Application Note

Interfacing to the ADC168M102R-SEP in Pseudo-Differential Operating Mode

Taiwo Arojojoye

ABSTRACT

This application note presents several methods for interfacing the ADC168M102R-SEP; a dual, 16-bit, 2x2 or 4x2 channel, radiation tolerant, simultaneous sampling analog-to-digital converter (ADC). This application note focuses on interfacing to the pseudo differential operating mode ADC168M102R-SEP, employing both the Multichannel Buffered Serial Port (McBSP) port and a combination of Serial Peripheral Interface (SPI) and pulse width modulator (PWM) peripheral, in the C2000 DSP to highlight the compatibility of the interface. The F28377D-SEP experiment kit is used to provide the source code in this application note.

Table of Contents

List of Figures

List of Tables

1 Introduction

The ADC168M102R-SEP is a dual, 16-bit, 1MSPS, radiation-tolerant, SAR ADC with eight input channels, which is configurable as eight pseudo-differential inputs (4x2) or four fully differential inputs (2x2). The 8 input channels are grouped into two pairs, followed by two independent sample and hold circuits for simultaneous signal acquisition. Based on the MODE pin M0/M1 configuration, this part can be configured as any combination of manual or automatic sequencing channel selection with two serial data output or single serial data output, as listed in Table 1-1.

Table 1-1. Channel Selection

M0	M1	Channel Selection	Output Pin Used
0	0	Manual (via SDI)	SDOA and SDOB
0	1	Manual (via SDI)	SDOA only
1	0	Automatic	SDOA and SDOB
1	1	Automatic	SDOA only

The channel information bits (one bit of channel indicator, followed by one bit of ADC indicator before the most significant bit of conversion result) are only available in fully-differential operating mode. For the consideration of channel identification, manual channel selection mode should be used when ADC168M102R-SEP is operating in pseudo-differential operating mode. Furthermore, for compatibility with most processors, single serial data output mode is selected in this application note, therefore, Mode II is mainly discussed in the following sections.

2 Hardware Platform

The ADC168M102REVM-PDK provides a platform to evaluate the functionality of the ADC168M102R-SEP ADC. This EVM can be used to connect with various DSPs or micro controllers from TI, while allowing access to both analog and digital signals for customized end-user applications. More information about the EVM is available in the ADC168M102REVM-PDK User Guide.

The *C2000 Experimenter Kits* from Texas Instruments are ideal products for initial device exploration and testing. The *DelfinoF28377D Experimenter Kit* has a docking station that features on-board USB JTAG emulation, and access to all control CARD signals, breadboard areas, and RS-232 and JTAG connectors. See http://www.ti.com/tool/tmdxdock28377d for more details about this kit.

The combination of the *DelfinoF28377D Experimenter Kit* and the *ADC168M102REVM-PDK* is a convenient way to experiment with interfacing the F28377D-SEP DSP to the ADC168M102R-SEP.

3 Hardware Interfaces

The F28377D-SEP comes with McBSP and SPI for communication with serial devices such as the ADC168M102R-SEP. In addition, it inherently brings multichannel enhanced pulse width modulator (ePWM) peripheral, which has the flexible capability of synchronization. Two methods for interfacing with the ADC168M102R-SEP are presented in the following sections.

The resistor between the DSP and ADC168M102R-SEP shown in the hardware connection (see Figure 3-1) is a reminder that high-speed digital signals can cause unwanted ringing in digital system which can ultimately degrade system performance. The requirement for the resistor depends on the clock speed, trace length and physical PCB layout in the end design.

3.1 Communication via McBSP

The McBSP consists of a data path and a control path to external devices. With the help of separated pins for transmission and reception, this supports full-duplex buffered data communication with a wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits. The McBSP also has four pins dedicated to the independent clocking and frame synchronizing for reception and transmission. More importantly, this integrates a programmable sample rate generator for internal generation and control of clock signals and frame synchronization signals. Because of these features, the McBSP simply communicates with different kinds of devices.

