

# Using the ADS8380 With the TMS320C6713 DSP

*Lijoy Philipose*
*Data Acquisition Applications*

## ABSTRACT

This application report presents a solution to interfacing the ADS8380 18-bit, 600-KSPS serial interface converter to the TMS320C6713 DSP. The hardware solution is made up of existing hardware: the ADS8380EVM, '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.

## Contents

1	Introduction .....	1
2	Hardware .....	2
3	Software Interface .....	3
4	Conclusion .....	7
5	References .....	7
Appendix A	Main.C .....	8
Appendix B	Functions.C .....	10

## List of Figures

1	Hardware Connection .....	3
2	Data Transfer Block Diagram .....	3
3	Config1.cdb .....	6
4	Screen Shot .....	6

## List of Tables

1	Jumper Settings for ADS8380EVM .....	2
---	--------------------------------------	---

## 1 Introduction

The ADS8380 converter is one of the first high-resolution, high-speed, serial interface SAR converters to come from Texas Instruments' Burr Brown line of products. It is an analog-to-digital converter with an 18-bit resolution, pseudo-differential input, 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 the TMS320C6713 DSP Starter Kit (DSK), 5-6K Interface Board, and the ADS8380 evaluation module (EVM). The hardware used in this application report is available from Texas Instruments.

### 2.1 TMS320C6713 DSK

The TMS320C6713 ('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 keyword *TMDSDSK6713* on the TI Web site at <http://www.ti.com>.

### 2.2 ADS8380EVM

The ADS8380 evaluation module (EVM), or ADS8380EVM, 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 ADS8380EVM**

REFERENCE DESIGNATOR	DESCRIPTION	JUMPER	
		Pins 1-2	Pins 2-3
SJP1	Buffer onboard reference, REF1004-2.5	Installed <sup>(1)</sup>	
	Buffer user reference voltage applied at P1 pin 20		Not installed
SJP2	Apply external reference directly to SJP4.	Installed <sup>(1)</sup>	
	Apply buffered external reference to SJP4.		Not installed
SJP3	Set negative supply of U2 to ground.	Not installed	
	Select negative supply of U2 to -VCC.		Installed <sup>(1)</sup>
SJP4	Apply internal reference to REFIN.	Installed <sup>(1)</sup>	
	Apply external reference to REFIN.		Not installed
SJP5	Set negative supply of U4 to ground.	Not installed	
	Set negative supply of U4 to -VCC.		Installed <sup>(1)</sup>
SJP6	Apply DC offset voltage to U4.	Installed <sup>(1)</sup>	N/A
SJP7	Short -IN pin to analog ground.	Installed <sup>(1)</sup>	N/A
W1	Set BVDD to +5VD.	Installed <sup>(1)</sup>	
	Set BVDD to +3.3VD.		Not installed
W2	Set signal SB#/2C High	Not installed	N/A
W3	Set signal PD High.	Not installed	N/A
W4	Set signal FS High.	Not installed	N/A
W5	Set signal $\overline{CS}$ LOW.	Not installed	N/A

<sup>(1)</sup> Install these jumpers/shorts.

### 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 connectors, two signal conditioning areas, and one set of parallel interface connectors. The ADS8380EVM 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. The 'C6713 DSP signals CLKX0, FSR0, and DRR0 are connected to ADS8380 signals SCLK, FS, and SDO, respectively. The BUSY signal is sent through two buffers to meet a 15-ns hold time, tH8. This is conveniently available on the ADS8380EVM as the busy signal buffer (U12) and FS buffer (U7). A small wire was used to short the buffered BUSY signal at connector P2 pin 15 to connector P2 pin 7. Chip select and the pin to select between single-ended or 2s complement output is set LOW. This interface is somewhat unusual in that it uses the busy signal as the frame sync to both devices and requires the DSP to be programmed in SPI mode, while the converter is operating in what looks like DSP mode. The ADS8380 interface specifications require this configuration.

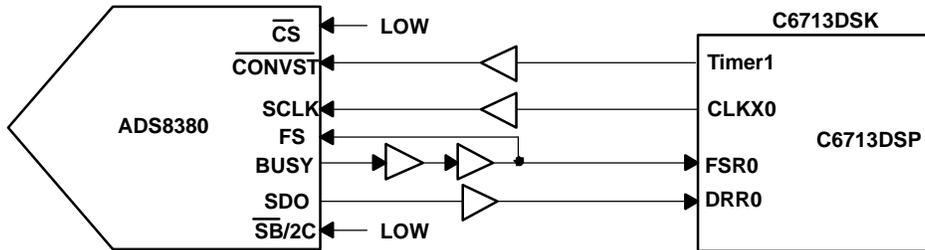


Figure 1. Hardware Connection

## 3 Software Interface

The software 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 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.

