AN-401: USING RTD'S WITH THE INTELLILOGGER

A resistance temperature detector (RTD) is a temperature sensing element commonly used in temperature measurement applications demanding high absolute accuracy. The RTD's resistance is a function of its temperature, meaning that if you measure the resistance, the temperature can then be calculated. Most RTD's are manufactured with platinum film or wire as the sensing element. This element is potted into a protective package more suitable for the final application.

This application note provides a background on RTD's as well as details on using RTD's with the IntelliLogger portable data acquisition and reporting instrument from Logic Beach Incorporated. Actual wired connection of RTD's to the IntelliLogger is explained and multiple RTD resistance to temperature conversion methods are detailed with advantages and disadvantages presented. The Summary section at the end of this Application Note provides guidance for method to employ for specific applications.

RTD CHARACTERISTICS:

RTD's come in a variety of materials, packages and types with various specifications.

MATERIAL CHARACTERISTICS

RTDs are specified by the material used for their resistive element and by their resistance at 0°C. The most common RTD's have resistances at 0°C of 100Ω, 500Ω, 1000Ω, or 2000Ω. They are commonly made from platinum, nickel, copper, tungsten, silver, and gold. Alloys of these metals are also used but platinum is by far the most popular for general use applications due to its stability and resistance characteristics.

RTD's are Positive Temperature Coefficient devices meaning that as the temperature increases, the resistance of the element also increases. Although much more linear in its temperature to output (resistance change in the case of an RTD) relationship than many other temperature sensing elements such as thermocouples or thermistors, the RTD is still a non-linear device.

Platinum RTD's elements have a typical operating temperature range of -250 to +650°C. Note that packaged RTD sensing probes and devices which have added materials (insulation, potting material, package) may not be usable over this temperature range and manufacturer's specifications should be followed.

ALPHA COEFFICIENT (TCR)

An additional specification that is used to classify an RTD is its alpha (α) coefficient… also commonly referred to as Temperature Coefficient of Resistance (TCR). The alpha coefficient represents the average resistance to temperature relationship over the temperature span of 0 to 100°C and is expressed as ohm/ohm/0°C. The larger this number is, the more sensitive the resistance of the RTD is to temperature changes. By multiplying this number by 100, one can see the approximate change in resistance per degree Celsius (if $\alpha = 0.00385$ then the resistance will change by around 0.385Ω per degree of temperature change). This conversion is approximate due to the non-linearity of the resistance to temperature characteristics of the RTD as mentioned above.

Platinum RTD's with an alpha of 0.00385 are high accuracy parts and are commonly used in general temperature measurement applications. The 0.00385 alpha platinum is a standard material as defined in DIN43760 and IEC751. Extremely pure, 99.999% (and hence costly) platinum is used in Standards Laboratories and has an alpha of 0.003926. Additionally, there are many more purity variations of platinum used for RTD fabrication.

NON-LINEARITY

Because the RTD resistance to temperature function is non-linear, if accurate conversions from resistance to temperature are to be made over a wide range, the resistance vs temperature relationship cannot be modeled with a simple straight line. As an alternative to linear modeling, the Callendar-Van Dusen equation can be used. This equation is a 4^{th} degree polynomial based on the RTD's resistance at 0°C, 100°C, and 260°C, and takes into account, the non-linearity of the function. This function is relatively simple for platinum RTDs, as platinum has a resistance to temperature curve that is very close to being linear. The CALLENDAR-VAN DUSEN equation is readily programmed into the IntelliLogger as will be shown later.

Alternatively, if the RTD is to be used over a fairly narrow temperature range and if slightly inaccurate readings are acceptable, then a linear interpolation can be used. This method effectively finds a "best-fit line" for the curve and approximates the temperature from the measured resistance using an mX+b linear conversion. The linear approximation method is easily programmed into the IntelliLogger and can meet many RTD application accuracy requirements.

RTD POTENTIAL MEASUREMENT ERRORS

Various factors must be considered in the use of RTD's for accurate temperature measurement.

