Column #107 March 2004 by Jon Williams:

Measure High, Measure Low

I'm pretty sure I've made this confession before, but if I haven't, here goes: I'm a bit of nut when it comes to temperature. Let's just say that I have an exceptional temperature curiosity. I have thermometers of one sort or another spread from one end of my home to the other, and I seem to be checking them constantly. I even found a useful little travel clock with an atomic clock and a thermometer built in; I can keep track of the exact time when I travel and monitor the environment at the same time -- I like that.

In science and industry, one of the most popular methods of measuring temperature is with a thermocouple. They're inexpensive, fairly accurate, and easy to use. They're easy, but the process of using them properly is not necessarily very simple. Let's back up a bit ... a thermocouple is constructed by joining two dissimilar metal wires at one end. A voltage will be developed between the joined and open end that is proportional to the temperature difference between the two ends. This is called the Seebeck voltage, named after Thomas Seebeck who discovered the effect in 1821.

The trick is that the Seebeck voltage is very small; on the order of fractional- to low-millivolts, so we just can't pull out our trusty DMM and measure it. Another thing is establishing a reference at what is called the "cold junction" (the point where we measure the

Seebeck voltage). This connection is called the cold junction because prior to electronic compensation, this connection point was placed in an ice bath to keep it at (or very near) zero degrees Celsius. If you look at a standard thermocouple table you'll see that the reference junction is specified at zero degrees Celsius.

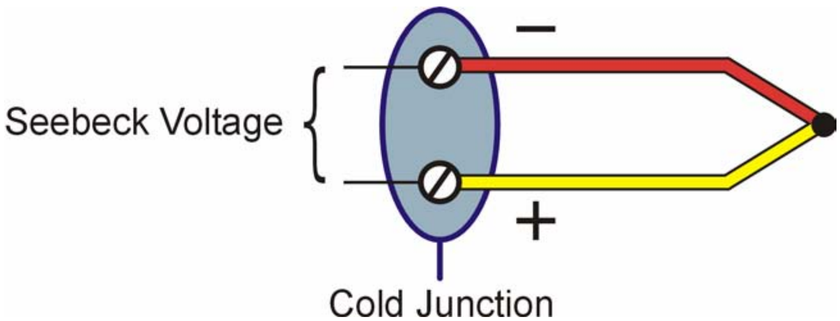

Figure 107.1: Thermocouple Connections

Lucky for us, technology is on our side. Dallas Semiconductor makes a neat little chip called the DS2760 which was actually designed for monitoring Lithium-Ion batteries, but works very nicely as a thermocouple interface. To the best of my knowledge, the use of the DS2760 as a thermocouple interface was originally presented by Dan Awtry of Dallas Semiconductor. What we're going to do this month is create a program for the BS2pe (or BS2p) that will talk to the DS2760 and display the thermocouple temperature in Celsius and Fahrenheit.

Temperature on a Wire

As you can see in Figure 107.2 the DS2760 is a one-chip solution for thermocouple interfacing. The BS2p/BS2pe makes talking to a 1-Wire device a snap; the rest is just assembling the code. Here's the plan.

- Measure the Cold Junction temperature (this comes from inside the DS2760)
- Measure the Seebeck voltage
- Find the thermocouple voltage that corresponds to the cold junction temperature
- Adjust the Seebeck voltage based on the cold junction temperature
- Look up the compensated temperature and display on LCD

All right ... you know the drill: We've planned our work, now let's work our plan.

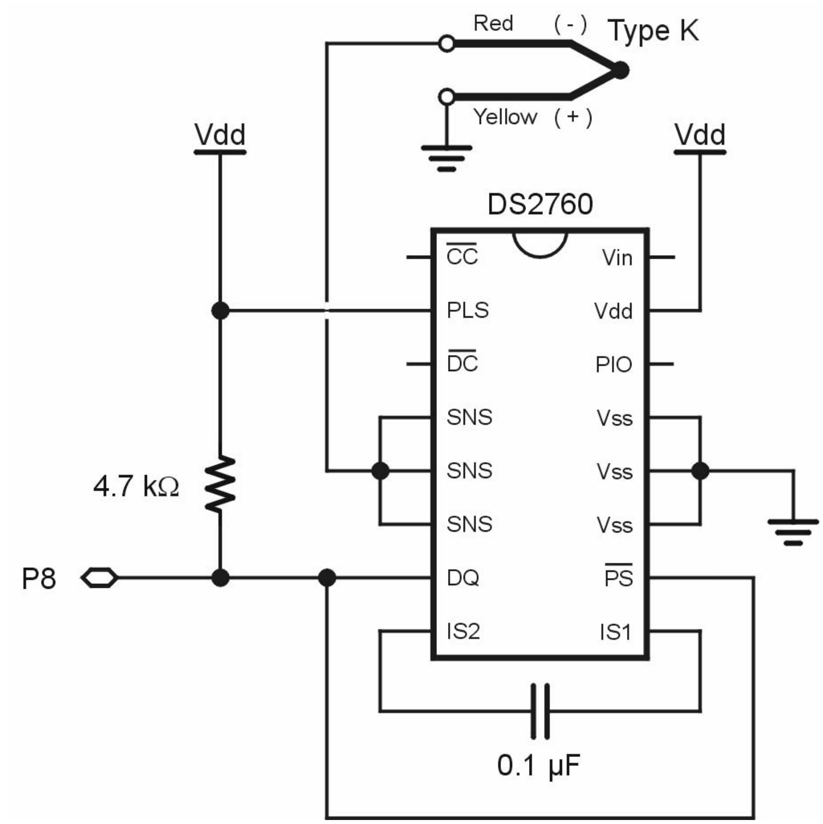

Figure 107.2: DS2760 Thermocouple Interface Schematic

Let's start at the top and make sure that we actually have a DS2760 connected. Note that this program is designed for just one sensor – it can be modified for multiple units but that's beyond the scope of what we're going to do here (You could, for example, put tables for various thermocouple types in different tables and select the type – you can download an example of this from Parallax). After checking to make sure that we're connected to a BS2p or BS2pe (required for 1-Wire communications), we initialize the 2x8 LCD and then retrieve the serial number from the 1-Wire device connected to P8.

```
Check_Device:
  OWOUT OW, %0001, [ReadNet]
  OWIN  OW, %0010, [SPSTR 8]
  GET 0, char
  IF (char <> $30) THEN
    LCDOUT E, LcdCls,   ["NO"]
    LCDOUT E, LcdLine2, ["  DS2760"]
    STOP
  ENDIF
```

We'll send the ReadNet ($33) command to the DS2760 using OWOUT, specifying a front-end reset (perform the reset process before sending data). ReadNet instructs the connected 1-Wire device to transmit its eight-byte serial number. Since we're not going to put the whole thing to use – but may want to display it later – we'll buffer it into the Scratchpad RAM using the SPSTR directive with OWIN. The first byte of the serial number string will be the device type; for the DS2760 this is $30. If that byte isn't $30 the program will put a message on the LCD and stop the program.

The reason we don't use END where STOP appears above is that END puts the Stamp into low-power mode. The Stamp's watchdog timer will interrupt the low-power mode every 18 milliseconds causing the pins to "glitch" (this is a known behavior). What I saw happen in testing is that the glitch on the LCD's E pin caused the display to be blanked, obliterating the message. STOP halts the program without placing the Stamp in low-power mode, so the IO pins remain in their current state without interruption.

Unless there's a problem we shouldn't get the "NO DS2760" message and we'll move right into the main loop that measures temperature using the process described earlier. The first step is to measure the cold junction temperature. This comes from inside the DS2760.

