Application Report

SLAA216 – February 2005

Implementing a Single-Chip Thermocouple Interface with the MSP430x42x

Mike Mitchell *MSP430*

ABSTRACT

This application report describes and shows how to implement a single-chip thermocouple interface without using the signal conditioning circuitry normally required for thermocouples. The thermocouple interfaces directly to the MSP430F42x devices from Texas Instruments, Inc. The MSP430 uses its internal 16-bit sigma-delta ADC to convert the thermocouple readings to a digital value, convert them to temperature, and display them on an LCD. A complete schematic and code listing are given at the end of this report.

1 Introduction

This application report describes and shows how to implement a single-chip, direct thermocouple interface, without using the signal conditioning circuitry normally required for thermocouples. The thermocouple interfaces directly to the MSP430F42x microcontroller from Texas Instruments, Inc. The MSP430F42x device is an ultralow-power microcontroller with integrated 16-bit sigma-delta ADC. The integrated ADC is used to convert the thermocouple voltage into digital values and the MSP430F42x CPU is used to convert the digital values into temperature and display them on an LCD. A complete schematic is given in the appendix and the complete code used for this report is downloadable from the Internet. For this application report a type K thermocouple was used and the measured temperature range was limited to 0 – 99.9°C.

2 Thermocouples

Thermocouples are constructed of two dissimilar metals welded at one end. They produce a voltage at the non-welded end relative to the temperature difference between the two ends of the thermocouple. So, it is not enough to simply measure the thermocouple voltage to determine its temperature. That will only tell you the temperature difference between the two ends of the thermocouple. The temperature of the cold junction (the connection of the thermocouple to the measuring device) must also be known. As a result, some type of cold junction compensation is required. Often, circuits are employed to produce a voltage proportional to the cold junction temperature. This voltage is injected into the circuit and is part of the typical thermocouple signal conditioning circuitry.

Another technique of cold junction compensation involves measuring the temperature of the junction with a temperature sensor. This is the technique employed in this report. In this technique, the temperature of the thermocouple is determined from knowing the cold junction temperature and measuring the thermocouple voltage.

Since this technique relies on a temperature sensor for the cold junction compensation, best results are obtained when the temperature of the sensor and the physical connection of the thermocouple to the circuit board are held at the same temperature. Often, commercial thermocouple measuring devices use an isothermal block to maintain or stabilize the temperature at the cold junction. The isothermal block may be as simple as pouring copper on the printed circuit board around the thermocouple connection and the cold junction temperature sensor, or may be more sophisticated and involve mechanical assemblies. This application uses the simpler PCB approach.

The voltage thermocouples produce is standardized by the National Institute of Standards and Technology (http://www.nist.gov). Data tables for thermocouple voltages are available from the NIST at http://srdata.nist.gov/its90/main/.

3 Hardware

The circuit used in this application report is shown in Appendix A. The MSP430F42x devices integrate three fully independent, 16-bit sigma-delta ADC converters, each with a programmable gain amplifier (PGA). This application uses two of them; one for measuring the thermistor and one for measuring the thermocouple. The resolution and PGA capability of the ADC allows the elimination of all but the simplest signal conditioning for the thermocouple; only a simple RC filter is used for noise filtering. No other external amplifiers or compensation circuits are required.

The thermistor is used for the cold junction compensation. The Thermistor used is from DigiKey, part number KC003T-ND. A resistor divider is formed with a 47-kΩ resistor and the thermistor to produce a voltage input to the ADC. The top of the divider is connect to the 1.25 V reference output and the full-scale input of the ADC converter is ±600 mV. The thermistor resistance decreases with increased temperature. When the thermistor is at 0°C, its resistance is 32.77 kΩ. The 47 Kohm resistor keeps the thermistor input voltage well below the ADC full-scale voltage for the temperature range of 0-40°C. An RC filter is used on the thermistor input voltage for noise filtering and the PGA setting was set to a gain of one.

The thermocouple is a type K. It is connected directly to the ADC converter, with the same RC filtering as the thermistor. No other signal conditioning is used for the thermocouple circuit, except for setting the PGA of the ADC to a gain of 16.

