Interfacing the ADS1100 to the MSP430F413

Andreas Dannenberg *MSP430*

ABSTRACT

This application report shows how the ADS1100 16-bit sigma-delta differential A/D converter can be interfaced with the MSP430F413 microcontroller. As an application example, the output of a resistive full-bridge strain gauge sensor is measured and displayed on an LCD.

Hardware Description

The MSP430F413 device used in this application is a low-cost MCU with integrated LCD driver [3]. It reads the external A/D converter and shows the computed measurement result on an LCD display. The ADS1100 has an I^2C interface for connecting to a master device. As the used MSP430F413 controller does not have a hardware I^2C communication module, port pins P2.0 and P2.1 have been used to implement an I^2C interface. The selection of these port pins is arbitrary, and can be changed by modifying the I2C_Master.h header file. 10-kΩ pullup resistors are used on the I^2C bus lines. Figure 1 shows the application schematic.

The bridge sensor is powered via port pins. This reduces power consumption by switching off the sensor excitation voltage when measurements are not performed. Three port pins each are connected in parallel for exciting the sensor.

A 3.5-digit static LCD from Varitronix displays measurement results and connects directly to the MSP430. The LCD manufacturer part number is VI-302. Pushbuttons PB1 and PB2 are connected to port pins P1.6 and P1.7 which are pulled up by 100-kΩ resistors.

Software Description

Overview

The demonstration software associated with this application report is provided in both C and assembly language. Both programs are functionally identical and make use of a common I^2C master communication library that is written in assembly language. The linked assembly-only project is about 20% smaller in code size compared with the mixed C/assembly project.

Table 1. File Overview

File Name	Description
ADS1100_Demo.c	ADS1100 readout demo application written in C
ADS1100_Demo.s43	ADS1100 readout demo application written in assembly
I2C_Master.s43	I^2C master communication library
I2C_Master.h	I^2C master communication library header file

Program Operation

On power-on reset, the MSP430 peripherals are initialized. This includes disabling the watchdog timer, configuring the LFXT1 oscillator load capacitors for the external watch crystal, and initializing the LCD controller and basic timer. In the source code, two 16-bit words are allocated in the MSP430 Flash Information memory segment A for holding calibration data. If both locations contain the same value (such as 0xFFFF after device programming on device start-up), the calibration mode is activated. Otherwise, the measurement mode is entered.

The ADS1100 is configured to continuously sample and convert with 16-bit precision. Its PGA gain is set to 8 to amplify the differential bridge sensor output voltage. Details on the ADS1100 operation can be found in [1]. Writing the configuration byte is accomplished by calling the I^2C communication library function I2CWrite8().

Next, the MSP430 enters low-power mode LPM3 with interrupts enabled. From now on, the entire program flow is interrupt driven. Two interrupt sources are enabled. The basic timer ISR is executed every 0.25 s and is mainly used for obtaining and displaying measurement results. The port 1 ISR is used for handling button events.

While in calibration mode, two data points are obtained. CAL LO or CAL HI is displayed to indicate which calibration data point is being handled. By pressing any button, the current ADS1100 A/D conversion result is read by calling the function I2CRead16() and stored into a temporary variable. After the calibration procedure, these two data points are programmed into the INFOA Flash Information memory segment using in-system self programming. The software now enters normal measurement mode. From now on, the ADS1100 conversion result is read every 0.25 s and the value is compared to the previous value. If the value changes, then a new display value is calculated and the display is updated. This avoids unnecessary 32-bit integer multiplications and divisions. The display value is calculated according to the following formula:

$$DisplayValue = \frac{CurrentADCValue - CalMin}{CalMax - CalMin} \times CAL\_MIN\_MAX\_SPAN$$

The range from CalMax to CalMin is projected into a range from 0 to CAL_MIN_MAX_SPAN. CAL_MIN_MAX_SPAN is set by default to 1,000. Calibration mode can be re-entered anytime by operating PB2.

Pushing PB1 disables conversions, switches off the LCD display, and the MSP430 enters the LPM3 mode. In this mode, the application circuit draws less than 1-µA current, with the 32-kHz oscillator still running. If required, the LPM4 mode can be used instead to reduce current consumption even further. When PB1 is pushed again, the application resumes normal operation.

I^2C Master Communication Library

To enable MSP430 devices without the hardware I^2C interface to communicate with external devices, a library has been created. This library uses bit-bang techniques to send and receive data over the I^2C bus lines. The three-state output scheme as required by the I^2C specification is implemented by having the port pin output latch cleared (PxOUT.x = 0), and only operating the port direction register (PxDIR.x). This way, the bus lines are either pulled up by the external resistor or driven low by the MSP430.

The library is written in assembly language to have a more predictable timing as can be achieved by using C, where the timing would be strongly dependent on compiler implementation and optimizer settings. When the CPU is operating at the default frequency of 1,048,576 Hz, the generated I^2C signal timing meets the standard mode speed specifications [2]. The library can be used with all available MSP430 devices.

The user-callable functions provided in the library follow C calling conventions and can be accessed easily from both C and assembly programs. A header file is provided to facilitate integration into the C environment. The following three functions can be called:

- `void I2CSetup(void)`

  This function initializes the MSP430 port pins that are used for I^2C communication. The port pin bit definitions SDA and SCL, as well as the port definitions PI2C_DIR, PI2C_OUT, and PI2C_IN, can be customized by modifying the header file I2C_Master.h. It is recommended to make use of the lower 4 bits of a port when selecting the I^2C signal pins. By doing so, the MSP430 CPU core is able to use the constant generator feature for performing the bit manipulations, thus reducing code size and increasing execution speed. For more information on the constant generator, refer to the RISC 16-Bit CPU section in the MSP430x4xx Family User's Guide [4].

