Application 3: Interfacing a Temperature Sensor to the PRIMER

Purpose:

To expose the student to rudimentary analog interface techniques.

Goals:

1. Build and test a simple temperature sensing circuit.
2. Load a program that will make use of the temperature sensor's output.
3. Calibrate the sensor and software to provide a temperature reading in approximate engineering units.
4. Control a simple process with temperature.

Materials required:

1) PRIMER trainer
1) fahrenheit thermometer
1) hair dryer
(A digital voltmeter may also prove helpful if available)

Component Description	**DIGI-KEY part number**
1) LM3911 Temperature controller	LM3911N
1) 100 uF 16 volt electrolytic capacitor	P5030
1) .001 uF ceramic disc capacitor	DH102
1) 8 pin soldertail dip socket	A9308
1) 10 Kohm 10 turn potentiometer	
1) 100 ohm 5% 1/4 watt resistor	
1) 12 Kohm 5% 1/4 watt resistor	
1) 15 Kohm 5% 1/4 watt resistor	
1) 22 Kohm 5% 1/4 watt resistor	
1) 82 Kohm 5% 1/4 watt resistor	
1) 82 Kohm 5% 1/4 watt resistor	
1) 1 Megohm 5% 1/4 watt resistor	
1) 1x2 inch chunk of perfboard	

The electronic components listed above may be ordered from DIGI-KEY$^{(R)}$ by phone by dialing 1-800-344-4539. They may also be found at electronic supply stores and other mail order houses.

Circuit Description:

The temperature sensing circuit used here is centered around the LM3911 chip. The LM3911 is a special purpose IC that is a temperature to voltage TRANSDUCER. The design of the LM3911 is extraordinarily flexible, and it may be used as a stand alone temperature controller as well. The chip has an op-amp, and internal voltage regulator, as well as the actual temperature sensing element. The LM3911 may be used as an accurate and rugged temperature sensor over the range of -25 to +85 degrees centigrade.

The LM3911 senses temperature by amplifying the voltage differential at the base-emitter junctions of two identical transistors, that are operating at different currents, with the same temperature applied to them. As the junction temperature changes, the curve of base-emitter voltage vs. temperature will differ between the two transistors, because they are operating at different currents. This differential would normally be a problem in conventional circuitry, but is taken advantage of here. The differential voltage is amplified by the LM3911, and presented to the input of an operational amplifier within the chip.

The output voltage change of the temperature sensing element is 10 millivolts per degree KELVIN. The kelvin temperature scale is somewhat equivalent to the CENTIGRADE scale, but the kelvin scale begins at ABSOLUTE ZERO.

Each degree kelvin is the same as one degree centigrade, but the scales start at different absolute temperatures. Zero degrees kelvin is -273 degrees centigrade, therefore, 0 degrees centigrade is +273 kelvin. As one degree change kelvin produces 10 millivolts change in the sensor output of the LM3911, and one degree centigrade is one degree kelvin, the LM3911 output will change 10 millivolts per degree centigrade as well. Only the baseline of each scale is different.

Although kelvin and centigrade are equivalent (for this application), fahrenheit degrees are entirely different. Both the scale shift, and the scale " gain" are different. Standard conversion formulas are used to convert centigrade to fahrenheit and vice versa. As nine fahrenheit degrees pass for 5 centigrade degrees (5/9 plus the 32 fahrenheit scale shift), each degree fahrenheit will produce an eighteen (18) millivolt change per degree fahrenheit.

The sensor element itself is internally tied directly to the non-inverting input of an op-amp built inside the LM3911. This op-amp may be wired as a direct buffer, which will pass the 10 millivolt per degree kelvin output directly to the outside world. It may also be configured as an amplifier with voltage gain, to permit greater voltage output for any given temperature change, or as a logic comparator, which will provide an on/off response to a temperature setpoint.

Also included in the chip is a 6.8 volt zener diode, to permit the LM3911 to provide it's own stabilized VCC supply. For this circuit application, the op-amp has been configured for approximately an 11 times gain, so a small temperature change will provide a larger output, which is around 100 millivolts per degree centigrade. Since the circuit shown here operates at 5 volts, the internal zener regulator will not be used. The comparator function is also not used.

Referring to the schematic, the LM3911 temperature sensor chip, U1, is powered by the 5 volt VCC supply of the PRIMER, which comes from the header connector plugged onto the analog port pins. As there is a lot of digital system noise present on the PRIMER VCC supply, an R-C decoupling network, consisting of R1 (100 ohm), and C1 (100 uF), forms a simple low pass filter that removes most of the power supply noise. This noise must be removed, else it will be amplified within the LM3911, and passed onto the analog output, which will cause erratic and unstable readings.