3.1.1 Using McBSP1

The 3.3V compatible digital interface of the ADC168M102R-SEP allows for a seamless connection between the F28377D-SEP Experiment Kit and the ADC168M102REVM-PDK. The hardware connections are shown in Figure 3-1. The chip select (CS) pin is controlled by the GPIO pin on the F28377D DSP to enable the ADC168M102R-SEP before any operation, thus allowing multiple devices to share the McBSP peripheral. Configure the internal sample rate generator (SRG) in McBSP to generate the transmit clock signal (CLKX). The Frame Sync Transmit (FSX) pulse can be used as the data read (RD) signal, since it is synchronized to the CLKX signal and its width can be set as 1 CLKX cycle. CLKR and FSR in the McBSP are defined as clock and frame synchronization signals for reception. To minimize the phase shift between receive data (DR) and receive clock(CLKR), CLKR is connected to the CLKX at the point which is very close to the ADC168M102R-SEP. FSR is connected in the same way.

In Mode II, the ADC168M102R-SEP needs 40 clock cycles to output the conversion result from both ADCs. If the start of the conversion (CONVST) signal is issued every 20 clock cycles (required for RD signal at that time) as in Mode I, every second pulse is ignored automatically, The CONVST signal can be connected to the FSX as well. To verify proper functionality and avoid corruption of the output data, the FSX signal should not be longer than one clock cycle.

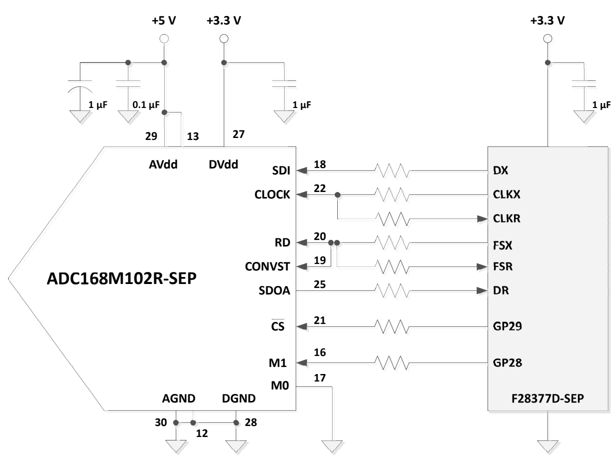

Figure 3-1. Hardware Connections Through McBSP

As shown in Figure 3-1 , the configuration command is sent through the DX pin of McBSP to the ADC168M102R-SEP SDI pin, including the reference voltage configuration, power mode configuration, input mode configuration, input channel selection, and so forth. The conversion result of both ADCs is transmitted through SDOA to DR pin in Mode II, the SDOB pin is not used in this particular operation mode.

For Mode II operation, control the M1 with GPIO, while connecting the M0 to ground, so that the time of changing the operation mode from Mode I to Mode II is precisely controlled by processor. With this method, you can foresee the channel information of the serial output, thus minimizing the software overhead of identifying the channel information of the conversion result.

Figure 3-2 shows the initialization process for Mode II using the previously mentioned hardware setup. Figure 3-3 shows magnification to one conversion cycle. The rising edge of RD and CONVST signal is synchronized to the rising edge of conversion clock and the width is set to be one clock period, the entire cycle repeats every 27 clock periods.

