In-Class Exercise: Thermocouple Conversion SubVI

As discussed in class, thermocouples are temperature measurement devices that output a low-level voltage proportional to the temperature at the sensing junction. Once the voltage has been cold-junction compensated, it can be converted to engineering units of temperature according to the various inverse polynomial functions outlined in Monograph 175 from the National Institutes of Standards and Technology (NIST), available at http://srdata.nist.gov/its90/main. Virtually all manufacturers of thermocouples and thermocouple measurement devices reference the NIST data for conversion purposes. For instance, Omega Engineering publishes the same thermocouple data in a more polished format, available online at http://www.omega.com/temperature/z/pdf/z198-201.pdf. Here is an excerpt from that reference for the K-type thermocouple:

Type K Thermocouples - coefficients of approximate inverse functions giving temperature, t_{90}, as a function of the thermoelectric voltage, E, in selected temperature and voltage ranges. The functions are of the form:

$$t_{90} = c_0 + c_1E + c_2E^2 \quad c_iE^i$$

where E is in microvolts and t_{90} is in degrees Celsius.

Temperature Range:	-200 to 0°C	0 to 500°C	500 to 1,372°C
Voltage Range:	-5891 to 0 µV	0 to 20,644 µV	20,644 to 54,886 µV
c_0 =	0.000 000 0	0.000 000	-1.318 058 x 10^{2}
c_1 =	2.517 346 2 x 10^{-2}	2.508 355 x 10^{-2}	4.830 222 x 10^{-2}
c_2 =	-1.166 287 8 x 10^{-6}	7.860 106 x 10^{-8}	-1.646 031 x 10^{-6}
c_3 =	-1.083 363 8 x 10^{-9}	-2.503 131 x 10^{-10}	5.464 731 x 10^{-11}
c_4 =	-8.977 354 0 x 10^{-13}	8.315 270 x 10^{-14}	-9.650 715 x 10^{-16}
c_5 =	-3.734 237 7 x 10^{-16}	-1.228 034 x 10^{-17}	8.802 193 x 10^{-21}
c_6 =	-8.663 264 3 x 10^{-20}	9.804 036 x 10^{-22}	-3.110 810 x 10^{-26}
c_7 =	-1.045 059 8 x 10^{-23}	-4.413 030 x 10^{-26}	
c_8 =	-5.192 057 7 x 10^{-28}	1.057 734 x 10^{-30}	
c_9 =		-1.052 755 x 10^{-35}	
Error Range:	0.04°C to -0.02°C	0.04°C to -0.05°C	0.06°C to -0.05°C

With this information, you are to create a LabVIEW subVI which will accept an input K-type thermocouple voltage (assumed compensated), and output the corresponding temperature that was measured. Additional requirements and notes follow.

- The inverse polynomials assume that the measured thermocouple voltage is in units of microvolts, yet it is likely that the values supplied to your subVI from the preceding data acquisition routine will be in plain volts. Therefore, you should somehow indicate to the user that he/she should supply a measured value in volts, and you should convert that value to microvolts in your subVI before passing it into the polynomial functions.

- Note that the excerpt for the K-type thermocouple identifies three different ranges for temperature and corresponding voltage, each of which has its own coefficients for the inverse polynomial. Your subVI should automatically determine which range to apply based upon the supplied input voltage, and use the appropriate coefficients.
- Provide an input to the subVI whereby the user can select which engineering units of temperature in which they wish the output to be expressed. Note that the polynomial functions output the value in degrees Celsius, so you will have to provide a means of converting to the other standard choices. These include degrees Fahrenheit, degrees Rankine, and Kelvin. If you do not know how all of these units are related, you can easily find their conversions online.
- Configure the temperature units input as an *enum* control, with the four basic choices identified (the *enum* control is an *enumerated list*). It is available under the *Modern→Ring & Enum* subpalette, and is configured by right-clicking and selecting *Edit Items…*
- For any scenario where an error exists, make the temperature output display “NaN,” which you can achieve by conditionally wiring to it the square root of negative one.
- Remember that a fully correct subVI must have its connector pane established, with error handlers, an appropriate icon, and documentation that will display in the context help window. Like all VI’s, yours should also have appropriate default values established for all front-panel controls and indicators. Also, it is generally good practice in all LabVIEW programs to include documentation of the code, and to follow the recommended style guidelines (straight wires, left-to-right flow, etc.).