TEXAS INSTRUMENTS

Application Report
SLAA239–March 2005

Using the ADS8382 With the TMS320C6713 DSP

Lijoy Philipose *Data Acquisition Applications*

ABSTRACT

This application report presents a solution to interfacing the ADS8382 18-bit, 600-KSPS serial interface converter to the TMS320C6713 DSP. The hardware solution is made up of existing hardware: the ADS8382EVM, ‘C6713 DSK, and 5-6K Interface Board. The software demonstrates how to use an EDMA and Timer peripheral to collect data at 598 kHz. Discussed also are some of the key points to remember when designing the hardware and writing the software. The software developed is available for download to involve the user in the discussion and as sample code to use in system development. Project collateral discussed in this application report can be downloaded from the following URL: www.ti.com/lit/zip/SLAA239.

Contents

List of Figures

List of Tables

Trademarks

Code Composer Studio is a trademark of Texas Instruments.

1 Introduction

The ADS8382 converter is the first high-resolution, high-speed, serial interface SAR converter to come from Texas Instruments’ Burr Brown line of products. It is an analog-to-digital converter with an 18-bit resolution and a 600-kHz sample rate. This application report presents a hardware and software solution to interfacing and using the converter with the TMS320C6713 digital signal processor (DSP). The software developed uses the EDMA, in combination with Timer1, to collect 2048 samples. Discussed also are some key points to remember when interfacing the converter to host processors.

2 Hardware

The hardware solution involves theTMS320C6713 DSP Starter Kit (DSK), 5-6K Interface Board, and the ADS8382EVM. The hardware used in this report is available from Texas Instruments.

2.1 TMS320C6713 DSK

The TMS320C6713 DSK (‘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.

2.2 ADS8382EVM

The ADS8382 evaluation module (EVM), or ADS8382EVM, is an easy way to test both the functional and dynamic performance of this 18-bit, analog-to-digital converter (ADC). The EVM includes those circuits essential to showing the performance of the converter and interfacing it to a serial bus. These circuits include the analog input, reference, power, and digital buffer circuits. Table 1 describes solder and pin jumper settings used in this application report.

Table 1. Jumper Settings for ADS8382EVM

REFERENCE DESIGNATOR[1]	DESCRIPTION	JUMPER	
		Pins 1-2	Pins 2-3
SJP1	Buffer onboard reference, REF1004-2.5	**Installed** [2]	
	Buffer user reference voltage applied at P1 pin 20		Installed
SJP2	Apply external reference directly to SJP4	**Installed** [2]	
	Apply buffered external reference to SJP4		Installed
SJP3	Set negative supply of U3 to ground	Installed	
	Select negative supply of U3 to –VCC.		**Installed** [2]
SJP4	Apply internal reference to REFIN	**Installed** [2]	
	Apply external reference to REFIN		Installed
SJP5	Apply common mode voltage to THS4131	**Installed** [2]	N/A
W1	Set BVDD to +5VD	**Installed** [2]	
	Set BVDD to +3.3VD		Installed
W3	Set signal PD High	Installed	N/A
W4	Set signal FS High	Installed	N/A
W5	Set signal $\overline{\text{CS}}$ LOW	**Installed** [2]	N/A

[1] Reference designator W2 is not used.

[2] Set jumpers as shown.

2.3 5-6K Interface Evaluation Module

Many TI data acquisition evaluation modules (EVM) are built to have a common set of connectors and signals at these connectors. The 5-6K Interface Board allows designers to easily connect these EVMs to the 'C5000 and 'C6000 family of digital signal processor starter kits(DSKs).

The 5-6K Interface Board consists of two serial, two signal conditioning areas, and one set of parallel interface connectors. The ADS8382EVM plugs onto connectors J10 (analog), J16 (serial control) and JP5 (Power). 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.

