Lab 7f Temperature Data Acquisition System

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, Second edition, by Jonathan W. Valvano, published by Thomson, copyright © 2006.

Goals • Study ADC conversion, Nyquist Theorem, Valvano Postulate,
• Develop a temperature measurement system using a thermistor,

Review • Operation of the ADC system in the 9S12DP512 data sheet,
• Data sheets on the TLC2274 quad op amp IC, and INA122P instrumentation amp
• Valvano Chapters 11,12 on thermistors, analog amplifiers, analog low pass filters, and ADC's, data acquisition systems.

Starter files • OC, ADC, LCD, therm10.xls, therm10Vref2_5.xls, lpf.xls, www.ti.com **FilterPro**

Requirements

This experiment will use an ADC converter on the 9S12 to construct a digital thermometer. The temperature range is 0 to 40 ˚C. The current temperature will be displayed as a fixed-point number on the LCD, using a 0.01 fixed-point format. Your temperature measurement resolution should be 0.1 ˚C or better. The average temperature accuracy should be 1 ˚C or better. The frequency components of the signal are 0 to 10 Hz.

Approach and Constraints

Figure 7.1 shows the data-flow graph of the temperature data acquisition system. The thermistor converts temperature to resistance, the analog amp and filter convert resistance to 0 to +5V analog voltage, and the ADC converts analog voltage to 10-bit integer. The periodic ISR creates the real time sampling, and data is passed to the foreground using any appropriate data structure (shown as **Data** in Figure 7.1). The main program converts integer to fixed-point data using a table lookup-interpolation scheme. The results are displayed on the LCD.

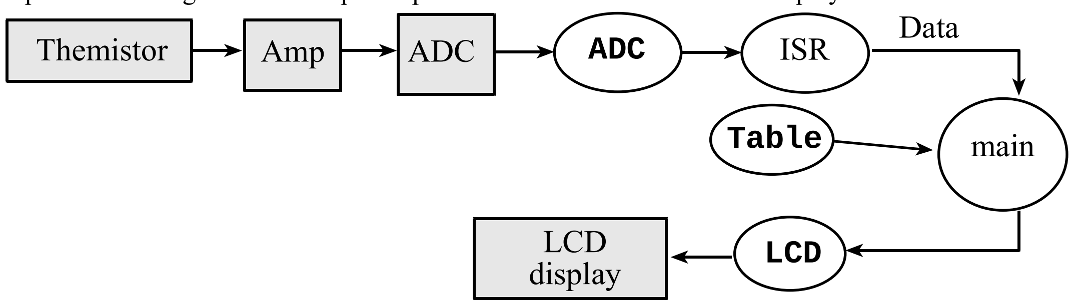

Figure 7.1. Data-flow graph of the data acquisition system.

Figure 7.2 shows one possible call-graph. Each of the modules (**LCD**, **ADC**, **Data**) has separate header and implementation files. Notice the main program does not directly access the **ADC** or **LCD** I/O port registers. The module **Data** can employ any appropriate method to pass data from background to foreground.

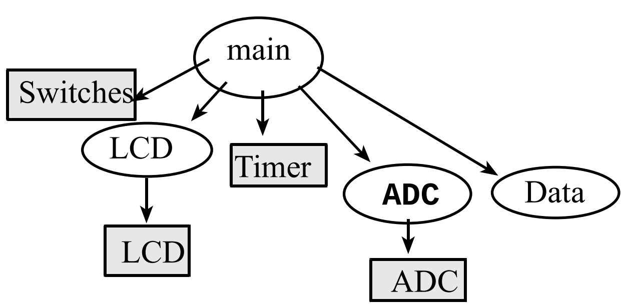

Figure 7.2. One possible call-graph of the data acquisition system.

This data acquisition system includes a thermistor, which converts temperature into a resistance. One possible way to convert resistance into voltage is to use a bridge. A second possibility is to use a constant current source. In either case, an analog amplifier boosts the voltage so the full scale temperature range (0 to 40 ˚C) maps into the full-scale range of the ADC (0 to 5V). A low-pass analog filter, with a cutoff frequency of about ½ the sampling rate, removes high frequencies that might otherwise cause aliasing. A low-pass analog filter must be implemented in this lab.