SELF-HEATING ERRORS

To measure the resistance of an RTD, a current is passed through the RTD and the corresponding voltage developed across the RTD is then measured and used for the temperature calculation. The excitation current passing through the resistive element results in power dissipation and "self-heating" of the element which will introduce measurement errors. For utmost accuracy, the power dissipation must be kept to a minimum in RTD measurement.

Manufacturers typically provide a specification indicating the RTD temperature rise per unit of power dissipation (eg 0.5°C/mW) under specified conditions (commonly specified for free air).

Self-heating is affected by the application. For example, if the RTD is in an insertion probe in a liquid filled pipe, the self-heating error is minimized due to efficient heat transfer to the heat conductive liquid which serves as a heat sink. Alternatively, if the RTD is suspended in still air, then the heat will transfer to the air more slowly due to the less efficient heat transfer to air resulting in larger self-heating errors. In the use of RTD's, the manufacturer's self-heating specifications, the thermal environment, and the desired measurement accuracy must be taken into consideration to achieve desired performance.

WIRING RESISTANCE ERRORS

Since the RTD resistance is the changing parameter in temperature indication, any added resistance in series with the RTD sensor will affect the perceived temperature. Hence, another potential source of inaccuracy in RTD measurement is due to the additive resistance developed in the wiring between the measurement instrument and the RTD sensing element.

In order to measure the resistance of an RTD (or any resistive component) with the IntelliLogger (or any meter instrument) a known "excitation" current is passed through the RTD and the voltage developed across it is measured. The resistance is then calculated per Ohm's law which states that the voltage across an element divided by the current through the element equals the resistance of the element… which is commonly written V / I = R; where V = Voltage, I = Current and R = Resistance

2-Wire Configuration

The simplest wiring configuration for an RTD is the use of two wires between the instrument and the RTD (Figure 1). In this configuration the resistance of the wiring to and returning from the RTD would be measured and included in the perceived RTD resistance value, effectively resulting in temperature measurement errors.

Resistance in wire is a function of the wire cross-sectional area (i.e. the wire gauge) and the length. Tables are readily available online that give the ohms/foot for various wire gauges.

For example, if a 2-wire configuration was used to connect an IntelliLogger to an RTD that was 50' away, a total of 100' of wire resistance would be measured and added to the RTD resistance. If 22AWG wire was used, per the wire resistance tables, 22AWG wire has a resistance of 0.01614Ω/foot…so 100 feet of the wire would result in an additional 1.614Ω. For a commonly used 0.00385 alpha 100Ω RTD, this would result in a measurement error of approximately 2.96°C (i.e. 1.614Ω /(0.385Ω/°C) = 2.96°C).

For shorter RTD connections simple 2-wire configurations with their slight errors (as described above) may be acceptable. Selection of heavier gauge wires can minimize these 2-wire resistance errors. Figure 1 shows a basic two-wire configuration with the parasitic resistances in the wires. In the measurement, the known current flows from the (+) Excitation/Sense terminal through the (+) wire resistance, the RTD resistance the (-) wire resistance and back to the (–) Excitation/Sense terminal of the measuring instrument. The current develops a voltage across all of the resistances in the series loop and hence the measured voltage across the +/- Excitation/Sense terminals will then be higher than the actual voltage across only the RTD. Using this erroneous voltage in Ohm's Law results in a calculated resistance that is greater than the RTD element and will result in a temperature error.

As mentioned above, with shorter wires of sufficient gauge (ie low resistance) this simple 2-wire configuration commonly suffices for lower absolute accuracy applications.