- `unsigned int I2CRead16(unsigned char Addr)`

  This function reads 16-bit data from the specified I^2C slave device. The LSB of the I^2C address byte that is sent out is set internally to indicate that slave data is requested. Then, data is read from the bus with the MSB first.

- `void I2CWrite8(unsigned char Addr, unsigned char Data)`

  By calling this function, the 8-bit parameter Data is send to the I^2C slave device indicated with Addr.

Furthermore, the library contains additional helper functions for generating I^2C start and stop conditions, for ACK and nACK indication, and for basic data transfer. These functions are used by the complex user callable functions that have been described above. These functions can create a customized version of the I^2C library for communication with different peripherals which might require reading 8-bit data, writing 16-bit data, or any other combination.

References

1. *ADS1100 Self-Calibrating 16-bit A/D Converter Data Sheet* (SBAS239)
2. *The I^2C Bus Specification V2.1*, Philips Semiconductors, 2000
3. *MSP430x41x Mixed Signal Microcontroller Data Sheet* (SLAS340)
4. *MSP430x4xx Family User's Guide* (SLAU056)

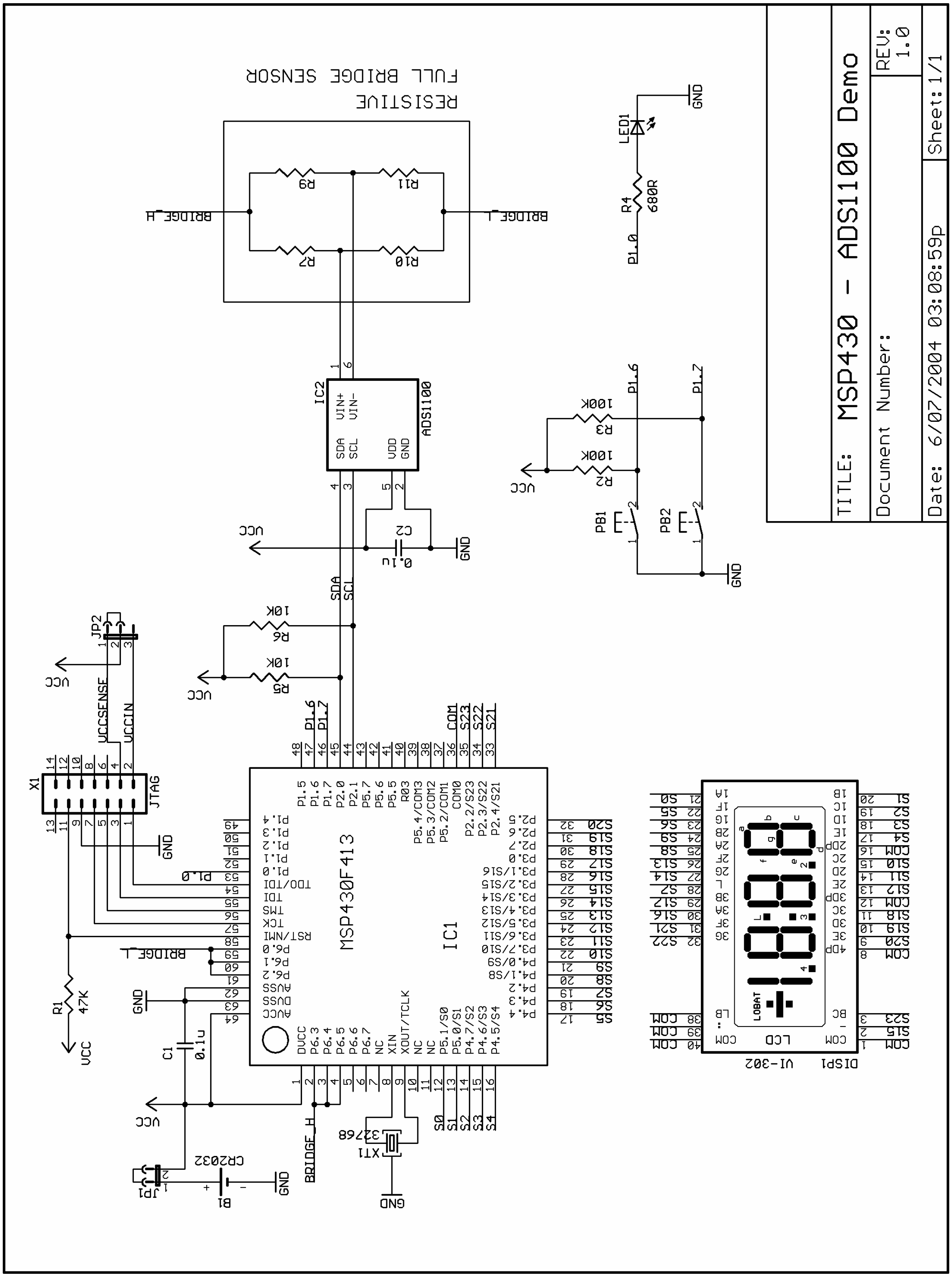

Figure 1. Application Schematic

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
		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

Copyright © 2004, Texas Instruments Incorporated