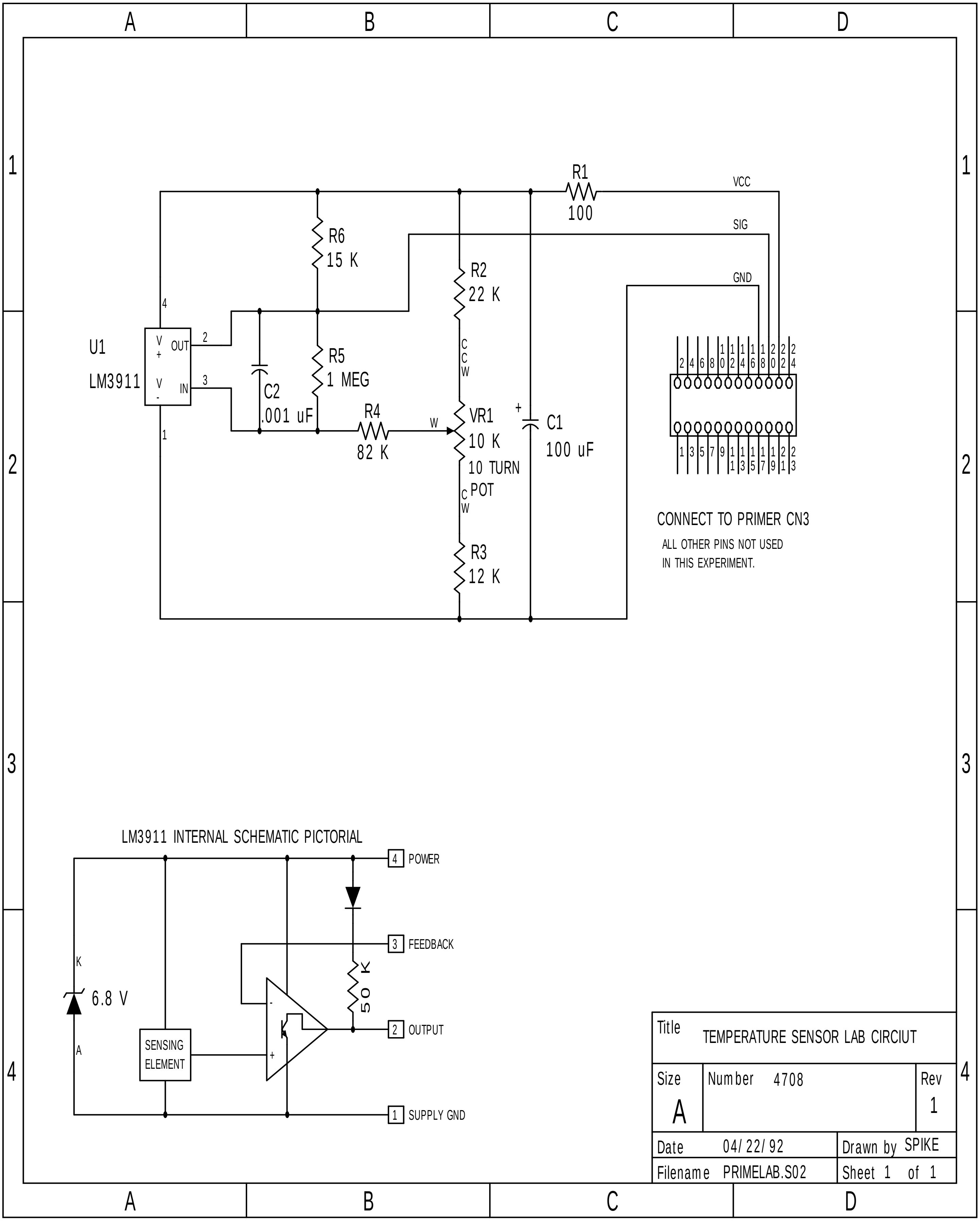
The voltage drop of R1 is negligible, due to the very low current consumption of the circuit. The amplifier gain is set by the ratio of resistances R5 and R4. Potentiometer VR1, and resistors R2 and R3 form a voltage divider that taps a reference voltage from the filtered VCC supply fed to the circuit. This tapped voltage goes to the inverting input of the op-amp in the LM3911. This voltage tap is adjustable, and is used to shift the op-amp output up/down scale. The gain of the circuit changes slightly as the shift pot, VR1 is adjusted. The parallel equivalent resistance of VR1, R2, and R3, is approximately 10.4 K when the pot is set at the center of its range. This equivalent resistance of 10.4 K, in series with the 82 K in R4, feeds 92.4 K into the inverting input node from this branch.