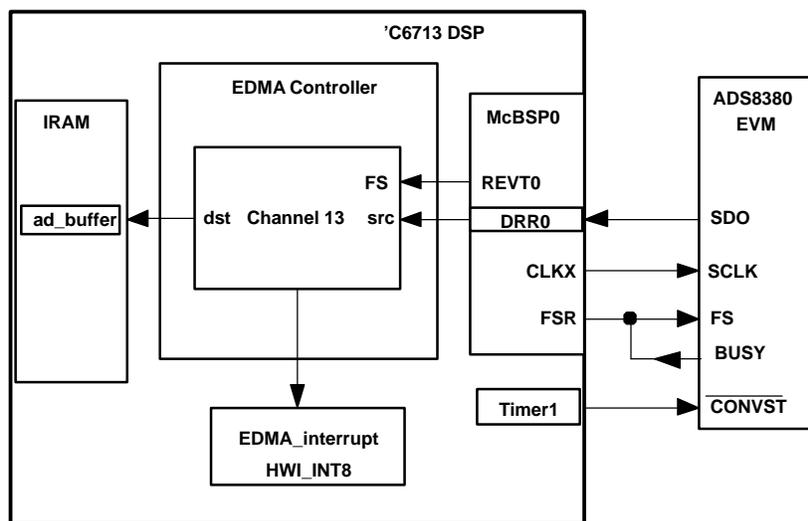


Figure 2. Data Transfer Block Diagram

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 ADS8380. The sampling frequency can be modified by writing to the timer period register. McBSP0 is programmed for clock-stop 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. The serial data output signal from the converter is shorted to the McBSP0 data receive pin. The EDMA channel 13 is synchronized to the McBSP0 receive event. 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*.

The following sections highlight some special features of the converter and the DSP settings that are relevant to this interface.

### 3.1 ADS8380 ADC

The ADS8380 is a high-performance, 18-bit, 600-kHz ADC with single-ended or pseudo-differential input. The term *pseudo-differential* may be new to some readers. This term refers to the characteristics of the analog input pins. Pseudo-differential basically means the difference between +IN and –IN pins will be captured on the sample and hold capacitors. It is pseudo because the –IN pin can only go plus or minus 200 mV around ground.

The output code format can be either 2s complement or straight binary. In this application report, the converter was set for straight binary output.

It has a CMOS serial interface that can be clocked at up to 40 MHz. This allows the user to read the data out of the part within 450 ns.

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 ADS8380 analog-to-digital converter has two additional modes, Wait and NAP. Figure 40 of the ADS8380 data sheet ([SLAS387](#)) 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.

For best performance, this converter requires the user to satisfy three quiet times. The digital bus must be inactive 30 ns before the convert-start signal goes low, 10 ns after the convert-start signal goes low, and 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. The last 600 ns of the conversion process is also important 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.

The ADS8380 product data sheet ([SLAS387](#)) provides detailed explanation of this device or any of the aforementioned items.

### 3.2 '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 in the sample rate generator register to 37.5 MHz. The calculation is as follows:

$$\text{SCLK} = (\text{CPU CLOCK} / 2) / \text{CLKDIV} = (225 \text{ MHz} / 2) / 3 = 37.5 \text{ MHz}.$$

The 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[]*.