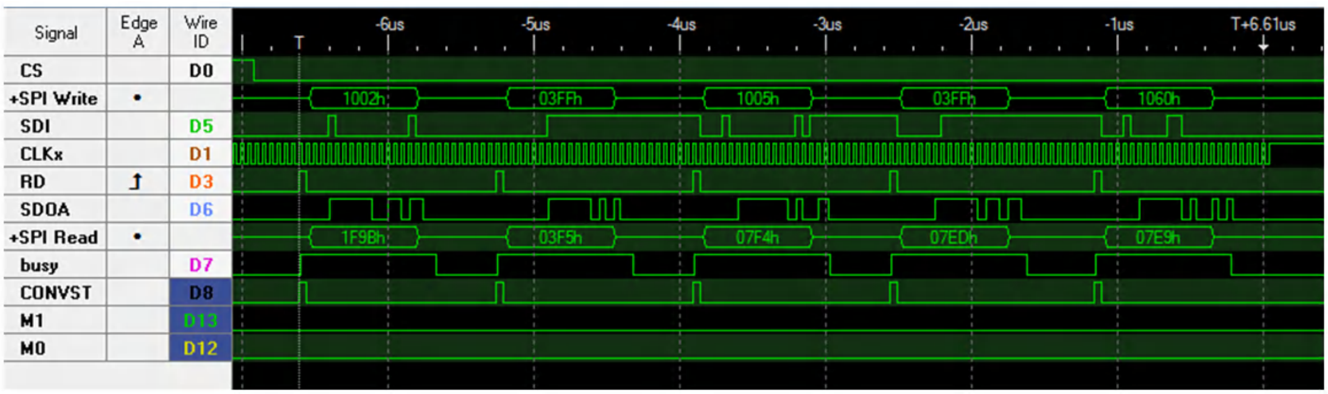

Figure 3-2. Mode II Waveforms Using McBSP

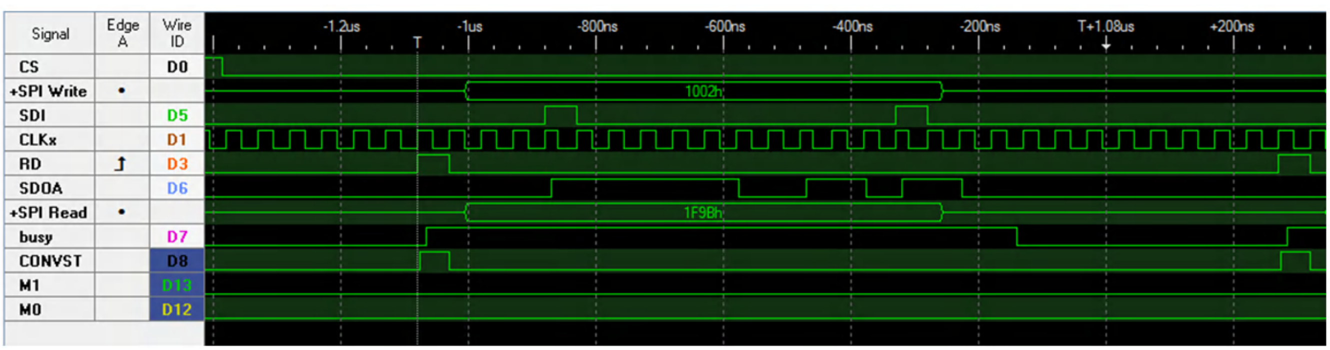

Figure 3-3. Expanded Mode II Waveform

The previously introduced interface scheme can be used in fully-differential input mode too, the only thing that needs to be changed is the *PDE* bit, which must be cleared in the CONFIG register.

3.1.2 Channel Identification

The fixed output sequence of the serial port can be used to identify the channel information of conversion results. In Mode II operation, after selecting the conversion channels via SDI at the beginning, another conversion cycle in needed to convert the selected channels. The following two conversion cycles are used to output the conversion result. For continuous conversion with channel sweeping in this operation mode, the channel selection command needs to be sent at the read cycle when the BUSY signal is low, in which case, the CONVST signal is ignored.

To maximize the operation efficiency, it’s suggested to set the M1 to high between the first two RD/CONVST signals shown as Figure 3-4, so that the first conversion cycle is used to set the channels which will be converted at the second conversion cycle, at which point the actual Mode II operation starts. The valid conversion result output starts with channel Ax (“x” depends on the channel configuration at the first conversion cycle, A0 in Figure 3-4) at the third conversion cycle.

