FAE Summit 2004 - Interfacing the ADS8361 to the MSP430F449 Low Power Micro Controller

Tom Hendrick *High Performance Analog - Data Acquisition Products Group*

LAB OBJECTIVES

This lab exercise presents a method for interfacing the ADS8361 16-bit SAR analog- to-digital converter to the MSP430F449 using SPI0 and SPI1 to achieve 2x2 channel, simultaneous operation of the ADS8361. You will review the initialization of the 32kHz watch crystal and how to derive a conversion clock from the DCO; see how to initialize the SPI ports (Master and Slave) and review the general operation of the IAR Embedded Workbench tool.

REQUIRED MATERIAL

HPA449 Development Board from SoftBaugh

ADS8361EVM with “HPA449” Header

Universal Front End (UFE) Board

Stimulus Board

DAP Power Supply

25 Pin “D” style male-female cable

FET Programmer

Optional – 9pin “D” style male-female cable for Serial Programming with “*Simple Flash*”

MSP430 Lab Introduction

The ADS8361 is a 2+2 channel, 16–bit upgrade for the ADS7861 (12-bit) 2+2 channel analog-to-digital converters. The converter is able to gluelessly interface to the MSP430F449 micro controller. This lab uses the HPA449 Development Board and an ADS8361EVM.

1 Hardware

The HPA449 Development Board is a convenient way to experiment with interfacing the MSP430F449 to the ADS8361. The ADS8361EVM plugs directly onto the HPA449 Board, which allows conversion results to be displayed on a custom LCD. Additional hardware is included for the purposes of this lab that eliminate the need to carry lab power supplies and signal generators.

1.1 HPA449 Development Board

The HPA449 Development Board was born through a joint effort between the DAP Applications Group and SoftBaugh, Inc. Boards are available for purchase directly through the manufacturer at www.softbaugh.com. This lab is specific to the MSP430F449 but can be ported to any MSP430 device with two serial ports.

1.2 UFE and Stimulus Board

The Universal Front End and Stimulus Boards were designed to accompany the Data Converter Seminar. These boards work together to provided buffered signals to the data converter and offer an effective means to demonstrate hardware without having to carry signal generators and excess lab equipment. These boards are not generally available to customers. To display voltage values on the LCD correctly, make sure the Stimulus Board is set for 1.2 – 120 Hz operation and the R1 potentiometer is set fully clockwise.

1.3 DAP PS Board

The DAP PS REV B board is a multi-output power supply that will operate from standard 5VDC wall transformers such as those supplied with the lab hardware. It is important to mention that the voltage applied to the DAP PS board goes directly to the 5V electronics on the HPA449. DO NOT use a 9V supply with this board! There is no over voltage protection and permanent damage may be inflicted on the hardware.

The DAP Power Supply provides +/-15 V using a PT5062A, -5V using a UCC284 and +3.3/+1.2V using a TPPM0110. The 5V applied to the board is used as the 5V system power as mentioned above.

1.4 ADS8361EVM

The ADS8361 is a member of the Motor Control Products family of serial ADCs available from Texas Instruments. The EVM provides a platform to demonstrate the functionality of the ADS8361 ADC with various Texas Instruments DSP's and Micro Controllers, while allowing easy access to all analog and digital signals for customized end-user applications. For more information on the EVM, search for document number SLAU094 from the main page of the Texas Instruments website at http://www.ti.com.

The ADS8361 is a simultaneous sampling converter designed for use in motor control applications, but is often used in non-motor related designs. Any application requiring a fast multi-channel converter can take advantage of the features of this device.

2 Hardware Interface

The hardware interface is seamless between the HPA449 and the ADS8361 EVM, meaning there is no "glue logic" required between the Micro and the ADC. The HPA449 Development Board provides direct access to the 449's SPI0/1 ports. In order to use both SPI ports, a simple wire-wrap header is provided.

The hardware connections are shown in Figure 3 via the HPA449 Board. The /CS and RD(+ CONVST) pins from the ADS8361 are connected to GPIO pins of the Micro. Chip select is also fed back to the STE1 pin to enable slave transfers. The SPI Clock and data from the OUTb pin are returned to SPI1 via the extension header.

