Application Report
SLAA129A – MARCH 2004

Implementing An Ultralow-Power Thermostat With Slope A/D Conversion

Carlos Muzzarelli and Neal Brenner MSP430

ABSTRACT

This application note describes what slope A/D conversion is and how it can be easily implemented using the MSP430. To demonstrate the slope A/D technique, two examples of ultra low power thermostats are also presented, one using the MSP430F11x1 and the other with the MSP430F41x. The examples both use the comparator of the MSP430 to implement slope A/D conversion.

Introduction

Because temperature is an analog measurement, it must be converted to a digital format before it can be analyzed by a processor. To make this conversion an ADC (analog-to-digital converter) can be used. However, to reduce cost, complexity and board size, the slope A/D technique can be used instead of an ADC.

In the examples that follow, the on-chip comparator of the MSP430 is used to perform the slope A/D conversion with an external thermistor to determine the ambient temperature. Once the temperature is converted to a digital form, it can be compared to a stored number. Then the MSP430 can determine whether to heat or cool a room.

Slope A/D Conversion

Slope A/D conversion is a technique used to perform analog-to-digital conversion. The technique is based on the charging/discharging of a capacitor with a known capacitance. The number of clock cycles it takes to charge/discharge the capacitor is then counted. The slope A/D conversion technique uses the number of clock cycles and the known capacitance of the capacitor to digitize unknown analog resistances, voltages, or currents.

Implementation of Slope A/D Conversion Using the MSP430

Figure 1 shows the hardware configuration of a Slope A/D implementation using the MSP430, the figure includes the sensor Rsens, the reference resistor Rref, and the capacitor Cm.

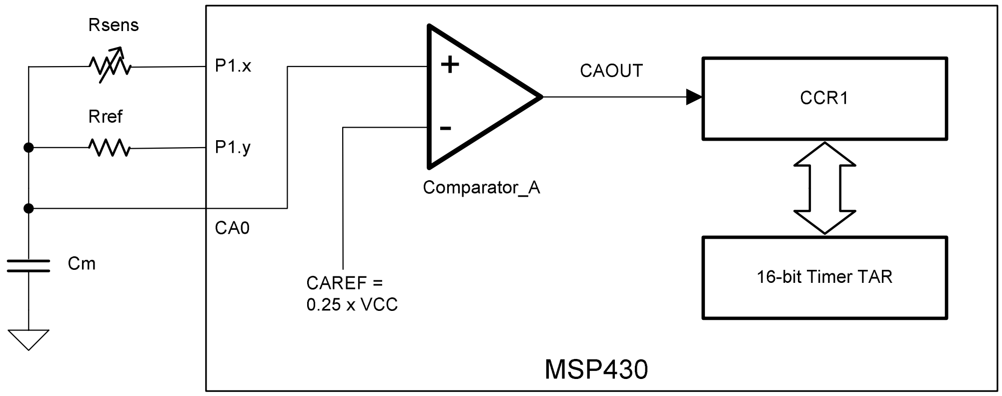

Figure 1. Measurement of Resistors

Capacitor Cm is charged to the Vcc voltage before each measurement. The timer is configured to run in continuous mode. The capacitor is first discharged through Rsens via a port pin. At the start of capacitor discharge, the value of timer register TAR is stored. When capacitor Cm voltage reaches the comparator reference value of 0.25 x Vcc, the negative edge of CAOUT causes the new TAR value to be captured in register CCR1. The difference between the value in CCR1 and the original start value of TAR represents the discharge time interval *tsens*. Next, the capacitor is again charged to Vcc, but this time it is discharged through Rref. The discharge time interval *tref* is found in the same way as previously described.

Figure 2 shows the voltage Vcm across capacitor Cm during the two measurements. The charge time tc must be between 5τ (for 1%) and 7τ (for 0.1%), depending on the accuracy required, where $\tau = Rref \cdot Cm$.

