Application Report
SNAA211–June 2013

MSP430 Interface to ADS1293 Code Library

Vishy Viswanathan

ABSTRACT

The MSP430 is an ideal microcontroller solution for low-cost, low-power precision sensor applications because it consumes very little power. The ADS1293 is a fully integrated signal chain for portable, low-power medical electrocardiogram (ECG), sports, and fitness applications. This library provides functions to facilitate the interfacing of any MSP430 device to the ADS1293. Any device within the MSP430 family can be used with this library, made possible by hardware abstraction. Similarly, any SPI-capable interface module within the MSP430 family is supported by the library. This allows the designer maximum flexibility in choosing the best MSP430 device for the application. This document provides descriptive information and instructions for using the library either for demonstration purposes or implementation into a project. This is the recommended starting point for developing software for the ADS1293 and MSP430 combination. The software examples have been developed for the ADS1293EVM but can easily be ported to another hardware platform.

Source code discussed in this application report can be downloaded from the ADS1293 product folder.

1 Introduction

This application note describes different ways to interface and use the TI ADS1293 devices with an MSP430. The accompanying software contains a function library allowing quick prototyping of ADS1293 setup and control. The software provided in this library is a starting point for developers wanting to get the most out of the MSP430 and the ADS1293 devices.

The ADS1293 incorporates all features commonly required in portable, low-power medical ECG, sports, and fitness applications. It provides a complete signal path solution for bio-potential measurements. The ADS1293 is equipped with a slave SPI port, through which it can communicate with an MSP430 and is an optimal match for the MSP430 ultra low power microcontrollers. The MSP430 is a great fit for applications where power conservation is a priority. The many power-saving mechanisms designed into the MSP430 make it ideal for such applications.

2 Purpose and Scope

To aid in interfacing these devices, TI has produced a code library that significantly reduces the need to write low-level interface functions. It provides a boost in the development of an MSP430/ADS1293-based product, saving time and allowing quick progression to the application-specific aspects of the project. This library is designed to be used with any MSP430 device. Since a SPI master can be implemented using one of many peripherals within the MSP430 family, and since the peripherals available may differ by device and application, library calls are provided for each of these interfaces. The chosen interface is selected by assigning a value to a system variable, which causes the compiler to conditionally include the appropriate function calls. As such, application code utilizing the library remains portable between various MSP430 devices, with minimal modification required.

Several complete example application projects are provided with the library. The purpose of these projects is to demonstrate use of the library. It is not intended as a comprehensive guide to using the ADS1293, and it does not make use of all the features of these devices. It does, however, use all the register access functions provided by the library.

3 File Organization

The library has been implemented with modular hardware abstraction. There is a header file specific to each of the hardware components (ADS1293, MSP430, and the board). The hardware definition header files are shown in Table 1. Table 2 shows the library code files and its header, and Table 3 shows the demonstration applications that accompanies the library.

Table 1. Hardware Definition Files

Filename	Description
TI_ADS1293.h	Definitions specific to the ADS1293 device, including register locations, bit definitions, and commonly-used masks for use with these registers.
TI_MSP430.h	Definitions specific to the MSP430 device; primarily, the pins used in the SPI interface. Definitions for USART0/1, USCI_A0/1/2/3, USCI_B0/1/2/3, USI and bit banging are included. Also, labels are defined for use with the system variable TI_ADS1293_SER_INTF. This label selects the modules to be used for accessing the ADS1293 SPI interface.
TI_MSP430_hardware_board.h	Definitions specific to the board being used; that is, the connections between the MSP430 and ADS1293EVM, such as the chip select, and LED pins. SPI connections to the ADS1293 are not defined here because they are defined inherently within TI_MSP430.h. The system variable TI_ADS1293_SER_INTF is defined in this file.

