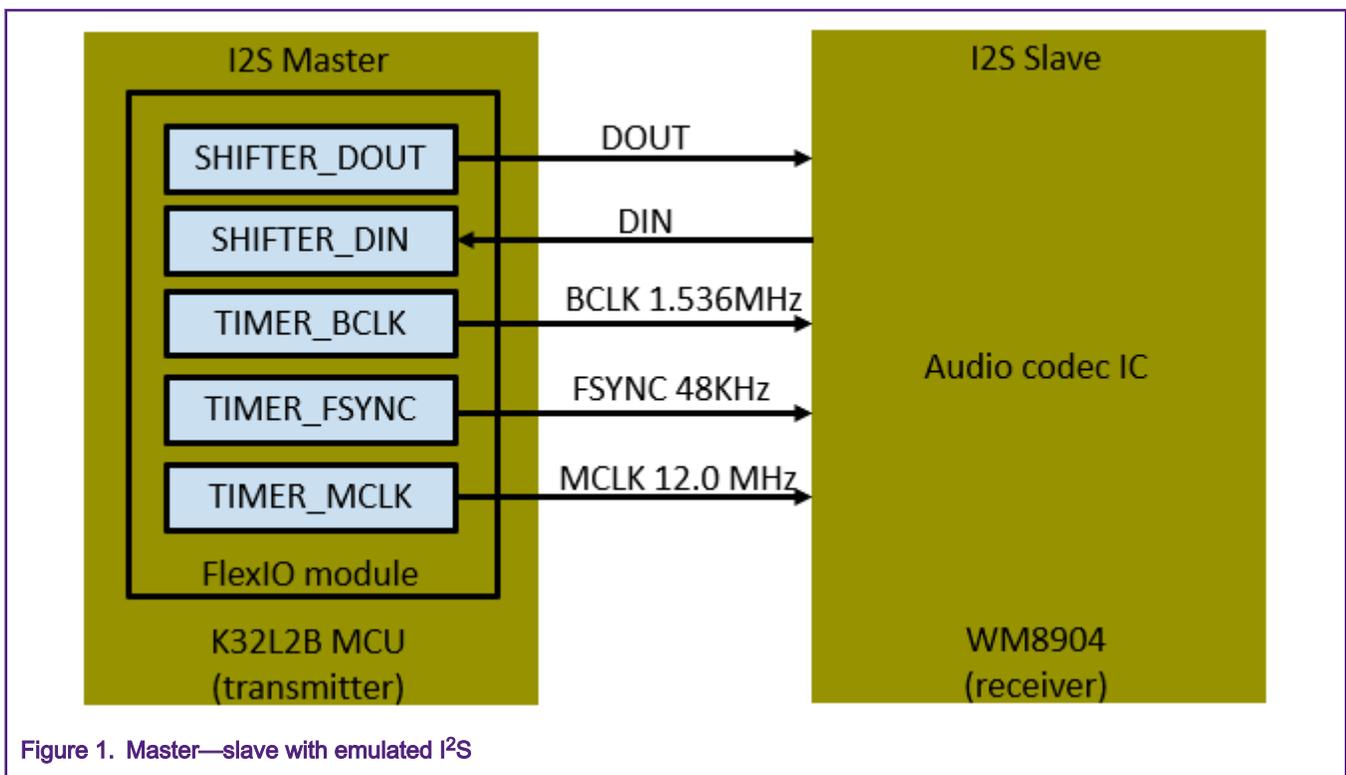
Transmit and receive are two basic modes of the shifters.

- If one shifter is configured as transmit mode, it loads data from its buffer register and shifts data out to its assigned pin bit by bit.
- If one shifter is configured as receive mode, it shifts data in from its assigned pin and stores data into its buffer register.

The load, store, and shift operations are all controlled by the shifter’s assigned timer.

The FlexIO module acts as the I²S bus master producing all required signals:

- Word Select (WS/FSYNC/LRCLK = 48 kHz)
- Bit Clock (BCLK = 1.536 MHz)
- Serial Data (SD/DOUT)
- FlexIO input frequency is the bus clock = 48.000 MHz
- Master Clock (MCLK = 12 MHz)



NOTE

- WM8904 could use its internal PLL to obtain 24.576MHz if there is an 12 MHz frequency connect with its **MCLK Pin**.
- 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).

3 I²S master emulation by FlexIO

In the application, FlexIO emulates an I²S interface to communicate with WM8904 codec, in which general I²C is integrated.