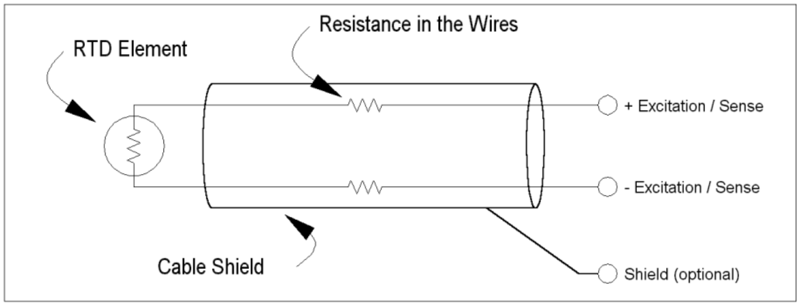


Figure 1; Two-Wire RTD Configuration

4-Wire Configuration

For applications requiring longer runs of lead wires and/or more accuracy in the actual temperature measurement, the “4-wire” configuration is commonly used. In this configuration, two pairs of wires are routed to each terminal of the RTD. One pair carries the known excitation current and the other pair is used to sense the voltage developed across the terminals of the RTD. The excitation current will develop a voltage across the RTD as well as across each of the parasitic wire resistances however the voltage is NOT measured with these Excitation leads. Instead, the second pair of Sense leads are used to measure the voltage across the RTD. As the current flowing in the Sense terminals during voltage measurement by a high input impedance rated instrument will be very low, the voltage developed in the parasitic resistors in the Sense leads will introduce negligible error.

Figure 2 shows the 4-wire configuration with the parasitic resistances in the wires. Note that since the excitation current is flowing in the Excitation lead pair, the lead resistances will develop error voltage. However, since the voltage across the RTD is measured by the Sense lead pair which does not have the Excitation current flowing in it, the error voltage developed across the Sense wire parasitic resistances is extremely low and negligible for almost all applications.

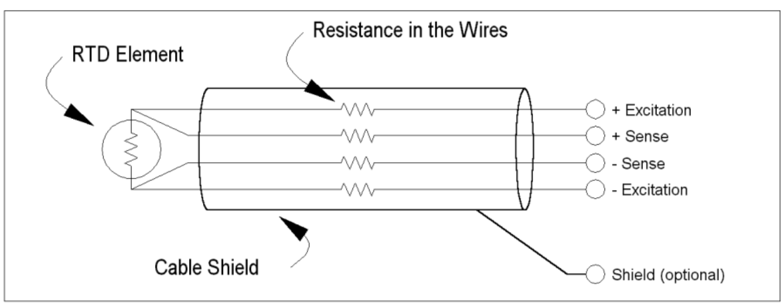


Figure 2; Four-Wire RTD Configuration

RTD 0˚C Resistance

Another method to minimize the effects of lead wire resistance is to utilize an RTD with a higher 0°C resistance. This effectively reduces the effect of parasitic lead resistances. For example, 1000Ω RTD’s are becoming more popular and their corresponding Ω/°C are approximately 10x higher than a comparable 100Ω RTD. This results in a reduction of the lead wire resistance effects by a factor of 10.

RTD RESISTANCE TO TEMPERATURE CONVERSION

Once the resistance of the RTD has been accurately measured by the data acquisition instrument , the resistance is commonly converted to temperature in one of two ways, using a Linear Approximation equation or by using the Callendar-Van Dusen equation[1].

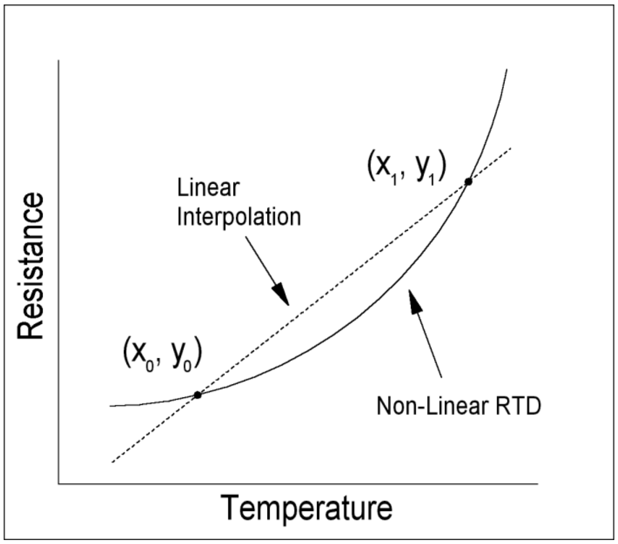


Figure 3; Graph of Linear Interpolation

LINEAR APPROXIMATION

Linear Interpolation is used to create the equation of a line in $y = mx + b$ form using Equation 1 and two points along the curve, (x_0,y_0) and (x_1,y_1), as seen in Figure 3. This method uses a straight line to approximate the RTD non-linear curve and can be quite accurate over narrow temperature ranges where the RTD approximates a straight line. This resistance to temperature conversion method is not as accurate as the Callendar-Van Dusen equation (below) but is simpler to implement.