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) */
};

```

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

$$\text{Timer1 Input Clock} = 225\text{E}6 / 4 = 56.25\text{MHz}$$

The formula for setting the sampling frequency (*F<sub>s</sub>*) is

$$F_s = 56.25\text{e}6 / (\text{Period Value})$$

or

$$\text{Period Value} = 56.25\text{e}6 / (F_s)$$

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 3](#). Right-click TimerCfg1, select properties, and select counter control tab.

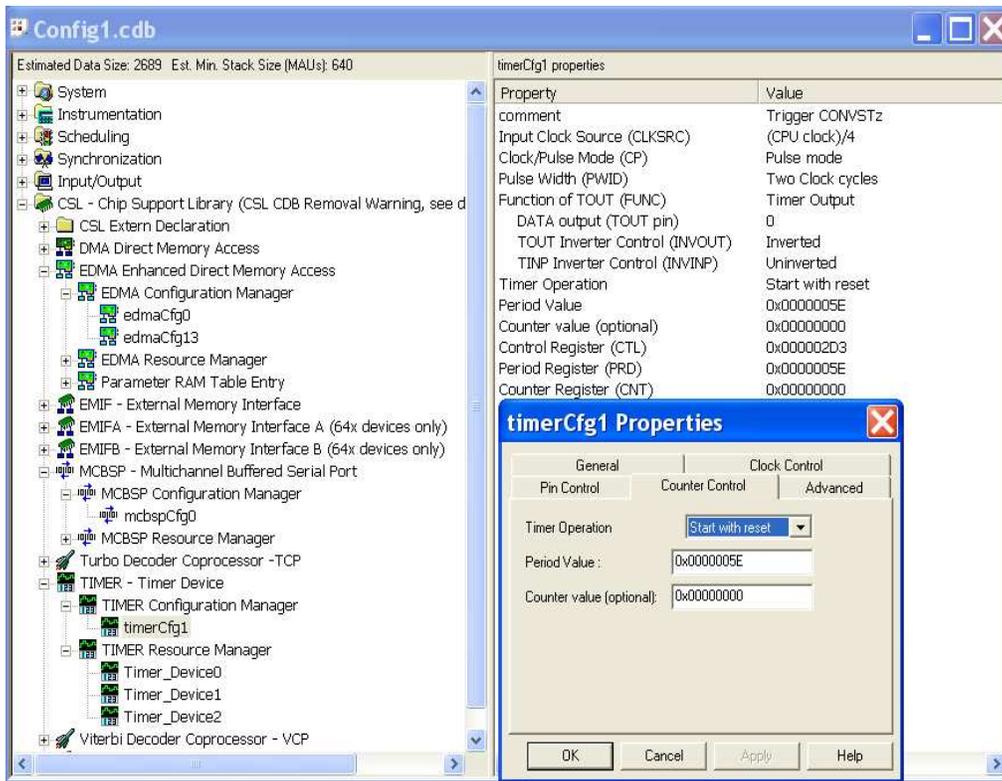


Figure 3. Config1.cdb

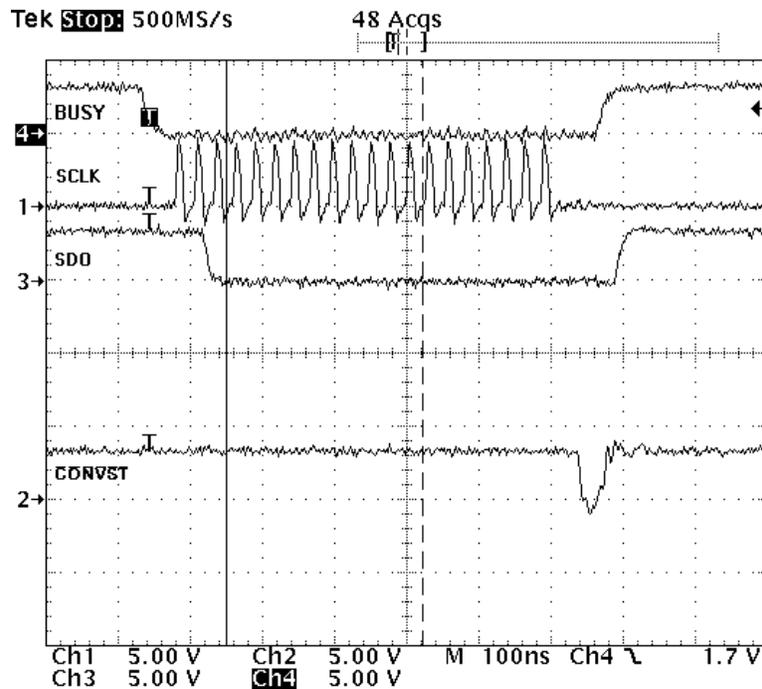


Figure 4. Screen Shot

#### 4 Conclusion

This application report presented one solution to interfacing the ADS8380 converter to the 'C6713 DSP. The ADS8380EVM 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. *18-Bit, 600-kHz, Pseudo-Differential Input, Micropower Sampling Analog-to-Digital Converter with Serial Interface and Reference* data sheet ([SLAS387](#))
5. *TMS320C6713, TMS320C6713B Floating-Point Digital Signal Processors* data sheet ([SPRS186](#))
6. *ADS8380EVM User's Guide* ([SLAU146](#))
7. *5-6K Interface Board EVM User's Guide* ([SLAU104](#))

## Appendix A Main.C

```

/*****
* FILENAME: main.c
* DESCRIPTION: This program uses McBSP0 of C6713 DSK to read
* 2048 samples continuously from the ADS8380 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 ADS8380EVM 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) => ADS8380EVM
* 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 ADS8380EVM.)
* 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 "Configlcfg.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 extended */

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(hEdmaChal3,&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 */
}
```

## Appendix B Functions.C

```

/*****
* FILENAME: functions.c
* DESCRIPTION: This program uses McBSP0 of C6713 DSK to read
* 2048 samples continuously from the ADS8380 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 ADS8380EVM 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) => ADS8380EVM
* 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 ADS8380EVM.)
* 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 "Configlcfg.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. edmaCf913 */
    /*to hEdmatb113 instead of hEdmatb10. Also comment out
    */
    /*Timer_pause() function call below.
    */
    TIMER_pause(htimer1);

    for (i=0;i<BLOCK_SZ;i++){
        // data[i]=(ad_buffer[i] & 0x00FFFFFF)>>2; /*LSB mapped in 20-bit xfer mode*/
        data[i]=(ad_buffer[i] & 0xFFFFF0); /*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

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:

<b>Products</b>		<b>Applications</b>	
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>	Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>	Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>	Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>	Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>	Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>	Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>	Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
		Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
		Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
		Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

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