2.4 Hardware Connections

The hardware connections used for this application report are shown in Figure 1. Normally, the 'C6713 DSP signals CLKX0, FSR0, and DRR0 are connected to ADS8382 signals SCLK, FS, and SDO, respectively. This particular converter has a number of timing restrictions call quiet times(explained later). In order to meet those restrictions with a DSP, the BUSY signal is used as the frame sync to the DSP and the converter. ADS8382 timing specification, tH8, calls for a 15-ns hold time between BUSY falling and frame sync . This is accomplished on the ADS8382EVM by using the BUSY signal buffer (U11) and FS buffer (U6). Each of these buffers will delay the signal about 8ns, giving us a total of 16ns which exceeds the tH8. Since there isn't a jumper to short BUSY to FS on the EVM, a small wire was used to short buffered BUSY signal at connector P2 pin 15 to connector P2 pin 7. This means the DSP has to be programmed for clock-stop mode and the converter for Read Frame controlled by FS. This interface is one way of meeting all the timing restrictions required to achieve optimal performance.

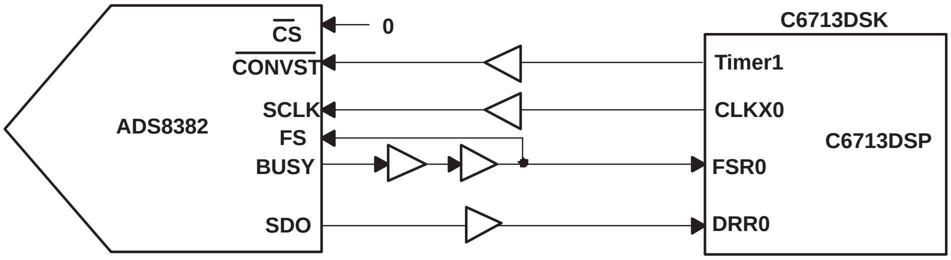


Figure 1. Hardware Connection

3 Software Interface

The objective here 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 with an EDMA channel, one of the timers, and an interrupt service routine. For this discussion, it is assumed that the reader is familiar with the DSP and its peripherals. If not, the reader should study the relevant documentation listed in the References section at the end of this application report.