The two exponential equations describing the capacitor discharge through the reference resistor Rref and the sensor Rsens can be reduced to the following simple equation for Rsens:

$$\frac{Rsens}{Rref} = -\frac{tsens}{Cm \times \ln \frac{Vcaref}{Vcc}} \times -\frac{Cm \times \ln \frac{Vcaref}{Vcc}}{tref} \quad \rightarrow$$

$$Rsens = Rref \times \frac{tsens}{tref}$$

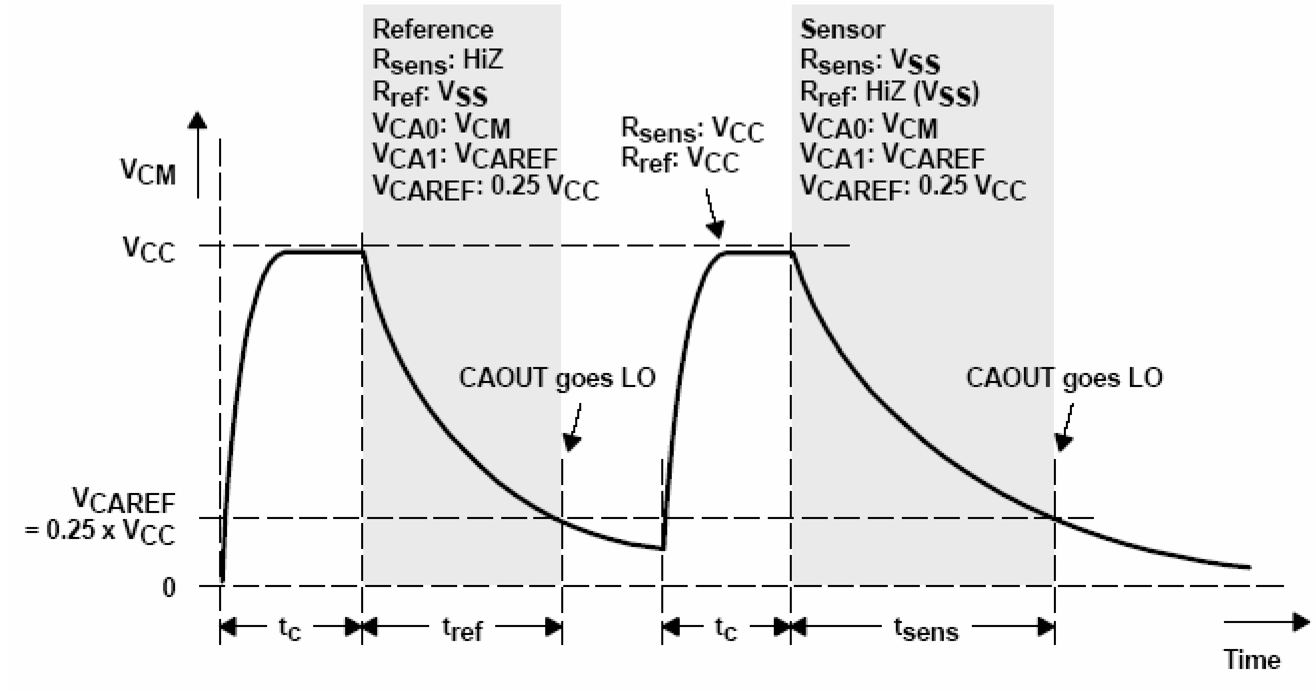

Figure 2. Voltage at Cm During Resistance Measurement

Thermostat Implementation Using Slope A/D Conversion

The next two sections show examples of how to use Slope A/D conversion with the MSP430 to build a thermostat. Both examples determine the temperature using Slope A/D to measure the resistance of a thermistor that changes as the temperature changes.

The first example is a low cost design using the MSP430F11x1. In this design, the user can set the desired temperature using a potentiometer. One of two LEDs light to show that the unit is cooling or heating. Slope A/D conversion is also used in this example to determine the potentiometer setting.

The second example offers more advanced features using the MSP430F41x. This example uses an LCD to display either current temperature or time. Two buttons are used to change the display and set the desired temperature and time.

Example 1: Thermostat Implementation using the MSP430F11x1

This circuit is a simple thermostat using the MSP430F11x1 microcontroller as shown in Figure 3. The software is interrupt driven, by a recurring five-second interrupt. The Timer-A module and capture/compare register 0 set the five-second interrupt. Between the five-second interrupts the circuit stays in low power mode three (LPM3) and draws an average current of only 1.5 µA.

The MSP430 reads the resistance value of the thermistor and the set-point potentiometer by performing a slope conversion as described previsouly. A thermistor resistance “look-up” table is then used to equate the resistance reading of the thermistor to a temperature. A different software routine is used to determine the temperature value corresponding to the resistance reading of the set-point potentiometer since there is a direct relationship between the resistance of the potentiometer (in series with the 4.7kohm resistor) and the temperature setting. The thermistor temperature is then compared to the set-point temperature. If the thermistor temperature is not within +/- 1 degree Fahrenheit of the set-point temperature, then port P1.1 or P1.2 goes high activating the Cooling or the Heating LED. Port P1.0 also goes high every five seconds to indicate that the system is on. Since the temperature in most rooms does not normally change quickly it is not necessary to measure the temperature more often, therefore allowing for the system to be in low power mode most of the time. This further conserves battery power since the slope A/D conversion is only performed every few seconds.

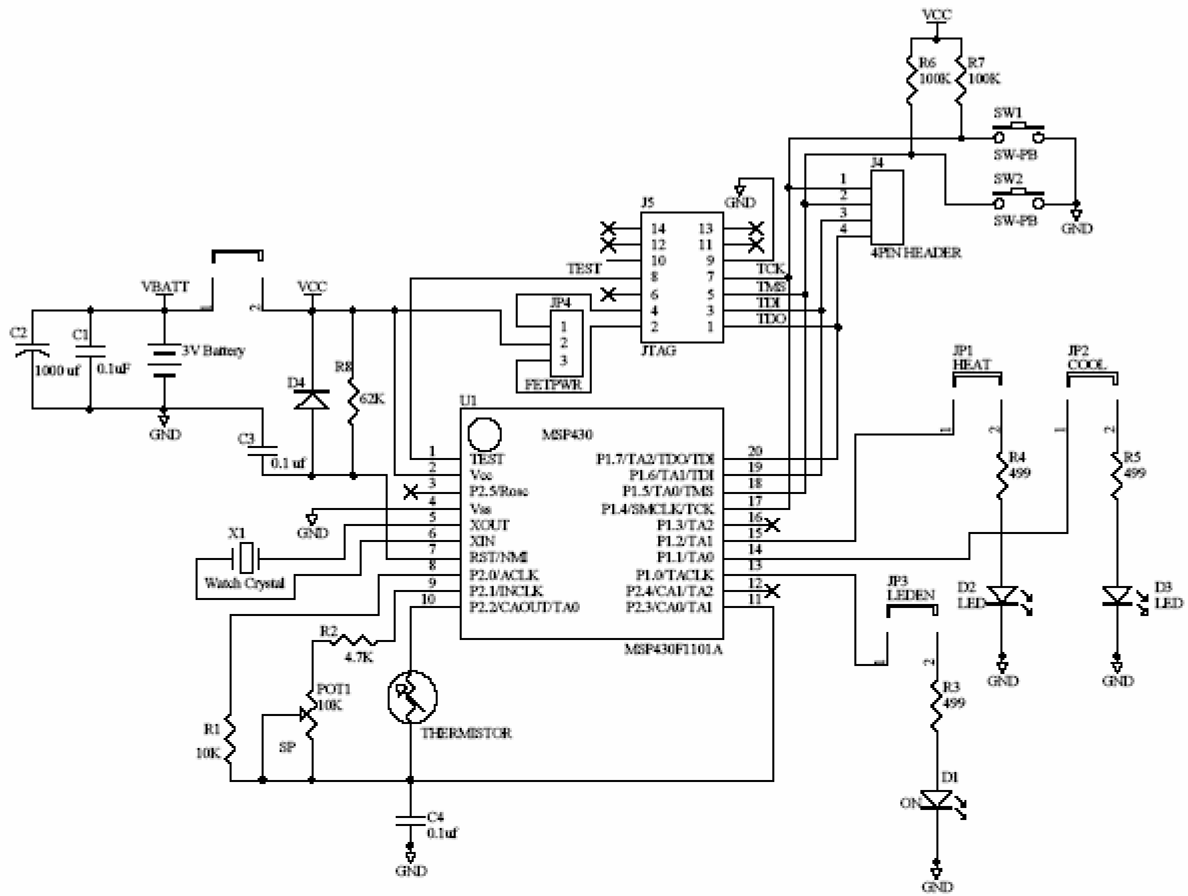

Figure 3. Thermostat using MSP430F11x1 Microcontroller

Example 2: Thermostat Implementation Using the MSP430F41x

The circuit in Figure 4 is an MSP430F41x thermostat implementation. It can operate in one of four modes determined by pressing the mode switch. The modes are thermostat, set point, time, and seconds. While in the set point and time modes, the user can change the set point or time using the set switch.

The software is interrupt driven, by a recurring one-second interrupt. Between interrupts the circuit stays in LPM3 and draws an average current of only 1.5 µA. The user is not aware the circuit is in low-power mode since the LCD remains active. The one-second interrupt is generated by one of the counters in the Basic Timer module, which continues to count even while the part is in LPM3.

The circuit maintains a real-time clock. The current hour and minute are displayed when the circuit is in time mode. The clock can be adjusted by pressing the set switch while in time mode. The time mode will automatically revert back to thermostat mode after ten seconds. There is also a “seconds” mode that simply counts seconds and displays them on the LCD.

The Basic Timer is clocked by a 32-kHz crystal that is remains active in LPM3. When the one second interrupt occurs, the on-board frequency locked loop (FLL+) starts so that code is executed at a high frequency. The real-time clock is then updated, even if the software is not in time mode. Next the low-power mode bits are cleared from the stack so that the part returns to the main loop in active mode, at the end of the interrupt routine. Once in the main loop, the operating mode is determined and the software branches to the associated routines. Once the routines are completed, the software returns to the main loop where it is immediately put into LPM3 to await the next one-second interrupt.

While in thermostat mode the MSP430 reads the resistance value of the thermistor by performing a slope conversion and determines temperature as described previously. The temperature is then compared to the set point. If the temperature is at or above the set point the LCD flashes the reading and pin P1.0 goes high and remains high as long as the reading is greater than or equal to the set point. The flashing LCD provides user feedback and the I/O pin could be used to control a heating/cooling system. In this application it drives an LED. Since the temperature in most rooms does not normally change quickly it is not necessary to measure the temperature every second. The software counts the number of seconds the circuit has been in thermostat mode and only measures the temperature every five seconds. This further conserves battery power.

While the circuit is in set point mode the symbol “-|“ is displayed on the LCD followed by the two-digit temperature set point. The symbol “-|“ was chosen because the 3.5-digit LCD does not allow us to display “SP” and then the two digit value of the set-point. The user is able to change the set point by pressing the set switch. This increments the set point one degree at a time through the range of the thermostat. If the maximum is exceeded, it rolls over to the lowest set point and increments from there. When the set point mode is exited, by pressing the mode switch, the software compares the new set point to the current one. If it did not change nothing is done. If it did change, a routine unlocks the flash to allow a write or erase. The software then checks to see if the entire flash information memory has been filled. If it is not full, a flash memory pointer is incremented to the next flash memory location and the new value is written to the information memory. If the information memory is full, it is first erased, the pointer is set back to the beginning of information memory, and the new set point is written. By using all 256 bytes of the information memory, the operating life of the application is extended. The MSP430 flash memory can be erased 100,000 times at 25 degree C. This means that the set point could be changed up to 25.6 million times before any flash memory errors occur.