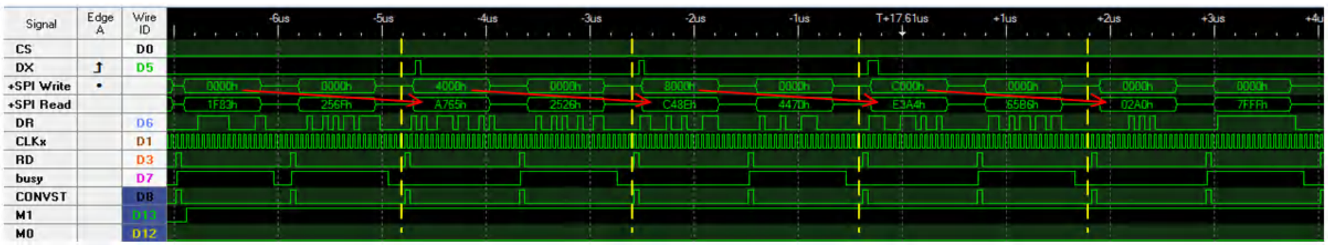

Figure 3-4. Continuous Conversion in Mode II

3.2 Communication via SPI and ePWM

For those processors which the McBSP peripheral is not available, it is possible to use the SPI and ePWM peripheral together to interface with the ADC168M102R-SEP. The ePWM peripheral used here needs to include at least two modules and have the capability of synchronizing to each other.

3.2.1 Using SPI and ePWM

One SPI port and two ePWM modules are combined together to interface with the ADC168M102R-SEP in this interfacing scheme, shown as Figure 3-5. The SPI port is configured as a slave device, the clock signal is provided from one of the ePWM modules, another ePWM module is used to control the RD and CONVST signal. These two ePWM modules are synchronized to each other with the time-base counter synchronization scheme.

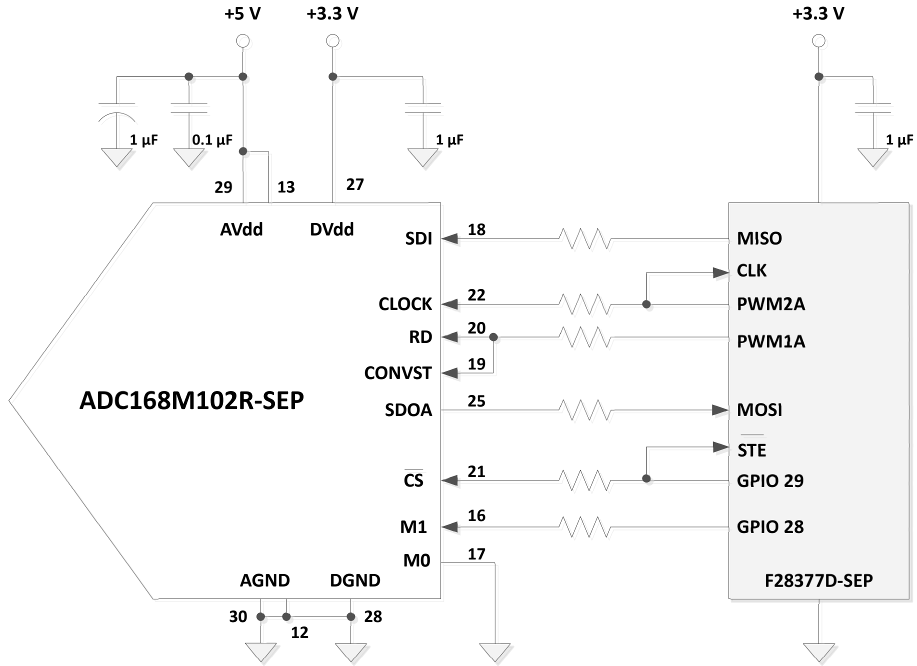

Figure 3-5. Hardware Connections for Mode II via SPI and ePWM

As shown in Figure 3-5, The SDI and SDOA signal of ADC168M102R-SEP are connected to the MISO and MOSI of the SPI port, respectively. Since the RD signal needs a falling clock edge to be validated, the transmitted data of the SPI port needs to be 1-bit right shifted. Detail of the software implementation is discussed in Section 4. Figure 3-6 shows one burst conversion series with eight conversion cycles and the corresponding six valid conversion results. The expanded first conversion cycle is shown in Figure 3-7.