1. Temperature-Resistance Calibration of the Thermistor:

The thermistor resistance varies nonlinearly with its temperature. It is very important to use temperature units of Kelvin in this equation and not ˚C.

$$R = R_O e^{+\beta/T} \qquad \text{(where T is temperature in degrees Kelvin)}$$

The thermistors in this lab have a resistance of about 30 kΩ at 25 ˚C. When attaching wires to the thermistor you will need to add water-proof insulation so the tip of the probe can be inserted into 5 mm of water. One water-proof method is to lightly paint the bare wires with paint or epoxy. Do not use electrical tape.

2. Choose a sampling rate

In this lab, we will process temperature signals (0 to 10 Hz). According to the Nyquist Theorem, we need a sampling rate greater than 20 Hz. Output compare interrupts will be used to sample the ADC in a background thread. This high priority interrupt will establish the sampling rate. This selection of sampling frequency will affect the design of analog and digital filters. So, when you change the sampling rate, you will have to redesign the filters.

Nyquist Theorem: If f_{max} is the largest frequency component of the analog signal, then you must sample more than twice f_{max} in order to faithfully represent the signal in the digital samples. For example, if the analog signal is

$$A + B \sin(2\pi ft + \phi)$$

and the sampling rate is greater than 2f, you will be able to determine A, B, f, and ϕ from the digital samples.

Valvano Postulate: If f_{max} is the largest frequency component of the analog signal, then you must sample more than ten times f_{max} in order for the reconstructed digital samples to look like the original signal when plotted on a voltage versus time graph.

3. Hardware Interface

Figure 7.3 shows one possibility for the analog electronics of the digital thermometer. You will add an analog filter in this lab. Choose the cutoff frequency of the LPF to be about ½ the sampling rate. The amplifier should convert the entire temperature range into the 0 to +5 V ADC range. Because you are using rail-to-rail op amps, the entire system can be powered by a single +5 V supply. PLEASE DO NOT USE +12 OR –12 V SUPPLIES IN THIS LAB. If you have received free samples from Analog Devices, TI and Maxim, you should design your system using state of the art components. The gain of the INA122P amplifier is determined by the resistor R_G. If you are using an AD623, connect the reference pin to 2.5V and select R2 to be the resistance of the mid-point temperature (see **Therm10Vref2_5.xls**). Go to TI.com and download their design tool called **FilterPro**.

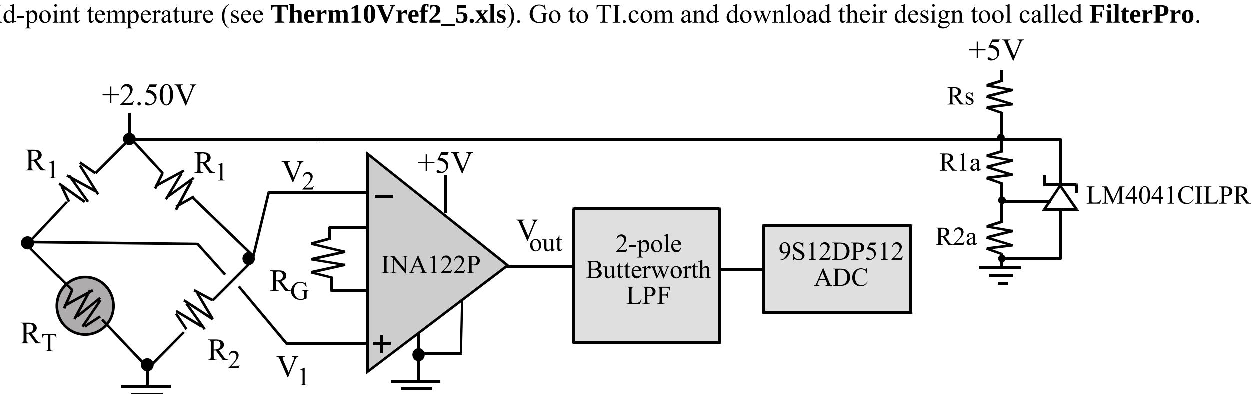

Figure 7.3. Possible thermistor interface (easy to construct, but more expensive).

