Using the ADS8328 in Auto Trigger and Auto Channel Mode With the C6713 DSP

Lijoy Philipose — *DAP Nyquist ADC*

ABSTRACT

This application report presents one solution for interfacing the ADS8328, configured for auto trigger and auto channel mode, to the TMS320C6713 DSP. The hardware solution comprises of the ADS8328EVM, TMS320C6713 DSP Starter Kit (DSK), and the 5-6K Interface Board. The software demonstrates how to use a McBSP, EDMA, and a timer peripheral to collect data at full speed. The software developed is available for download to involve the user in the discussion and as sample code to use in system development.

Contents

List of Figures

List of Tables

1 Introduction

This application report presents a hardware and software solution for interfacing the ADS8328 16-bit analog-to-digital converter with the TMS320C6713 DSP. The software solution uses the EDMA, McBSP, and Timer1 peripherals of the DSP to collect 32,768 samples from each channel.

The hardware solution comprises of the TMS320C6713 DSK (‘C6713 DSK), 5-6K Interface Board, and the ADS8328EVM. The hardware described in this report is available from Texas Instruments.

1.1 TMS320C6713 DSK

The TMS320C6713 DSP Starter Kit ('C6713 DSK) not only provides an introduction to C6000™ technology, but is powerful enough to use for fast development of networking, communications, imaging, and other applications like data acquisition. For more information, search for part number TMDSDSK6713 on the TI Web site at www.ti.com.

1.2 ADS8328EVM

The ADS8328 Evaluation Module (ADS8328EVM) provides an easy way to test both the functional and dynamic performance of this 16-bit analog-to-digital converter (ADC). The ADS8328EVM includes those circuits essential to showing the performance of the converter and interfacing it to a serial bus. These circuits include the analog driver, reference, and nominal power supply filtering. Table 1 shows how the ADS8328EVM was configured.

The ADS8328EVM was powered with ±6-V, +5-V, and +3.3-V supplies. The ±6-V supply was used to power the bipolar amplifiers. The +5-V source powered the ADS8328 analog supply pin and the reference ICs on the board. Lastly, the +3.3-V supply powered the digital interface pin of the ADS8328. This allows the ADS8328 to communicate properly to the ‘C6713DSK 3.3-V logic.

Table 1. Jumper Settings for ADS8328EVM

Reference Designator		Description	Jumper Setting	
			1-2	2-3
W1		Connect output from U6 to REF+ pin of DUT.	**Installed**	
		Connect pin 8 of DUT to REF+ pin of DUT.		**Not installed**
W2		Connect output from U2 to REF+ pin of DUT.	**Not installed**	
		Connect External reference to REF+ pin of DUT.		**Not installed**
W3		Connect reference chip (U2) to reference buffer U6.	**Installed**	
		Connect pin 8 of DUT to reference buffer U6.		**Not installed**
W4		Connect SDO of DUT to I/O socket (P2B) pin 13.	**Installed**	
		Connect SDO of DUT to I/O plug (P2T) pin 13.		**Not installed**
W5		Connect SDO from socket to EOC/INT	**Installed**	
		Connect EOC/ INT to P2T and P2B pin 15		**Not installed**
W6		Connect +3.3VD to W7 pin 1	**Installed**	
		Connect +1.8VD to W7 pin 1		**Not installed**
W7		Connect pin 2 of W6 to +VBD	**Installed**	
		Connect +5VD to +VBD		**Not installed**
W8	1-2	Short CNTL1 to CS	Not installed	
	3-4	Short FSX to CS	Installed	
	5-6	Connect pin 19 of P2T and P2B to CS	Not installed	
	7-8	Connect pin 12 of P2T and P2B to CS	Not installed	
	9-10	Connect pin 14 of P2T and P2B to CS	Not installed	

1.3 5-6K Interface Evaluation Module

Texas Instruments builds many data acquisition evaluation modules (EVM) that have a common set of connectors and signals at those connectors. The 5-6K Interface Board allows designers to easily connect those EVMs to the C5000™ and C6000™ family of digital signal processor starter kits (DSK).

The 5-6K Interface Board consists of two serial connectors, two signal-conditioning areas, and a parallel interface. See TI literature number SLAU104 for more information on the 5-6K Interface Board, or search for keyword *5-6K Interface* on the TI Web site www.ti.com.