The temperature is read from registers $18 and $19. This value is 11 bits (10 bits plus sign), and interestingly, Bit10 (sign) is left-aligned with the MSB (Bit15) of our variable tmpCJ. Let's look at the code, and then go through it.

```
Read_CJ_Temp:
  OWOUT OW, %0001, [SkipNet, $69, $18]
  OWIN  OW, %0010, [tmpCJ.BYTE1, tmpCJ.BYTE0]
  IF (tmpCJ.BIT15) THEN
    tmpCJ = 0
    error = 1
  ELSE
    tmpCJ = tmpCJ.HIGHBYTE
    error = 0
  ENDIF
  RETURN
```

To retrieve the temperature we send the SkipNet ($CC) command (only one device is connected, so no serial number is required), followed by $69 (read), and then the register. Since 1-Wire devices work with bytes, our OWIN instruction breaks the tmpCJ variable into bytes using internal PBASIC aliases BYTE1 (high byte) and BYTE0 (low byte).

Remember that the temperature is left-aligned within tmpCJ, so the sign is currently sitting in Bit15. If this bit is one the temperature is negative. To keep things simple we will disallow negative cold junction temperatures (in theory it should be zero C, not lower) and set tmpCJ to zero and the error flag to one.

When the temperature is – as it will be in most cases – positive, we can convert the raw value to degrees by taking the high byte of the raw temperature. Yes, I know what you're thinking: "Huh?" Okay, here goes.... The raw value needs to be right shifted by five bits to correct the alignment. Okay, that's easy. Then we have to multiply by 0.125 to get whole degrees. As it turns out, 0.125 is a convenient fractional value because multiplying by 0.125 is the same as dividing by eight. And, as luck would have it, dividing by eight is the same thing as a right shift by three bits. So, in total, we have a right shift of eight bits which means that our whole degrees result is simply the high byte of the raw temperature value.

Let me interrupt this broadcast for a minute and talk about those "convenient fractional values." While the Stamp has operators (*/ and **) that can help with fractions there, are times when we don't need to take that route. In this case, for example, we could have used the */ operator with $40 to multiply by 0.125, but it's simpler to divide by eight. Now I admit, 0.125 is a common value and easy to recognize, but what about a value like 0.0769? Here's a tip: When in doubt about a fraction (that is less than one), enter it into your scientific calculator and then press the reciprocal [1/x] key. If the value is a whole number (or very very close), cha-ching! ... divide by the whole number. And if that value happens to be an even power of two (2, 4, 8, 16, 32 ...) then we can use the shift operator instead of divide since it's faster.

Okay, back to work. The next step is measuring the Seebeck voltage from the thermocouple. The process is identical to measuring temperature.

```
Read_TC_Volts:
  OWOUT OW, %0001, [SkipNet, $69, $0E]
  OWIN  OW, %0010, [tCuV.BYTE1, tCuV.BYTE0]
  signTC = tCuV.BIT15
  tCuV = tCuV >> 3
  IF signTC THEN
    tCuV = tCuV | $F000
  ENDIF
  tCuV = ABS tCuV */ 4000
  RETURN
```

The voltage is stored as a 13 bit (12 bits plus sign) value in the current registers ($0E and $0F) of the DS2760. The reason it's in the current register is that the DS2760 uses a shunt to convert a current to voltage for reading. In our application we're using the external sense resistor version of the DS2760 which lets us measure a voltage with a resolution of 15.625 microvolts per bit.

After retrieving the voltage into the variable tCuV, we save the sign by making a copy of bit 15. As with the temperature, the voltage is going to be left-aligned when in our word variable, and the sign bit is the MSB. After the sign is saved we correct the bit alignment in tCuV by shifting right three bits.

Now, if the sign bit is one, that means the voltage is negative and the value in tCuV is represented in two's-compliment format. Keep in mind that the shift process pads the opposite end with zeros (the high-end bits in this case), so we need to put ones in those positions to make the two's-compliment value of tCuV correct. This will let the ABS function return the right value.

The final step is to multiply by 15.625 to get microvolts. As the factor is fractional and greater than one, we'll uses the star-slash (*/) operator. The parameter for star-slash is calculated by multiplying 15.625 and 256.

Okay, we have the cold junction temperature and the Seebeck voltage; now it's time to do a bit of math and determine the actual thermocouple temperature.

Turning the Tables on Tough Math

A key element of this program is the use of large tables to hold the thermocouple data. The reason we use a table is that the thermocouple output voltage versus temperature is not linear and, in fact, would require a multi-order equation to maintain accuracy. One of my favorite features of the BS2p family is the ability to use READ and WRITE across program slots. This lets us put our code in slot zero, and our table(s) slots one and higher. The STORE instruction is used to select a table.

To compensate for a cold junction value above zero degrees Celsius, we'll determine the voltage that would be generated by that temperature for our thermocouple. This is simple; we point at our table with STORE, and then calculate the address within the table by multiplying our cold temperature value by two. This is necessary since we are using words (two bytes) to store the thermocouple output voltages.

```
STORE PosTable
READ (tmpCJ * 2), Word cjComp
```

Notice that we're taking advantage of a new PBASIC 2.5 feature: using the Word modifier with READ. The only caveat is that data must be stored in the table as low-byte, high-byte. This is not a problem for us as we're creating the table using the Word modifier. At this point, cjComp holds the cold junction compensation voltage for our thermocouple.

Now it's time to combine the compensation voltage with the Seebeck voltage. After we've done that, we can do a reverse lookup in the table to determine the thermocouple temperature.

```
IF (signTC = 0) THEN
  ' TC is above cold junction
  cjComp = cjComp + tCuV
  STORE PosTable
  tblHi = 1023
  signC = 0
ELSE
  ' TC is below cold junction
  IF (tCuV <= cjComp) THEN
    cjComp = cjComp - tCuV
    STORE PosTable
    tblHi = 1023
    signC = 0
```

```
    ELSE
      cjComp = tCuV - cjComp
      STORE NegTable
      tblHi = 270
      signC = 1
    ENDIF
  ENDIF
```

A bit of logic is used to do the combining for a couple of reasons. The first is that we've simplified other aspects of the code by maintaining a separate sign bit from the Seebeck voltage. The other reason is that we actually have two tables: one for positive temperatures (up to 1023 degrees C), one for negatives (down to -270 degrees C).

Things are easiest when the Seebeck voltage is positive. In this case we simply add the compensation voltage to the thermocouple voltage, and point to the positives table. We'll set the upper end of our table search to 1023 (this is the last Word in the table) and set the sign bit for Celsius degrees to zero since we know it's positive.

When the Seebeck voltage sign is one (voltage is negative) this indicates that the temperature is lower than the cold junction temperature, but we don't know if it is below zero Celsius, so we need to apply a bit of additional logic. If the Seebeck (absolute value) voltage is less than or equal to the compensation voltage, we can subtract it from the compensation voltage and point to the positive tables as before. When the Seebeck voltage is greater than the compensation voltage this means that the thermocouple temperature is below zero Celsius. We calculate the compensated voltage by subtracting the original compensation voltage from the Seebeck value, then pointing to the negatives table. Notice that the high end of the search for the negatives table is only 270. The reason for this, of course, is that absolute zero is 270 degrees Celsius. And since the temperature is negative we will set the Celsius sign bit accordingly.

Where In the Table is My Value?

With the compensated voltage (cjComp) in hand, all we have to do now is find that value – or it's closest match – in the table and that position will be our actual thermocouple temperature. Okay, how do we find it? One approach, the easiest, is to start at the bottom of the table and scan upward until we find a match or exceed our search value. The trouble with this method is that it can take a very long time to find a value that is in the high end of the table.

