

Interfacing an I2S Device to an MSP430 Device

Michael Burns

Keywords

- TLV320AIC26
- Codec
- MSP430
- I2S Bus

1 Introduction

The MSP430 series of microprocessors are fast powerful devices ideally suited for use with the various wireless transceivers offered by Texas Instruments. In streaming audio applications, however, it is desirable that the microprocessor support an audio bus to allow for interconnection to an audio CODEC. This application note describes how to create an I2S-like bus (left-justified mode) from an SPI bus using a 4-bit counter, a 'D' type flip-flop, and a dual inverter gate.

2 Abbreviations

BCLK – Bit Clock

Codec – Compressor-decompressor

DIN – Data In

DMA – Direct Memory Access

DO – Digital Output

DOUT – Data Out

I2S – Inter-IC Sound Bus

LRCLK – Left\Right Clock

MCLK – Master Clock

MHz – Megahertz

MSB – Most Significant Bit

SPI – Serial Peripheral Interface Bus

μs – Microseconds

3 SPI vs I2S

The fundamental difference between the I2S bus protocol and the SPI bus protocol is that I2S and its derivatives require a 'LR Clock' (Left\Right Clock) to indicate whether the 16 bits of data pertain to the left or to the right channel. An additional difference is that the SPI bus works in 8-bit bytes, whereas the I2S bus works in 16-bit words.

The I2S bus standard states that "In I2S mode, the MSB of the left channel is valid on the second rising edge of BCLK after the falling edge of LRCLK. Similarly, the MSB of the right channel is valid on the second rising edge of BCLK after the rising edge of LRCLK" (see [Figure 1](#)).

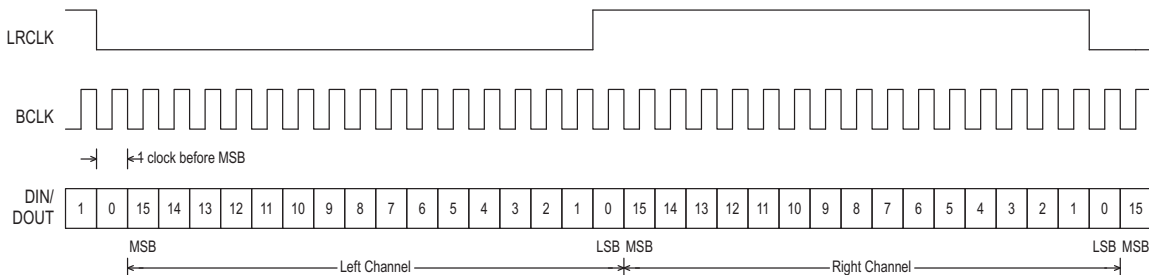


Figure 1. I2S Bus Timing

4 Left-Justified Mode

The I2S protocol is difficult to implement, due to the one BCLK delay between LRCLK and Data In ('DIN') and Data Out ('DOUT'). A much easier to implement, but similar protocol is 'left-justified mode' (see [Figure 2](#)).

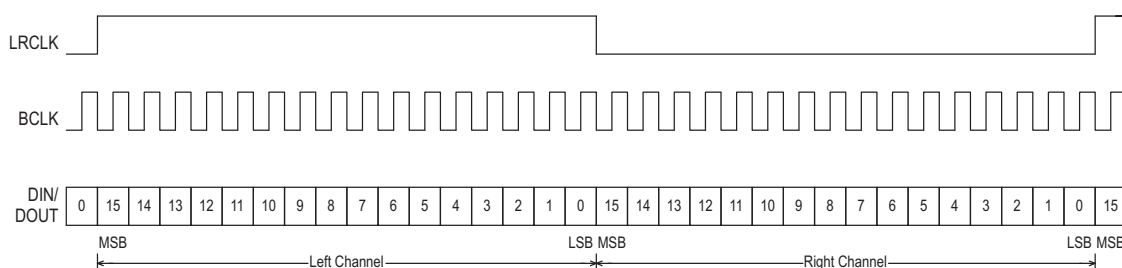


Figure 2. Left-Justified Mode Timing

The left-justified mode standard states that "In left-justified mode, the MSB of the right channel is valid on the rising edge of BCLK following the falling edge of LRCLK. Similarly, the MSB of the left channel is valid on the rising edge of BCLK following the rising edge of LRCLK". For more details on I2S mode and its derivatives, see the Overview section of the TLV320AIC26 data sheet, *Low Power Stereo Audio Codec with Headphone/Speaker Amplifier and 12 Bit Battery/Temperature/Auxiliary ADC* ([SLAS412](#)).