The ADS8328EVM plugs onto the 5-6K Interface Board. Align the board such that P1 of the ADS8328EVM plugs into J10 of the 5-6K Interface Board. Likewise, align P2B of the ADS8328EVM onto J16 of the 5-6K Interface Board.

Power for both the 5-6K Interface Board and ADS8328EVM can be applied at J1 and J2. The 5-6K Interface Board brings up +5 V and +3.3 V from the ‘C6713DSK board. This can be used to power the digital supply pin of ADS8328; simply short W2 and W3 across pins 2 and 3. In this case, the user needs only to supply ±6-V and +5-V supplies at J1 and J2.

Table 2. 5-6k Interface Card Rev B

Reference Designator	Jumper Settings		Comments
	1-2	2-3	
W1	**Open**	N/A	
W2		Shorted	
W3		Shorted	
W4		Shorted	
W5		Shorted	
W6	Shorted		
W7	Shorted		
W8	Shorted		
W9	Shorted		
W10	Shorted		
W11	Shorted		
W12	Shorted		
W13	Shorted		
J14	Shorted across pins 1 and 2		
J13	Shorted across pins 7 and 8		

1.4 Hardware Connections

The hardware connections described in this document are shown in Figure 1. The ‘C6713 McBSP signals CLKX0, FSX0, DXR0, and DRR0 are connected to ADS8328 signals SCLK, FS/CS, SDI, and SDO, respectively. The EOC signal is connected to the external interrupt six.

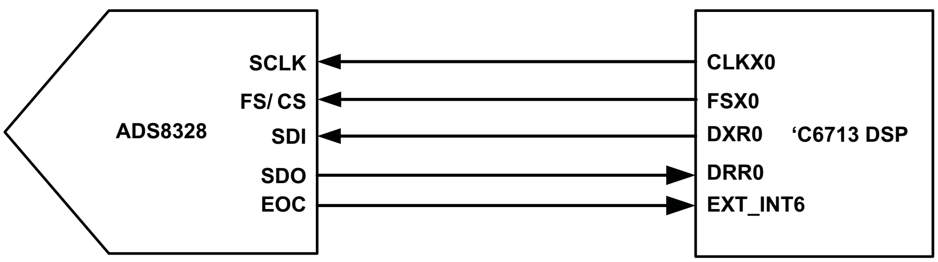

Figure 1. Hardware Connection

2 Software Interface

The software interface objective is to collect a block of samples as quickly as possible, while freeing up the processor to perform other tasks. The processor should be alerted only when the samples are ready for processing. The most efficient way of doing this is to use an EDMA channel, one of the timers, and an interrupt routine. For this discussion, it is assumed that the reader is familiar with the DSP and its peripherals. If that is not the case, the reader should study the application reports and data sheets listed in the references section at the end of this document.