Searching large tables is nothing new, and we can borrow from computer science solutions. When the table is ordered, we can use what is called a binary search. This is a divide-andconquer approach to searching a table (or array). To do a binary search we'll need three pointers: the low end of the search, the high end of the search, and the midpoint. We find the midpoint by adding low and high together, then dividing by two. Then we'll check the value at the midpoint against our search value (cjComp). If we find a match we jump right out of the search. If the midpoint value is not a match we will compare it against the search value. When the search value is lower than the midpoint value, we'll reset the high end of the table search to the midpoint. If the search value is higher than the midpoint table value, we'll reset the low end of the search to the midpoint. As you can see, we get rid of half of the available search values with every iteration of the search loop. This makes the binary search very fast and lets us find any value within 10 loop iterations.

```
TC_Lookup:
  tblLo = 0
  tempC = 22

  READ (tblHi * 2), Word testVal
  IF (cjComp > testVal) THEN
    error = 1
  ELSE
    DO
      eePntr = (tblLo + tblHi) / 2
      READ (eePntr * 2), Word testVal

      IF (cjComp = testVal) THEN
        EXIT
      ELSEIF (cjComp < testVal) THEN
        tblHi = eePntr
      ELSE
        tblLo = eePntr
      ENDIF

      IF ((tblHi - tblLo) < 2) THEN
        eePntr = tblLo
        EXIT
      ENDIF
    LOOP
    tempC = eePntr
  ENDIF
  RETURN
```

Our code is actually modified a bit from the traditional binary search. In typical application, the search will report the position or "not found." We want the closest position if the actual value is not in the table. This is accomplished by monitoring the span between the high and low pointers. When it falls to one or zero, we've searched the whole table and we will use the low pointer as our search result.

Now that we have the correct Celsius temperature, we can convert to Fahrenheit and send the values to an LCD.

```
Display_Temps:
  IF (tempC = 0) THEN
    signC = 0
  ENDIF

  tempF = tempC * 9 / 5
  IF (signC) THEN
    tempF = 32 - tempF
  ELSE
    tempF = tempF + 32
  ENDIF
  signF = tempF.BIT15
  tempF = ABS tempF

  LOOKDOWN tempC, >= [1000, 100, 10, 0], idx
  LCDOUT E, LcdLine1, [223, "C ", REP " "\idx,
                      signC * 13 + 32, DEC tempC]

  LOOKDOWN tempF, >= [1000, 100, 10, 0], idx
  LCDOUT E, LcdLine2, [223, "F ", REP " "\idx,
                      signF * 13 + 32, DEC tempF]
```

Before we do the conversion well fix the sign bit for Celsius if required. There may be times when the temperature is just a hair below zero and the sign bit will get set. It's an easy fix.

There's no magic in converting from Celsius to Fahrenheit; we use the formula F = C x 9 / 5 + 32. As our program uses absolute values with a separate sign bit, an IF-THEN structure will take care of the "+ 32" part of the equation. And this actually points to another reason for using absolute values: the divide operator (required in the Fahrenheit conversion) cannot be used with two's-compliment (negative) values.

To keep things on the LCD neat, I use Tracy Allen's right justification trick with REP (repeat) modifier for serial output instructions (SEROUT, I2COUT, OWOUT, and LCDOUT [even though it uses a parallel buss]). A LOOKDOWN table is used to determine the width of our value, and then the width is used to pad the display with spaces ahead of the printed value. The sign bit is used with a bit of math to print a space for positive values and a hyphen for negatives. The DEC modifier finishes the process.

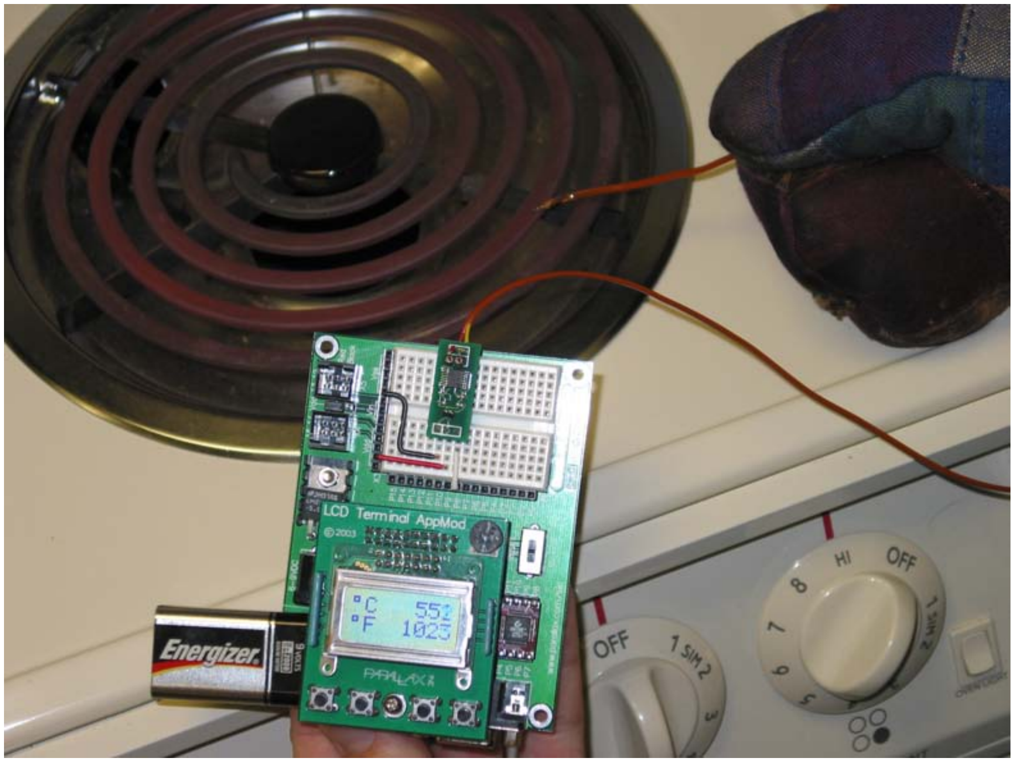

Figure 107.3: Don’t Touch – the Stove is Hot!

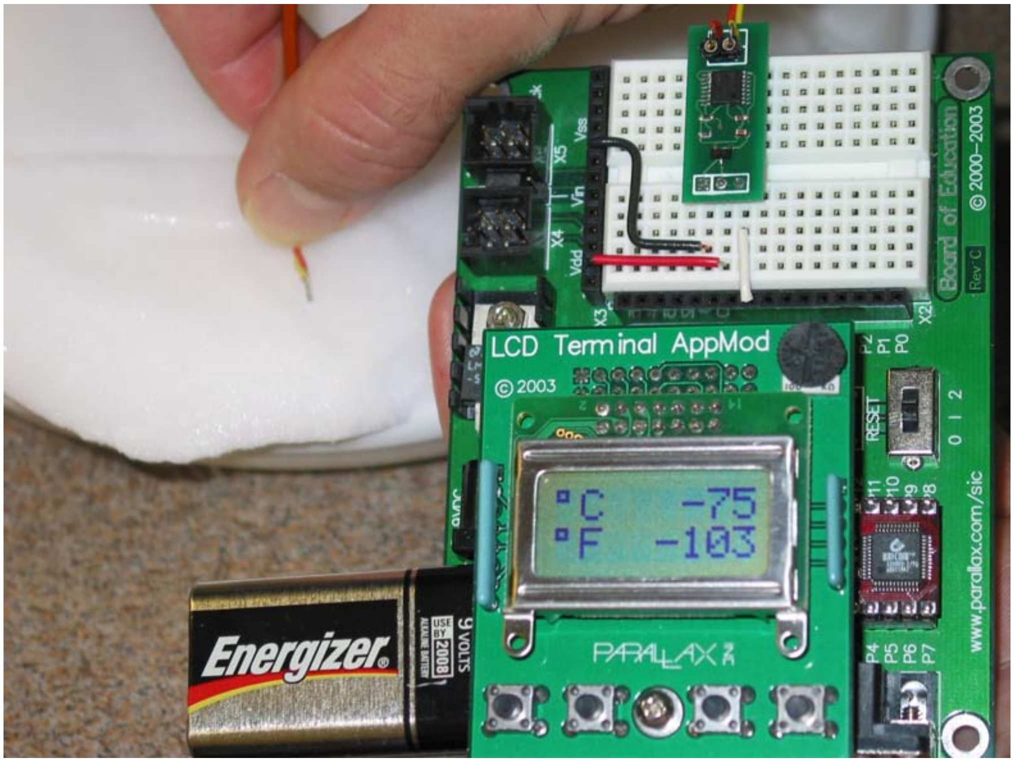

Figure 107.4: Dry Ice is Really Cold!