5 Implementation

[Figure 3](#) shows one possible implementation of a SPI to left-justified mode converter. Because a 5-bit counter in a single package is not readily available, a 4-bit counter is used (CD74AC161) along with SN74LVC1G175 D-type flip flop. Both the CD74AC161 and the SN74LVC1G175 are positive-edge triggered. The RCO (carry out) output of the counter goes high when a count of 15 is reached, and returns low when the counter resets to 0. An inverter (SN74LVC2G04) is placed between RCO and the clock of a D-type flip flop connected toggle mode, so that LRCLK toggles shortly after the counter resets (i.e., after 16 SCLK clocks) (see [Figure 4](#)).

The Px.x line is controlled by a DO bit of the MSP430. It should be held low (logic true) until the SPI control registers of the MSP430 are properly initialized.

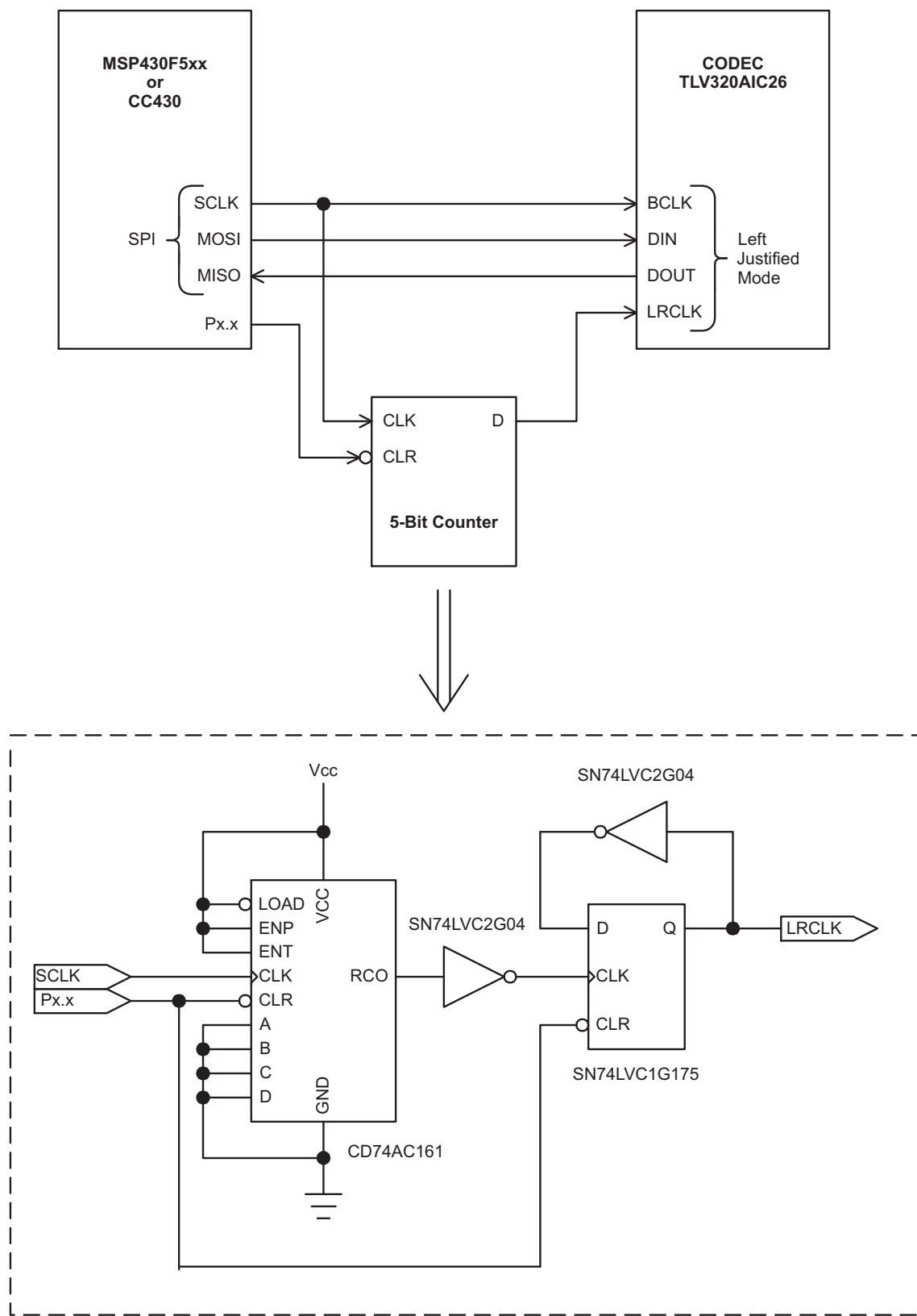


Figure 3. Possible Hardware Implementation

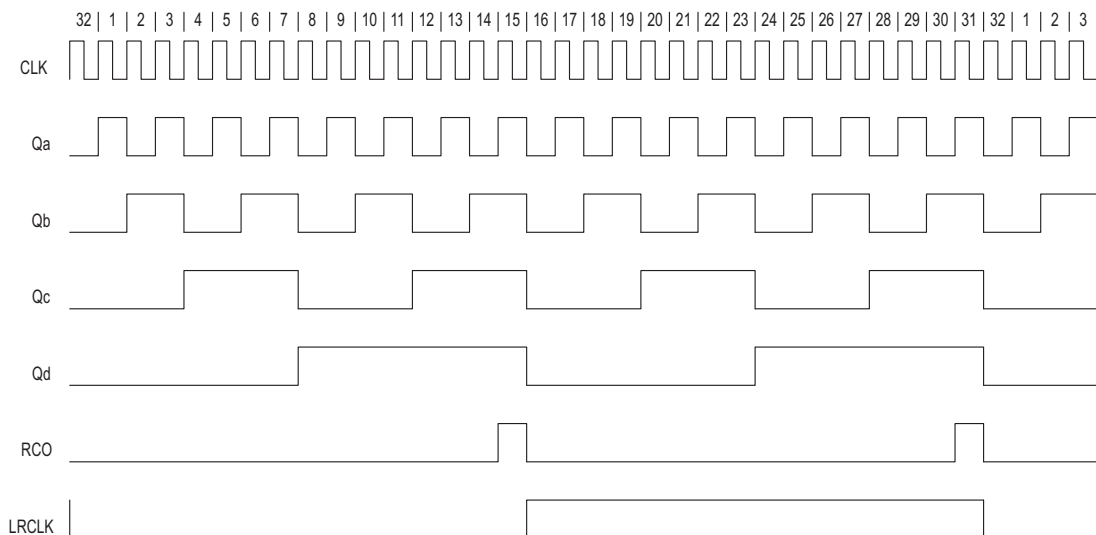


Figure 4. Converter Timing Diagram

Follows is sample 'C' code to initialize the UCB1 registers of a MSP430F5418 microprocessor for use with the hardware logic described previously.

// The I2S bus to/from the Codec is created using an SPI bus (UCB1)

```
UCB1CTL1 = UCSWRST;           // Hold USCI in Reset state
```

```
// UCB1 Settings:
// Master
// 3 Pin SPI
// 8 data bits
// Clock Source: SMCLK (8.192 MHz)
// Bit Rate Divider (BR0, BR1): 32 (Bit Rate = 256 kbps)
// No Modulation
// MSB First
// Clock Phase - UCCKPL = 0, UCCKPH = 1
// MSB first
```

```
UCB1CTL0 = UCCKPH | UCMSB | UCMST | UCMODE_0 | UCSYNC;
UCB1CTL1 |= UCSSEL_2;
UCB1BR0 = 32;
UCB1BR1 = 0;
```

```
UCB1CTL1 &= ~UCSWRST;        // Enable USCI
```

It is imperative in any I2S bus application that the data stream continuously; that is, that data be sent to the TX buffer as it empties, and that RX buffer be read and stored before the next byte becomes available. One way to assure this is through the use of the MSP430 DMA channels.