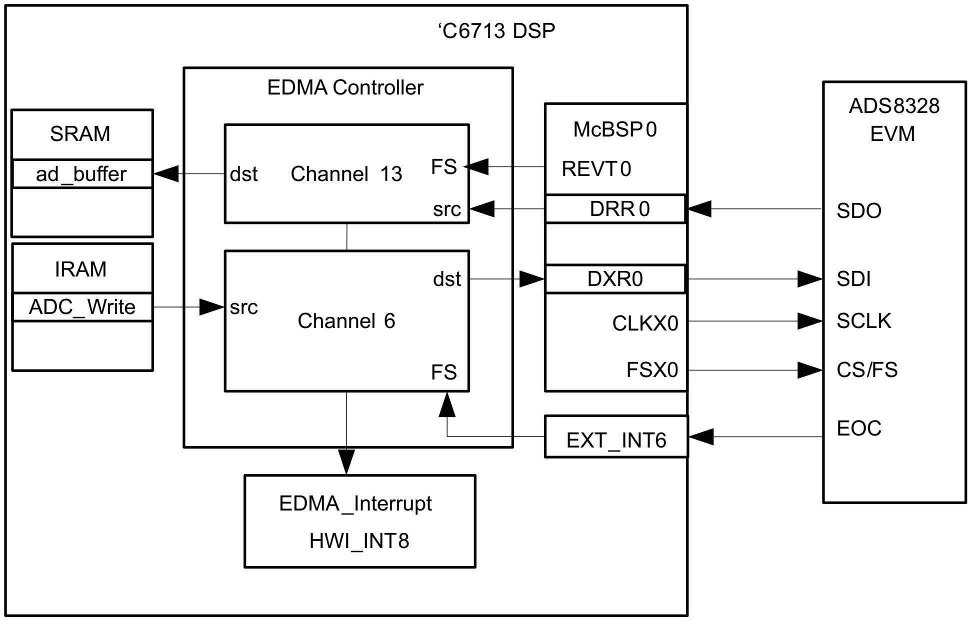

Figure 2. Data Transfer Block Diagram

Figure 2 provides a visual representation of the data flow through the hardware.

The EDMA channel 13 is synchronized to the McBSP0 receive data ready event. EDMA channel 6 is synchronized to the EOC/INT pin of the ADS8328. Every interrupt event (i.e., falling edge of EOC) triggers a transfer of the *read data* instruction to the data transmit register of the McBSP0. This then triggers a serial data transfer of that command out to the converter. At the end of that 16-bit cycle, a receive event occurs. As a result, EDMA channel 13 transfers data from the receive data register to the memory location of ad_buffer. Once all 65,536 samples are captured, the EDMA transfers future data points to location *temp*. Also at this point, the EDMA controller signals the processor via EDMA interrupt. The processor executes a number of instructions in the EDMA interrupt service routine including sorting the data, resetting peripherals, and re-enabling the data transfer system.

2.1 ADS8328

The ADS8328 is a 16-bit analog-to-digital converter with a high-speed serial interface. It offers users many modes of operations. The following code snippet shows how the ADS8328 is configured in this document.

```
adc_cfgReg =WRITE_CFR |                /*write REQUIRED setting to CFR */

  !AUTO_CHAN|CCLK_INT|MAN_TRG|D8_DNC|  /*1101*/

  !POL_INT_EOC_LOW|PIN10_EOC|PIN10_OUTPUT|ANAP_DISABLE|  /*0111*/

  RESUME_NAP|RESUME_DEEP|!TAG|NO_RESET;  /*1101 */
```

The device is set up for internal conversion clock mode. This allows the user to set the I/O clock to be as fast or as slow as desired.

The ADS8328 is configured to trigger conversions and to toggle between channels automatically.

Pin 10 is configured to behave as an End-of-Conversion signal and to be active HIGH. Whenever this signal goes low – indicating the end of the conversion process – the EDMA channel 6 triggers a data read of the converted data.