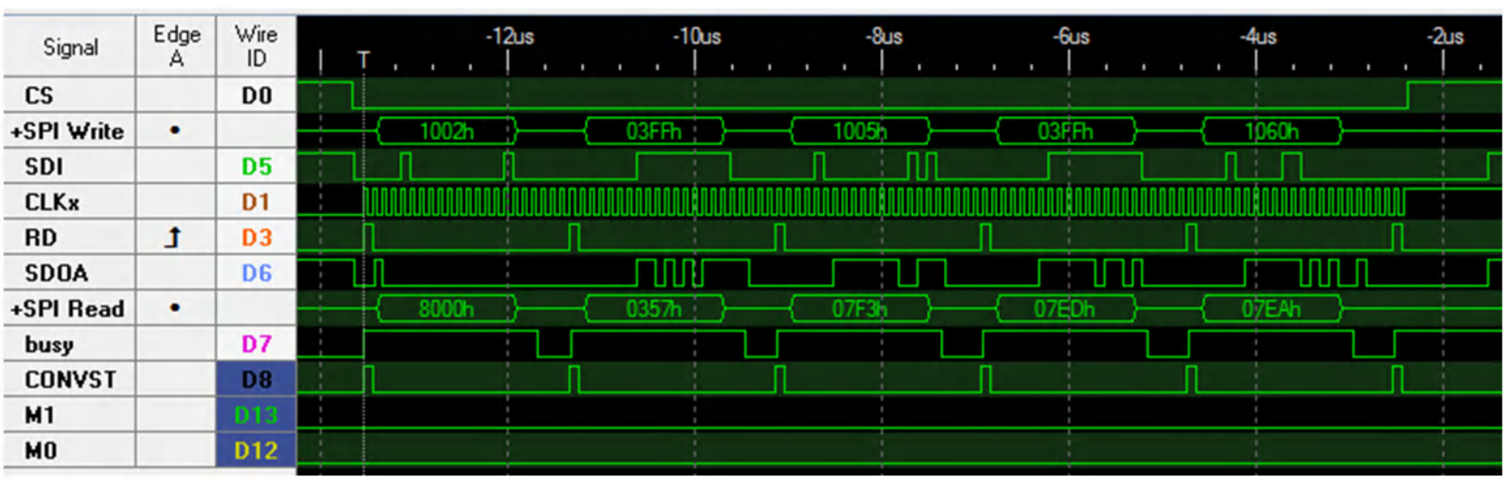

Figure 3-6. Mode II Waveforms using SPI and Epwm

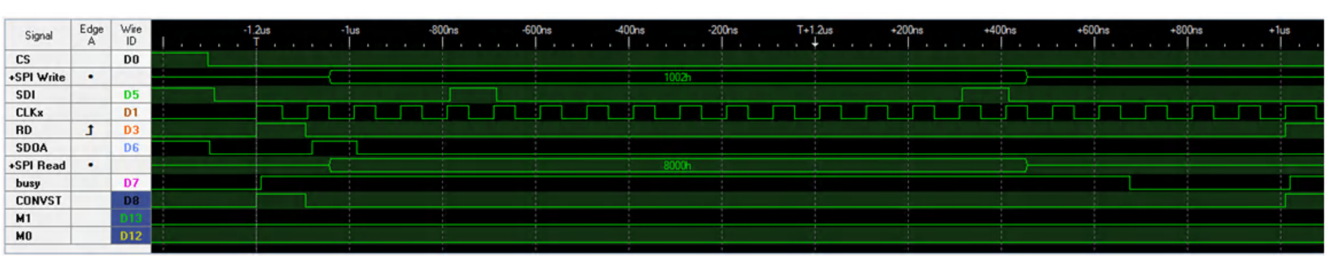

Figure 3-7. Expanded First Conversion Cycle

If there are two serial ports (McBSP + SPI or 2 SPI) available in the application, it is possible to communicate with the ADC168M102R-SEP with both the SDOA and SDOB serial outputs. Details are found in chapter 3.2 of SLAA167.

3.2.2 SPI Peripheral Requirement

The ADC168M102R-SEP needs at least 20 clock cycles to finish one conversion cycle. To ensure the conversion accuracy or maintain the constant conversion speed, it would be helpful if the SPI peripheral used here has the capability of handling at least 20 bits of data size or integrates with transmit and receive buffers. Otherwise, the software implementation should specify that the idle time between the two conversion portion, t_{idle}, is less than the maximum clock period (2 μs in the ADC168M102R-SEP while operating at half-clock mode). Figure 3-8 shows the waveform using an 8-bit SPI without transmit and receive buffer interfacing to the ADC168M102R-SEP.

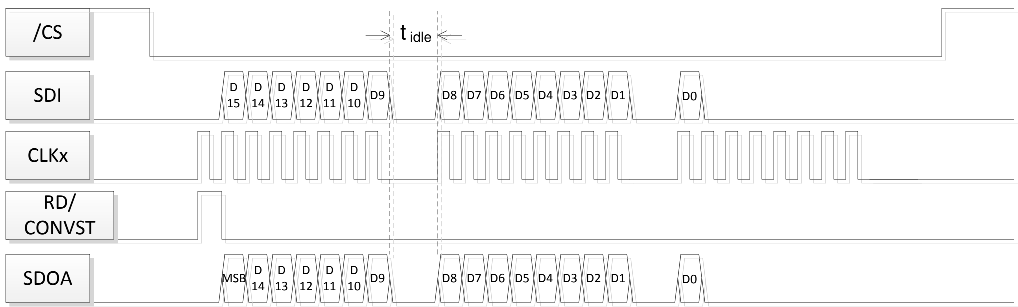

Figure 3-8. Interfacing to ADC168M102R-SEP with an 8-bit SPI without Buffer

4 Software Interface

All of the software is written in C language and compiled with Code Composer Studio version 5.5. The following sections describe the setup of the McBSP port, as well as the setup of SPI and ePWM peripheral for interfacing with Mode II operation of the ADC168M102R-SEP. Both polling and interrupt methods for McBSP received data process are used in the source code.

4.1 McBSP Settings

The McBSP is programmed as a serial port with the internal sample rate generator (SRG) configured to generate the internal data clock (CLKG), which is used as internal transmit clock (CLKX). The transmitter and receiver are set as 16-bit operation with a one-bit delay after frame synchronization and before the transmission/reception of the first bit of the frame.

The transmit frame-synchronization pulse (FSX) is generated by the SRG, the width of the FSX should be set as one CLKX clock cycle. Depending on whether the polling or interrupt mechanism is used to process the transmission and reception data, the FSX needs to be set as: (1) generated when the transmit register is written or (2) periodically based on the setting of FPER bits in SRGR2 register. Table 4-1 lists the key register settings of McBSP interfacing with the ADC168M102R-SEP using polling or interrupt methodology.

Table 4-1. Key Register Setting of McBSP

Register	Setting for Polling Operation	Setting for Interrupt Operation	Comments
SRGR2.bit.CLKSM	1	1	SRG use LSPCLK as input clock
PCR.bit.SCLKME	0	0	
SRGR1.bit.CLKGDV	9	9	Clock divider
PCR.bit.CLKXM	1	1	CLKX is driven by CLKG
RCR1.bit.RWDLEN1	2	2	Set as 16-bit operation
XCR1.bit.XWDLEN1	2	2	
RCR2.bit.RDATDLY	1	1	one-bit delay on data receive and transmit
XCR2.bit.XDATDLY	1	1	
PCR.bit.FSXM	1	1	FSX is driven by CLKG
SRGR1.bit.FWID	0	0	FSX is 1 CLKG cycle width
SRGR2.bit.FSGM	0	0	Different ways to generate FSX
SRGR2.bit.FPER	–	21	Period between two FSX signal
MFFINT.bit.XINT	–	1	Enable Tx/Rx interrupt.
MFFINT.bit.RINT	–	1	

In the sample code, the ADC168M102R-SEP is running with a serial clock of 20 MHz, which is achieved by divide ten from low-speed peripheral clock (LSPCLK). The fixed data rate continuous conversion of ADC168M102R-SEP is realized by using interrupt mechanism to process the data in McBSP, while the ADC168M102R-SEP is configured using polling mechanism to process the data in McBSP, shown as the following lines of code.

```
for(loopcount = 0; loopcount<5; loopcount ++)
{
    mcbsp_xmit(Txdata[Txcounter++],0);
    while(McbspaRegs.SPCR1.bit.RRDY == 0 ) { ; }    // Check for receive available?
    Rxdata[Rxcounter++] = McbspaRegs.DRR1.all;      // read out result.
    while(McbspaRegs.SPCR2.bit.XRDY == 0){ ; }    // Check for transmit ready?
}
```

4.2 SPI and ePWM Settings

F28377D-SEP integrates with high-speed synchronous SPI that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device, furthermore, it supports a 16-level receive and transmit FIFO.

In this interfacing scheme, the SPI port is configured as a slave device with 4-wire mode operation. For compatibility with ADC168M102R-SEP timing diagram, the SPI port is set to output data on rising edge and input data on falling edge with non-delayed clock.

In the source code, the character length of the SPI port is set as 11 bits, a buffer is employed in the transmission and reception, the next word in the buffer is transferred immediately upon completion of transmission of the previous one, thus combining two SPI transmissions into one ADC168M102R-SEP conversion cycle; the following several lines of code shows the translation between them. Table 4-2 lists the key register settings of the SPI port previously described.

```
1.One ADC168M102R-SEP configure command split into two SPI transmission cycles:
        for(Cnt=0; Cnt < number; Cnt++)
        {
           SpiaRegs.SPITXBUF = (*(data+Cnt) >> 1) & 0x7FE0 ;
           SpiaRegs.SPITXBUF = (*(data+Cnt) << 10) & 0xFC00;
        }
```

```
2. Two SPI reception cycles combine together to be one ADC168M102R-SEP conversion result:
        for(Counteri = 0; Counteri < Rxcounter/2; Counteri ++ )
        {
           RxInterpreter[Counteri] = ( Rxdata[Counteri * 2] << 6 )
                                     | ( Rxdata[Counteri * 2 +1] >> 5 );
        }
```

Table 4-2. Key Register Setting o SPI

Register	Setting	Comments
SPICTL.all	0x0002	As a slave, normal SPI clocking scheme without delay
SPIPRI.bit.TRIWIRE	0	Normal 4-wire SPI mode
SPIFFCT	0x0000	The next word in buffer is transferred immediately upon completion of transmission of previous one
SPICCR.all	0x008A	11 bits; data output on rising edge, input on falling edge

The two ePWM modules used is set to clock at the same rate and counter mode, with different counter period so that they could be used as RD/CONVST signal or clock signal. The counter value of ePWM2 module is synchronized to be zero when the counter value of ePWM1 module reaches to zero. Key register setting for ePWM module is shown in Table 4-3.

Table 4-3. Key Register Setting of ePWM

Register	ePWM1 Settings	PWM2 Settings	Comments
TBCTL.bit.HSPCLKDIV	1	1	Set the time base clock rate
TBCTL.bit.CLKDIV	1	1	
TBCTL.bit.CTRMODE	0	0	Set the counter mode as up-count mode
TBPRD	219	9	Set the period of the time-base counter
AQCTLA.bit.ZRO	2	2	Set the ePWM module output action
AQCTLA.bit.CAU	1	1	
CMPA.half.CMPA	10	5	Set compare register to control duty cycle
TBCTL.bit.PHSEN	0	1	Synchronization control, ePWM1 as master, ePWM2 as slave
TBCTL.bit.SYNCOSEL	1	0	
TBPHS.half.TBPHS	0	0	Set the counter value after synchronization

4.3 Software Flow

The sample code presented in this application report comes in two project files and two workspaces, corresponding to using the McBSP port or combining an SPI port with ePWM peripheral interfacing with the ADC168M102R-SEP.

In the program using McBSP port to interface with the ADC168M102R-SEP, the McBSP and other peripherals are configured appropriately once entering the main function. The configuration command for the ADC168M102R-SEP is sent, after that, the McBSP port is reset to use interrupt instead of polling to process the transmission and reception data. At this time, the channel configuration command can be sent and the corresponding conversion result is retrieved through interrupt routine. The program ends with 1026 data (128 data per channel, plus with two invalid data at the beginning) received. Figure 4-1 shows the example flow chart.