$$y = mx + b \qquad m = \frac{y_1 - y_0}{x_1 - x_0} \qquad b = y_0 - x_0\left(\frac{y_1 - y_0}{x_1 - x_0}\right)$$

Equation 1; Linear Interpolation

[1] For applications requiring extremely high accuracy and with sufficient mathematical processing power, high order (eg 20th order) polynomials can also be used. The use of high order polynomial conversions is not addressed in this Application Note as the prior simpler methods meet all but the most stringent applications.

CALLENDAR-VAN DUSEN EQUATION APPROXIMATION

In the mid 1920's an equation relating platinum RTD temperature and resistance was derived by Hugh Callendar, and refined by M. S. Van Dusen… hence its name. The standard form of the equation which solves for resistance knowing temperature is shown in Equation 2.

$$R_T = R_0(1 + AT + BT^2 – 100CT^3 + CT^{4})$$

Equation 2; Callendar-Van Dusen equation in a form to solve for Resistance (R_T) with a known Temperature (T) in Celcius degrees

The included variables and equation usage are described in detail later in this Application Note. For a calculation of temperature knowing resistance, an inverse form of the Callendar-Van Dusen equation is required and addressed following.

The Callendar-Van Dusen equation is more simply implemented for positive (Celsius) temperatures. For negative temperatures an iterative form of the equation is required however that also can readily be handled via the Math Icon in HyperWare-II. Once the Callendar-Van Dusen equation is programmed into the IntelliLogger it then executes the measured RTD resistance to temperature conversion.

INTELLILOGGER RTD RESISTANCE MEASUREMENT…AN EXAMPLE APPLICATION

It is desired to accurately log the temperature of a 1000Ω platinum, 0.00385 alpha RTD that has a specified power dissipation rating (P_{diss}) of 50mW/˚C. It is desired to make free air temperature measurements over the temperature range of 0 to 100˚C (+/-0.1˚C) using an IntelliLogger data logging system.

Two steps are required to measure the RTD temperature:

1. Determine the RTD resistance
2. Mathematically convert the RTD resistance to temperature using:
    - mX+b linear approximation (good accuracy only for narrow temperature ranges)
    - Callendar-Van Dusen equation (high accuracy over any range)

RTD RESISTANCE DETERMINATION CONCEPT

To measure the resistance of an RTD a known current (I_{RTD}) can be routed through the RTD and the voltage across the RTD (V_{RTD}) then measured.

Per Ohm's law, the RTD voltage (V_{RTD}) divided by the current (I_{RTD}) is the resistance (R_{RTD}).

$$V_{RTD} / I_{RTD} = R_{RTD}$$

This Ohm's Law resistance calculation is simply performed in the IntelliLogger per a program developed by the user within HyperWare-II and loaded into the IntelliLogger where it executes.

RTD Resistance Measurement Using the IntelliLogger

The high resolution isolated analog input channels on the IntelliLogger ILIM-7 Analog Expansion Module or the IL-80 (which includes an integral ILIM-7 module) are well suited for the measurements necessary for RTD use. The wiring configuration will be as shown in Figure 4.

The current through the RTD will be measured by one IntelliLogger channel configured in mAdc mode (which utilizes an internal precision 100 ohm shunt resistor) and the voltage across the RTD will be measured with a second channel configured for Vdc input.

Additionally, the excitation current can be provided by the program controlled +5Vdc output power supply integral to the IntelliLogger and ILIM-7 modules.

RTD Excitation Current

A nominal +5Vdc supply is built into the IntelliLogger ILIM-7 module. This supply will be used to provide the excitation current to the RTD. As the current through the RTD will be measured precisely each time a measurement is performed, the stability over time and temperature of this supply voltage is not critical.