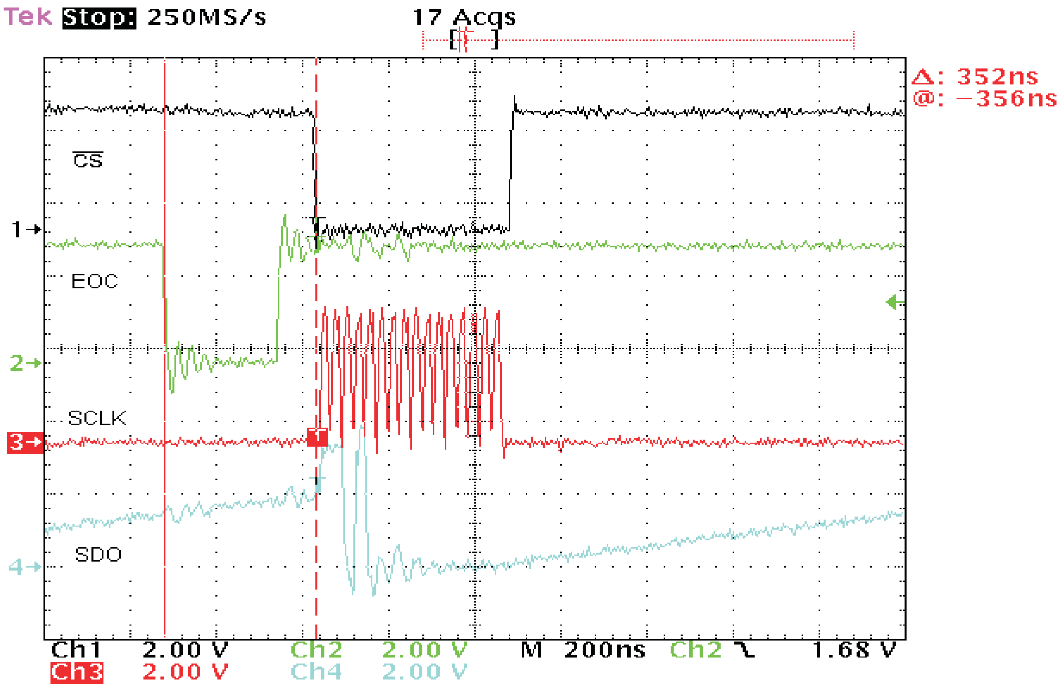

Figure 3. ADS8328 Interface Screen Shot

All the power-down modes and the TAG feature are disabled.

This device supports a high-speed serial clock (SCLK). SCLK is typically the serial interface clock. It can be as high as 50 MHz when the analog supply is set to +5 V and up to 33 MHz when the analog supply is +2.7 V. When SCLK is used as the interface and conversion clock, then the maximum SCLK rate is reduced to 21 MHz. This document uses the internal clock as the conversion clock and SCLK as the serial interface clock. This is why SCLK is set to 37.5 MHz.

See the ADS8328 product data sheet (SLAS415) for detailed explanation of the device or any of the aforementioned items.

2.2 'C6713 DSP

The TMS320C6713 DSP and its respective peripherals (i.e., Timer1, EMIF, and EDMA) need to be set up to work with the converter. It is assumed the reader has a working understanding of the host processor; therefore, the DSP setup is not covered in much detail in this document. See the downloadable code and references listed at the end of this document for more information.

The settings for the various peripherals can be found and modified in the *config1.cdb* file. The seed file for the DSP/BIOS configuration file was the *dskC6713.cdb* file. All the peripheral register settings and interrupt control were handled within this configuration file, allowing the user code to be simple, containing only statements to enable the peripherals.

```
MCBSP_Config mcbspCfg0 = {
    0x03001000,        /*  Serial Port Control Reg. (SPCR)   */
    0x00050040,        /*  Receiver Control Reg. (RCR)   */
    0x00050040,        /*  Transmitter Control Reg. (XCR)   */
    0x200F0002,        /*  Sample-Rate Generator Reg. (SRGR)   */
    0x00000000,        /*  Multichannel Control Reg. (MCR)   */
    0x00000000,        /*  Receiver Channel Enable(RCER)   */
    0x00000000,        /*  Transmitter Channel Enable(XCER)   */
    0x00000A0D         /*  Pin Control Reg. (PCR)   */
};
```

The register settings for the Multi-Channel Buffered Serial Port (McBSP) is shown in the preceding code. The serial port is set up for 16-bit clock stop mode with no delay. CLKX0 is an output pin, and the SCLK (CLKX0) signal is generated by the sample rate generator. The data shifted out is delayed by one clock. In clock stop mode, CLKX0 and CLKR0 signals are tied together internally. The serial shift clock rate was set in the sample rate generator register to 37.5 MHz.

SCLK = (CPU CLOCK/2)/CLKDIV = (225 MHz/2)/3 =37.5 MHz.

```
EDMA_Config edmaCfg13 = {
    0x20360003,         /*  Option  */
    0x00000000,         /*  Source Address - Numeric   */
    0x00000000,         /*  Transfer Counter - Numeric  */
    (Uint32) ad_buffer,         /*  Destination Address - Extern Decl. Obj  */
    0x00000000,         /*  Index register - Numeric  */
    0x00010000          /*  Element Count Reload and Link Address  */
};
```

Register settings for EDMA channel 13 is shown in the preceding code. The EDMA channel is configured to transfer 65,536 data samples from the McBSP0 receive data register. Each of those transfers are triggered by a receive event from McBSP0. The EDMA channel is linked to the EDMA configuration shown in the following code. After 65,536 samples are transferred, the EDMA loads settings which transfer any additional data samples to the location of *temp*. This is done to ensure that the EDMA does not overwrite the data stored in array ad_buffer.

```
EDMA_Config edmaCfg6 = {
    0x48060003,         /*  Option  */
    (Uint32) &ADC_Write,         /*  Source Address - Extern Decl.Obj  */
    0x00000000,         /*  Transfer Counter - Numeric  */
    0x00000000,         /*  Destination Address - Numeric   */
    0x00000000,         /*  Index register - Numeric  */
    0x00010000          /*  Element Count Reload and Link Address  */
};
```