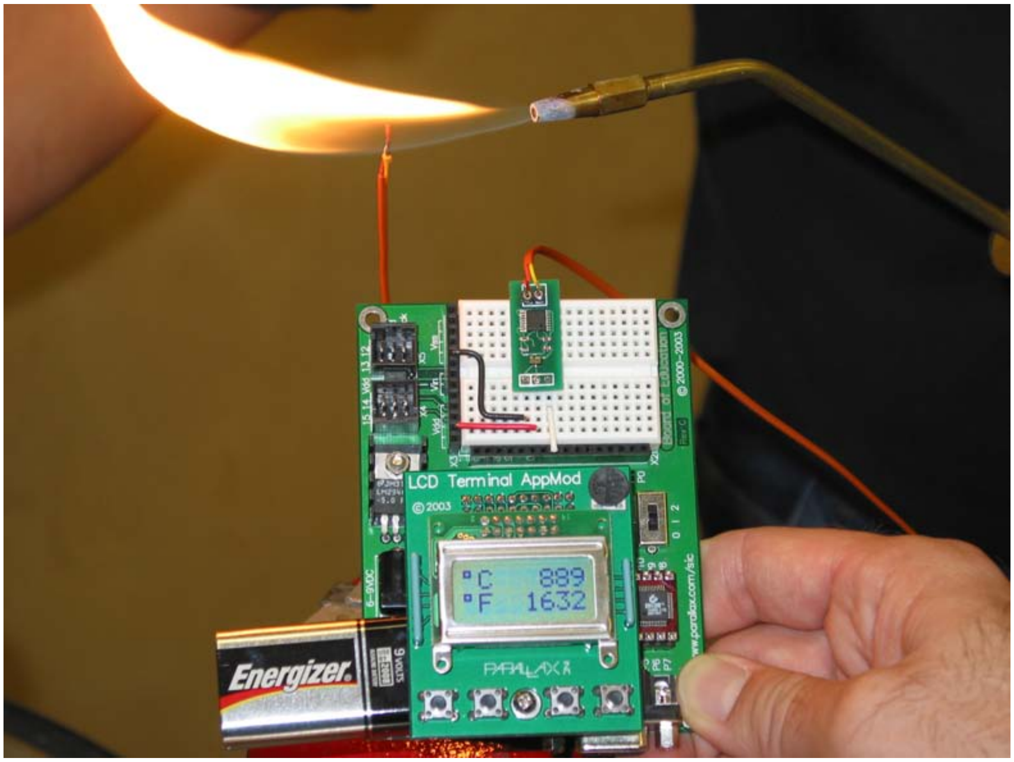

Figure 107.5: And torches get really hot!!!!

Temperature Hunting

As you can see in the photos, I assembled my test unit on a standard BOE. By using a nine volt battery I was able to roam around and test temperatures. My first spot of interest was the hot water coming out of the tap in my hotel room (I'm visiting the California office as I write this). How hot is it? A whopping 140 degrees Fahrenheit! That's hot. But I've got access to hotter things, like that burner on the stove: over 800 degrees. And what about measuring cold temperatures? I picked up some dry ice at the supermarket and measured it at around minus 100 degrees Fahrenheit. Wow, that is cold.

Please ... before you go off on your own temperature hunting expeditions, be aware that you can be burned by extreme heat or extreme cold (like dry ice). I had a friend take the photos for me so that I could focus on not getting too close to the "danger zone" with my hands.

Even if your thing isn't thermocouples or temperature measuring, I do hope that you found the use of tables interesting. After finishing this project, I thought of a couple other projects that could be simplified with a table, and I would get better resolution than using integer math. You can use your favorite PC programming language to calculate values and output your table (as text that can be copied into the Stamp editor). I'm currently experimenting with a very interesting multi-platform language called Python. Check it out, you might find it interesting and useful too.