Power Dissipation Limiting

To limit the self-heating power dissipation in the RTD during the measurement such that desired accuracy is maintained for the application, a series resistor is necessary to limit the current flow through the RTD sensor.

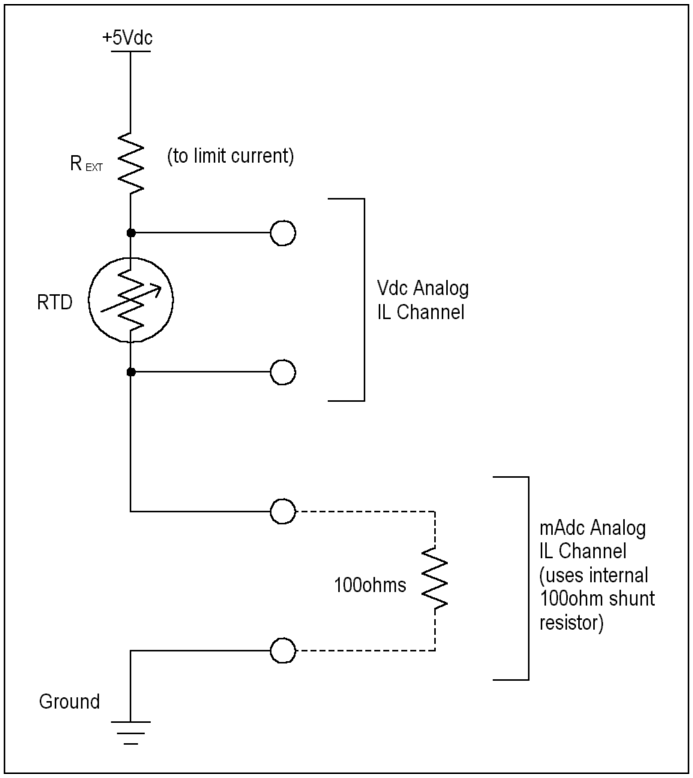


Figure 4; Schematic of RTD Measurement with the IntelliLogger

Selection of the external current limiting resistor value should take into account:

- Excitation voltage to be used
- Power dissipation rating of the RTD (and corresponding RTD current over the temperature measurement range)
- Thermal heat sinking characteristics of the medium to which the RTD is in contact (eg still air or flowing water) in the application
- To a lesser degree but for optimum resolution and accuracy, if the voltage and currents measured over the RTD temperature measurement range approach full use of the IntelliLogger channel input range, that is preferred

For the described application, a 1000 ohm RTD is specified. After review of the manufacturer's specifications for the RTD it is noted that the power dissipation specification for the part in free air is 50mW/˚C. The application requires that the part is to be used in free air so to ensure that the maximum error goal of +/-0.1°C is achieved, it is necessary that the part not dissipate more than 5mW (ie 1/10th of 50mW) during reading and preferably less to allow for other error factors..

External Resistor Value Calculation

Per many readily available RTD Resistance to Temperature tables[2], the resistances of the RTD when at either end of the desired measurement range are 1000Ω at 0°C and 1385.1Ω when at 100°C.

Based on the power equation… $P_{RTD} = I_{RTD}^2 \times R_{RTD}$ which can be massaged into the following equation to solve for $I_{RTD\ (max)}$. Note that the highest RTD resistance over the specified temperature range is used (rather than the lowest) since the power dissipation will be at a maximum when the R_{RTD} term is at its maximum (for the same current). The maximum current calculation follows:

$$I_{\max} = \sqrt{\frac{P_{\max}}{R_{RTD}}} = \sqrt{\frac{5mW}{1385.1\Omega}} = 1.9mA$$

To limit I_{max} to 1.9mA, the total external series resistance through which the excitation current will flow is calculated as follows based on Ohm's law:

$$\frac{5V}{1.9mA} = 2631\Omega$$

As shown in Figure 4, this total series resistance includes the RTD resistance, the 100Ω IntelliLogger internal mAdc input channel shunt and the external current limiting resistor R_{EXT}. For this current calculation we use the lowest resistance of the RTD (i.e. 1000Ω) over its operating temperature range since with a fixed voltage supply the maximum current will flow with the lowest total resistance in the string.

$$R_{EXT} = 2631-100-1000 = 1571\Omega$$

Rounding up the calculated 1571Ω to the next commercially available standard resistor value results in use of an external resistor, R_{EXT} = 1580 (i.e. 1.58KΩ).