We need an instrumentation amp (premade or built) because of its high input impedance and good CMRR. This semester, TI has an INA122P instrumentation amplifier that can be obtained as a free sample. If you did not order an INA122P, then you will need to build your own instrumentation amplifier using 3 op-amps, as shown in Figure 7.4. The gain of the 3-op-amp instrumentation amplifier is a function of the resistors R_3 R_4 R_5 and R_6. The TLC2274 operates rail-to-rail, which means its output can swing all the way from 0 to +5 V. Any of the rail-to-rail op amps mentioned in class can be used in this lab: AD8032, TLC2274, or OPA2350. These op amps can operate on a single +5V supply. *In fact, if you connect the TLC2274 up to the usual +12 –12 V supplies, you will damage the device and damage your 9S12.*

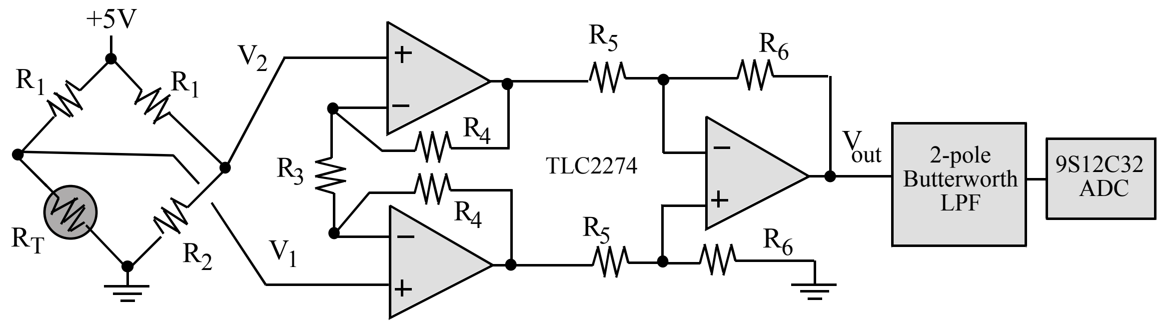

Figure 7.4. A good thermistor interface (harder to construct, but cheaper to build).

The advantage of using an instrumentation amp (with its high Z_{in}) is you can analyze the transfer function from R_T to V_1 considering only the bridge input (+2.50 or +5V), R_1 and R_T. Normally, the R_1 resistor in the bridge is chosen large enough to prevent self-heating the thermistor. Assume the dissipation constant to be about 1 mW/˚C. Limit the thermistor power to 0.1 mW so that the self-heating error is below 0.1 ˚C. The R_2 resistor in the bridge establishes one of the extreme values of the temperature range. For example, when the thermistor resistance, R_T, equals R_2, then the bridge output voltage is 0. The gain of the differential amplifier, along with the ADC range (0 to +5 V in our case) will determine the temperature range of the system.

4. Software Conversion:

Using the calibration data, the nonlinear thermistor equation, the characteristics of your analog circuit and the response of the 9S12's ADC, determine the ADC output sample for each temperature for about 5 to 10 temperature points within the 0 to 40 ˚C temperature range. There is an Excel spreadsheet to assist you called **Therm10.xls** or **Therm10Vref2_5.xls** depending on whether your instrumentation reference pin is ground or 2.5V. Show both, a table of figures and a plot of this data. Include appropriate intermediate voltages in the table (e.g., thermistor resistance, bridge output, and analog circuit output.) Design a software conversion routine that calculates temperature from the ADC sample. You should consider various methods:

a) linear equation (don't use because it has errors too large),
b) nonlinear equation,
c) large table lookup (one entry for each ADC value, i.e., 1024 entries),
d) small table lookup (≈16 entries) with linear interpolation in between.

The 9S12 has special table lookup and interpolation op codes, so you may wish to consider writing the conversion in assembly language. See the **`TBL.RTF`** and **`ETBL.RTF`** examples on the **TExaS** simulator.