Until next time, Happy Stamping.

```
' =========================================================================
'
'   File...... KTableNeg.BPE
'   Purpose... K-type (Chromel/Alumel) thermocouple data (0C reference)
'   Author.... Compiled by Jon Williams
'   E-mail.... jwilliams@parallax.com
'   Started...
'   Updated... 19 JAN 2004
'
'   {$STAMP BS2pe}
'   {$PBASIC 2.5}
'
' =========================================================================

' tC                 -0          -1          -2          -3          -4
'                    -5          -6          -7          -8          -9

Kn000   DATA    Word 00000, Word 00039, Word 00079, Word 00118, Word 00157,
                Word 00197, Word 00236, Word 00275, Word 00314, Word 00353

Kn010   DATA    Word 00392, Word 00431, Word 00470, Word 00508, Word 00547,
                Word 00586, Word 00624, Word 00663, Word 00701, Word 00739

Kn020   DATA    Word 00778, Word 00816, Word 00854, Word 00892, Word 00930,
                Word 00968, Word 01006, Word 01043, Word 01081, Word 01119

Kn030   DATA    Word 01156, Word 01194, Word 01231, Word 01268, Word 01305,
                Word 01343, Word 01380, Word 01417, Word 01453, Word 01490

Kn040   DATA    Word 01527, Word 01564, Word 01600, Word 01637, Word 01673,
                Word 01709, Word 01745, Word 01782, Word 01818, Word 01854

Kn050   DATA    Word 01889, Word 01925, Word 01961, Word 01996, Word 02032,
                Word 02067, Word 02103, Word 02138, Word 02173, Word 02208

Kn060   DATA    Word 02243, Word 02278, Word 02312, Word 02347, Word 02382,
                Word 02416, Word 02450, Word 02485, Word 02519, Word 02553

Kn070   DATA    Word 02587, Word 02620, Word 02654, Word 02688, Word 02721,
                Word 02755, Word 02788, Word 02821, Word 02854, Word 02887

Kn080   DATA    Word 02920, Word 02953, Word 02986, Word 03018, Word 03050,
                Word 03083, Word 03115, Word 03147, Word 03179, Word 03211

Kn090   DATA    Word 03243, Word 03274, Word 03306, Word 03337, Word 03368,
                Word 03400, Word 03431, Word 03462, Word 03492, Word 03523

Kn100   DATA    Word 03554, Word 03584, Word 03614, Word 03645, Word 03675,
                Word 03705, Word 03734, Word 03764, Word 03794, Word 03823
```

```
Kn110   DATA    Word 03852, Word 03882, Word 03911, Word 03939, Word 03968,
                Word 03997, Word 04025, Word 04054, Word 04082, Word 04110

Kn120   DATA    Word 04138, Word 04166, Word 04194, Word 04221, Word 04249,
                Word 04276, Word 04303, Word 04330, Word 04357, Word 04384

Kn130   DATA    Word 04411, Word 04437, Word 04463, Word 04490, Word 04516,
                Word 04542, Word 04567, Word 04593, Word 04618, Word 04644

Kn140   DATA    Word 04669, Word 04694, Word 04719, Word 04744, Word 04768,
                Word 04793, Word 04817, Word 04841, Word 04865, Word 04889

Kn150   DATA    Word 04913, Word 04936, Word 04960, Word 04983, Word 05006,
                Word 05029, Word 05052, Word 05074, Word 05097, Word 05119

Kn160   DATA    Word 05141, Word 05163, Word 05185, Word 05207, Word 05228,
                Word 05250, Word 05271, Word 05292, Word 05313, Word 05333

Kn170   DATA    Word 05354, Word 05374, Word 05395, Word 05415, Word 05435,
                Word 05454, Word 05474, Word 05493, Word 05512, Word 05531

Kn180   DATA    Word 05550, Word 05569, Word 05588, Word 05606, Word 05624,
                Word 05642, Word 05660, Word 05678, Word 05695, Word 05713

Kn190   DATA    Word 05730, Word 05747, Word 05763, Word 05780, Word 05797,
                Word 05813, Word 05829, Word 05845, Word 05861, Word 05876

Kn200   DATA    Word 05891, Word 05907, Word 05922, Word 05936, Word 05951,
                Word 05965, Word 05980, Word 05994, Word 06007, Word 06021

Kn210   DATA    Word 06035, Word 06048, Word 06061, Word 06074, Word 06087,
                Word 06099, Word 06111, Word 06123, Word 06135, Word 06147

Kn220   DATA    Word 06158, Word 06170, Word 06181, Word 06192, Word 06202,
                Word 06213, Word 06223, Word 06233, Word 06243, Word 06252

Kn230   DATA    Word 06262, Word 06271, Word 06280, Word 06289, Word 06297,
                Word 06306, Word 06314, Word 06322, Word 06329, Word 06337

Kn240   DATA    Word 06344, Word 06351, Word 06358, Word 06364, Word 06370,
                Word 06377, Word 06382, Word 06388, Word 06393, Word 06399

Kn250   DATA    Word 06404, Word 06408, Word 06413, Word 06417, Word 06421,
                Word 06425, Word 06429, Word 06432, Word 06435, Word 06441

Kn260   DATA    Word 06441, Word 06444, Word 06446, Word 06448, Word 06450,
                Word 06452, Word 06453, Word 06455, Word 06456, Word 06457

Kn270   DATA    Word 06458
```

```
' =========================================================================
'
'   File...... KTablePos.BPE
'   Purpose... K-type (Chromel/Alumel) thermocouple data (0C reference)
'   Author.... Compiled by Chuck Gracey and Jon Williams
'   E-mail.... jwilliams@parallax.com
'   Started...
'   Updated... 19 JAN 2004
'
'   {$STAMP BS2pe}
'   {$PBASIC 2.5}
'
' =========================================================================

' tC                 +0          +1          +2          +3          +4
'                    +5          +6          +7          +8          +9

K0000   DATA    Word 00000, Word 00039, Word 00079, Word 00119, Word 00158,
                Word 00198, Word 00238, Word 00277, Word 00317, Word 00357

K0010   DATA    Word 00397, Word 00437, Word 00477, Word 00517, Word 00557,
                Word 00597, Word 00637, Word 00677, Word 00718, Word 00758

K0020   DATA    Word 00798, Word 00838, Word 00879, Word 00919, Word 00960,
                Word 01000, Word 01040, Word 01080, Word 01122, Word 01163

K0030   DATA    Word 01203, Word 01244, Word 01284, Word 01326, Word 01366,
                Word 01407, Word 01448, Word 01489, Word 01530, Word 01570

K0040   DATA    Word 01612, Word 01653, Word 01694, Word 01735, Word 01776,
                Word 01816, Word 01858, Word 01899, Word 01941, Word 01982

K0050   DATA    Word 02023, Word 02064, Word 02105, Word 02146, Word 02188,
                Word 02230, Word 02270, Word 02311, Word 02354, Word 02395

K0060   DATA    Word 02436, Word 02478, Word 02519, Word 02560, Word 02601,
                Word 02644, Word 02685, Word 02726, Word 02767, Word 02810

K0070   DATA    Word 02850, Word 02892, Word 02934, Word 02976, Word 03016,
                Word 03059, Word 03100, Word 03141, Word 03184, Word 03225

K0080   DATA    Word 03266, Word 03307, Word 03350, Word 03391, Word 03432,
                Word 03474, Word 03516, Word 03557, Word 03599, Word 03640

K0090   DATA    Word 03681, Word 03722, Word 03765, Word 03806, Word 03847,
                Word 03888, Word 03931, Word 03972, Word 04012, Word 04054

K0100   DATA    Word 04096, Word 04137, Word 04179, Word 04219, Word 04261,
                Word 04303, Word 04344, Word 04384, Word 04426, Word 04468

K0110   DATA    Word 04509, Word 04549, Word 04591, Word 04633, Word 04674,
```

```
                Word 04714, Word 04756, Word 04796, Word 04838, Word 04878

K0120   DATA    Word 04919, Word 04961, Word 05001, Word 05043, Word 05083,
                Word 05123, Word 05165, Word 05206, Word 05246, Word 05288

K0130   DATA    Word 05328, Word 05368, Word 05410, Word 05450, Word 05490,
                Word 05532, Word 05572, Word 05613, Word 05652, Word 05693

K0140   DATA    Word 05735, Word 05775, Word 05815, Word 05865, Word 05895,
                Word 05937, Word 05977, Word 06017, Word 06057, Word 06097

K0150   DATA    Word 06137, Word 06179, Word 06219, Word 06259, Word 06299,
                Word 06339, Word 06379, Word 06419, Word 06459, Word 06500

K0160   DATA    Word 06540, Word 06580, Word 06620, Word 06660, Word 06700,
                Word 06740, Word 06780, Word 06820, Word 06860, Word 06900

K0170   DATA    Word 06940, Word 06980, Word 07020, Word 07059, Word 07099,
                Word 07139, Word 07179, Word 07219, Word 07259, Word 07299

K0180   DATA    Word 07339, Word 07379, Word 07420, Word 07459, Word 07500,
                Word 07540, Word 07578, Word 07618, Word 07658, Word 07698

K0190   DATA    Word 07738, Word 07778, Word 07819, Word 07859, Word 07899,
                Word 07939, Word 07979, Word 08019, Word 08058, Word 08099

K0200   DATA    Word 08137, Word 08178, Word 08217, Word 08257, Word 08298,
                Word 08337, Word 08378, Word 08417, Word 08458, Word 08499

K0210   DATA    Word 08538, Word 08579, Word 08618, Word 08659, Word 08698,
                Word 08739, Word 08778, Word 08819, Word 08859, Word 08900

K0220   DATA    Word 08939, Word 08980, Word 09019, Word 09060, Word 09101,
                Word 09141, Word 09180, Word 09221, Word 09262, Word 09301

K0230   DATA    Word 09343, Word 09382, Word 09423, Word 09464, Word 09503,
                Word 09544, Word 09585, Word 09625, Word 09666, Word 09707

K0240   DATA    Word 09746, Word 09788, Word 09827, Word 09868, Word 09909,
                Word 09949, Word 09990, Word 10031, Word 10071, Word 10112

K0250   DATA    Word 10153, Word 10194, Word 10234, Word 10275, Word 10316,
                Word 10356, Word 10397, Word 10439, Word 10480, Word 10519

K0260   DATA    Word 10560, Word 10602, Word 10643, Word 10683, Word 10724,
                Word 10766, Word 10807, Word 10848, Word 10888, Word 10929

K0270   DATA    Word 10971, Word 11012, Word 11053, Word 11093, Word 11134,
                Word 11176, Word 11217, Word 11259, Word 11300, Word 11340

K0280   DATA    Word 11381, Word 11423, Word 11464, Word 11506, Word 11547,
```

```
                Word 11587, Word 11630, Word 11670, Word 11711, Word 11753

K0290   DATA    Word 11794, Word 11836, Word 11877, Word 11919, Word 11960,
                Word 12001, Word 12043, Word 12084, Word 12126, Word 12167

K0300   DATA    Word 12208, Word 12250, Word 12291, Word 12333, Word 12374,
                Word 12416, Word 12457, Word 12499, Word 12539, Word 12582

K0310   DATA    Word 12624, Word 12664, Word 12707, Word 12747, Word 12789,
                Word 12830, Word 12872, Word 12914, Word 12955, Word 12997

K0320   DATA    Word 13039, Word 13060, Word 13122, Word 13164, Word 13205,
                Word 13247, Word 13289, Word 13330, Word 13372, Word 13414

K0330   DATA    Word 13457, Word 13497, Word 13539, Word 13582, Word 13624,
                Word 13664, Word 13707, Word 13749, Word 13791, Word 13833

K0340   DATA    Word 13874, Word 13916, Word 13958, Word 14000, Word 14041,
                Word 14083, Word 14125, Word 14166, Word 14208, Word 14250

K0350   DATA    Word 14292, Word 14335, Word 14377, Word 14419, Word 14461,
                Word 14503, Word 14545, Word 14586, Word 14628, Word 14670

K0360   DATA    Word 14712, Word 14755, Word 14797, Word 14839, Word 14881,
                Word 14923, Word 14964, Word 15006, Word 15048, Word 15090

K0370   DATA    Word 15132, Word 15175, Word 15217, Word 15259, Word 15301,
                Word 15343, Word 15384, Word 15426, Word 15468, Word 15510

K0380   DATA    Word 15554, Word 15596, Word 15637, Word 15679, Word 15721,
                Word 15763, Word 15805, Word 15849, Word 15891, Word 15932

K0390   DATA    Word 15974, Word 16016, Word 16059, Word 16102, Word 16143,
                Word 16185, Word 16228, Word 16269, Word 16312, Word 16355

K0400   DATA    Word 16396, Word 16439, Word 16481, Word 16524, Word 16565,
                Word 16608, Word 16650, Word 16693, Word 16734, Word 16777

K0410   DATA    Word 16820, Word 16861, Word 16903, Word 16946, Word 16989,
                Word 17030, Word 17074, Word 17115, Word 17158, Word 17201

K0420   DATA    Word 17242, Word 17285, Word 17327, Word 17370, Word 17413,
                Word 17454, Word 17496, Word 17539, Word 17582, Word 17623

K0430   DATA    Word 17667, Word 17708, Word 17751, Word 17794, Word 17836,
                Word 17879, Word 17920, Word 17963, Word 18006, Word 18048

K0440   DATA    Word 18091, Word 18134, Word 18176, Word 18217, Word 18260,
                Word 18303, Word 18346, Word 18388, Word 18431, Word 18472

K0450   DATA    Word 18515, Word 18557, Word 18600, Word 18643, Word 18686,
```

```
                 Word 18728, Word 18771, Word 18812, Word 18856, Word 18897

K0460    DATA    Word 18940, Word 18983, Word 19025, Word 19068, Word 19111,
                 Word 19153, Word 19196, Word 19239, Word 19280, Word 19324

K0470    DATA    Word 19365, Word 19408, Word 19451, Word 19493, Word 19536,
                 Word 19579, Word 19621, Word 19664, Word 19707, Word 19750

K0480    DATA    Word 19792, Word 19835, Word 19876, Word 19920, Word 19961,
                 Word 20004, Word 20047, Word 20089, Word 20132, Word 20175

K0490    DATA    Word 20218, Word 20260, Word 20303, Word 20346, Word 20388,
                 Word 20431, Word 20474, Word 20515, Word 20559, Word 20602

K0500    DATA    Word 20643, Word 20687, Word 20730, Word 20771, Word 20815,
                 Word 20856, Word 20899, Word 20943, Word 20984, Word 21027

K0510    DATA    Word 21071, Word 21112, Word 21155, Word 21199, Word 21240,
                 Word 21283, Word 21326, Word 21368, Word 21411, Word 21454

K0520    DATA    Word 21497, Word 21540, Word 21582, Word 21625, Word 21668,
                 Word 21710, Word 21753, Word 21795, Word 21838, Word 21881

K0530    DATA    Word 21923, Word 21966, Word 22009, Word 22051, Word 22094,
                 Word 22137, Word 22178, Word 22222, Word 22265, Word 22306

K0540    DATA    Word 22350, Word 22393, Word 22434, Word 22478, Word 22521,
                 Word 22562, Word 22606, Word 22649, Word 22690, Word 22734

K0550    DATA    Word 22775, Word 22818, Word 22861, Word 22903, Word 22946,
                 Word 22989, Word 23032, Word 23074, Word 23117, Word 23160

K0560    DATA    Word 23202, Word 23245, Word 23288, Word 23330, Word 23373,
                 Word 23416, Word 23457, Word 23501, Word 23544, Word 23585

K0570    DATA    Word 23629, Word 23670, Word 23713, Word 23757, Word 23798,
                 Word 23841, Word 23884, Word 23926, Word 23969, Word 24012

K0580    DATA    Word 24054, Word 24097, Word 24140, Word 24181, Word 24225,
                 Word 24266, Word 24309, Word 24353, Word 24394, Word 24437

K0590    DATA    Word 24480, Word 24523, Word 24565, Word 24608, Word 24650,
                 Word 24693, Word 24735, Word 24777, Word 24820, Word 24863

K0600    DATA    Word 24905, Word 24948, Word 24990, Word 25033, Word 25075,
                 Word 25118, Word 25160, Word 25203, Word 25245, Word 25288

K0610    DATA    Word 25329, Word 25373, Word 25414, Word 25457, Word 25500,
                 Word 25542, Word 25585, Word 25626, Word 25670, Word 25711

K0620    DATA    Word 25755, Word 25797, Word 25840, Word 25882, Word 25924,
```

```
                Word 25967, Word 26009, Word 26052, Word 26094, Word 26136

K0630   DATA    Word 26178, Word 26221, Word 26263, Word 26306, Word 26347,
                Word 26390, Word 26432, Word 26475, Word 26516, Word 26559

K0640   DATA    Word 26602, Word 26643, Word 26687, Word 26728, Word 26771,
                Word 26814, Word 26856, Word 26897, Word 26940, Word 26983

K0650   DATA    Word 27024, Word 27067, Word 27109, Word 27152, Word 27193,
                Word 27236, Word 27277, Word 27320, Word 27362, Word 27405

K0660   DATA    Word 27447, Word 27489, Word 27531, Word 27574, Word 27616,
                Word 27658, Word 27700, Word 27742, Word 27784, Word 27826

K0670   DATA    Word 27868, Word 27911, Word 27952, Word 27995, Word 28036,
                Word 28079, Word 28120, Word 28163, Word 28204, Word 28246

K0680   DATA    Word 28289, Word 28332, Word 28373, Word 28416, Word 28416,
                Word 28457, Word 28500, Word 28583, Word 28626, Word 28667

K0690   DATA    Word 28710, Word 28752, Word 28794, Word 28835, Word 28877,