Figure 2 shows the hardware platform and data flow.

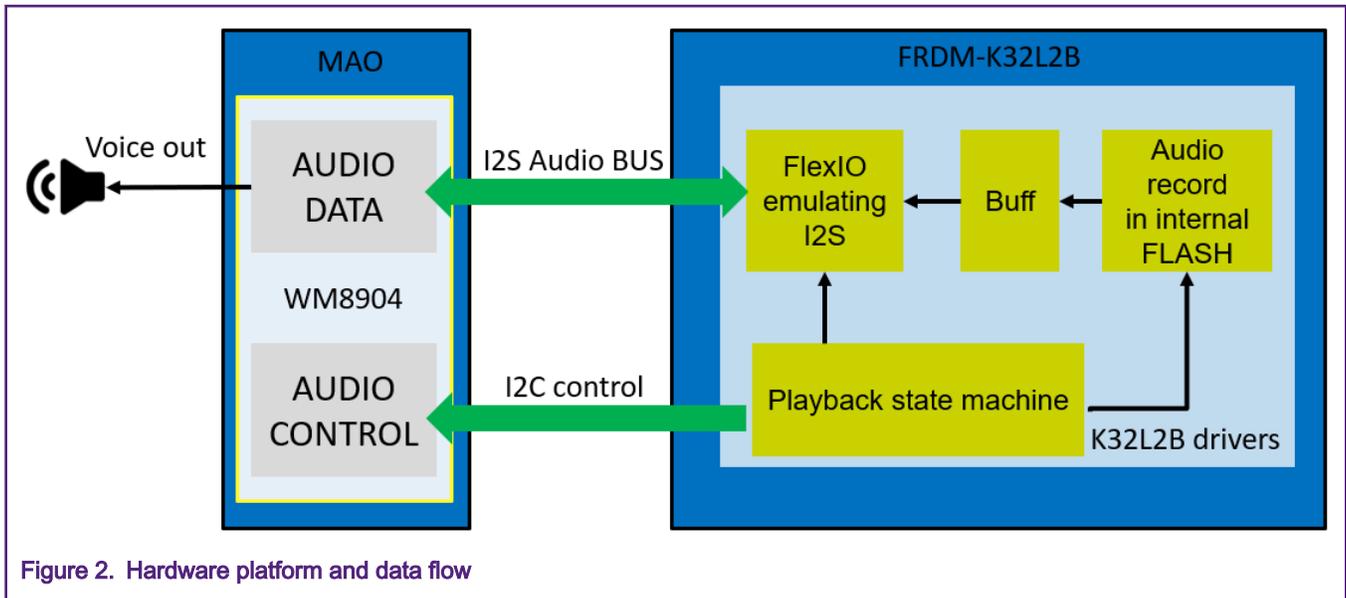


Figure 2. Hardware platform and data flow

3.1 General description

I²S master mode can be supported by using three timers, two shifters, and four pins.

- One timer is used to generate the bit clock and control the shifters,
- The second timer is used to generate the frame sync.
- The third timer and one pin are additionally used to generate the MCLK output.

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 by using the DMA controller, and the shifter error flag will be set if there is a transmit underrun or receive overflow.

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 has generated one clock cycle before the first output data.

Figure 3 shows the resource assignment.