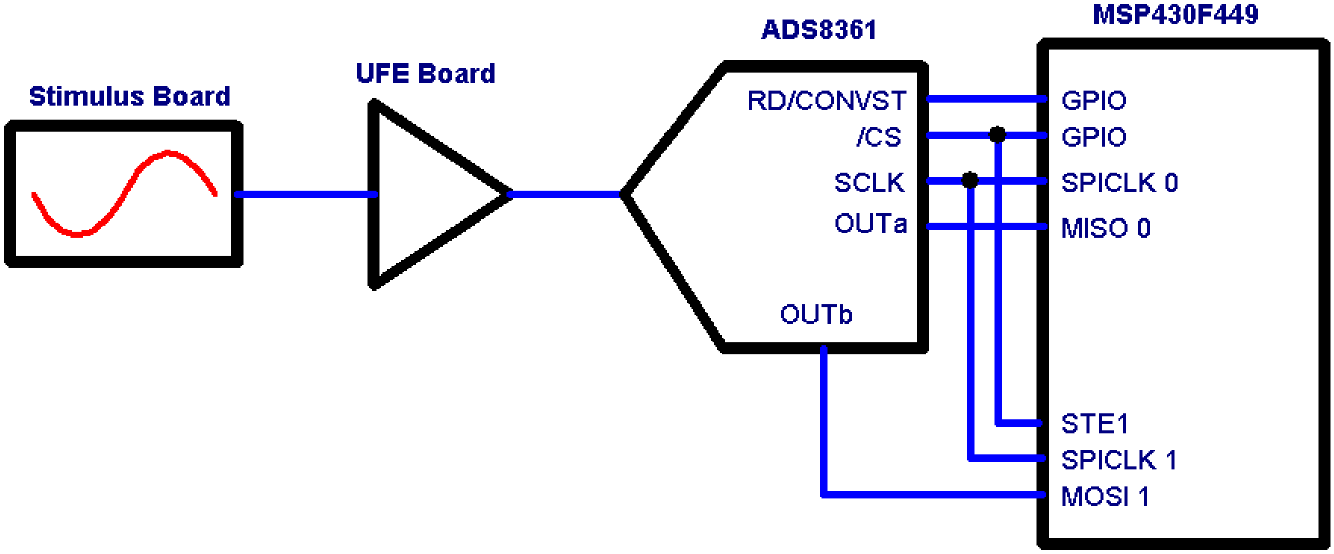

Figure 1. MSP430 Interface Block Diagram

3 Software Interface

All of the software was written and compiled using IAR Embedded Workbench Kickstart for the MSP430. The most involved portion of writing the code for this simple interface is programming the conversion clock and setting up the SPI ports. Getting the data properly formatted can also be a challenge with the MSP430 since the SPI ports only operate in 8bit mode. In this example, 3 data variables are defined which receive the conversion results byte by byte. They are then concatenated and shifted before being stored into a data array – see the function at the end of void adc_convert (void) for more details.

3.1 SPI Port and Clock Settings

The SPI ports are set in the function “void setupSPI (void)”. SPI 0 is set to be the master, while SPI 1 is set for slave operation. SPISTE1 must be low in order for the slave to receive data and since both serial outputs of the ADS8361 work together, the CS line can be used for this function through general purpose IO.

This example uses the DCO in the F449 chip to generate a four-megahertz clock from the 32kHz watch crystal X1. The code uses a conditional compile statement “#define Resonator (0)” to determine how the F449 device should configure its main clock. Note the function “void setupClock (int)” passes “Resonator” as the argument. If a high-speed resonator or crystal is used on the MSP430, the user should set up a “wait” loop that tests the oscillator fault flag state. The MSP430 can determine if the clock has stabilized before continuing its initialization. This is particularly useful when waking up from one of the extended sleep modes where the resonator may have been shut down to conserve battery power.