Table 2. Library Code

Filename	Description
TI_MSP430_spi_USCIA1_5xx.c	Function for accessing the ADS1293 registers via SPI_USCIA1 module from the MSP430 5xx family
TI_MSP430_other_spi_modules\ TI_MSP430_spi_USCIA0_5xx.c TI_MSP430_spi_USCIA2_5xx.c TI_MSP430_spi_USCIA3_5xx.c TI_MSP430_spi_USCIB0_5xx.c TI_MSP430_spi_USCIB1_5xx.c TI_MSP430_spi_USCIB2_5xx.c TI_MSP430_spi_USCIB3_5xx.c TI_MSP430_spi_USCIA0.c TI_MSP430_spi_USCIA1.c TI_MSP430_spi_USCIB0.c TI_MSP430_spi_USCIB1.c TI_MSP430_spi_USART0.c TI_MSP430_spi_USART1.c TI_MSP430_spi_USI.c TI_MSP430_spi_BITBANG.c TI_MSP430_spi_eUSCIA0_FR57xx.c TI_MSP430_spi_eUSCIA1_FR57xx.c TI_MSP430_spi_eUSCIB0_FR57xx.c	Functions for accessing the ADS1293 registers via other MSP430 SPI modules like SPI_USART0, SPI_USART1, SPI_USCIA0, etc.
TI_MSP430_spi.h	Function declarations for TI_MSP430_spi_*.c

Table 3. Demo Applications Included with the Library

Demo Application	Filename	Description
Application1: Read/Write ADS1293 Registers	demo-app01\main.c	Application code with functions to demonstrate read/write of the ADS1293 register
	demo-app01\ TI_ADS1293_register_settings.h(1)	Application specific initialization values for the ADS1293 registers
Application2: Auto Increment Read/Write ADS1293 Registers	demo-app02\main.c	Application code with functions to demonstrate auto increment read and writes of ADS1293 Registers.
	demo-app02\ TI_ADS1293_register_settings.h(1)	Application specific initialization values for the ADS1293 registers
Application3: Stream Read ADC Data with interrupt	demo-app03\main.c	Application code with functions to stream read ADC Data with interrupt
	demo-app03\ TI_ADS1293_register_settings.h(1)	Application specific initialization values for the ADS1293 registers

(1) The register settings values for TI_ADS1293_register_settings.h can easily be obtained from the "Register configuration file" saved from Medical AFE Software [2]. Demo Application code reads the values stored in register settings file to initialize the ADS1293 device register.

Figure 1 shows a stack diagram of the library. Note that one of the files displayed in the stack is the standard definition file for the specific MSP430 device being used. This file is included with the development environment being used to create the MSP430 software.