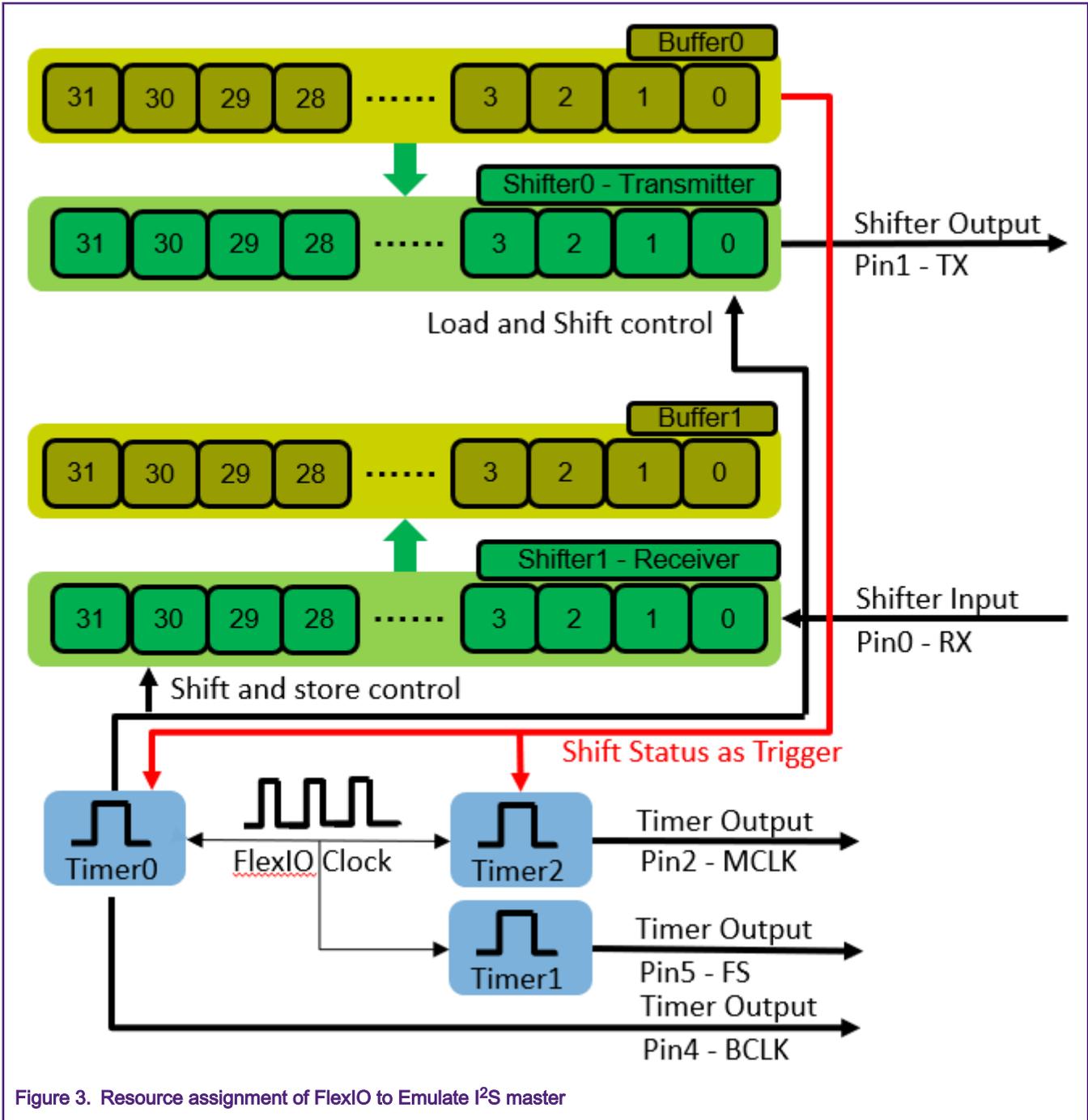


Figure 3. Resource assignment of FlexIO to Emulate I²S master

3.2 Configurations of shifters and timers

3.2.1 Configurations of Shifter 0 (TX)

Shifter 0 is used as the transmitter and its initial configurations are as described in Table 2.

Table 2. Initial configurations and register information of Shifter 0

Register	Value	Items	Configurations/Description
SHIFTCTL0	SMOD = 2	Shifter mode	Transmit
	PINPOL = 0	Shifter Pin Polarity	Pin is active high.
	PINSEL = 3	Shifter Pin Select	Pin 3
	PINCFG = 3	Shifter Pin Configuration	Shifter pin output.
	TIMPOL = 0	Timer Polarity	Shift on posedge of Shift clock.
	TIMSEL = 0	Timer Select	Timer 0 is used for controlling the logic/shift register and generating the shift clock.
SHIFTCFG0	START = 1	Shifter Start bit	Transmitter loads data on first shift.
	SSTOP = 0	Shifter Stop bit	Stop bit disabled for transmitter/receiver/match store.
	INSRC = 0	Input Source	Selects the PIN as input source for the shifter.

3.2.2 Configurations of Shifter 1 (RX)

Shifter 1 is used as the receiver and its initial configurations are as described in [Table 3](#).

Table 3. Initial configurations and register information of Shifter 1

Register	Value	Items	Configurations/Description
SHIFTCTL1	SMOD = 1	Shifter mode	Transmit
	PINPOL = 0	Shifter Pin Polarity	Pin is active high.
	PINSEL = 6	Shifter Pin Select	Pin 6
	PINCFG = 0	Shifter Pin Configuration	Shifter pin output disabled.
	TIMPOL = 1	Timer Polarity	Shift on negedge of Shift clock.
	TIMSEL = 0x00	Timer Select	Timer 0 is used for controlling the logic/shift register and generating the shift clock.
SHIFTCFG1	START = 1	Shifter Start bit	Transmitter loads data on enabled.
	SSTOP = 0	Shifter Stop bit	Stop bit disabled for transmitter/receiver/match store.
	INSRC = 0	Input Source	Selects the PIN as input source for the shifter.

3.2.3 Configurations of Timer 1 (FS 48 KHz)

Timer 1 is used as the receiver and its initial configurations are as described in [Table 4](#).

Table 4. Initial configurations of Timer 1

Register	Value	Items	Configurations/Description
TIMCTL1	TIMOD = 3	Timer Mode	Single 16-bit counter mode.
	PINPOL = 1	Timer Pin Polarity	Pin is active low.
	PINSEL = 0x05	Timer Pin Select	Pin 5
	PINCFG = 3	Timer Pin Configuration	Timer pin output.
	TRGSRC = 0	Trigger Source	External trigger selected.
	TRGPOL = 0	Trigger Polarity	Trigger active high.
	TRGSEL = 0x02	Trigger Source	—
TIMCFG1	TSTART = 0	Timer Start Bit	Start bit disabled.
	TSTOP = 0	Timer Stop Bit	Stop bit disabled.
	TIMENA = 1	Timer Enabled	Timer enabled on Timer 0 enable.
	TIMDIS = 0	Timer Disabled	Timer never disabled.
	TIMRST = 0	Timer Reset	Timer never reset.
	TIMDEC = 0	Timer Decrement	Decrement counter on FlexIO clock. Shift clock equals to Timer output.
	TIMOUT = 0	Timer Output	Timer output is a logic one when enabled and is not affected by timer reset.

3.2.4 Configurations of Timer 0 (BCLK 1.536 MHz)

Timer 0 is used control Shifter 0 and Shifter 1 and its initial configurations are as described in [Table 5](#).

Table 5. Initial configurations of Timer 0

Register	Value	Items	Configurations/Description
TIMCTL0	TIMOD = 1	Timer Mode	Dual 8-bit counters baud/bit mode.
	PINPOL = 0	Timer Pin Polarity	Pin is active high.
	PINSEL = 0x04	Timer Pin Select	Pin 4
	PINCFG = 3	Timer Pin Configuration	Timer pin output
	TRGSRC = 1	Trigger Source	Internal trigger selected
	TRGPOL = 1	Trigger Polarity	Trigger active low
	TRGSEL = 0x01	Trigger Select	Shifter 0 status flag

Table continues on the next page...

Table 5. Initial configurations of Timer 0 (continued)

Register	Value	Items	Configurations/Description
TIMCFG0	TSTART = 1	Timer Start Bit	Start bit enabled.
	TSTOP = 0	Timer Stop Bit	Stop bit disabled.
	TIMENA = 2	Timer Enable	Timer enabled on Trigger high.
	TIMDIS = 0	Timer Disable	Timer never disabled.
	TIMRST = 0	Timer Reset	Timer never reset.
	TIMDEC = 0	Timer Decrement	Decrement counter on FlexIO clock. Shift clock equals to Timer output.
	TIMOUT = 0	Timer Output	Timer output is a logic one when enabled and is not affected by timer reset.

3.2.5 Configurations of Timer 2 (MCLK 12 MHz)

Timer 2 is used to generate MCLK output and its initial configurations are as described in [Table 6](#).

Table 6. Initial configurations of Timer 1

Register	Value	Items	Configurations/Description
TIMCTL1	TIMOD = 1	Timer Mode	Dual 8-bit counters baud/bit mode.
	PINPOL = 0	Timer Pin Polarity	Pin is active low.
	PINSEL = 0x02	Timer Pin Select	Pin 2
	PINCFG = 3	Timer Pin Configuration	Timer pin output.
	TRGSRC = 1	Trigger Source	Internal trigger selected.
	TRGPOL = 1	Trigger Polarity	Trigger active low.
	TRGSEL = 0x01	Trigger Select	Shifter 0 status flag.
TIMCFG1	TSTART = 1	Timer Start Bit	Start bit enabled.
	TSTOP = 0	Timer Stop Bit	Stop bit disabled.
	TIMENA = 2	Timer Enabled	Timer enabled on Trigger high.
	TIMDIS = 0	Timer Disabled	Timer never disabled.
	TIMRST = 0	Timer Reset	Timer never reset.
	TIMDEC = 0	Timer Decrement	Decrement counter on FlexIO clock. Shift clock equals to Timer output.
	TIMOUT = 0	Timer Output	Timer output is a logic one when enabled and is not affected by timer reset.