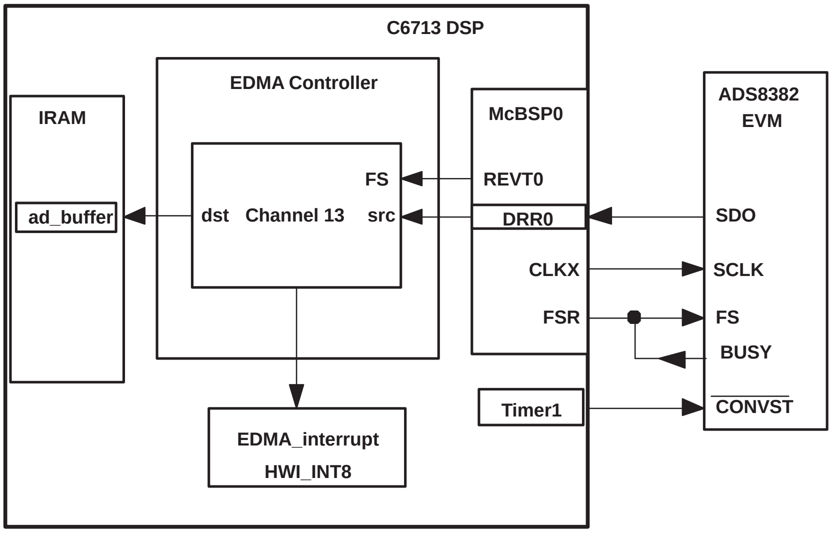


Figure 2. Data Transfer Block Diagram

Figure 2 provides a visual representation of the data flow through the hardware. Timer1 is programmed to generate a pulse at 598 kHz and wired to the convert-start pin of the ADS8382. The sampling frequency can be modified by writing to the Timer period register. McBSP0 is programmed for clock-stop mode, sometimes call SCI. mode. As previously mentioned, the BUSY signal is used to frame each serial transfer. The McBSP0 transmit clock is configured as an output and is wired to SCLK. Serial data output signal from the converter is shorted to data receive register. EDMA channel 13 is synchronized to the McBSP0 receive event. Such that each time a receive event occurs, the EDMA transfers data from the receive data register to a data array called *ad_buffer*. Once all 2048 samples are captured, the EDMA transfers future data points to the location *temp*. Forcing samples 2049+ to temp prevents data in ad_buffer from being corrupted.

The ADS8382 is a high-performance, 18-bit, 600-kHz ADC with fully differential, pseudo-bipolar input. The term *fully differential, pseudo-bipolar* may be new to some readers. This phase refers to the characteristics of the analog input pins. Fully differential means that each input pin must be 180 degrees out of phase. Pseudo-bipolar means the input cannot go below ground reference or zero volts (see Figure 3). The input span(+IN - (-IN)) is from -VREF to +VREF. The maximum differential voltage between +IN and -IN is 2*VREF. The converter outputs code in 2's complement format.

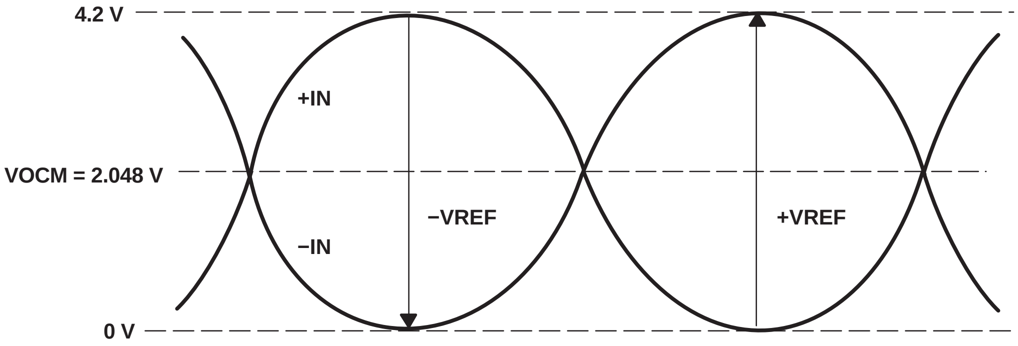


Figure 3. Analog Input Signal Range

Traditionally, SAR-type converters have only two modes of operation. The device is either sampling the input voltage or converting the analog input to its digital representation. The ADS8382 analog-to-digital converter has two additional modes, Wait and NAP. Figure 40 of the ADS8382 data sheet (SLAS416) shows a graphical representation of the states. The device requires a certain combination of signals to enter into sample mode and another to enter into convert mode. If the convert-start signal is HIGH and chip select is LOW, the device enters sample mode. The device has a minimum sample time and a maximum sample time. The minimum sample is 500 ns and maximum of 1 ms. If the device is held in sample mode for more than 1 ms, the conversion result is adversely affected. Therefore, if continuous sampling is not wanted, place the device in either NAP or WAIT mode.

The ADS8382 has a CMOS serial interface that can be clocked at up to 40 MHz. This high rate allows the user to read the data out of the device within 450 ns after conversion is complete. The remaining time can therefore be used to allow the input to settle to half an 18-bit LSB level.

For best performance, this converter requires the user satisfy three quiet times. The quiets times state the digital bus must be inactive 1) 30 ns before the convert-start signal goes low, 2)10 ns after the convert-start signal goes low, and 3) 600 ns before the BUSY signal goes low. The quiet time around convert start going low is to ensure that the ground reference plane is settled and quiet when the device goes to capture or hold the input voltage. Any digital glitch or ground bounce during this critical time will cause the voltage captured on the internal sample and hold circuit to vary. The last 600 ns of the conversion process is also critical because it is then that the device resolves the last few bits. A glitch in the ground plane, or substrate of the device, will show itself as an offset error in the digital word because the minus reference changed during the critical bit decision time. It is therefore very important the timing specifications be met.

The ADS8382 product data sheet (SLAS416) provides detailed explanation of this device or any of the aforementioned items.

3.1 3.1.2 C6713 DSP

The TMS320C6713 DSP and its respective peripherals (i.e., timer1, EMIF, and EDMA) need to be configured properly 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 detail in this application report. For more information, see the downloadable code and references listed in section 5 .

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 and format data.

The register settings for the Multi-Channel Buffered Serial Port (McBSP) follow:

```
MCBSP_Config mcbspCfg0 = { 0x00303800, /* Serial Port Control Reg. (SPCR) */ 0x00020060, /*
Receiver Control Reg. (RCR) */ 0x00010060, /* Transmitter Control Reg. (XCR) */ 0x201F0002, /*
Sample-
Rate Generator Reg. (SRGR) */ 0x00000000, /* Multichannel Control Reg. (MCR) */ 0x00000000, /*
Receiver Channel Enable(RCER) */ 0x00000000, /* Transmitter Channel Enable(XCER) */ 0x0000060D /*
Pin Control Reg. (PCR) */ };
```

The serial port is set up for a 20-bit, clock-stop mode with a delayed clock. CLKX0 is an output pin, and the clock signal was 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 37.5 MHz, because it's the highest frequency allowed that's less than 40MHz. The calculation is as shown below:

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

Register settings for EDMA channel 13 follow:

```
EDMA_Config edmaCfg13 = { 0x40360003, /* 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 */ };
```

The EDMA channel is configured to transfer 2048 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 as shown in the following code listing. After 2048 samples are captured, the EDMA loads these settings, takes any additional data samples, and copies them over to the location *temp*. This is done to ensure that the EDMA does not overwrite the data stored in the array ad_buffer.

```
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 2048 samples have been gathered, the EDMA interrupts the CPU. After some time, the CPU executes function *hwiDMA_isr()*. In this function, the CPU stops the timer, which initiates new conversions. Before resetting the various interrupts and exiting the function, the DSP clears out the last two bits of the 20-bit transfer, thereby leaving only the 18-bit digital representations of the input. The data is MSB-mapped in the array *ad_buffer[]* and *data[]*. This means that the 18-bit data word is mapped MSB to MSB in a 32-bit word.

The following is the register setting for timer1 which sets the sampling rate:

```
TIMER_Config timerCfg1 = { 0x000002D3, /* Control Register (CTL) */ 0x0000005E, /* Period
Register (PRD) */ 0x00000000 /* Counter Register (CNT) */ };
```

Timer1 input clock source is the CPU CLOCK divided by 4.

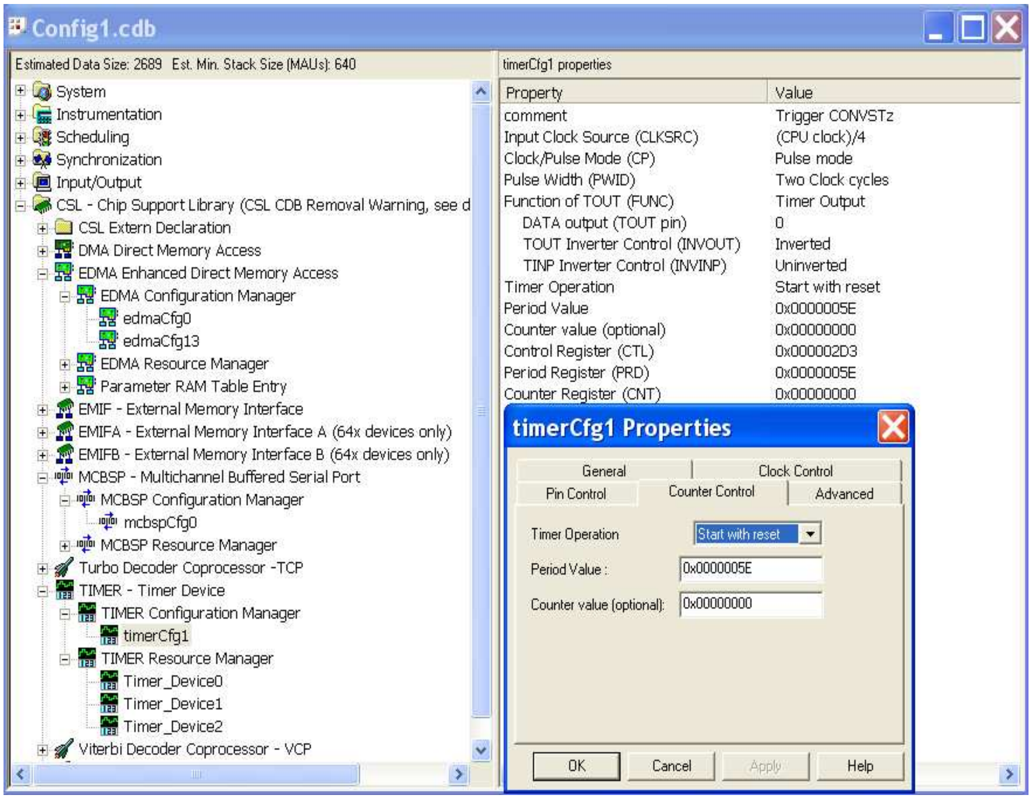
Timer1 Input Clock = 225E6/4 = 56.25MHz

The formula for setting the sampling frequency (Fs) is

Fs = 56.25e6 /(Period Value)
OR
Period Value = 56.25e6/(Fs)

The timer1 output frequency is set to achieve the highest possible sample rate of 598.4 kHz. Timer1 is set to CPU clock divided by 4, pulse mode, Inverted, and two clocks per pulse. The period register is set to 94.

To change the timer frequency, open file *Config1.cdb* in Code Composer Studio™ and expand as shown in Figure 4. Right click timerCfg1, select properties, and select counter control tab.


Config1.cdb
Estimated Data Size: 2689 Est. Min. Stack Size (MAUs): 640
System
Instrumentation
Scheduling
Synchronization
Input/Output
CSL - Chip Support Library (CSL CDB Removal Warning, see d
CSL Extern Declaration
DMA Direct Memory Access
EDMA Enhanced Direct Memory Access
EDMA Configuration Manager
edmaCfg0
edmaCfg13
EDMA Resource Manager
Parameter RAM Table Entry
EMIF - External Memory Interface
EMIFA - External Memory Interface A (64x devices only)
EMIFB - External Memory Interface B (64x devices only)
MCBSP - Multichannel Buffered Serial Port
MCBSP Configuration Manager
mcbspCfg0
MCBSP Resource Manager
Turbo Decoder Coprocessor -TCP
TIMER - Timer Device
TIMER Configuration Manager
timerCfg1
TIMER Resource Manager
Timer_Device0
Timer_Device1
Timer_Device2
Viterbi Decoder Coprocessor - VCP
timerCfg1 properties
timerCfg1 Properties
General
Clock Control
Pin Control
Counter Control
Advanced