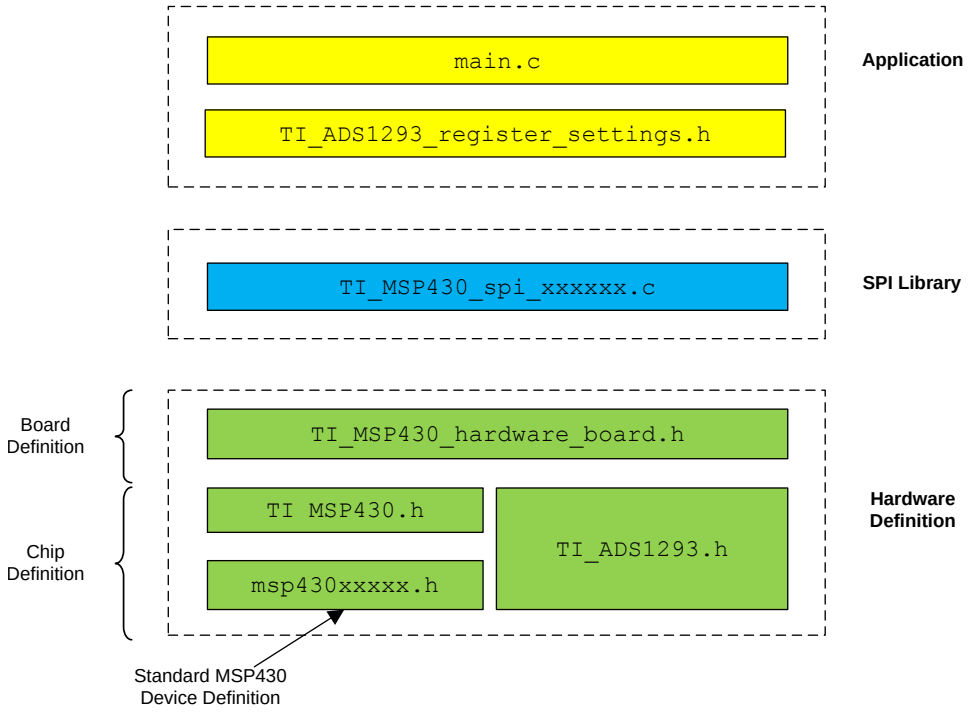

Figure 1. Code Library Stack

4 Functions

Table 4 shows the SPI register-access functions provided in the library, with a brief description.

Table 4. Register Access & Control Functions Provided by the Library

Function Name	Description
void TI_ADS1293_SPISetup (void)	Configures the SPI port assigned by the TI_ADS1293_SER_INTF system variable. Must be called before calling any of the other functions.
void TI_ADS1293_SPIWriteReg (uint8_t addr, uint8_t value)	Writes "value" to the ADS1293 configuration register at address “addr”.
uint8_t TI_ADS1293_SPIReadReg (uint8_t addr)	Reads a single register at address “addr” and returns the 8-bit value read.
void TI_ADS1293_SPIAutoIncWriteReg(uint8_t addr, uint8_t *buffer, uint8_t count)	Writes values to multiple configuration registers, the first register being at address “addr”. First data byte is at “buffer”, and both addr and buffer are incremented sequentially (within the ADS1293 and MSP430 respectively) until “count” writes have been performed.
void TI_ADS1293_SPIAutoIncReadReg(uint8_t addr, unit8_t *buffer, uint8_t count)	Reads multiple configuration registers, the first register being at address “addr”. Values read are deposited sequentially starting at address “buffer”, until “count” registers have been read.
void TI_ADS1293_SPIStreamReadReg(uint8_t *buffer, uint8_t count)	Special read function for reading status, pace, and ecg data registers of selected channels. Channels to be read must be selected in CH_CNFG before calling this function. Data Loop Register read is extended “count+1” times where “count” is number of source bytes enabled in CH_CNFG. Data read are deposited sequentially starting at address “buffer” until “count” bytes have been read.

A version of these functions is provided for all the MSP430 peripherals that are capable of communicating using the SPI protocol. These peripherals are:

- USART0 for 1xx, 2xx, and 4xx families
- USART1 for 1xx, 2xx, and 4xx families
- USCI_A0 for 5xx and 6xx families
- USCI_A1 for 5xx and 6xx families
- USCI_A2 for 5xx and 6xx families
- USCI_A3 for 5xx and 6xx families
- USCI_B0 for 5xx and 6xx families
- USCI_B1 for 5xx and 6xx families
- USCI_B2 for 5xx and 6xx families
- USCI_B3 for 5xx and 6xx families
- USCI_A0 for 2xx and 4xx families
- USCI_A1 for 2xx and 4xx families
- USCI_B0 for 2xx and 4xx families
- USCI_B1 for 2xx and 4xx families
- USI for G2xx value series family
- Bitbang using GPIO pins
- eUSCIA0 for FRAM 57xx family
- eUSCIA1 for FRAM 57xx family
- eUSCIB0 for FRAM 57xx family

5 Using the Software

5.1 Prerequisites

To successfully compile, download and run the software described in this document, the following material is needed:

- ADS1293 Evaluation Board ADS1293EVM with onboard MSP430F5529 MCU
- MSP430 USB Debugging Interface MSP430-FET430UIF
- IAR Embedded Workbench or TI Code Composer Studio for MSP430

The software can be adapted to run on other MSP430 hardware boards as well. See Section 5.3 for instructions.