4 Software description

NOTE

- Several driver functions have been implemented in this application, which based on the Hardware Abstraction Layer (HAL) of the NXP Kinetis Software Development Kit (KSDK).
- SDK version 2.6.0

4.1 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.

In the `FLEXIO_I2S_Init` function the FlexIO shifters and timers are initialized. User can define timers, shifters, and pins are used for emulating the I²S according real requirement.

The I²C bus is used for initial configuration of the WM8904 audio codec IC placed on the MAO, which is used for this application's demonstration and testing.

4.2 I²S bus emulation in software

I²S 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 Pin3.
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 Pin6. 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 Pin4 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. 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 Pin5 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.

4.3 Running software code and testing the demo

The user can download a program image to the microcontroller to test current demo. The PC host obtains a serial port after a USB cable is connected between the PC host and the USB interface on FRDM-K32L2B Board. Open a serial terminal, with these settings: 115200 baud rate, 8 data bits, no parity, one stop bit, no flow control.

Table 7. Materials in Firmware development

List	Description
PC	Host device connected to the development board

Table continues on the next page...

Table 7. Materials in Firmware development (continued)

List	Description
Debugger	Default CMSIS-DAP firmware in Debugger onboard.
IDE	MDK (V5.26.2.0)
Demos	...\\SDK_2.6.0_FRDM-K32L2B_FlexIOI2S\boards\frdmk32l2b\driver_examples\flexio\i2s_dma\dma_i2s_transfe

- Users could use an audio record stored in the MCU's internal flash memory, or follow the steps below to test current demo.
 1. Connect **LINE IN** with mobile phone using audio line.
 2. Connect **LINE OUT** with Headset using USB line.
 3. Download software code and press **RESET** to run the demo.
 4. Play music on PC.
- The waveforms of MCLK and BCLK.

User can use an oscilloscope or a logic analyzer to capture the waveforms of MCLK and BCLK. [Figure 4](#) captured by a logic analyzer when running the demos.

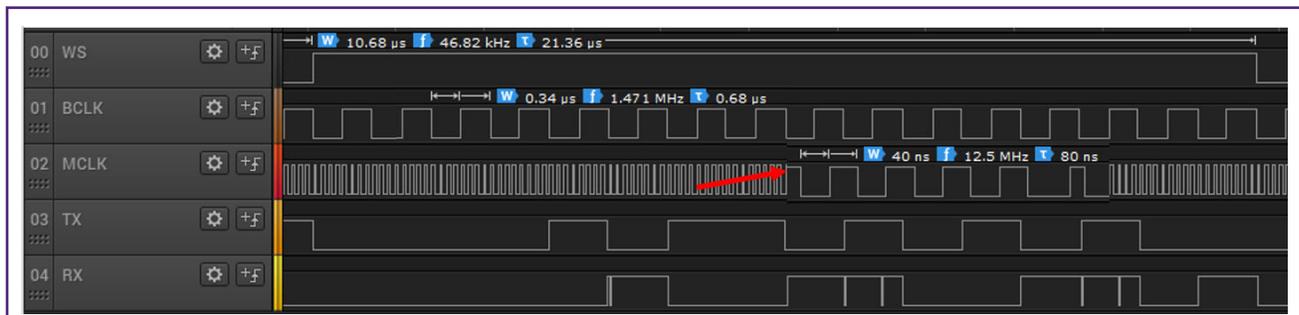


Figure 4. MCLK and BCLK of emulated I²S

5 Conclusions

This application shows the FlexIO peripheral, available on the K32L2B MCU, emulating the I²S audio bus in the role of I²S master transmitter. An audio record is stored in the MCU's internal flash memory and reproduced by an I²S slave device WM8904 audio codec. The application is demonstrated using the FRDM-K32L2B Board. The I²S bus functionality can be successfully emulated using the described method. An application software example is available on the NXP website for free download.

6 References

Reference documents include:

- *K32 L2B Sub-Family Reference Manual* (document [K32L2B3xRM](#))
- *Emulating the I²S Bus Master with the FlexIO Module* (document [AN4955](#))

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 02/2020

Document identifier: AN12644