3.2 Software Flow

The software presented in this lab reads 100 samples at approximately 17 kHz continuously. Throughput is limited by the 4MHz clock applied to the SPI ports and could be increased by a factor of two using an 8MHz resonator and configuring both SPI ports as “slave” devices. In slave mode, there is no restriction on the input clock speed. In master mode, the clock speed is controlled by the baud rate settings, which have a divide-by-two maximum speed (4MHz max with 8MHz max main clock).

After the samples are received, they are stored and sorted into four data arrays: CHA0_buffer through CHB1_buffer. Channel 0/1 data is determined by the presence of the address bit in the serial stream received by the SPI ports. Once the buffers are full, the values received are displayed on the LCD based on the level of the interrupt received at port2. This is controlled by push-button switch SW2, which counts 0-3 and displays channel A0 through B1 data via function “sortFloat(display_volts(CHB1_buffer, Samples))” found at the end of the convert data sub routine. The index pointers are then reset and the process starts over again in a never-ending loop.

4 Running the Example Program

4.1 Start Kick Start for the MSP430

Launch the IAR Systems tool by double clicking on the Embedded Workbench shortcut from the PC desktop.

4.2 Open the Dual_Ads8361 Workspace

From the "File – Open Workspace" menu, choose ADS78_8361.eww. This file is located in the C:\Program Files\IAR Systems\EW32\430\FET_examples\HPA449\FAE_ADS8361 folder. The tool is configured to bring up the last active workspace, so it should not be necessary to "find" the workspace.

4.3 Expand the Project

Expand the project in the project manager pane and review the various entries. Review the "**ADS78_8361.C**" file for the general structure on setting up the MSP430. Find the function "*void setupports (void)*" and see how the port 3 and 4 function select bits must be set for "SPI Mode" before this software will work correctly.

4.4 Build/Load and Run the Project

From the project menu, choose "**ReBuild All**". Alternatively, you can use the "ReBuild All" Icon from the toolbar. Start the debugger by choosing "**Debug**" from the project menu or click on the Debug Icon on the toolbar. Note – If the project does not load onto the HPA449 Target board, press the reset button (S3) and try again.

4.5 View CHA0 Data in the LCD

After the file is loaded, the LCD screen should be blank. From the debug menu, choose "**GO**" to start running the code. You could also use the "*Go*" Icon on the toolbar. The RTD output, CHA0 data, will be displayed on the LCD. Squeeze the RTD bead between your thumb and index finger to observe a change in the display.

4.6 Force an Interrupt – Change the Display

Force an interrupt to the MSP430 by pressing SW2. The display should now show channel A1 data. This is the sine/triangle mix that we saw in the DSP – the display should swing between +2.499 and -2.500. Pressing the button again changes the display to channel B0, which is the 4-20mA transmitter. The display should show approximately 0-2V depending on the position of the potentiometer (R15) located in the lower right corner of the Stimulus Board. The last channel to be displayed (press SW2 again) will be the square wave/0-5V combination. Changing the potentiometer (R2, upper right corner) will force the display to read between 0 and +2.499 at one extreme to 0 and -2.500 at the other. Try to center the display at +/-2.00 Volts.

4.7 Using "Simple Flash" to program the HPA449

To use "Simple Flash" to program the HPA449, stop the debugger and highlight the "***ADS78_8361 – Debug***" line item in the IAR workspace manager. Right click and choose "*Options…*". Under the *XLINK* category, change the output format to "other" and choose the "intel-standard" option. Click the OK button then go ahead and "Re-Build All" from the project menu. This action will create a hex file with the extension .a43.

Start Simple flash and browse to the folder where the .a43 file was placed in the last step. Under the conditions of this lab, that would be C:\Program Files\IAR Systems\EW32\430\FET_examples\HPA449\FAE_ADS8361\Debug\Exe. Double click on ADS78_8361.a43 to load the file, and then choose the "Flash" option. The key to successful flashing with this method is to ensure the shunt jumpers located on J4 are all installed at the lower two pins.

The Simple Flash tool can be downloaded by customers at the following URL:

http://www.mrdataesq.net/proj/simpleflash/

This concludes the "Mini-EVM" Hands On session.