EDMA channel 6 transfers the read command to the data transmit register of the McBSP0. This action triggers the McBSP to trigger an I/O cycle with the ADS8328. The *ADC_Write* variable is loaded with the read data command.

```
EDMA_Config edmaCfg0 = {
    0x20160003,         /*  Option  */
    0x00000000,         /*  Source Address - Numeric   */
    0x00000000,         /*  Transfer Counter - Numeric  */
    (Uint32) &temp,         /*  Destination Address - Extern Decl. Obj  */
    0x00000000,         /*  Index register - Numeric  */
    0x00010000          /*  Element Count Reload and Link Address  */
};
```

After taking 65,536 sample points, EDMA channel 13 is configured to transfer any future interrupts to the location of *temp*. The EDMA controller triggers an interrupt to the DSP. The processor responds by executing the *hwiDMA_isr()* function. The hardware interrupt service routine resets the convert start trigger (Timer1) and the EDMA channels. It reconfigures the EDMA channels and flushes the McBSP receive buffers. After which point, it executes the necessary steps to start the process all over again.

3 Conclusion

This application report presents one solution to interfacing the ADS8328 converter to the ‘C6713 DSP. The ADS8328EVM plugs onto the 5-6K Interface Board, which in turn plugs onto the ‘C6713 DSK. All the hardware described in this application report can be ordered from Texas Instruments. The software solution involves using the DSP/BIOS and configuration tool (i.e., *config1.cdb* file) to visually set up the EDMA and interrupts. The C-language code development with this report is available for download on the TI Web site at www.ti.com.

4 References

1. *TMS320C621x/TMS320C671x EDMA Architecture* application report (SPRA996)
2. *Applications Using the TMS320C6000 Enhanced DMA* application report (SPRA636)
3. *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (SPRU234)
4. *ADS8327, ADS8328, Low Power, 16-Bit 500-kHz, Single/Dual Unipolar Input, Analog-to-Digital Converters With Serial Interface* data sheet (SLAS415)
5. *TMS320C6713 Floating Point Digital Signal Processor* data sheet (SPRS186)
6. *5-6K Interface Board* user's guide (SLAU104)