Since feedback resistor R5, which is one megohm, is approximately 10 times the 92.4 K fed by the R4 branch, the resistor ratio is roughly 10 to 1. The gain of a non-inverting op-amp circuit of this configuration is the ratio of feedback +1, so the total gain is roughly 11. Therefore, for each 10 millivolts output change produced by a one degree kelvin change at the sensor, the output voltage will swing 110 millivolts. This high voltage gain, while producing a increased change in voltage per change in temperature, slightly limits the temperature range that can be measured.

Resistor R6 is used to help the op-amp with the positive output swing. Refer to the schematic, and observe the internal schematic pictorial of the LM3911 itself. The op-amp has an open collector transistor output. A 50 K resistor and diode provide pullup for positive output swings. This 50 K does not provide enough current to drive the PRIMER's 100 K input sufficiently high, so an external resistor, R6, provides extra pullup power.

Capacitor C2 provides stabilization feedback for high frequency stability of the op-amp circuit, by rolling off the frequency response well below that frequency where the op-amp becomes unstable with the gain such as it is. The internal op-amp is not frequency compensated, and with all the stray capacitances present on this circuit, the op-amp sees some pretty shaky "poles".

The output voltage from the LM3911 and accompanying circuit is then fed to the PRIMER analog input via the connector header.



Procedure:

The temperature circuit should be built on perfboard, and connected to the PRIMER's analog port connector header. The circuit may be connected by wire-wrapping, soldering or by using a female connector. The circuit will draw power from the PRIMER, and feed its analog output to the PRIMER. Carefully check the wiring of the circuit, and be sure it is properly connected to the PRIMER.

HINT: Allow the circuit to thoroughly cool after soldering and handling. A surprising amount of residual heat will remain in the LM3911 package, which will deter attempts to adjust the setpoint correctly. If you set VR1, and the reading slowly drifts down, (lower temperature) it is probably due to this effect.

Load the following program into memory:

```
         ; This program shows the fahrenheit temperature in the
         ; left four displays
leds     equ      11h      ; output port for digital output LEDs
adcin    equ      9        ; ADCIN service number
leddec   equ      13h      ; LEDDEC service number
mult     equ      7        ; MULT service number
div      equ      8        ; DIV service number
mos      equ      1000h    ; address of MOS services
adjst    equ      123      ; #of fahrenheit degrees * 100 per
                           ; change in value returned from ADCIN

         org      0ff01h
loop:    mvi      c,adcin
         call     mos      ; get the digital value of analog input voltage
         mvi      h,0
         lda      mxanlg   ; maximum analog value (this may be different on
                           ;   other PRIMERs, or with different temp sensors)
         sub      l        ; invert the analog conversion
         mov      l,a      ; HL = analog value
         lxi      d,adjst  ; load D with the adjustment factor
         mvi      c,mult
         call     mos      ; DE = HL * DE
         xchg              ; HL = DE
         lxi      d,100
         mvi      c,div
         call     mos      ; divide HL by 100
         lda      basetmp  ; get the base temperature
         add      l        ; now A is the actual temperature
         mov      e,a      ; E = temperature
         mov      a,e      ; A = temperature
         lhld     lotemp   ; L = low temp limit, H=high temp limit
         cmp      l        ; see if analog value is below L
         jnc      chkhi    ; check the high value if not
         mvi      a,0
         out      leds     ; turn on LEDs
chkhi:   mov      a,e      ; A = temperature
         cmp      h
         jc       noled    ; if A<H then don't turn off LEDs
         mvi      a,0FFh
         out      leds     ; H > = A so turn off LEDs
noled:   mvi      d,0      ; clear D register
         mvi      c,leddec
         call     mos      ; display the temp in DE
         jmp      loop     ; read it again

mxanlg:  ds       1        ; max analog value given by temp sensor
basetmp: ds       1        ; base temperature
lotemp:  ds       1        ; lower limit temperature
hitemp:  ds       1        ; upper limit temperature
         end
```