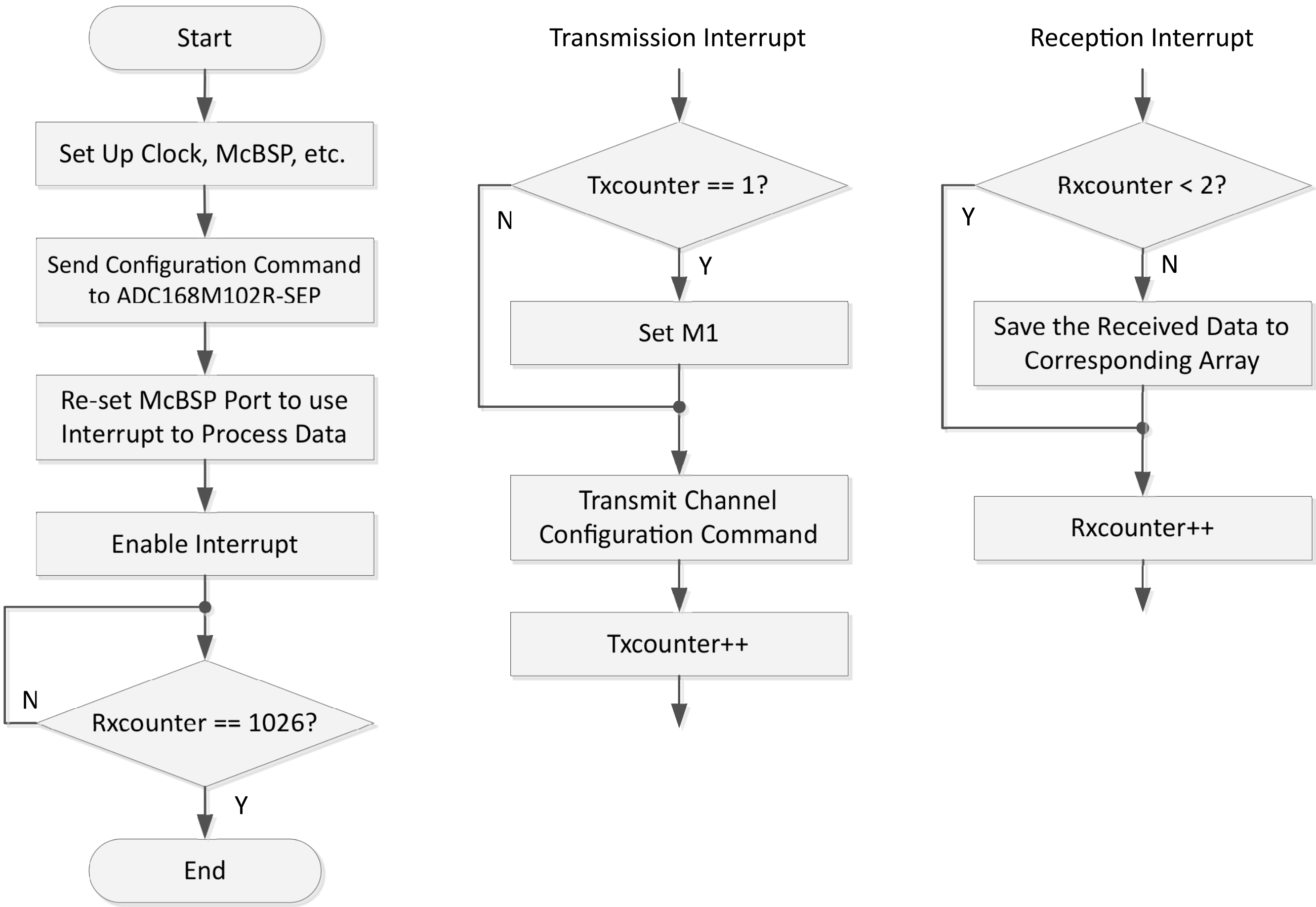

Figure 4-1. Software Flow Chart

5 Summary

This application note outlined multiple interfacing methods for the ADC168M102R-SEP. Specifically, this document examines the pseudo differential operating mode, demonstrating interface compatibility with the C2000 DSP through both the McBSP port and an SPI/PWM peripheral combination.

6 References

1. Texas Instruments, *ADC168M102R-SEP Radiation-Tolerant, 8-Channel, 1MSPS, 16-Bit ADC*, datasheet.
2. Texas Instruments, *TMS320F2837xD Dual-Core Real-Time Microcontrollers*, technical reference manual.
3. Texas Instruments, *TMS320x281x DSP Multichannel Buffered Serial Port (McBSP) Reference Guide*, reference guide.
4. Texas Instruments, *F2837xD Firmware Development Package User's Guide (F2837xD-FRM-EX-UG-100)*, user's guide.