Appendix A MAIN.C

```
/*************************************************************
* FILENAME: main.c
* DESCRIPTION: This program uses McBSP0 of C6713 DSK to read
* samples continuously from the ADS8328 16-bit, 500KSPS
*    Analog-to-Digital Converter.  The 16-bit words read during
*   each data frame is stored at "ad_buffer" by a EDMA transfer.
*  The ADS8328EVM plugs onto the 5-6k interface card.  The 5-6k
* interface card plugs onto the C6713 DSK.
* Conversions are initiated by Timer1 programmed to be pulse.
* One EDMA channel is triggered by Timer1 to send the read data
* command to the ADS8328.  Another EDMA channel is used to transfer
* data from the McBSP DRR0 register to memory.  This program is setup
* to continuously collect data.  A probe point can be placed in the HWI
* function to watch the data.
* Hardware connections:
*    6713DSK (DSP/5-6K interface card) => ADS8328EVM (RevA)
*    CLKX0 -> SCLK
*    FSX -> CSn\FS
*    DXR0 -> SDI
*    DRR0 -> SDO
*   Timer1 => CONVST#
* AUTHOR    : DAP Application Group, L. Philipose, Dallas
*    CREATED 2006 © BY TEXAS INSTRUMENTS INCORPORATED.
* VERSION: 1.0
*************************************************************/
/*  Include Header File  */
#include "Config1cfg.h"
#include "dsk6713.h"
#include "dc_conf.h"
#include "target.h"
/* Create the buffers. We want to align the buffers to be cache friendly */
/* by aligning them on an L2 cache line boundary.                        */
#pragma DATA_SECTION(ad_buffer,".mydatasection");
#pragma DATA_ALIGN(ad_buffer,(BLOCK_SZ));
unsigned int ad_buffer[BLOCK_SZ];  /*data from 16-bit read */
unsigned ad_buffer[BLOCK_SZ];  /*data from 16-bit read */
#pragma DATA_SECTION(ad_chan0,".mydatasection");
#pragma DATA_ALIGN(ad_chan0,(BLOCK_SZ/2));
unsigned short ad_chan0[BLOCK_SZ/2];  /*data from 16-bit read */
#pragma DATA_SECTION(ad_chan1,".mydatasection");
#pragma DATA_ALIGN(ad_chan1,(BLOCK_SZ/2));
unsigned short ad_chan1[BLOCK_SZ/2];  /*data from 16-bit read */
unsigned int ADC_Write=0xd000;
unsigned int temp=0;
int readout, adc_cfgReg0=0, adc_cfgReg1=0;
/*****************************************************************************\
* Function: main()
* Description: Initializes and enables C6713DSK, McBSP0, EDMA, timer, and
*  interrupts.
\*****************************************************************************/
void main()
{  int I=0;
    /* Initialize the board support library, must be first BSL call */
    DSK6713_init();
```

```
    /* Set McBSP0 for use with daughtercard */
    DSK6713_rset(DSK6713_MISC,MCBSP1SEL);
   /* Initialize the ad_buffer */
    for (i=0;i<BLOCK_SZ;i++){
      ad_buffer[i]= 0x0;
    }
    TIMER_reset(hTimer1);
/* Enable the EDMA controller interrupt */
      IRQ_reset(IRQ_EVT_EDMAINT);                    /*Reset EDMA interrupt         */
      IRQ_enable(IRQ_EVT_EDMAINT);
    EDMA_intDisable(TCCINTNUM6);                /*Disable EDMA interrupt     */
      EDMA_intClear(TCCINTNUM6);                    /*Clear EDMA interrupt         */
      EDMA_intEnable(TCCINTNUM6);                  /*Enable EDMA interrupt        */
/*Configure EDMA Channel and McBSP0*/
    EDMA_config(hEdmaCha6,&edmaCfg6);        /*read sync'ed with interrupt*/
    EDMA_config(hEdmaCha13,&edmaCfg13);
       MCBSP_config(hMcbsp0,&mcbspCfg0);
//     TIMER_config(hTimer1,&timerCfg1);
/*Start McBSP0*/
    MCBSP_start(hMcbsp0,MCBSP_RCV_START | MCBSP_XMIT_START
                | MCBSP_SRGR_START| MCBSP_SRGR_FRAMESYNC, 0x1000);
adc_cfgReg0 = WRITE_CFR |  /*write REQUIRED settings to CFR */
   AUTO_CHAN       | CCLK_INT    | !MAN_TRG     | D8_DNC|  /* 1101*/
   !POL_INT_EOC_LOW| PIN10_EOC   | PIN10_OUTPUT | ANAP_DISABLE| /*0111*/
   RESUME_NAP      | RESUME_DEEP| !TAG          | NO_RESET;  /* 1101*/
//        ADC_Write =READ_CFR;        //Read back the configuration register.
                                //Check datasheet for how readback functions.
//        while (!MCBSP_xrdy(hMcbsp0));
//        MCBSP_write(hMcbsp0,ADC_Write);
//        while (!MCBSP_rrdy(hMcbsp0));
//        readout=MCBSP_read(hMcbsp0);
ADC_Write =READ_DATA;
while (!MCBSP_xrdy(hMcbsp0));
    MCBSP_write(hMcbsp0,adc_cfgReg0);
    while (!MCBSP_rrdy(hMcbsp0));
    temp=MCBSP_read(hMcbsp0);
//Clear any pending EDMA interrupts on Channel 6 and 13.
    EDMA_clearChannel(hEdmaCha6);
    EDMA_clearChannel(hEdmaCha13);
//Start EDMA channel.
    EDMA_enableChannel(hEdmaCha6);        /*Enable EDMA channel 6 -Triggered from ADC
interrupt, writes read ADC command    */
    EDMA_enableChannel(hEdmaCha13);        /*Enable EDMA channel 13 -REVT0    */
//Start Convert start pulse.
//    TIMER_start(hTimer1);              //uncomment for manual trigger mode.

}
```

