ECE263

Lab 7

Spring 2013

Temperature Display Module
TDM

Revised: 29 Mar 2013 DAL

(intentionally left blank)

Object:

In this lab you will design a Temperature Display Module (TDM) that will measure the ambient temperature from two locations and then display the results as degrees Fahrenheit (F) using the program you generated for Lab 5.

Material:

Dragon 12 Plus Development Board
10K NTC Thermistor
Users choice

Background:

Sensor

In order to measure "real world" information, computers need a sensor that will convert whatever phenomena that is being measured into an electrical signal that can be converted into digital format. For measuring temperature, you will be using a *thermistor* which is a device that is made from a material that changes its resistance as a function of temperature. By placing the thermistor in a simple voltage divider circuit (Figure 1), resistance values can be converted into voltage levels that can be converted into digital format by using an analog to digital converter (ADC).

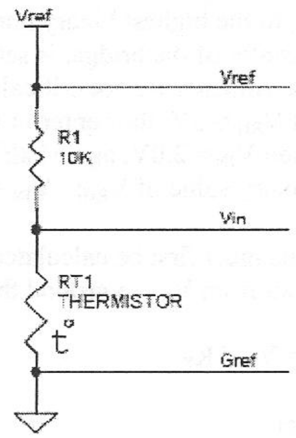

Figure 1.
Voltage Divider Circuit

The voltage levels can then be converted into temperature readings with some simple calculations. Manufacturers of thermistors usually provide a graph, mathematical function, and/ or a table that will translate the resistance reading for a specific device into the corresponding temperature value in either Celsius or Fahrenheit units. The thermistors that will be used for this lab are negative temperature coefficient (NTC) devices that are nominally 10K at 25C (77F). With NTC thermistors, the resistance is inversely proportional to the temperature which means that as the temperature increases the resistance of the material decreases. Table 1, which is attached at the end of this write-up, has the corresponding resistance reading for temperatures from -50F to 300F in 1F steps.

A/D Conversion

The analog to digital conversion process operates by comparing the voltage to be measured (V_{IN}) to a known voltage. An ADC has two reference voltage inputs, V_{RH} and V_{RL}, that establish a range of voltages which is divided into a number of discrete steps by using a resistor ladder or equivalent. Each step represents a portion of the full range and the larger the number of steps, the finer the resolution of the result.

RANGE = $V_{RH} - V_{RL}$ (e.g. 5V- 0V = 5V range)
STEPS = 2^N, where N = the number of binary bits of the result (e.g. 10-bit => 1023 steps)
RESOLUTION = RANGE / STEPS (e.g. 5V / 1023 steps = 4.9 mV/step)

The lowest voltage in the range (V_{RL}) is assigned the lowest binary value and the highest voltage (V_{RH}) is assigned the highest binary value, for example, with an unsigned results 0V = 000_{16} and 5V = $3FF_{16}$. The voltage that is being measured is sampled and compared to the voltage of each step and the result is the number of the step that most closely matches the sampled input voltage. The final result is obtained by multiplying the binary result by the resolution. If the result of a 10-bit ADC with a range of 5 volts was $0F4_{16}$ the voltage would be:

V_{IN} = $0F4_{16}$ steps x 5V / 1023 steps = 244_{10} steps x 4.9 mV/step = 1.196V

<u>Ratiometric Measurement Circuit</u>

The resistor divider circuit that you are going to use to measure the resistance of the thermistor, shown in Figure 1, is a ratiometric circuit. The reference voltage of the divider is also used as the reference voltage V_{RH} of the ADC. Since the ADC converts V_{RH} to the highest binary number of the range and since the input voltage being measured, in this case the center of the bridge, is set to a binary number in proportion to the relative voltage of the range, the absolute voltages are not critical. For example, if the reference resistor and the thermistor are both 10K, and if V_{RH} = 5V, the center of the bridge (V_{IN}) would be 2.5V or ½ of V_{RH} . If , on the other hand, V_{RH} = 4V, then V_{IN} = 2.0V, again half of V_{RH}. In either case, the binary result of the conversion would be half of the binary value of V_{RH}: $V_{IN} = 3FF_{16} / 2 = 1FF_{16}$.

To measure the temperature, the resistance value must first be calculated using Ohm's law: V = I x R. Using the ratiomentric circuit, the current I flows from V_{RH} to ground through both resistors R_{REF} and R_T which gives us :

$$I = V / R \Rightarrow I = V_{REF} / (R_{REF} + R_T) = V_{IN} / R_T$$

When solving the above equation for R_T you get:

$$R_T = V_{IN} \times R_{REF} / (V_{REF} - V_{IN})$$

Since the ADC result for V_{IN} is a discrete number, you can pre-calculate all of the possible resistance reading that can be measured with the ADC by using a program such as EXCEL to solve the above equation substituting in all numbers for V_{IN} from 000_{16} to $3FF_{16}$. Column 1 of the spreadsheet would be VIN and would contain the numbers 0 through 127, column 2 would contain the formula for RT, and column 3 would be manually filled by looking up the resistances values in the *Resistance vs. Temperature Chart* and selecting the appropriate temperature value. You will notice that in several cases a 1 bit change in the binary ADC result does not give a full 1F change in temperature reading. You could interpolate to fractional degrees but for the sake of simplicity, just choose the closest approximation.

In order to use the table you create, the numeric ADC result would be used as an index or pointer into a look-up table with the corresponding temperature values replacing the resistance values. To create a lookup table for your program, use the define constant (DC) assembler directive to create a series of values in successive memory locations in ROM space at the end of your program code. The table would resemble the following:

```
TEMP_TABLE: DC.B  99   ;99F
            DC.B  98   ;98F
            DC.B  97   ;97F
```

With this method, to display a temperature you would simple do an A/D conversion, read the result, use it as a pointer into the Temperature Table and then read out the correct temperature.

ADC Module

The MC9S12DG256 microcontroller used on the Dragon 12-Plus board has two 8-channel, 8 or 10-bit successive approximation ADC modules. The pins used by the two ADC modules are are shared with a general purpose digital input port. Each ADC module has several control registers that are used to select the port pins for either analog or digital operation and to configure the operating characteristics of the ADC operation, such as:

1. ATD Enable / Power Down
2. Result Length: 8-bit or 10-bit
3. Signed / Unsigned Result
4. Result Justification : Right or Left
5. Clock Rate (for successive approximation)
6. Scan: single conversion or continuous
7. Mult: single channel of multiple channels per conversion
8. Channel Select Codes

For this lab you should select: single conversion, single channel, 8-bit, unsigned result, with either justification. The 8-bit result will be sufficiently accurate and will cut down on the number of values you need to have in the conversion table. You should also limit the size of the table by restricting the display to a smaller temperature range: 40F to 90F for example. Note: you will need to modify the pointer if you use less that 256 entries in the table.

Pre-Lab

Program

The program for this lab will measure the temperature of two 10K NTC thermistors which will be monitoring two different locations (e.g. inside temp and outside temp) and then it will save the results in memory so they can be displayed. In addition, the highest and lowest values read on chan 1 (inside) will also be saved for display. The code for these functions will be incorporated into the program that you generated for Lab 5. The thermistors will be attached to AN2 and AN3.

The temperature measurement and conversion routine will be a new task that will be called by the task manager Whenever the TMP_CON flag is set in T_FLG. It will do an A/D conversion of each channel and then determine the temperature by means of a look-up table as described in lecture. The results from the first channel will be placed in the variable named NUM_0 in Lab 5 and the results from the conversion of the second channel will be placed in NUM_1. After storing the result for the first channel, you also compare it to the values in NUM_2 (MAX). If the current reading is higher than the stored value in NUM_2 then replace it with the current value. Then compare the current reading to the value in NUM_3 (MIN) and if it is lower, replace NUM_3 with the current value.

Tasks 1 and Task 2 will remain unchanged. In the RTI_ISR you will use a third counter to keep track of the timing for the temperature measurements. Every time this 8-bit counter rolls over to zero, you will set the TMP_CON bit in T_FLG which will cause the new task to be called every 262 ms.

The decode functions in Task 3 will be slightly changed. The UP and DWN keys will no longer be used so those routine can simply do a return. The ENTER key will be used to clear the MAX and MIN values by re-initializing the MAX to be equal to the lowest possible temp and MIN to be equal to the highest possible temp. The MODE key will retain it's function and will allow you to cycle between the four different temperature displays

1. Create an EXCEL spreadsheet using the formula described above and create a table of resistances and temperatures for all possible A/D values of V_{IN} from 00_{16} to FF_{16} (remember V_{RH} would be FF_{16} for 8-bit, unsigned results)
2. Using the Thermistor Chart attached, add a column that substitutes the closest temperature in place of the resistance.
3. Create an assembly code table of the EXCEL results as described previously.

Questions

1. What is an NTC thermistor
2. What is the resolution of an 8-bit A/D where V_{RH} is 3.3V, V_{RL} is 0V?
3. What is the resistance of the thermistor at 200F?
4. What is a ratiometric circuit?

Before you come to the lab, do the following:

1. Answer questions
2. Draw a flow chart or write the pseudo code of the new sections of the program as described above.
3. Generate the assembly code for the program.

At the start of the lab, turn in a copy of the flow chart and the assembly code that will be your starting point. Remember, the pre-lab is to be the combined effort of both (all) lab partners.

Procedure: Working with your lab partner(s), complete the following steps:

Using the procedures you learned in the previous labs,

Setup

1. Get a thermistors and measure the resistance:
   a. What is the resistance?
   b. What is the temperature?
   c. hold the thermistors in your hand, how does the resistance change?
2. Attach the thermistors to the Dragon 12P board:
   a. attach the thermistor to terminal block T6 between GND (center) and AN2.
   b. insert a 10K resistor into the proto board
   c. attach a wire between VRH and one end of the 10K resistor
   d. attach a wire between the other end of the 10K resistor and AN2 along with the thermistor lead
3. Create a new project and enter your program as *main.asm*. Generate the executable object file using the *Make* facility and then download the program on the Dragon 12 Plus.

4. Initialize NUM_0, NUM_1, and NUM_2 variables to 0 and NUM_3 variable to $FF.

Debug

5. Start testing the program by verifying that you are displaying the initial temperature on the seven segment display.
6. Verify that you can perform A/D conversions and return a valid numeric result.
7. Verify that you can convert the A/D result to a valid temperature.
8. Verify that the temperature meaurment is being done every 262 msec.
9. Verify the overall operation of the program.
10. Verify that your program can display valid results as the temperature is changed.
11. Demonstrate your results to the TA or instructor.

Lab Report:

NOTE: DO NOT INCLUDE A COPY OF THIS LAB WRITE-UP IN THE LAB REPORT

°F	OHMS	°F	OHMS	°F	OHMS	°F	OHMS	°F	OHMS	°F	OHMS	°F	OHMS
-50	481,142	0	85,387	50	19,900	100	5,827	150	2,044	200	829	250	378
-49	472,642	1	82,719	51	19,377	101	5,697	151	2,005	201	815	251	373
-48	454,909	2	80,142	52	18,870	102	5,570	152	1,966	202	802	252	367
-47	437,907	3	77,656	53	18,377	103	5,446	153	1,929	203	788	253	362
-46	421,602	4	75,266	54	17,899	104	5,326	154	1,892	204	776	254	357
-45	405,965	5	72,937	55	17,435	105	5,208	155	1,856	205	763	255	352
-44	390,966	6	70,698	56	16,985	106	5,094	156	1,821	206	750	256	347
-43	376,577	7	68,535	57	16,548	107	4,982	157	1,787	207	738	257	342
-42	362,770	8	66,447	58	16,123	108	4,873	158	1,753	208	726	258	337
-41	349,522	9	64,428	59	15,711	109	4,767	159	1,720	209	714	259	332
-40	336,804	10	62,479	60	15,310	110	4,664	160	1,688	210	702	260	327
-39	324,597	11	60,595	61	14,921	111	4,563	161	1,657	211	691	261	323
-38	312,876	12	58,774	62	14,543	112	4,464	162	1,626	212	680	262	318
-37	301,622	13	57,014	63	14,176	113	4,368	163	1,596	213	669	263	314
-36	290,813	14	55,313	64	13,820	114	4,274	164	1,567	214	658	264	309
-35	280,433	15	53,669	65	13,473	115	4,183	165	1,538	215	648	265	305
-34	270,460	16	52,078	66	13,136	116	4,094	166	1505	216	637	266	301
-33	260,878	17	50,541	67	12,809	117	4,007	167	1,482	217	627	267	296
-32	251,670	18	49,054	68	12,491	118	3,922	168	1,455	218	617	268	292
-31	242,821	19	47,616	69	12,182	119	3,839	169	1,429	219	607	269	288
-30	234,316	20	46,225	70	11,882	120	3,758	170	1,402	220	598	270	284
-29	226,138	21	44,879	71	11,589	121	3,679	171	1,377	221	588	271	280
-28	218,276	22	43,577	72	11,305	122	3,602	172	1,352	222	579	272	276
-27	210,716	23	42,318	73	11,029	123	3,527	173	1,328	223	570	273	273
-26	203,445	24	41,099	74	10,761	124	3,454	174	1,304	224	561	274	269
-25	196,451	25	39,919	75	10,500	125	3,383	175	1,281	225	553	275	265
-24	189,722	26	38,777	76	10,246	126	3,312	176	1,258	226	544	276	262
-23	183,246	27	37,671	25°C $\boxed{77}$ 10K	9,999	127	3,244	177	1,235	227	536	277	258
-22	177,019	28	36,601	78	9,758	128	3,177	178	1,213	228	527	278	255
-21	171,023	29	35,565	79	9,526	129	3,112	179	1,192	229	519	279	251
-20	165,251	30	34,561	80	9,297	130	3,049	180	1,171	230	511	280	248
-19	159,696	31	33,590	81	9,076	131	2,987	181	1,150	231	503	281	244
-18	154,347	32	32,648	82	8,861	132	2,926	182	1,130	232	496	282	241
-17	149,197	33	31,737	83	8,651	133	2,867	183	1,110	233	488	283	238
-16	144,236	34	30,853	84	8,447	134	2,809	184	1,091	234	481	284	235
-15	139,458	35	29,998	85	8,249	135	2,752	185	1,072	235	473	285	232
-14	134,855	36	29,169	86	8,056	136	2,697	186	1,054	236	466	286	229
-13	130,420	37	28,365	87	7,867	137	2,643	187	1,035	237	459	287	225
-12	126,147	38	27,587	88	7,684	138	2,591	188	1,017	238	452	288	223
-11	122,030	39	26,832	89	7,506	139	2,539	189	1,000	239	445	289	220
-10	118,061	40	26,100	90	7,333	140	2,489	190	983	240	439	290	217
-9	114,235	41	25,391	91	7,164	141	2,440	191	966	241	432	291	214
-8	110,547	42	24,704	92	6,999	142	2,392	192	950	242	426	292	211
-7	106,991	43	24,037	93	6,839	143	2,345	193	933	243	420	293	208
-6	103,561	44	23,391	94	6,683	144	2,299	194	918	244	413	294	206
-5	100,254	45	22,764	95	6,530	145	2,254	195	902	245	407	295	203
-4	97,063	46	22,156	96	6,382	146	2,210	196	887	246	401	296	200
-3	93,986	47	21,566	97	6,238	147	2,167	197	872	247	395	297	198
-2	91,017	48	20,993	98	6,097	148	2,125	198	857	248	390	298	195
-1	88,152	49	20,438	99	5,960	149	2,084	199	843	249	384	299	193
												300	190

Table 2: Temperature vs. Resistance Chart