Sample code for setting up two DMA channels to handle the transfer of data to and from the codec is shown below. In this scheme, DMA channel 0 is set up to generate an interrupt every 'ADC Samples' (number of ADC samples). For the TLV320AIC26 codec, two 16-bit ADC samples are sent to the MSP430 every 1 / sample_rate seconds (e.g., 125 μ s for a sample rate of 8 kHz). Four memory bytes are allocated for each 'ADC_Sample'. The same is true for the Data Out stream; 4 bytes are allocated for every 'ADC_Sample'. In this case, ADC_SAMPLES has been set to 58; therefore, DMA channel 0 generates an interrupt every $58 \times 125 = 7250 \mu$ s.

```

#define ADC_SAMPLES 58                                // ADC samples per packet

// allocate memory for the codec (audio) buffers

char codec_in_bufferA[4*ADC_SAMPLES];                // Audio from Codec (4 bytes per sample)
char codec_in_bufferB[4*ADC_SAMPLES];                // Audio from Codec (4 bytes per sample)
char codec_out_bufferA[4*ADC_SAMPLES];               // Audio to Codec (4 bytes per sample)
char codec_out_bufferB[4*ADC_SAMPLES];               // Audio to Codec (4 bytes per sample)

char act_codec_in_bufnr = 0;                          // active codec_in buffer (0 | 1)
char act_codec_out_bufnr = 0;                         // active codec_out buffer (0 | 1)

// Set up the DMA Controller. Channel 0 is used for I2S bus D0 transfers (audio data
// to codec). Channel 1 is used for I2S DI bus transfers (audio data from codec)

DMACTL0 = 0x1617;          // DMA0TSEL = 23 (UCB1TXIFG)
                          // DMA1TSEL = 22 (UCB1RXIFG)

// DMA Channel 0 Control (I2S D0 using UCB1)
// DMADT = Repeated single transfer
// DMADSTINCR = Destination address is unchanged
// DMASRCINCR = Source address is incremented
// Transfer byte to byte
// DMA interrupts enabled

DMA0CTL = DMADT_4 | DMADSTINCR_0 | DMASRCINCR_3 | DMASBDB | DMAIE;
// Source block address
__data16_write_addr((unsigned short) &DMA0SA, (unsigned long) &codec_out_bufferA);
// Destination Address
__data16_write_addr((unsigned short) &DMA0DA, (unsigned long) &UCB1TXBUF);
DMA0SZ = sizeof(codec_out_bufferA);

// DMA Channel 1 Control (I2S DI using UCB1)
// DMADT = Repeated single transfer
// DMADSTINCR = Destination address is incremented
// DMASRCINCR = Source address is unchanged
// Transfer byte to byte
// DMA interrupts disabled

DMA1CTL = DMADT_4 | DMADSTINCR_3 | DMASRCINCR_0 | DMASBDB;
// Source block address
__data16_write_addr((unsigned short) &DMA1SA, (unsigned long) &UCB1RXBUF);
// Destination block address
__data16_write_addr((unsigned short) &DMA1DA, (unsigned long) &codec_in_bufferA);
DMA1SZ = sizeof(codec_in_bufferA);

```

The interrupt handler for DMA channel 0 is set up to switch the buffers used by both DMA channels to the 'other' buffer:

```
/ DMA Interrupt Service Routine
//-----

#pragma vector=DMA_VECTOR
__interrupt void DMA_ISR(void)
{
    DMA0CTL &= ~DMAIFG;

    if (act_codec_in_bufnr == 0) {
        // Destination block address
        __data16_write_addr((unsigned short) &DMA1DA, (unsigned long) &codec_in_bufferA);
    }
    else {
        // Destination block address
        __data16_write_addr((unsigned short) &DMA1DA, (unsigned long) &codec_in_bufferB);
    }
    act_codec_in_bufnr ^= 0x01;    // switch buffers

    if (act_codec_out_bufnr == 0)
        // Source block address
        __data16_write_addr((unsigned short) &DMA0SA, (unsigned long) &codec_out_bufferA);
    else
        // Source block address
        __data16_write_addr((unsigned short) &DMA0SA, (unsigned long) &codec_out_bufferB);
    act_codec_out_bufnr ^= 0x01;    // switch buffers

    sendpacket = 1;
}
```

Variable 'sendpacket' is a flag, used to indicate that 'ADC_SAMPLES' samples have been received, and that the codec buffers have been switched.

6 Conclusion

This application note describes a way to interface a standard CODEC (for example, TLV320AIC26), to an MSP430 microcontroller (for example, MSP430F5418) using a SPI port. It uses the left-justified mode variant of the standard I2S bus protocol and can be implemented simply and inexpensively using a 4-bit counter, a 'D' type flip-flop, and a dual inverter gate.

7 References

1. TLV320AIC26 specification sheet, *Low Power Stereo Audio Codec with Headphone/Speaker Amplifier and 12 Bit Battery/Temperature/Auxiliary ADC* ([SLAS412](#))

8 Document History

Revision	Date	Description/Changes
SWRA295	2010.03.01	Initial release

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps