Implementation of Temperature Sensor on PICM4520 Microcontroller

Application Note

Brad Pasbjerg

Design Team 7

March 30th, 2012
Table of Contents

Cover ................................................................................................................................. 1
Table of Contents ........................................................................................................... 2
Abstract ............................................................................................................................ 3
Keywords .......................................................................................................................... 3
Introduction and Background ......................................................................................... 3
Objective .......................................................................................................................... 4
Configuring the ADC ...................................................................................................... 4
Programming the PIC18F4520 ....................................................................................... 6
Include Files ..................................................................................................................... 6
Define Variables ............................................................................................................... 6
Main Program ................................................................................................................... 6
Conclusion ....................................................................................................................... 8
References ....................................................................................................................... 9
Abstract

The TC1046VNBTR is a powerful temperature sensor that can transmit temperature information. This application note discusses how to implement the TC1046VNBTR to an ADC port on the PIC18F4520 Microcontroller (MCU).

Keywords

Analog to Digital Converter (ADC), Microcontroller Unit (MCU), MegaHertz (MHz), MilliVolts (mV)

Introduction and Background

The TC1046 is a linear output temperature sensor whose output voltage is directly proportional to measured temperature. The TC1046 can accurately measure temperature from -40 °C to +125 °C. The reading off this device will give us the actual temperature where the sensor is located.

The PIC18F4520 is a Microcontroller that has a high computational performance, with the addition of high endurance, Enhanced Flash program memory. Along with these features the PIC chip also has an Analog-to-Digital (A/D) converter module. This module has 13 input ports show in Figure 1. The ADC has a resolution of 10 bits, so the digital reading range is from 000 to 3FF as show in Figure 2. Based on the ADC reading and the known reference voltage you are able to derive the voltage reading by multiplying the reading by a constant.

Figure 1 – PIC18F4520 ADC Ports
Objective

The objective of this document is to demonstrate how to implement the TC1046 temperature sensor and program the PIC18F4520 ADC ports, including initializing the Microcontroller ADC and reference ports, collecting data from the port, and how to interpret that data.

Configuring the ADC

The Microcontroller first needs to be powered in order to work. There are 4 pins that need to be connected to power the chip. Pins 11 & 32 need to be connected to 5V, pins 12 & 32 need to be connected to ground. Also without a clock the MCU will be unable to perform commands, Pin 13 needs to be connected to an Oscillator. The frequency of the Oscillator is typically 4 MHz.

The next steps are to power the TC1046 and create a reference voltage. In order to power the temperature sensor, it is easiest to attach to a surface mount board due to its small size. As seen in figure 3, Pin1 will connect to the 5V power supply, Pin 3 will connect to ground, and the Pin 2, V_{out}, will connect to the input ADC Pin (for this application, the ADC port will be Pin 2 on the MCU).
For creating the reference voltage the maximum voltage at ADC port ($V_{OUT}$, Pin 2, of the temperature sensor) must be known. From the TC1046 data sheet we find the voltage range to be from 174mV to 1.205V. With a max of 1.205V, the reference should be made as close as possible to have the highest accuracy. The design in figure 4 shows a voltage divider that will produce 1.513V. The output of the voltage divider should go to Pin 5 on the MCU.
Programming the PIC18f4520

Include Files

The programming requires several modules in order to function. The required modules are ADCON1.c, SetChanADC.c, ConvertADC.c, BusyADC, and ReadADC. These can all be found within ADC.h. Useful libraries for this application are

```c
#include <p18cxxx.h>
#include <string.h>
#include <stdio.h>
#include <ADC.h>
#include <stdlib.h>
```

Define Variables

Variables must be defined to do the necessary calculations to retrieve the temperature from the ADC. A double is used in order to obtain a decimal value. All of the variables below like voltage and temperature will be determined within a few decimal points of accuracy. Integers like ‘count’ and ‘adc_result’ are used in order to follow a desired flow of programming.

```c
int adc_result;
long int count;
double number = 0;
double voltage = 0;
double tempC = 0;
double tempF = 0;
double test = 0;
```

Main Program

The first part of the program initializes the functions I call in the main program. These call functions help the main look clean and help visualize the flow of the program.

```c
void TempCalc ( void );
void VoltageAvg ( void );
```
This part of the code is the main which shows the flow of the program itself. The first thing the main does is initialize the ADC ports so the conversion uses the reference voltage we made at Pin 5 and ground to calculate the hexadecimal result.

The main then goes into a continuous while loop that continuously checks the ADC. This process does not stop until the system is forced to stop. So the temperature is being tracked in real time.

```c
void main ()
{

    //Temp Sensor Initial
    OpenADC(ADC_FOSC_32 & ADC_RIGHT_JUST & ADC_12_TAD,
            ADC_CH0 & ADC_INT_OFF, 0);  //open adc port for reading
    ADCON1 =0x10; //set VREF+ to Pin 5 and VREF- to GND (VSS)
    while ( 1 )
    {
        SetChanADC(ADC_CH0); //Set ADC to Pin 2
        VoltageAvg();
        TempCalc();
    }
}
```

The VoltageAvg function first enters a for loop and retrieves the ADC result in hexadecimal form. The variable holding this is ‘adc_result’. The result is then transformed into a decimal using ‘number’. The voltage is then calculated using the formula for ADC conversions for a 10-bit number and reference voltage measured to be 1.509V. The purpose of the for loop is to retrieve an averaged value for the temperature sensor. This helps to cancel the effect of noise from the sensor and voltage supplies, which often causes inaccurate readings.

The TempCalc function takes the voltage from the previous function and uses an equation from the TC1046 datasheet to calculate the temperature in Celsius and Fahrenheit. These values can now be used in combination with other applications to produce the temperature control system.

```c
void VoltageAvg ( void )
{
    for(count = 0; count < 50; count++)
    {
        ConvertADC(); //perform ADC conversion
        while(BusyADC()); //wait for result
        adc_result = ReadADC(); //get ADC result
        if(adc_result & 0x0008) number=8+number;
        if(adc_result & 0x0004) number=4+number;
    }
}
```
void TempCalc ( void )
{
    tempC = (voltage*1000 - 424)/6.25;
    tempF = tempC*9/5 + 32;
    voltage = 0;
}

Conclusion

As demonstrated by this program, temperature sensing applications can be handled by simple wiring and programming. The above program is able to calculate temperatures in Celsius and Fahrenheit that would be recommended to be used in an LCD display or a PID controller applications.
References

“TC1046 Temperature sensor data sheet”

“PIC18F4520 Microcontroller data sheet”