The sensor reads in degrees C

The thermistor circuit is accurate to within +/- 0.1 C or better when read with a 16bit ADC like the ADS1115, but still very accurate with a 12bit ADC.

We only use the Murata thermistor, the TDK coeffs were left over from a teaching project; it can be deleted.

2500  is the mV excitation of the thermistor circuit, its hard wired into the sensor.

The routine for measuring the thermistor is:

Apply voltage to the sensor, 3 to 6 v

delay(100)

Read the voltage on Vout from the thermistor using your microcontrollers ADC

Typically your result is in volts, but the function I sent you uses mV as an input. So you see me multiply the result by 1000 to get mV

If you are reading millivolts from your ADC, then don’t mult by 1000

Also, the statement logRt = log(Rt); needs to calculate the natural log  of the resistance, NOT log base 10.

Finally, a good way to test your function is to put in a value for Vout that is half the excitation or 1250 mV in our case, this should give you a Temperature of 25C.

Here is the function I run on our boron:

// convert to thermistor resist to temp with steinhart hart

float thermistorCalc(float Vo, float Vex) {

  // vout in mV,  circuit excitation in mV

  float logRt,Rt,T;

  float R = 10000.0;            // fixed resistor, 10K

  // c1, c2, c3 are calibration coefficients for a particular thermistor

  //float c1 = 0.89771073E-03, c2=2.4993536E-4, c3 = 1.9750764E-7; // TDK NTCG163JF103

  float c1 = 0.901747748E-03, c2=2.489190310E-04, c3 = 2.043213857E-07; // Murata NCP18XH103F03RB

  //Rt=R\*(Vex/Vo-1);         // calc thermistor resistance,base resistor next to gnd

  Rt=R/(Vex/Vo-1);           // calc thermistor resistance, therm next to gnd

  if (Rt>0) {

    logRt = log(Rt);           //calc log of R

    T = ( 1.0 / (c1 + c2\*logRt + c3\*logRt\*logRt\*logRt ) ) - 273.15; // Steinhart Hart, 3nd order

  } else

  T=-99.9;

  return T;

}

Here is where I calc voltage and then temp on 5 sensors.

for (byte i=0; i<5; i++)   {                           // convert to volts and temps here

    current.voltSoil[i]= (3300.0\*rawSoil[i]) / 4095.0;           // convert to millivolts here

    voltTemp[i]= (3300.0\*rawTemp[i]) / 4095.0;

    current.soilTemp[i] = thermistorCalc(voltTemp[i], 2500.0); // calc temperature

  }

Note my, rawTempl[i] variable is the average of 32 subsamples  - for example

for (byte i=0; i<=soilSubSamples; i++)   {                 // average readings from adc channel A2, temperature V

    int rawADC = analogRead(A2);

    myRA.addValue(rawADC);

  }

  rawTemp[0]=myRA.getAverage();

  myRA.clear();

I use

#include "RunningAverage.h"