The 32768-Hz crystal is used to provide a low frequency clock for the Basic Timer. It supports low power operation by allowing the rest of the device to be in LPM3 while only the LCD driver and a timer are operational. The crystal also provides the time base for the LCD driver.

The LCD used is the SBLCDA4 from Softbaugh (http://www.softbaugh.com). It is a 4-Mux, 7.1-digit LCD. Resistors R12, R14, and R17 provide the voltage levels for the integrated LCD driver.

Two switches are provided for a calibration function. Each switch is de-bounced with a 0.1-µF capacitor across its pull-up resistor.

4 Software

The software flow is shown below in Figure 1 and discussed in the following sections.

4.1 Overview

The software is written to provide a low-power system. The MSP430 is normally in low power mode 3 (LPM3) with only a 32768-Hz clock running. The Basic Timer uses the 32768-Hz clock to provide a one-second interrupt that wakes the MSP430 CPU for the temperature measurement.

This application uses two of the three integrated SD16 modules. They are grouped together to provide simultaneous sampling of the thermistor and thermocouple. During sampling, 32 conversions are performed on both the thermistor and thermocouple voltage and then averaged into a single conversion value. Next, the thermistor value is converted to temperature using a look-up table. The resulting thermistor temperature is then converted to an equivalent ADC count value for the thermocouple. This equivalent value is added to the ADC conversion value for the thermocouple, which provides the cold-junction compensation. Finally, the resulting temperature is determined using the adjusted thermocouple ADC value and a look-up table.

After the temperature is determined, it is converted to BCD format and displayed on the LCD. After display, the MSP430 is put back into LPM3 to wait for the next Basic Timer interrupt. The complete application consumes only 2.5 µA in LPM3, even with the LCD active.

A calibration routine is also provided. This allows for a single-point calibration of the initial temperature reading.

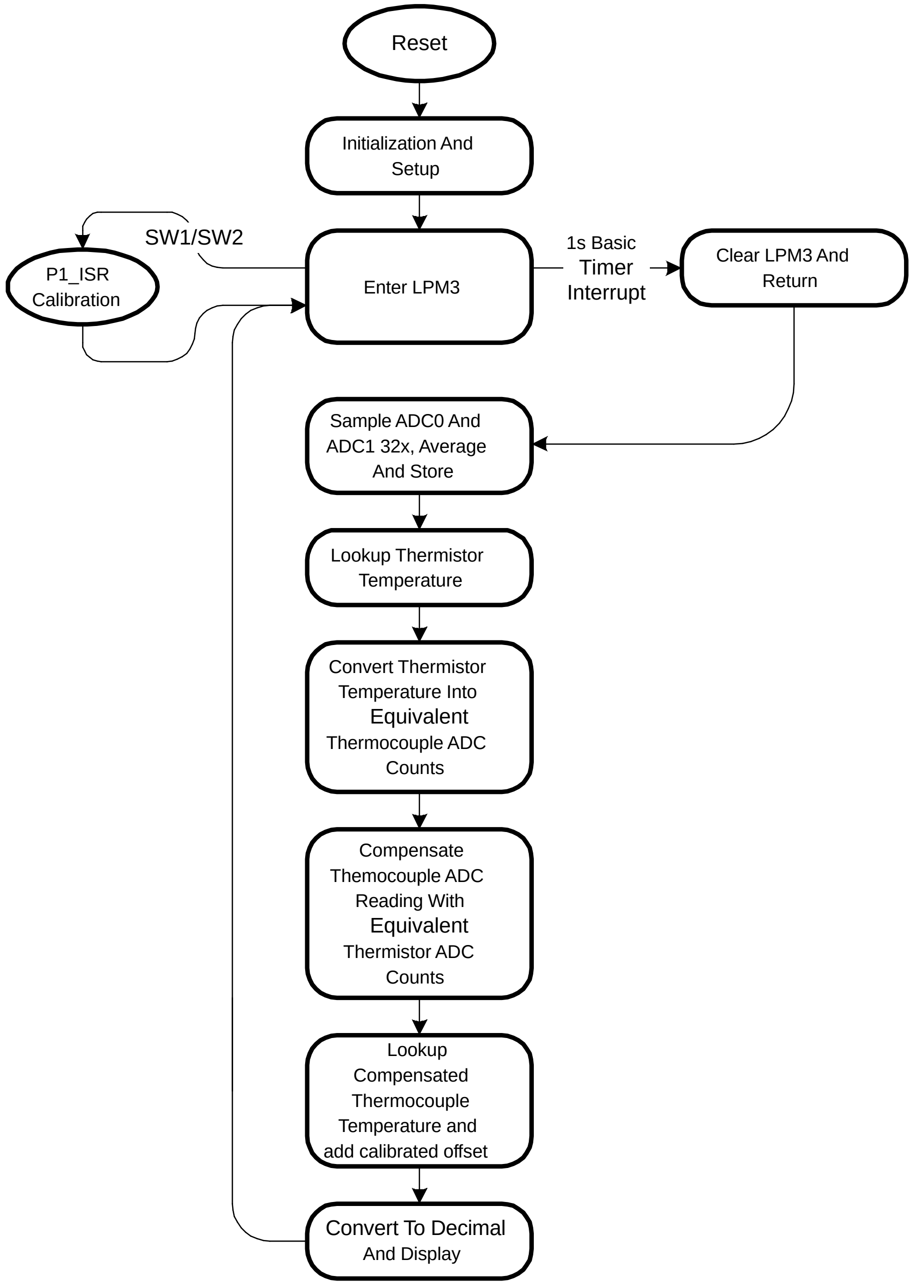


Figure 1. Software Flow

4.2 Initialization and Setup

During setup, the MSP430 watchdog timer is disabled, the stack pointer initialized, and the FLL is left at its default setting resulting in a CPU clock speed of 1.048 MHz. The LFXT1 crystal oscillator capacitors are selected so the 32768-Hz crystal oscillates properly. Next the Basic Timer is initialized for a one-second interrupt and each SD16 is initialized for clock source, data format, grouping, interrupt enable, and gain setting. Lastly, the MSP430 enters LPM3.

4.3 Mainloop

The Mainloop is a simple loop of subroutine calls. This loop gets executed once per second, triggered by the Basic Timer interrupt. First the sample routine is called. Then the thermistor temperature is determined. If the thermistor temperature is in-range, it is translated to thermocouple ADC counts. Then the thermocouple ADC value is adjusted using the translated thermistor counts and the thermocouple temperature is determined. Finally, the temperature is converted to decimal format and displayed on the LCD.

4.4 Sample

The sample routine first measures the offset of each ADC by selecting the shorted channel of each ADC and taking and buffering 32 ADC conversions. Afterwards the data is averaged into one offset value for each ADC and stored.

After offset measurement, the thermistor and thermocouple are measured 32 times each. Each ADC conversion is buffered and then averaged into a single value and stored. Finally, the measured offset of each ADC is subtracted from the measurements.

Since the software is written for low-power, the reference is only enabled during the Sample routine. After the measurements, the reference is turned off. The SD16 automatically enters a low-power state if it is not actively converting, so no other reconfiguration is required for low-power operation.

5 Get_TR_Temp – Determining the Thermistor Temperature

To determine the thermistor temperature from it’s ADC value, first a table lookup is performed to determine the temperature to the nearest degree. Then, an interpolation is done to determine the temperature to the nearest tenth of a degree. The whole and tenths portion of the thermistor temperature are stored in separate variables and used separately to adjust the thermocouple ADC value.

To interpolate the tenths portion the following equation is used:

((higher-ADCvalue)x10)/(higher-lower) where:

ADCvalue = ADC conversion value of thermistor voltage

higher = the next higher value in the table

lower = the next lower value in the table

The table of thermistor values shown in the code in the appendix is specific to this application. It uses measured values for the 47-kΩ resistor and for the ADC reference. The general formula for computing the table is:

$$\text{ADCvalue} = \text{hex}\left[2^{N} \times \frac{\text{voltage}}{2 \times \text{Vref}}\right] \text{ where :} \tag{1}$$

N = ADC resolution in bits

voltage = resulting voltage from the voltage divider

Vref = reference voltage

For this application the equation for the voltage divider is:

$$\text{voltage} = \left(\frac{\text{V} \times \text{Rt}}{\text{Rt} + 47\text{k}}\right) \text{ where :} \tag{2}$$

Rt = thermistor resistance and

V = the voltage source for the divider – Vref in this application.

Combining the two equations results in:

$$\text{ADCvalue} = \text{hex}\left[\left(\frac{2^N}{2 \times \text{Vref}}\right)\left(\frac{\text{Vref} \times \text{Rt}}{\text{Rt} + 47\text{k}}\right)\right\} \text{ which reduces to :} \quad (3)$$

$$\text{ADCvalue} = \text{hex}\left[\frac{2^{N-1} \times \text{Rt}}{\text{Rt} + 47\text{k}}\right] \quad (4)$$

The Get_TR_Temp also includes range checking to determine if the thermistor temperature is beyond the allowable range. If so, the appropriate flag is set to alert other sections of the program to the out-of-range condition.

6 Adjust_TC_Val – Compensating The Thermocouple Measurement

The Adjust_TC_Val routine is the cold junction compensation for the thermocouple. To compensate the thermocouple, the temperature of the cold junction must be known and it must be translated into an equivalent thermocouple voltage, and then added to the thermocouple voltage. Since this application performs the cold junction compensation in software, after measuring the cold junction temperature with the thermistor, the cold junction temperature is translated into equivalent ADC counts of thermocouple voltage.

The Adjust_TC_Val routine takes the whole portion of the thermistor temperature, TR_Whole, and pulls the corresponding thermocouple ADC value from the thermocouple data table. It then subtracts it from the next higher value in the table and multiplies this difference by the tenths portion of the thermistor temperature, TR_Tenths. Then this value is added to the table value that corresponds to the whole portion of the temperature. This yields an equivalent thermocouple ADC count value for the cold junction temperature, to the nearest tenth of a degree, even though the thermocouple data table is in one degree steps. Finally, the equivalent thermocouple ADC count value for the cold junction is added to the actual thermocouple ADC reading.

The Adjust_TC_Val routine is only executed if the thermistor temperature is in-range.

7 Get_TC_Temp – Determining the Thermocouple Temperature

Converting the adjusted thermocouple ADC value into temperature traditionally involves complicated math routines. However, this application report simplified the task by using a table lookup rather than a mathematical calculation. A table was constructed in Excel with the desired temperatures to be measured and the corresponding type K thermocouple voltage. Then the voltage was converted into equivalent ADC counts. Determining the thermocouple temperature then becomes a simple table lookup with tenths interpolation just like was done for the thermistor. After the thermocouple temperature is determined, the temperature adjustment, set in the calibration routine, is subtracted to give the final corrected temperature.

As with the thermistor, the table of values used for the thermocouple temperature lookup is specific to this application and uses a measured value for the reference. The general formula for computing the table is:

$$\text{ADCvalue} = \text{hex}\left[\frac{2^N \times \text{PGA} \times \text{Vtc}}{2 \times \text{Vref}}\right] \text{ where :} \quad (5)$$

N = 16

PGA = 28.35 (see device datasheet)

Vref = Reference for the ADC

Vtc = Thermocouple voltage

The Get_TC_Temp also includes range checking to determine if the thermocouple temperature is beyond the allowable range. If so, the appropriate flag is set to alert other sections of the program to the out-of-range condition.

8 P1_ISR – Calibration Routine

The P1_ISR routine, shown in Figure 2, is the interrupt service routine for the two switches and it implements a calibration routine. The calibration is a single-point calibration intended to calibrate initial temperature error. The calibration routine is entered by pressing SW1 and SW2 simultaneously. When calibration mode is entered, the last measured temperature is displayed on the LCD. Next, pressing SW1 or SW2 individually increments or decrements the display. The user should increment or decrement the displayed value until it reaches the actual known temperature and then press SW1 and SW2 simultaneously again to exit calibration mode. When calibration mode is exited, the difference between the measured temperature and the user-set temperature is calculated and stored as a temperature adjustment for the measured thermocouple temperature value.