Self-Heating Minimization with Cycled RTD Excitation Current

Another method to minimize the RTD self-heating induced error can implemented by turning the RTD excitation current on, quickly taking a reading then shutting the current off. The RTD requires time to increase in temperature due to the thermal mass of the sensor body mitigating the RTD generated heat. By keeping the RTD powered ON for short periods of time, the thermal effect is minimized. This cycled application of the excitation current is addressed in the programming of the IntelliLogger described later in this App Note.

[2] Many RTD manufacturers as well as resellers publish RTD Resistance to Temperature conversion tables

INTELLILOGGER WIRING CONFIGURATIONS

As described earlier, 2-wire or 4-Wire physical wiring configurations can be used with the IntelliLogger. The following wiring diagrams (Figure 5 and Figure 6) illustrate the wiring variations.

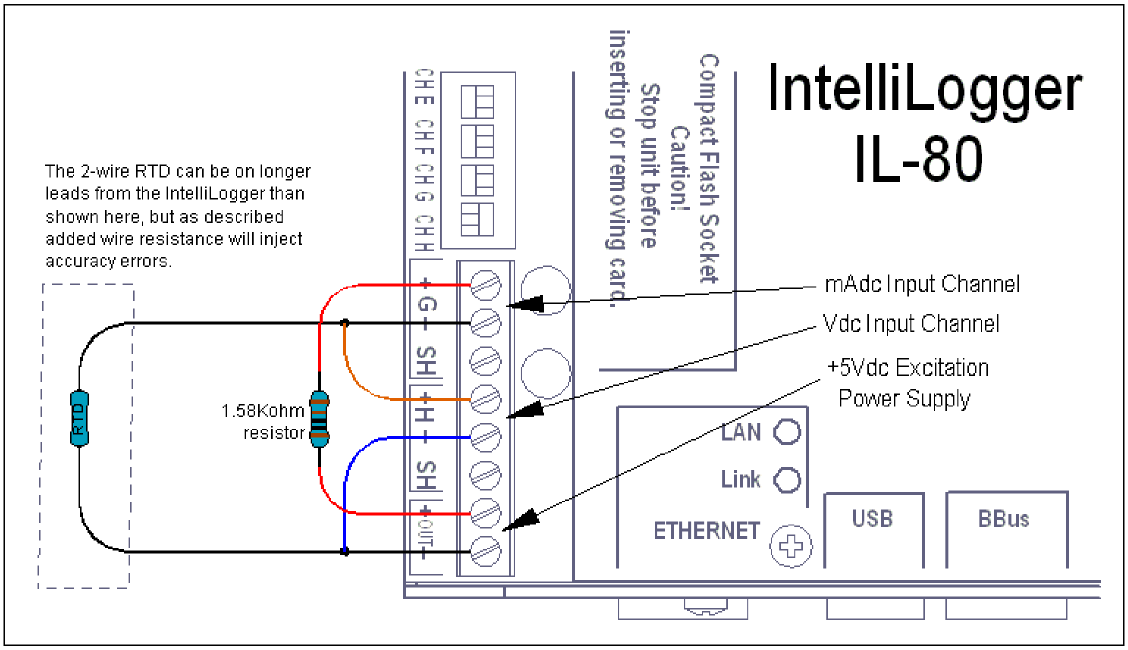


Figure 5; 2-Wire RTD Interface to the IntelliLogger using Vexc from the ILIM-7 Module

Both the 2-wire (Figure 5) and 4-wire (see Figure 6) RTD with current limiting external resistor configurations are shown connected to an IL-80.

In the 2-wire configuration, the two wires to the RTD supply the excitation current and are used to sense the voltage. As previously stated, this method is not advised for long cable runs as the voltage developed on the leads will be impressed at the voltage reading terminals of the IntelliLogger connection introducing an offset error in the V_{RTD} measurement.