5. Real-time ADC sampling:

The program will continuously sample the ADC at a sampling frequency, f_S, selected in part 2. The ADC must be implemented using interrupts. There are two good approaches to implementing real-time ADC sampling. The first approach is to implement a periodic interrupt using output compare at the desired sampling rate. In the ISR you start the ADC conversion, wait for it to complete, read the ADC result, and pass the data from the ISR to the main program using a FIFO queue.

The second approach is to create a digital squarewave with a frequency matching the desired sampling rate using a PWM channel. The output of the PWM is connected to an input at PAD7. The ADC is configured to start a conversion on a transition of PAD7. The ADC is also armed to generate an interrupt on completion. In this second approach, the ADC ISR does not need to start the conversion. In fact, the ADC completion flag triggers the interrupt. This ISR simply reads the data, acknowledges the interrupt and passes the data to the foreground using a FIFO queue.

The main or foreground thread will get data from the FIFO, calculate temperature from the ADC sample, and then display the measured temperature on the LCD display. The LCD may be too slow to be able to erase and redraw each measurement. Rather, it will be faster to move the LCD cursor and redraw only the new temperature measurement.

6. Temperature Resolution (skip this section Spring 2010):

To measure temperature resolution, we use the student's t-test to determine if the system is able to detect the change. To use the student's t test we need to make the following assumptions:

1) the errors in one data set are independent (not correlated to) the errors in the other data set;
2) the errors in each data sample are independent (not correlated to) the errors in other data within that set;
3) the errors are normally distributed;
4) the variance is unknown;
5) the variances in the two sets are equal.

If a random variable, X, is normally distributed with a mean is μ and a standard deviation of σ, then the probability that it falls between $\pm 1\ \sigma$ is 68 %. I.e.,

$P(\mu-\sigma < X < \mu+\sigma) = 0.68$

Similarly,

$P(\mu-1.96\sigma < X < \mu+1.96\sigma) = 0.95$
$P(\mu-2\sigma < X < \mu+2\sigma) = 0.954$
$P(\mu-2.58\sigma < X < \mu+2.58\sigma) = 0.99$
$P(\mu-3\sigma < X < \mu+3\sigma) = 0.9997$

The square of the standard deviation is called variance, σ^2. In most situations, we do not know the mean and standard deviation, so we collect data and estimate them. In particular, we take multiple measurements assuming the temperature is constant. Let $\boldsymbol{X_i}$ be repeated measurements under the same conditions, and $\boldsymbol{N}$ is the number of measurements (e.g., $\boldsymbol{N} = 10$).

$$\overline{X} = \frac{1}{N}\sum_i X_i \qquad S^2 = \frac{1}{N-1}\sum_i (X_i - \overline{X})^2$$

The N-1 term is used in the calculation of S because there are N-1 degrees of freedom. These expressions are unbiased estimates of μ and σ, meaning as the sample size increases the estimates approach truth. Formally, we say the expected value of $\overline{X}$ is μ, or $E(\overline{X}) = \mu$. Similarly, the expected value of S^2 is σ^2, or $E(S^2) = \sigma^2$.

For example, we collect two sets of data (e.g., 10 measurements in each set, $\boldsymbol{N} = 10$), and we want to know if the means of two sample sets are different. Consider the measurements in the two data sets as the sum of the true value plus an error:

$X_{01} = \mu_0 + e_{0i}$
$X_{i1} = \mu_1 + e_{1i}$

Assumption 1 states that e_{0i} are not correlated to e_{1i}. Assumption 2 states that e_{0i} are not correlated to e_{0j} and e_{1i} are not correlated to e_{1j}. Thermal noise will satisfy these assumptions. We employ a test statistic to test the hypothesis H_0: $\mu_0 = \mu_1$. First, we estimate the means and variances of the data (assuming equal sized samples)

$$\overline{X_0} = \frac{1}{N}\sum_i X_{0i} \qquad S_0^2 = \frac{1}{N-1}\sum_i (X_{0i} - \overline{X_0})^2$$

$$\overline{X_1} = \frac{1}{N}\sum_i X_{1i} \qquad S_1^2 = \frac{1}{N-1}\sum_i (X_{1i} - \overline{X_1})^2$$

From these, we calculate the test statistic **t**:

$$t = \frac{\overline{X_1} - \overline{X_0}}{\sqrt{S_0^2/N + S_1^2/N}}$$

The two sets of data, together, have 2N-2 degrees of freedom. The student's t table, shown as Table 7.1, has two dimensions. In the vertical direction, we specify the degrees of freedom, **df**. For example, if there are 10 data points in each data set, then **df** equals 18. In the horizontal direction we select the probability of being correct. For example, if we wish to be 99% sure of the test, then we select the 99% column. Selecting the row and the column allows us to pick a number threshold. For example, the number in the **df**=18 row, **confidence**=99% column is 2.878. This means if H_0 is true, then

Probability of **t** < -2.878 = 0.005 and Probability of **t** > 2.878 = 0.005

Therefore

Probability of -2.878 < **t** < 2.878 = 0.99 (confidence interval of 99%)

If we collect data and calculate **t** such that the test statistic **t** is greater than 2.878 or less than -2.878, then we claim "we reject the hypothesis H_0". If the test statistic **t** is between -2.878 and 2.878 we do not claim the hypothesis to be true. In other words we have not proven the means to be equal. Rather, we say "we do not reject the hypothesis H_0".

confidence	80%	90%	98%	**99%**	99.8%	99.9%
df p=	0.10	0.05	0.01	**0.005**	0.001	0.0005
8	1.397	1.860	2.896	**3.355**	4.501	5.041
9	1.383	1.833	2.821	**3.250**	4.297	4.781
10	1.372	1.812	2.764	**3.169**	4.144	4.587
11	1.363	1.796	2.718	**3.106**	4.025	4.437
12	1.356	1.782	2.681	**3.055**	3.930	4.318
13	1.350	1.771	2.650	**3.012**	3.852	4.221
14	1.345	1.761	2.624	**2.977**	3.787	4.140
15	1.341	1.753	2.602	**2.947**	3.733	4.073
16	1.337	1.746	2.583	**2.921**	3.686	4.015
17	1.333	1.740	2.567	**2.898**	3.646	3.965
18	**1.330**	**1.734**	**2.552**	**2.878**	**3.610**	**3.922**
19	1.328	1.729	2.539	**2.861**	3.579	3.883
20	1.325	1.725	2.528	**2.845**	3.552	3.850
21	1.323	1.721	2.518	**2.831**	3.527	3.819
22	1.321	1.717	2.508	**2.819**	3.505	3.792
23	1.319	1.714	2.500	**2.807**	3.485	3.767
24	1.318	1.711	2.492	**2.797**	3.467	3.745
25	1.316	1.708	2.485	**2.787**	3.450	3.725
26	1.315	1.706	2.479	**2.779**	3.435	3.707
27	1.314	1.703	2.473	**2.771**	3.421	3.690
28	1.313	1.701	2.467	**2.763**	3.408	3.674
29	1.311	1.699	2.462	**2.756**	3.396	3.659
30	1.310	1.697	2.457	**2.750**	3.385	3.646
40	1.303	1.684	2.423	**2.704**	3.307	3.551
50	1.299	1.676	2.403	**2.678**	3.261	3.496
60	1.296	1.671	2.390	**2.660**	3.232	3.460
80	1.292	1.664	2.374	**2.639**	3.195	3.416
100	1.290	1.660	2.364	**2.626**	3.174	3.390
120	1.289	1.658	2.358	**2.617**	3.160	3.373
∞	1.282	1.645	2.326	**2.576**	3.090	3.291

Table 7.1. Student's t distribution table.

Preparation (do this before your lab period)

1. The thermistor resistance varies nonlinearly with its temperature. Perform a very crude temperature calibration experiment with two points somewhere in your temperature range of 0 to 40 ˚C. One temperature standard can be created with a mixture of crushed ice and water in an insulated vessel (e.g., a Styrofoam cup). If you have access to the 2nd floor lab, you should use the temperature controller set at 37 ˚C for the second point. To use the temperature controller, put tap water into the chamber to a depth of about 5 mm (¼ inch). Place your thermistor into the water so it touches the aluminum plate and gently cover the top of the hole with an insulating material. Keeping the insulation dry will yield better results. You can use paper towel to clean the chamber after use. If you do not have access to the 2nd floor, you can use your skin temperature in the axilla region (arm pit), which is about 36 ˚C, the room temperature as measured by the heating/AC system, or the outside temperature as measured by the weather man. Use the Excel spreadsheet to determine R_0 and β from the two calibration points. In particular, **Therm10.xls** or **Therm10Vref2_5.xls** depending on whether your instrumentation reference pin is ground or 2.5V. This quick calibration will only be used to choose resistors in the circuit. A real calibration will be performed as procedure 5.

2. Review the technical information on the ADC system of 9S12. List three ways you could use to initiate the ADC conversion process. What are two ways of knowing that the conversion process has been completed? Place the answers to these two questions in the beginning comment section of your main program. You may use the

example ADC programs, but it is your responsibility to understand the modes. In particular, you should know 8/10-bit (ATD0CTL4 bit 7), right/left justified (DJM), signed/unsigned format (DSGN), multichannel mode (MULT), and continuous mode (SCAN).

3. Choose one of the options as discussed in hardware section and design the appropriate thermistor amplifier. Be prepared during checkout to discuss the reasons for your choice of design. You must add a two-pole Butterworth low-pass anti-aliasing analog filter. Show name and number of all the pins involved including power. Add bypass capacitors on all chips. Why is it is important to connect bypass capacitors across the power pins for the analog IC components? Label all resistance and capacitance values and types. For example, 1kΩ 5% carbon, or 0.01μF 10% X7R ceramic. Include the interface to the LCD. Draw the circuit using the CAD tool PCB Artist.

4. This program is not used to measure temperature. Rather it is used to study the Nyquist Theorem as described in Procedure 1). Create a real-time data acquisition system that takes 100 samples at 1000 Hz, then prints out the data. Download the SCI0_DP512.zip starter file. You can either add your ADC and real-time sampling to this SCI0 project, or add the SCI0.c SCI0.h files to your project. You do not have to do it exactly like the following code, but you should first collect the data, next print it out, then stop.

```
unsigned short DataBuffer[100];
unsigned short Count=0;
void back(void){ unsigned short data;
  if(Count<100){
    data = ADC0_In(0x85); // your program that samples channel 5
    DataBuffer[Count++] = data;
  }
}
void main(void){unsigned short i;
  PLL_Init();          // 24 MHz
  ADC0_Init();         // your module
  SCI0_Init(115200);   // SCI output to PC
  OC_Start(1,&back);   // your module sampling at 1000 Hz
  while(Count<100){};  // copy ADC to buffer in background
  for(i=0; i<100; i++){
    SCI0_OutUDec(DataBuffer[i]); SCI0_OutChar(10);SCI0_OutChar(13);
  }
  for(;;){};
}
```

5. Write the software system that samples the ADC at a rate at or above 20 Hz. Pass the data from background into the foreground using a fifo queue. In the main program convert the ADC data to fixed point temperature and display it on the LCD. A “syntax-error-free” hardcopy listing for the software is required as preparation. The TA will check off your listing at the beginning of the lab period. You are required to do your editing before lab. The debugging will be done during lab. Document clearly the operation of the routines.

Procedure (do this during your lab period)

1. *Basic understanding*: The purpose of this section is to verify the **Nyquist Theorem** and the **Valvano Postulate**. Generate a continuous waveform (1 to +4V) with an adjustable frequency from 10 Hz to 10 kHz. Use a function generator but make sure the output is 1 to +4V. Please connect the analog waveform to a scope and verify the voltage range is between 0 and +5V before connecting to the 9S12. VOLTAGES OUTSIDE THIS RANGE WILL DAMAGE THE 9S12. Next connect the signal to an analog input, e.g., PAD5. In this part, we will not be using the thermistor or analog amplifier. Use the software system from Preparation 4) to capture 100 data points at 1 kHz sampling. Collect data for frequencies about 100 Hz (Valvano Postulate), 500 Hz (Nyquist), and 2 kHz (aliased). Use the SCI output to transfer data to the PC. Plot the results by connecting the data points with a straight line. Describe the concepts of Nyquist Theorem, Valvano Postulate, and aliasing using this specific data. Be prepared to explain your results during checkout.

2. *Static analog circuit test*: Perform these tests before connecting the circuit to the 9S12. Construct and evaluate the thermistor circuit. One by one replace the thermistor with 4 regular resistors that have resistances within the typical range of your thermistor. One test resistor should have a resistance equal to the resistance of the thermistor near the maximum temperature, and another test resistor should have a resistance equal to the resistance of the thermistor near the minimum temperature. Record the voltage values at strategic places in your analog circuit. What voltage output do you get when the thermistor is disconnected? What voltage output do you get when the thermistor wires

are shorted? You should modify your temperature measurement software to output specific error conditions if the thermistor is shorted or disconnected.

3. *Dynamic analog circuit test*: Again, perform these tests before connecting the circuit to the 9S12. Disconnect the thermistor, and connect a sine-wave signal generator in its place. Make sure the voltage level of the signal generator is within range, so that the inputs and outputs of your analog circuit are not saturated. Record the sine-wave amplitudes of the input and output voltages. Start at about 1 Hz and collect measurements at ten different frequencies. Make sure you choose frequencies large enough to see the gain roll off. Calculate the gain at each frequency. Plot the gain versus frequency response of your circuit. In systems where the shape of the signal is important, such as audio or video, the phase versus frequency response is important. You do not have to measure the phase response of your analog circuit.

4. *Analog-to-Digital Conversion*: Modify your real-time system so it displays ADC sample on the LCD as a decimal number from 0 to 1023. Connect the output of your thermistor amplifier to the input of the 9S12 ADC system. Use your four fixed resistors and collect digital samples. Add the ADC sample to the data collected in part 2.

5. *Calibration*. In this section your software will output the ADC sample as a decimal number. We define the temperature as measured by the Fluke 87V multimeter as reference truth. There is a K-type thermocouple that plugs into the Fluke allowing it to measure temperature. Its range is -40 to 260 °C (-40 to 500 °F), and its accuracy is 2.2 °C or 2 %. There are two of these meters on the 4th floor lab, so record with which Fluke meter you calibrated. You are allowed to use another temperature reference, as long as it is as good as this Fluke. Place your thermistor and the reference thermometer into an insulated cup with crushed ice and water. Wait for both your ADC measurement and the reference thermometer to stabilize. Record both the true temperature and the ADC sample as measured by your system. Next, place your thermistor and the reference thermometer in the chamber of the temperature controller set at 37 ˚C (see preparation 1 for now to use the controller). Record both the true temperature and the ADC sample as measured by your system. Incorporate this calibration data into a header file called `calib.h`. In particular, if you change thermistors or recalibrate, only changes to this header file will be required. You can use or not use the spreadsheets **Therm10.xls** or **Therm10Vref2_5.xls** as long as you incorporate the nonlinearity of the thermistors in an appropriate manner. You could take a third calibration point at room temperature in air. One possible solution is to use the 0 and 37 ˚C to create a 53-point table from Therm10.xls. Then, use the third calibration point as a temperature offset to adjust the therm10.xls. I.e., sample the ADC, use the therm10.xls lookup tables to convert ADC into temperature, then add a temperature correction as determined by the one air temperature calibration. In order to reduce the time for all EE345L students to finish this lab, please leave the temperature controller set at 37 ˚C. It takes at least 30 minutes for the controller to stabilize. This way, multiple groups can collect data simultaneously.

Figure 7.5. The Fluke 87V can measure temperature.