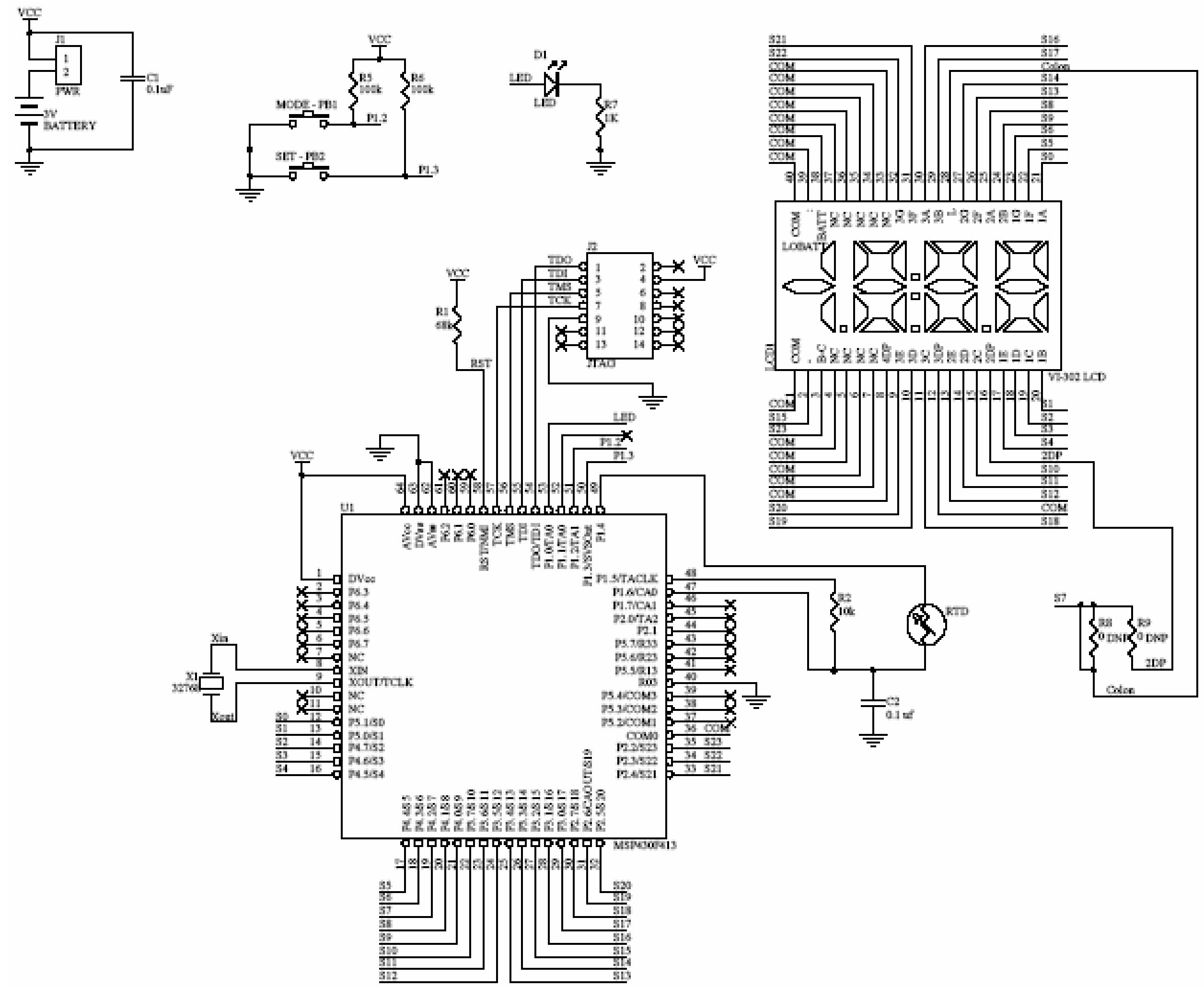

Figure 4. Thermostat Using MSP430F41x and a LCD

References

1. *MSP430F41x data sheet, SLAS340*
2. *MSP430F11x1 data sheet, SLAS241*
3. *MSP430x1xx User's Guide, SLAU049*
4. *MSP430x4xx User's Guide, SLAU056*

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