OK
Cancel
Apply
Help

Property	Value
comment	Trigger CONVSTz
Input Clock Source (CLKSRC)	(CPU clock)/4
Clock/Pulse Mode (CP)	Pulse mode
Pulse Width (PWID)	Two Clock cycles
Function of TOUT (FUNC)	Timer Output
DATA output (TOUT pin)	0
TOUT Inverter Control (INVOUT)	Inverted
TINP Inverter Control (INVINP)	Uninverted
Timer Operation	Start with reset
Period Value	0x0000005E
Counter value (optional)	0x00000000
Control Register (CTL)	0x000002D3
Period Register (PRD)	0x0000005E
Counter Register (CNT)	0x00000000

Timer Operation: Start with reset

Period Value : 0x0000005E

Counter value (optional): 0x00000000

Figure 4. Config1.cdb

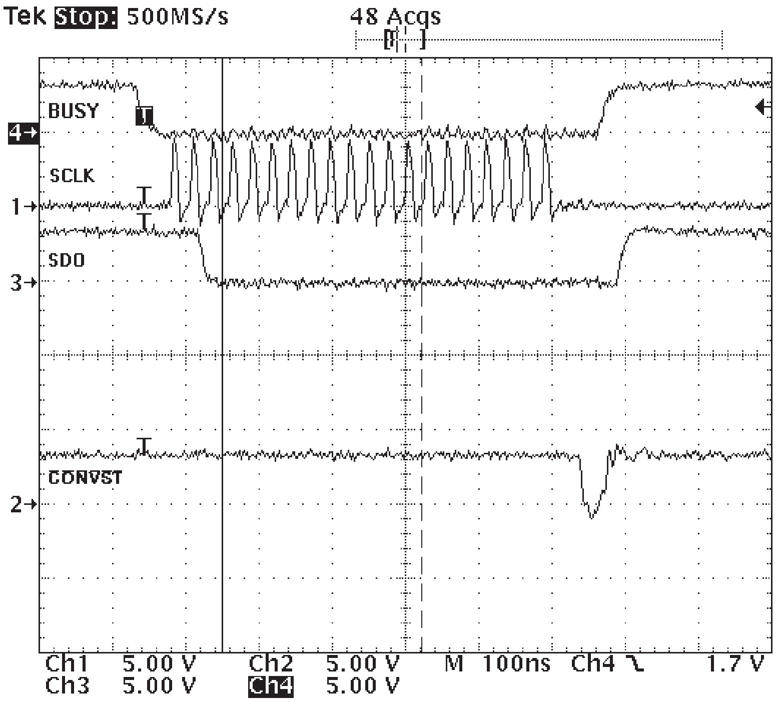


Figure 5. Screen Shot

4 Conclusion

This application report presented one solution to interfacing the ADS8382 converter to the ‘C6713 DSP. The ADS8382EVM plugs onto the 5-6k Interface Board, which in turn plugs onto the ‘C6713 DSK. All the hardware used for this application report can be ordered from Texas Instruments. The software solution involves using DSP/BIOS and configuration tool (i.e., config1.cdb file) to visually set up the EDMA, Timer, and interrupt. The EDMA and Timer1 are used to trigger a conversion cycle at 598 kHz. The C-language code developed with this report is available for download.

5 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. *16-Bit, 600-kHz Fully Differential, Pseudo-Bipolar Input, Micropower Sampling Analog-to-Digital Converter with Serial Interface and Reference* data sheet (SLAS416)
5. *TMS320C6713, TMS320C6713B Floating-Point Digital Signal Processors* data sheet (SPRS186)
6. *ADS8382EVM User's Guide* (SLAU143)
7. *5-6K Interface Board EVM User's Guide* (SLAU104)

Appendix A

Main.C

A.1 Main.C