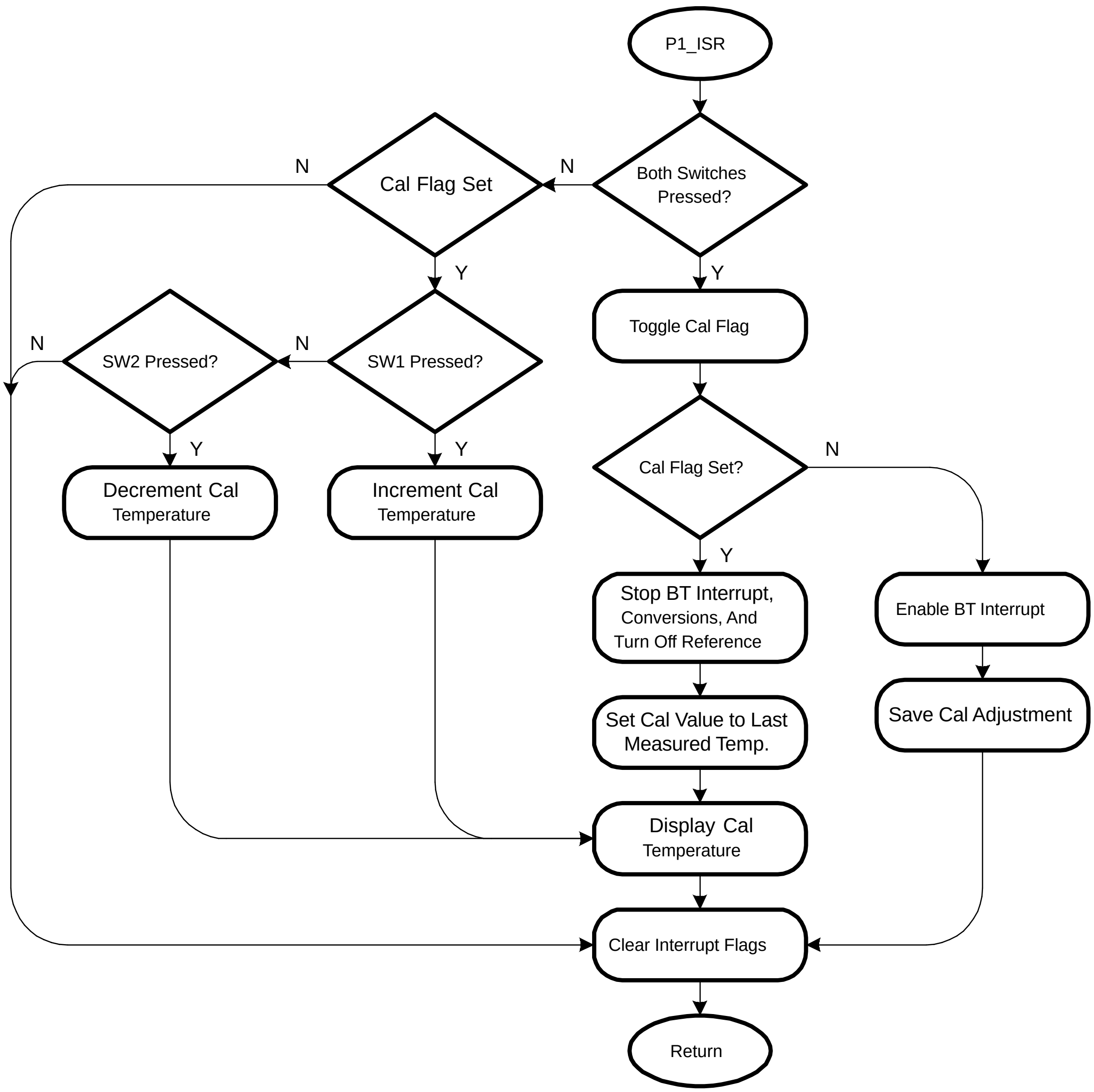


Figure 2. P1_ISR – Calibration Routine

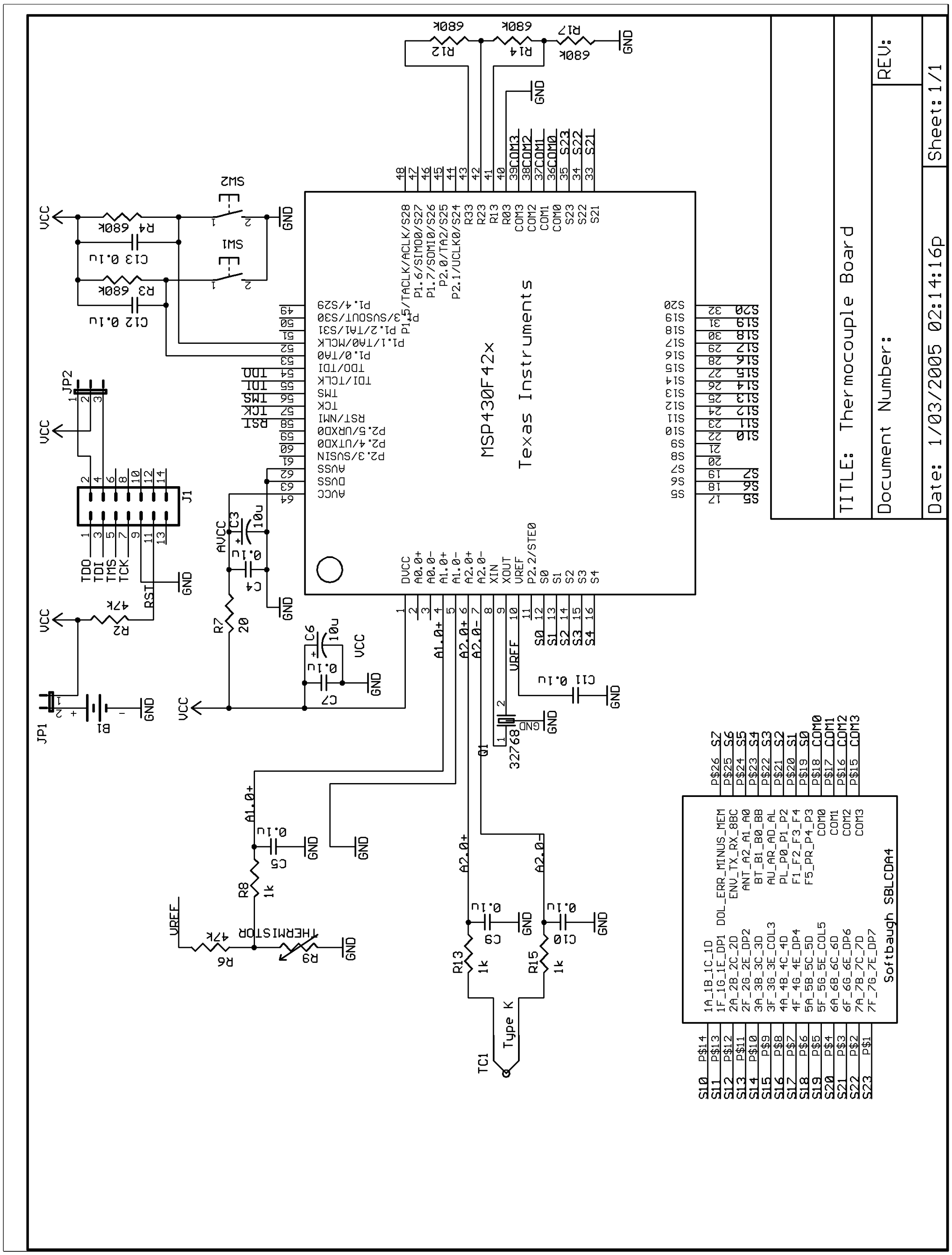
9 Binary-to-BCD and Display

After the temperature is determined, it is converted from binary to BCD format, and then displayed on the LCD. The LCD display routine used is generic for the chosen LCD. Unused portions of the LCD were blanked for this application. HI or LO is displayed if the thermistor or thermocouple temperature was out-of-range.

10 References

1. MSP430x4xx Family User's Guide (SLAU056)
2. MSP430x42x data sheet (SLAS421)

Appendix A Schematic



IMPORTANT NOTICE

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