Appendix B Functions.C

```
/*************************************************************
* FILENAME: functions.c
* DESCRIPTION: This program uses McBSP0 of C6713 DSK to read
* 65k samples continuously from the ADS8328 16-bit, 500KSPS
*    Analog-to-Digital Converter.  The 16-bit words read during
*   each data frame is in "ad_buffer".
*  The ADS8328EVM plugs onto the 5-6k interface card.  The 5-6k
* interface card plugs onto the C6713 DSK.
* Hardware connections:
* Hardware connections:
*    6713DSK (DSP/5-6K interface card) => ADS8328EVM (RevA)
*    CLKX0 -> SCLK
*    FSX -> CSn\FS
*    DXR0 -> SDI
*    DRR0 -> SDO
*   Timer1 => CONVST#  (optional)
* AUTHOR    : DAP Application Group, L. Philipose, Dallas
*    CREATED 2006 © BY TEXAS INSTRUMENTS INCORPORATED.
* VERSION: 1.0
*************************************************************/
/*  Include Header File  */
#include "Config1cfg.h"
#include "dsk6713.h"
#include "dc_conf.h"
extern unsigned int ADC_Write, readout, adc_cfgReg0, adc_cfgReg1;
extern unsigned short ad_chan0[BLOCK_SZ/2];  /*data from 16-bit read */
extern unsigned short ad_chan1[BLOCK_SZ/2];  /*data from 16-bit read */
extern unsigned ad_buffer[BLOCK_SZ];
/*************************************************************/
/*hwiDMA_isr():                                               */
/*     Hardware Interrupt Function disables EDMA channels and     */
/*    Timer1, Then post software interrupt.                   */
/*************************************************************/
void hwiDMA_isr()
{
int i=0,j=0,temp;
/*Reset Timer1*/
//     TIMER_reset(hTimer1);    //uncomment when using manual trigger mode.
/*Reset EDMA channels*/
   EDMA_reset(hEdmaCha6);
   EDMA_reset(hEdmaCha13);
//Sort data channels
    while(j<BLOCK_SZ/2)
    {
        while(i<BLOCK_SZ)
        {
            ad_chan0[j] = (unsigned short) ad_buffer[i];
            ad_chan1[j++] = (unsigned short) ad_buffer[i+1];
            i=i+2;
        }
    }
//Reset ADC
    while (!MCBSP_xrdy(hMcbsp0));        //Reset A/D
    MCBSP_write(hMcbsp0,WRITE_RESET);
```

```
    while (!MCBSP_rrdy(hMcbsp0));
    temp=MCBSP_read(hMcbsp0);
IRQ_reset(IRQ_EVT_EDMAINT);                    /*Reset EDMA interrupt         */
    EDMA_intDisable(TCCINTNUM6);                /*Disable EDMA interrupt     */
      EDMA_intClear(TCCINTNUM6);                    /*Clear EDMA interrupt         */
      EDMA_intEnable(TCCINTNUM6);                  /*Enable EDMA interrupt        */
    IRQ_enable(IRQ_EVT_EDMAINT);
//Configure EDMA channels.
    EDMA_config(hEdmaCha6,&edmaCfg6);        /*read sync'ed with interrupt*/
    EDMA_config(hEdmaCha13,&edmaCfg13);
temp=MCBSP_read(hMcbsp0);            //flush receiver buffer
    temp=MCBSP_read(hMcbsp0);
    temp=MCBSP_read(hMcbsp0);
//Write and enable ADC
    while (!MCBSP_xrdy(hMcbsp0));        //enable ADC
    MCBSP_write(hMcbsp0,adc_cfgReg0);        //auto trigger and channel
    while (!MCBSP_rrdy(hMcbsp0));
    temp=MCBSP_read(hMcbsp0);
EDMA_clearChannel(hEdmaCha6);
    EDMA_clearChannel(hEdmaCha13);
EDMA_enableChannel(hEdmaCha6);        /*Enable EDMA channel 6 -Triggered from ADC
interrupt, writes read ADC
command    */
EDMA_enableChannel(hEdmaCha13);        /*Enable EDMA channel 13 -REVT0    */
/*Start Timer1 */
//    TIMER_config(hTimer1,&timerCfg1);  //uncomment next two lines for manual trigger
mode.
//    TIMER_start(hTimer1);

}
```

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 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.

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.

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
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265