                Word 28919, Word 28961, Word 29003, Word 29045, Word 29087

K0700   DATA    Word 29129, Word 29170, Word 29213, Word 29254, Word 29297,
                Word 29338, Word 29379, Word 29422, Word 29463, Word 29506

K0710   DATA    Word 29548, Word 29589, Word 29631, Word 29673, Word 29715,
                Word 29757, Word 29798, Word 29840, Word 29882, Word 29923

K0720   DATA    Word 29964, Word 30007, Word 30048, Word 30089, Word 30132,
                Word 30173, Word 30214, Word 30257, Word 30298, Word 30341

K0730   DATA    Word 30382, Word 30423, Word 30466, Word 30507, Word 30548,
                Word 30589, Word 30632, Word 30673, Word 30714, Word 30757

K0740   DATA    Word 30797, Word 30839, Word 30881, Word 30922, Word 30963,
                Word 31006, Word 31047, Word 31088, Word 31129, Word 31172

K0750   DATA    Word 31213, Word 31254, Word 31295, Word 31338, Word 31379,
                Word 31420, Word 31461, Word 31504, Word 31545, Word 31585

K0760   DATA    Word 31628, Word 31669, Word 31710, Word 31751, Word 31792,
                Word 31833, Word 31876, Word 31917, Word 31957, Word 32000

K0770   DATA    Word 32040, Word 32082, Word 32124, Word 32164, Word 32206,
                Word 32246, Word 32289, Word 32329, Word 32371, Word 32411

K0780   DATA    Word 32453, Word 32495, Word 32536, Word 32577, Word 32618,
                Word 32659, Word 32700, Word 32742, Word 32783, Word 32824

K0790   DATA    Word 32865, Word 32905, Word 32947, Word 32987, Word 33029,
```

```
                Word 33070, Word 33110, Word 33152, Word 33192, Word 33234

K0800   DATA    Word 33274, Word 33316, Word 33356, Word 33398, Word 33439,
                Word 33479, Word 33521, Word 33561, Word 33603, Word 33643

K0810   DATA    Word 33685, Word 33725, Word 33767, Word 33807, Word 33847,
                Word 33889, Word 33929, Word 33970, Word 34012, Word 34052

K0820   DATA    Word 34093, Word 34134, Word 34174, Word 34216, Word 34256,
                Word 34296, Word 34338, Word 34378, Word 34420, Word 34460

K0830   DATA    Word 34500, Word 34542, Word 34582, Word 34622, Word 34664,
                Word 34704, Word 34744, Word 34786, Word 34826, Word 34866

K0840   DATA    Word 34908, Word 34948, Word 34999, Word 35029, Word 35070,
                Word 35109, Word 35151, Word 35192, Word 35231, Word 35273

K0850   DATA    Word 35313, Word 35353, Word 35393, Word 35435, Word 35475,
                Word 35515, Word 35555, Word 35595, Word 35637, Word 35676

K0860   DATA    Word 35718, Word 35758, Word 35798, Word 35839, Word 35879,
                Word 35920, Word 35960, Word 36000, Word 36041, Word 36081

K0870   DATA    Word 36121, Word 36162, Word 36202, Word 36242, Word 36282,
                Word 36323, Word 36363, Word 36403, Word 36443, Word 36484

K0880   DATA    Word 36524, Word 36564, Word 36603, Word 36643, Word 36685,
                Word 36725, Word 36765, Word 36804, Word 36844, Word 36886

K0890   DATA    Word 36924, Word 36965, Word 37006, Word 37045, Word 37085,
                Word 37125, Word 37165, Word 37206, Word 37246, Word 37286

K0900   DATA    Word 37326, Word 37366, Word 37406, Word 37446, Word 37486,
                Word 37526, Word 37566, Word 37606, Word 37646, Word 37686

K0910   DATA    Word 37725, Word 37765, Word 37805, Word 37845, Word 37885,
                Word 37925, Word 37965, Word 38005, Word 38044, Word 38084

K0920   DATA    Word 38124, Word 38164, Word 38204, Word 38243, Word 38283,
                Word 38323, Word 38363, Word 38402, Word 38442, Word 38482

K0930   DATA    Word 38521, Word 38561, Word 38600, Word 38640, Word 38679,
                Word 38719, Word 38759, Word 38798, Word 38838, Word 38878

K0940   DATA    Word 38917, Word 38957, Word 38996, Word 39036, Word 39076,
                Word 39115, Word 39164, Word 39195, Word 39234, Word 39274

K0950   DATA    Word 39314, Word 39353, Word 39393, Word 39432, Word 39470,
                Word 39511, Word 39549, Word 39590, Word 39628, Word 39668

K0960   DATA    Word 39707, Word 39746, Word 39786, Word 39826, Word 39865,
```

```
                Word 39905, Word 39944, Word 39984, Word 40023, Word 40061

K0970   DATA    Word 40100, Word 40140, Word 40179, Word 40219, Word 40259,
                Word 40298, Word 40337, Word 40375, Word 40414, Word 40454

K0980   DATA    Word 40493, Word 40533, Word 40572, Word 40610, Word 40651,
                Word 40689, Word 40728, Word 40765, Word 40807, Word 40846

K0990   DATA    Word 40885, Word 40924, Word 40963, Word 41002, Word 41042,
                Word 41081, Word 41119, Word 41158, Word 41198, Word 41237

K1000   DATA    Word 41276, Word 41315, Word 41354, Word 41393, Word 41431,
                Word 41470, Word 41509, Word 41548, Word 41587, Word 41626

K1010   DATA    Word 41665, Word 41704, Word 41743, Word 41781, Word 41820,
                Word 41859, Word 41898, Word 41937, Word 41976, Word 42014

K1020   DATA    Word 42053, Word 42092, Word 42131, Word 42169
```

```
' =========================================================================
'
'   File...... Thermo-K.BPE
'   Purpose... Type-K Thermocouple temperature measurement using the DS2760
'   Author.... Jon Williams
'   E-mail.... jwilliams@parallax.com
'   Started...
'   Updated... 19 JAN 2004
'
'   {$STAMP BS2pe, KTablePos.BPE, KTableNeg.BPE}
'   {$PBASIC 2.5}
'
' =========================================================================


' -----[ Program Description ]---------------------------------------------
'
' This program lets a BS2p or BS2pe read the temperature from the Parallax
' DS2760 thermocouple module.  User interface is through the Parallax LCD
' AppMod.  This program uses separate tables for positive and negative
' temperatures allowing for a wide range of measurement values.


' -----[ Revision History ]------------------------------------------------


' -----[ I/O Definitions ]-------------------------------------------------

E               PIN     1                       ' LCD Enable (1 = enabled)
RW              PIN     2                       ' Read/Write\
RS              PIN     3                       ' Reg Select (1 = char)
LcdDirs         VAR     DIRB                    ' dirs for I/O redirection
LcdBusOut       VAR     OUTB
LcdBusIn        VAR     INB

OW              PIN     8                       ' 1-Wire buss pin


' -----[ Constants ]-------------------------------------------------------

LcdCls          CON     $01                     ' clear the LCD
LcdHome         CON     $02                     ' move cursor home
LcdCrsrL        CON     $10                     ' move cursor left
LcdCrsrR        CON     $14                     ' move cursor right
LcdDispL        CON     $18                     ' shift chars left
LcdDispR        CON     $1C                     ' shift chars right

LcdDDRam        CON     $80                     ' Display Data RAM control
LcdCGRam        CON     $40                     ' Custom character RAM
LcdLine1        CON     $80                     ' DDRAM address of line 1
LcdLine2        CON     $C0                     ' DDRAM address of line 2
```

```
BtnUp           CON     0                       ' for AppMod buttons
BtnDn           CON     1

ReadNet         CON     $33                     ' read OW net address
SkipNet         CON     $CC                     ' skip OW net address
RdReg           CON     $69                     ' read register

PosTable        CON     1                       ' slot for postives table
NegTable        CON     2                       ' slot for negative table


#DEFINE _DebugOn = 1                            ' show data on DEBUG window


' -----[ Variables ]-------------------------------------------------------

idx             VAR     Nib                     ' loop counter
char            VAR     Byte                    ' for display

buttons         VAR     Nib                     ' LCD AppMod buttons