6. *Accuracy*. **Accuracy** is defined as the absolute difference between the true temperature and the value measured by your device. Accuracy is dependent on the same parameters as resolution, but in addition it is also dependent on the stability of the transducer and the quality of the calibration procedure. Wait at least one hour between calibration and accuracy measurements, but please use the same Fluke meter. In this section your software will output the temperature as a fixed-point number with a 0.01 ˚C format. Collect measures in crushed ice/water, room air, and in the temperature controller set at 37 ˚C, creating a table showing the true temperature (x_{ti} as determined by the temperature controller), and measured temperature (x_{mi} using your device). Calculate average accuracy by calculating the average difference between truth and measurement,

$$\text{Average accuracy (with units in ˚C)} = \frac{1}{n}\sum_{i=1}^{n}\left| x_{ti} - x_{mi} \right|$$

7. *Reproducibility*: Place the thermistor in either crushed ice/water or in the chamber of the temperature controller set the temperature to 37 ˚C, and record 10 independent temperature measurements. Calculate the standard deviation of these data and report S (estimation of σ) as reproducibility.

8. **Skip this part** *Resolution*: Now, set the temperature to 37.1 ˚C, and record 10 more independent temperature measurements. Again, calculate the standard deviation of these data. Next calculate the student's t statistic and use it to determine if your system as a temperature resolution of 0.1 ˚C or better.

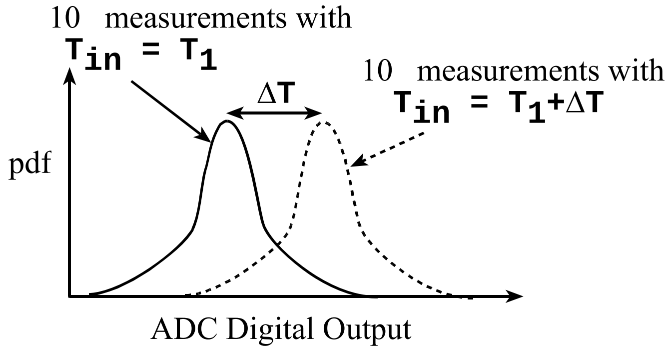

Figure 7.6. Resolution means if the temperature increases by ΔT, the system will probably notice.

Deliverables (exact components of the lab report)
A) Objectives (1/2 page maximum)
B) Hardware Design
Circuit diagram of the thermistor and LCD interfaces
C) Software Design (a hardcopy software printout is due at the time of demonstration)
1) Calibration data (procedure 5 and the **`calib.h`** file)
2) Low level ADC interface (**`ADC.c`** and **`ADC.h`** files)
3) Main program used to measure temperature
D) Measurement Data
1) Sketch three waveforms (procedure 1)
2) Static circuit performance (procedure 2,4)
3) Dynamic circuit performance (procedure 3)
4) Accuracy (procedure 6)
5) Reproducibility (procedure 7)
E) Analysis and Discussion (1 page maximum)

Checkout (show this to the TA)
You should be able to demonstrate the proper operation of digital thermometer. We expect your accuracy to be at least 1 ˚C and your reproducibility to be less than 0.1 ˚C. *(However, since the entire class is using the same calibration process, the TAs can adjust these expectations based on experimental conditions)*

A hardcopy printout of your software will be given to your TA, and graded for style at a later time.

Hints
1) This is a long lab with many parts, so start early.
2) Don't try to complete the experiment in one full swoop. Run the hardware test program given above before testing your software. Debug the system in an analytical, step-wise manner.
3) Please do not use regular op amps that require ±12 V supplies. An advantage of the single supply op amp with rail-to-rail outputs is that the output of your analog amplifier will remain within the 0 to +5V ADC range.
4) Circuits built with either the INA122P are very easy to implement and have much improved performance over instrumentation amps built with three op amps.
5) The temperature controller is should be set to 37 ˚C, so all students can share, and no one has to wait.
6) I built the Therm10.xls spread sheet to handle up to 13-bit ADCs. In order to use the **`etbl`** instruction, there is a place in the software where B=(256*(xL-x1))/(x2-x1), containing the 8-bit binary fraction of how far between the points is the sample. The closer the points are together, the more accurate the conversion. Notice that 53*256 > 8192. See the etbl.rtf example file created while TExaS is installed, or. I posted etbl.rtf on the lab manual site. http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2009/etbl.rtf. I duplicated entries at the beginning and the end, so the search phase of the conversion always gives a solution.
7) The underlined sections identify components that must be performed and included in the lab report.