```
/************************************************************ * FILENAME: main.c * DESCRIPTION:
This program uses McBSP0 of C6713 DSK to read * 2048 samples continuously from the ADS8382 18-
bit, 598KSPS * Analog-to-Digital Converter. The 20-
bit words read during * each data frame is in "ad_buffer". The 18-
bit digital sample * is stored in array Data. * The ADS8382EVM plugs onto the 5-
6k interface card. The 5-
6k * interface card plugs onto the C6713 DSK. * Hardware connections: * 6713DSK (DSP/5-
6K interface card) => ADS8382EVM * CNTLb =>CS# (CS# can be tied LOW permanently) * CLKX(J16.3)=>
SCLK (P2.3) * FSX(J16.7) => BUSY (short BUSY P2.15 to P2.7 on ADS8382EVM.) * DRR(J16.13)=> SDO
(P2.13) * INT6 => BUSY * Timer1 => CONVST# * AUTHOR : DAP Application Group, L. Philipose, Dallas
* CREATED 2005 © BY TEXAS INSTRUMENTS INCORPORATED. * VERSION: 1.0
************************************************************/

/* Include Header File */ #include "Config1cfg.h" #include "dsk6713.h" #include "dc_conf.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_ALIGN(ad_buffer,BLOCK_SZ); int ad_buffer[BLOCK_SZ];
/*raw data from 20-bit read */ #pragma DATA_ALIGN(data,BLOCK_SZ); int data[BLOCK_SZ]; /*18-
bit data from AD with sign extented */

unsigned int temp=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; data[i]= 0x0; }

/* 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 */ EDMA_intEnable(TCCINTNUM6); /*Enable EDMA interrupt */

/*Configure EDMA Channel and McBSP0*/ EDMA_config(hEdmaCha13,&edmaCfg13);
MCBSP_config(hMcbsp0,&mcbspCfg0);

/*Start McBSP0*/ MCBSP_start(hMcbsp0,MCBSP_RCV_START | MCBSP_XMIT_START | MCBSP_SRGR_START|
MCBSP_SRGR_FRAMESYNC, 0); EDMA_clearChannel(hEdmaCha13); TIMER_start(hTimer1); /*Set convert
start to 598kHz */ EDMA_enableChannel(hEdmaCha13); /*Enable EDMA channel 13 -REVT0 */ }
```

Functions.C

B.1 Functions.C

```
/************************************************************ * FILENAME: functions.c *
DESCRIPTION: This program uses McBSP0 of C6713 DSK to read * 2048 samples continuously from the
ADS8382 18-bit, 598KSPS * Analog-to-Digital Converter. The 20-
bit words read during * each data frame is in "ad_buffer". The 18-
bit digital sample * is stored in array Data. * The ADS8382EVM plugs onto the 5-
6k interface card. The 5-
6k * interface card plugs onto the C6713 DSK. * Hardware connections: * 6713DSK (DSP/5-
6K interface card) => ADS8382EVM * CNTLb =>CS# (CS# can be tied LOW permanently) * CLKX(J16.3)=>
SCLK (P2.3) * FSX(J16.7) => BUSY (short BUSY P2.15 to P2.7 on ADS8382EVM.) * DRR(J16.13)=> SDO
(P2.13) * INT6 => BUSY * AUTHOR : DAP Application Group, L. Philipose, Dallas * CREATED 2005 © BY
TEXAS INSTRUMENTS INCORPORATED. * VERSION: 1.
************************************************************/

/* Include Header File */ #include "Config1cfg.h" #include "dsk6713.h" #include "dc_conf.h"

extern int ad_buffer[BLOCK_SZ]; extern int data[BLOCK_SZ];

/************************************************************/ /*hwiDMA_isr(): */ /* Hardware
Interrupt Function disables EDMA channels and */ /* Timer0, Then post software interrupt. */
/************************************************************/

void hwiDMA_isr() { int i=0;

/*To continuously collect samples. Link EDMA config. edmaCfg13 */ /*to hEdmatbl13 instead of
hEdmatbl0. Also comment out */ /*Timer_pause() function call below. */ TIMER_pause(hTimer1);

 for (i=0;i<BLOCK_SZ;i++){ // data[i]=(ad_buffer[i] & 0x000FFFFF)>>2; /*LSB mapped in 20-
bit xfer mode*/ data[i]=(ad_buffer[i] & 0xFFFFFFFC); /*MSB mapped in 20-
bit xfer mode*/ } 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);
// TIMER_resume(hTimer1); }
```

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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