btn1            VAR     buttons.BIT0
btn2            VAR     buttons.BIT1
btn3            VAR     buttons.BIT2
btn4            VAR     buttons.BIT3

vIn             VAR     Word                    ' DS2760 voltage input
tmpCJ           VAR     Word                    ' cold junction temp in C
tCuV            VAR     Word                    ' thermocouple millivolts
signTC          VAR     Word                    ' TC sign bit

cjComp          VAR     Word                    ' temp compensation
tempC           VAR     Word                    ' temp in Celsius
signC           VAR     Bit
tempF           VAR     Word                    ' temp in Fahrenheit
signF           VAR     Bit

tblLo           VAR     Word                    ' table pointers
tblHi           VAR     Word
eePntr          VAR     Word
testVal         VAR     Word                    ' test value from table
error           VAR     Bit                     ' 1 = out of range


' -----[ EEPROM Data ]-----------------------------------------------------


' -----[ Initialization ]--------------------------------------------------

Stamp_Check:
  #IF ($stamp < BS2P) #THEN
```

```
    #ERROR "This program requires BS2p or BS2pe"
  #ENDIF

Setup:
  DIRL = %11111110                              ' setup pins for LCD

LCD_Init:
  PAUSE 500                                     ' let LCD settle
  LCDCMD E, %00110000 : PAUSE 5                 ' 8-bit mode
  LCDCMD E, %00110000 : PAUSE 0
  LCDCMD E, %00110000 : PAUSE 0
  LCDCMD E, %00100000 : PAUSE 0                 ' 4-bit mode
  LCDCMD E, %00101000 : PAUSE 0                 ' 2-line mode
  LCDCMD E, %00001100 : PAUSE 0                 ' no crsr, no blink
  LCDCMD E, %00000110                           ' inc crsr, no disp shift


' -----[ Program Code ]----------------------------------------------------

Intro:
  LCDOUT E, LcdCls, ["THERMO-K"]                ' splash

  #IF _DebugOn #THEN
    DEBUG CLS, "Thermo-K", CR
  #ENDIF

  PAUSE 1500

Check_Device:
  OWOUT OW, %0001, [ReadNet]                    ' get serial number
  OWIN  OW, %0010, [SPSTR 8]                    ' store in SPRAM
  GET 0, char                                   ' read OW device type
  IF (char <> $30) THEN                         ' if not $30, wrong device
    LCDOUT E, LcdCls,   ["NO"]                  ' display error message
    LCDOUT E, LcdLine2, ["  DS2760"]

    #IF _DebugOn #THEN
      DEBUG CLS, "No DS2760 found."
    #ENDIF

    STOP                                        ' stop program
  ENDIF

Main:
  DO
    GOSUB Read_TC_Volts                         ' read Seebeck voltage
    GOSUB Read_CJ_Temp                          ' read cold junction temp
    STORE PosTable
    READ (tmpCJ * 2), Word cjComp               ' get compensation voltage

    ' combine cjComp and tCuV
```

```
  '
  IF (signTC = 0) THEN
    ' TC is above cold junction
    cjComp = cjComp + tCuV
    STORE PosTable
    tblHi = 1023
    signC = 0
  ELSE
    ' TC is below cold junction
    IF (tCuV <= cjComp) THEN
      cjComp = cjComp - tCuV
      STORE PosTable
      tblHi = 1023
      signC = 0
    ELSE
      cjComp = tCuV - cjComp
      STORE NegTable
      tblHi = 270                             ' absolute zero
      signC = 1
    ENDIF
  ENDIF

  GOSUB TC_Lookup                             ' get temp via table search

  IF error THEN
    GOSUB Display_OOR                         ' out of range
  ELSE
    GOSUB Display_Temps                       ' put temps on LCD
  ENDIF

  #IF _DebugOn #THEN                          ' program report
    DEBUG HOME,
          "DS2760 Demo", CR,
          CLREOL, CR,
          "tmpCJ  ", DEC tmpCJ, " C", CLREOL, CR,
          "tCuV   ", signTC * 13 + 32, DEC tCuV, " uV", CLREOL, CR,
          CLREOL, CR

    IF error THEN
      DEBUG "tC     ?", CLREOL, CR
      DEBUG "tF     ?", CLREOL, CR
    ELSE
      DEBUG "tC     ", (signC * 13 + 32), DEC tempC, CLREOL, CR
      DEBUG "tF     ", (signF * 13 + 32), DEC tempF, CLREOL, CR
    ENDIF
  #ENDIF

  PAUSE 100
LOOP
END
```

```
' -----[ Subroutines ]-----------------------------------------------------

' Read and debounce the LCD AppMod buttons

LCD_Get_Buttons:
  LcdDirs = %0000                               ' make LCD bus inputs
  buttons = %1111                               ' assume all pressed
  FOR idx = 1 TO 10
    buttons = buttons & LcdBusIn                ' make sure button held
    PAUSE 5                                     ' debounce 10 x 5 ms
  NEXT
  LcdDirs = %1111                               ' return bus to outputs
  RETURN


' Reads device input voltage (Vin pin)
' -- mV in millivolts (max reading is 4.75 volts)

Read_Vin:
  OWOUT OW, %0001, [SkipNet, RdReg, $0C]
  OWIN  OW, %0010, [vIn.BYTE1, vIn.BYTE0]
  IF (vIn.BIT15) THEN                           ' check sign
    vIn = 0                                     ' disallow negative
  ELSE
    vIn = vIn >> 5 */ $4E1                      ' x 4.88 millivolts
  ENDIF
  RETURN


' Reads current register to get TC voltage
' -- each raw bit = 15.625 uV
' -- tCuV in microvolts

Read_TC_Volts:
  OWOUT OW, %0001, [SkipNet, RdReg, $0E]
  OWIN  OW, %0010, [tCuV.BYTE1, tCuV.BYTE0]
  signTC = tCuV.BIT15                           ' save sign bit
  tCuV = tCuV >> 3                              ' correct alignment
  IF signTC THEN
    tCuV = tCuV | $F000                         ' pad 2's-compliment bits
  ENDIF
  tCuV = ABS tCuV */ 4000                       ' x 15.625 uV
  RETURN


' Reads cold junction (device) temperature
' -- each raw bit = 0.125 degrees C
' -- returns tmpCJ in whole degrees C

Read_CJ_Temp:
```

```
  OWOUT OW, %0001, [SkipNet, RdReg, $18]
  OWIN  OW, %0010, [tmpCJ.BYTE1, tmpCJ.BYTE0]
  IF (tmpCJ.BIT15) THEN                         ' check sign
    tmpCJ = 0                                   ' disallow negative
    error = 1                                   ' flag temp too low
  ELSE
    tmpCJ = tmpCJ.HIGHBYTE                      ' >> 5 x 0.125 (>> 3)
    error = 0
  ENDIF
  RETURN


' Search currently selected table for nearest entry
' -- uses modified binary search algorithm to find cjComp
' -- high end of search set before calling (tblHi)
' -- successful search sets tempC

TC_Lookup:
  tblLo = 0                                     ' low entry of table
  tempC = 22                                    ' default to room temp

  READ (tblHi * 2), Word testVal                ' check max temp
  IF (cjComp > testVal) THEN
    error = 1                                   ' out of range
  ELSE
    DO
      eePntr = (tblLo + tblHi) / 2              ' midpoint of search span
      READ (eePntr * 2), Word testVal           ' read value from midpoint

      IF (cjComp = testVal) THEN
        EXIT                                    ' found it!
      ELSEIF (cjComp < testVal) THEN
        tblHi = eePntr                          ' search lower half
      ELSE
        tblLo = eePntr                          ' search upper half
      ENDIF

      IF ((tblHi - tblLo) < 2) THEN             ' span at minimum
        eePntr = tblLo
        EXIT
      ENDIF
    LOOP
    tempC = eePntr
  ENDIF
  RETURN


Display_OOR:
  LCDOUT E, LcdLine1, [" OUT OF "]
  LCDOUT E, LcdLine2, [" RANGE! "]
  RETURN
```

```
Display_Temps:
  IF (tempC = 0) THEN
    signC = 0                                  ' fix sign error if needed
  ENDIF

  ' calculate Fahrenheit

  tempF = tempC * 9 / 5
  IF (signC) THEN                              ' C temp is negative
    tempF = 32 - tempF
  ELSE
    tempF = tempF + 32
  ENDIF
  signF = tempF.BIT15                          ' save sign
  tempF = ABS tempF                            ' work with absolute value

  ' send temps to LCD

  LOOKDOWN tempC, >= [1000, 100, 10, 0], idx
  LCDOUT E, LcdLine1, [223, "C ", REP " "\idx,
                      signC * 13 + 32, DEC tempC]

  LOOKDOWN tempF, >= [1000, 100, 10, 0], idx
  LCDOUT E, LcdLine2, [223, "F ", REP " "\idx,
                      signF * 13 + 32, DEC tempF]

  RETURN
```