ADDRESS	**DATA**	**INSTRUCTION**
FF01	0E	MVI C,09
FF02	09	
FF03	CD	CALL 1000
FF04	00	
FF05	10	
FF06	26	MVI H,00
FF07	00	
FF08	3A	LDA FF42
FF09	42	

```
FF0A         FF
FF0B         95          SUB  L
FF0C         6F          MOV  L,A
FF0D         11          LXI D,007B
FF0E         7B
FF0F         00
FF10         0E          MVI  C,07
FF11         07
FF12         CD          CALL 1000
FF13         00
FF14         10
FF15         EB          XCHG
FF16         11          LXI  D,0064
FF17         64
FF18         00
FF19         0E          MVI  C,08
FF1A         08
FF1B         CD          CALL 1000
FF1C         00
FF1D         10
FF1E         3A          LDA  FF43
FF1F         43
FF20         FF
FF21         85          ADD  L
FF22         5F          MOV  E,A
FF23         7B          MOV  A,E
```

```
ADDRESS      DATA        INSTRUCTION
FF24         2A          LHLD FF44
FF25         44
FF26         FF
FF27         BD          CMP  L
FF28         D2          JNC  FF2F
FF29         2F
FF2A         FF
FF2B         3E          MVI  A,0
FF2C         00
FF2D         D3          OUT  11
FF2E         11
FF2F         7B          MOV  A,E
FF30         BC          CMP  H
FF31         DA          JC   FF38
FF32         38
FF33         FF
FF34         3E          MVI  A,FF
FF35         FF
FF36         D3          OUT  11
FF37         11
FF38         16          MVI  D,00
FF39         00
FF3A         0E          MVI  C,13
FF3B         13
FF3C         CD          CALL 1000
FF3D         00
FF3E         10
FF3F         C3          JMP  FF01
FF40         01
FF41         FF
FF42         3F          (max analog val)

FF43         00          (base temp data)
FF44         5A          (lo temp limit)
FF45         64          (hi temp limit)
```

After loading in the program, you must calibrate the temperature sensor circuit and the program. Start the program running at FF01 and observe the left four numeric output LEDs. A decimal number should be displayed there. With a small screwdriver, turn the potentiometer (VR1) clockwise. If after 20 turns the output hasn't changed, turn VR1 counterclockwise for 20 turns (VR1 has mechanical stops that don't care if you turn them too many times). Adjust VR1 until the value on the display is as low as it can go. As soon as the value on the display stops decreasing, stop turning VR1. Subtract the value that is on the displays from 64 (decimal), stop the program then convert that value to hexadecimal and store it at FF42. Since the value returned by the A/D convertor decreases as the temperature increases, it is subtracted from the maximum value the A/D convertor can produce (normally 63 decimal) thereby inverting the value. The temperature sensor, though, does not produce the 5 volts required to give the maximum value, and for this reason the value at FF42 must be changed.

Now check the temperature of the sensor using a thermometer and convert this value to hex and store it at FF43. This is the base temperature. If you start the program at FF01 again, the base temperature (or within 1 or 2 degrees of it) will be shown on the displays. Heat up the sensor with the hair dryer and you will see that when the displayed temperature reaches 100 degrees the digital output LEDs turn off. Let the sensor cool down to below 90 degrees and they will turn on again. It is possible for the digital output connector (connected to the digital output LEDs) to control external devices such as fans or heaters, if you know how to build relay drivers that will turn such devices on and off (do not attempt this if you are not proficient in electronics). If a fan is connected to the output connector, the program can turn on the fan when the temperature reaches 100 degrees and turn it off when the temperature drops below 90 degrees. Likewise, if a heater is connected, the program can turn on the heater when the temperature drops below 90 and turn it off when the temperature reaches 100 degrees.

You may be wondering by now why the program is written in such a way as to turn the LEDs on at one temperature and turn them off at another. This is done to keep the output device from rapidly oscillating on and off. Rapid oscillation is fine when dealing with LEDs but it can be destructive to relays. This technique of using different turn on and turn off temperatures is commonly used in environment control systems. To see what would happen if there was one turn on and turn off temperature, store 5A at address FF45 and run the program. Heat up the sensor to 89 degrees and while watching the digital output LEDs, slowly heat the sensor to 90 degrees. You should see that as the temperature approaches 90 degrees the LEDs will start to oscillate rapidly for a moment (the LEDs may appear to dim) until the temperature is stable at 90 degrees.

Program Description:

The program reads the analog to digital convertor and then inverts the value that was returned from it so that as the temperature increases, the value will increase. This value is then scaled to provide an accurate fahrenheit temperature. It was found through experimentation, that a change of 69 degrees from the base temperature causes the A/D convertor value to change by 56 decimal. This means that for each change in A/D convertor value there is a 69/56 or 1.23 degree change in the temperature. Since MOS only does integer math, a trick had to be used to perform floating point math. The inverted A/D convertor value was multiplied by 123 and then the product was divided by 100 which effectively scaled the value by 1.23 and removed the tenths and hundredths digits. After the A/D convertor value is converted to fahrenheit, the base temperature is added to it to give the actual value. After this, it is compared to the low and high temperature values. If the temperature is below the low temperature value, zero is sent to the port for the digital ouput LEDs (which causes them to turn on), and if the temperature is at the high temperature limit, FF hex is sent the to the port (which turns the LEDs off). Finally the temperature is displayed on the left 4 displays and the program starts all over again.