A free, code size limited, but fully functional edition of IAR Embedded Workbench (IAR Kickstart) is available from the IAR Systems website (www.iar.com) or from the TI MSP430 software tools page.

As an alternative to IAR Embedded Workbench, it would be possible to use Code Composer. A trial edition of the Code Composer is available from the TI MSP430 software tools page.

5.2 Getting Started

Follow these simple steps to get your application up and running:

1. Install IAR Workbench
2. Download the source code for this application note and unzip the files to your working directory.
3. Open IAR Embedded Workbench and create a new project:
   (a) Select Project -> Create new project…
   (b) Select tool chain MSP430
   (c) Base the project on the empty project template.
   (d) Save (you will also be asked to save the current workspace).
4. Add to the project the following C files from the software that you downloaded and unzipped in step 2:
   (a) All C files from the code\library folder
   (b) All C files from the code\library\TI_MSP430_other_spi_modules folder
   (c) All C files from the code\demo-application-examples\demo-app01 folder
5. Open the “options…” dialog for the new project by right clicking the project name in the workspace window. A window should appear.
   (a) Under “General options”, select the MSP device. For the ADS1293EVM target board, use MSP430F5529.
   (b) Under “C/C++ compiler”, click on the preprocessor pane.
      (i) The “Ignore standard include directories” tick box should not be ticked.
      (ii) In the “Additional include directories”, add include paths telling the compiler where to find the header files included by the C files. You should add the $PROJ_DIR$\code\include folder.
   (c) Under “Debugger”, select “FET Debugger” from the “Driver” drop down list.
   (d) Under “FET Debugger”, in the “Connection” section, choose the connection type of the FET tool (e.g. Texas Instruments USB-IF). Leave the rest of the settings as is.
6. Click “OK” to close the options window.
7. Select “Project -> Rebuild All”. There should be no errors or warnings when IAR rebuilds the executables (if not done already, you will also be asked to save the current workspace).
8. The configuration of the hardware definition files in the library as distributed by TI is for an ADS1293EVM with MSP430F5529 MCU. The system variable TI_ADS1293_SER_INTF defined within TI_hardware_board.h identifies USCIA1 as the connected SPI port to control the ADS1293. Peripheral pinouts can change slightly between the individual MSP430 devices and families. For this reason TI_MSP430.h identifies the pins which correspond to a peripheral for any given device.
9. The ADS1293 Evaluation Board can be seen in Figure 2.

10. Attach the MSP430 FET to your PC. If you are running Windows and using the USB FET tool for the first time, you will asked to install some drivers for the tool. For Windows they are located in $IAR_INSTALL_DIR$\430\drivers\TIUSBFET
11. Attach the MSP430 FET to the ADS1293EVM using the JTAG connector.
12. Select Project -> Debug. IAR will now establish a connection with the target MCU, download the application and program the MSP430. The debugger will be started, halting the target at main().
13. Demo_app01 is a simple example that demonstrates the SPI calls to write and successfully read back a ADS1293 register.
14. Steps 3 to 13 can be followed to exercise other demo applications included with the library as well.

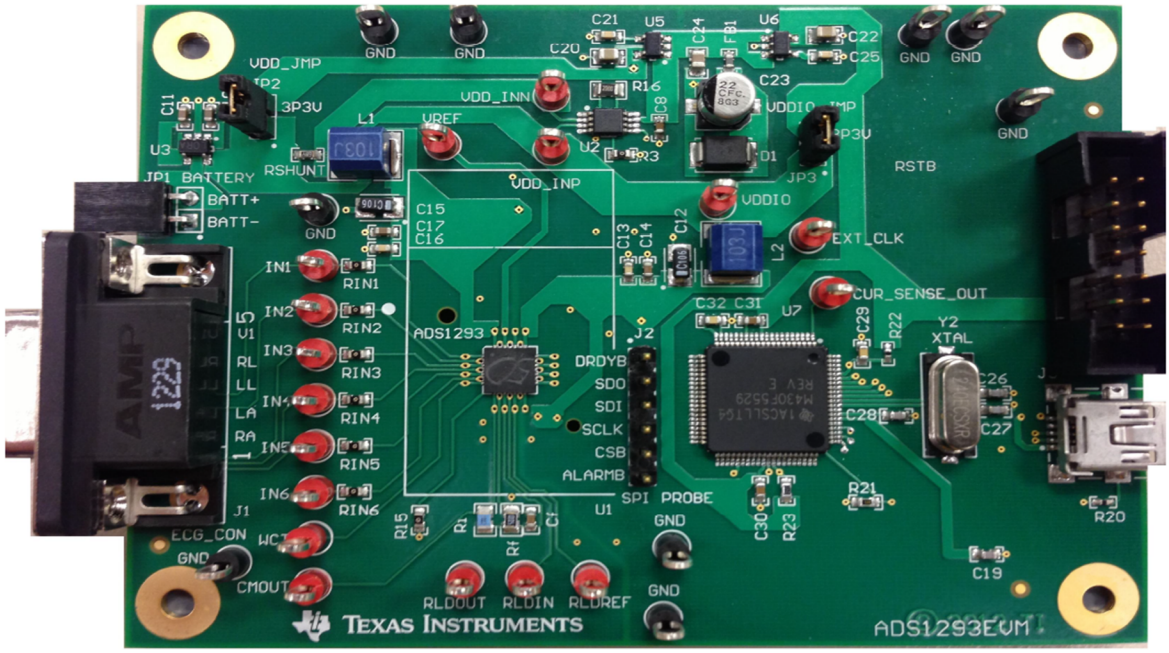

Figure 2. ADS1293EVM

5.3 Adapting the Hardware

The procedure for adapting this code to other hardware is as follows:

- Edit the pin assignments within TI_MSP430.h for the interface modules being used. It is not necessary to modify the pins for the interfaces not selected for use with the SPI bus, as they will not be referenced by the library. The labels being referenced in the #define assignments will be drawn from the standard definition file (msp430.h) listed at the top of TI_MSP430.h.
- Edit the pin assignments in TI_MSP430_hardware_board.h, taking into account all the necessary connections on the board being used. The assigned labels are drawn from the standard definition file (msp430.h) listed at the top of TI_MSP430.h.
- Assign the proper values to TI_ADS1293_SER_INTF in TI_MSP430_hardware_board.h. The labels available for assignment can be found at the bottom of TI_MSP430.h.
- Set up appropriately the function to configure the system clock source and clock rate. This will depend on your hardware and the particular MSP430 MCU in use.
- Make sure the physical hardware connections between the MSP430 and the ADS1293 are modified according to the pin assignments above.

After making these changes, rebuild the project and download the code image. The application should function as described earlier.

5.4 Using the Library with an Application

The same procedure as described in the section above should be applied in order to adapt the library to the new hardware.

The function TI_ADS1293_SPISetup() should always be called after a POR event within the MSP430. After this the access of registers is straightforward.

6 References

1. ADS1293: Low Power, 3-Channel, 24-Bit Analog Front End for Biopotential Measurements, Data Sheet SNAS602
2. ADS1293 Medical AFE Software, SNAC050
3. ADS1293EVM User's Guide, SNAU138
4. MSP430F551x/MSP430F552x Mixed Signal Microcontroller Data Sheet SLAS590
5. MSP430x5xx/MSP430x6xx Family User's Guide SLAU208
6. MSP430 USB Debugging Interface MSP-FET430UIF, SLAU278N

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions of TI 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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