In the 4-wire configuration, a four conductor shielded cable connects the RTD to the IL-80, providing two wires for the excitation current path, and two parallel wires for sensing the voltage directly across the remote RTD. The 5Vdc excitation output on the ILAD (or integrated ILIM-7) is used to provide a current source for the RTD. The current through the RTD is limited by a 1.58KΩ resistor to minimize RTD self-heating. The excitation current is also routed through one of the ILIM-7 current measurement inputs. As the IntelliLogger +5Vdc excitation source is not a current source, the current will vary as the RTD resistance changes in normal operation. Additionally, over time and temperature, the +5Vdc excitation source may vary slightly. Using this additional mAdc input channel provides an accurate measurement of the instantaneous current flowing through the RTD. The Sense leads are connected to a voltage input on the IL-80 so that we can measure the voltage developed across the RTD. Once programmed, the IntelliLogger can then accurately calculate the resistance of the RTD by dividing the voltage across the RTD by the current flowing through the RTD, i.e. $R_{RTD} = V_{RTD} / I_{RTD}$.

This resulting RTD resistance is then used within the IntelliLogger Program Net in either a straight line approximation or a Callendar-Van Dusen calculation of the RTD temperature.

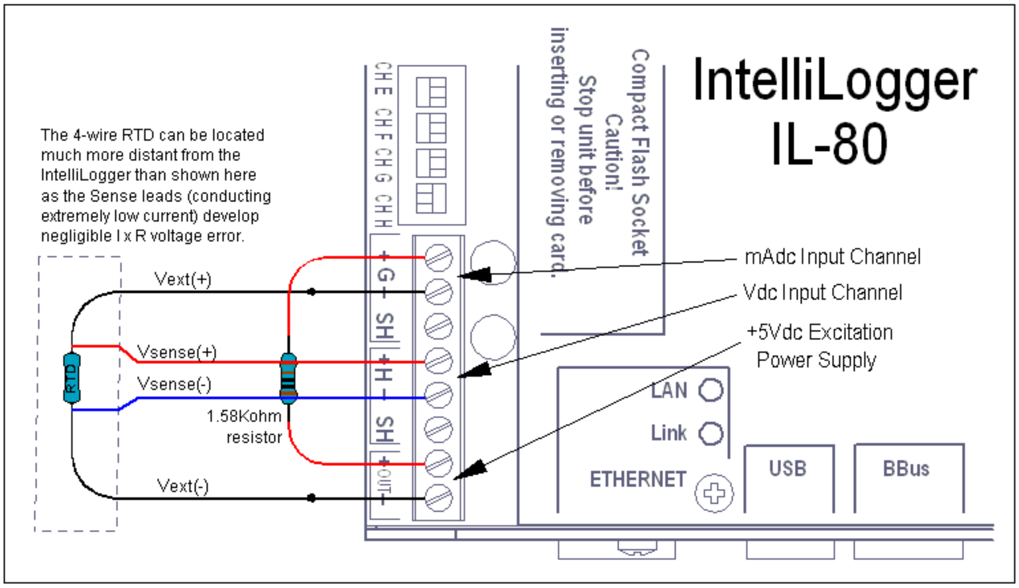


Figure 6; 4-Wire RTD Interface to the IntelliLogger using Vexc from ILIM-7 Module

RTD EXCITATION CURRENT

The IL-80 base has a +5Vdc regulated supply output on the base unit as well as a second output that is integral to the integrated ILIM-7 module. External ILIM-7 modules also have this +5Vdc output. In these wiring diagrams, the +5Vdc output on the ILIM-7 module us shown being used for the excitation current. Under software control, this +5Vdc output cycles on before the isolated analog input voltage and current readings are taken and shuts off automatically after the readings. This excitation power supply operation is enabled within each channel's Configuration Dialog (Figure 7) by checking the *Enable Excitation During Reading* checkbox. Although not necessary for RTD circuits as they have nearly immediate response, for some excitation applications, additional time delay between the excitation power supply turning on and the actual channel reading can be added. This additional settling time is defined in the channel Configuration Dialog via the "Extended Settling Time" setting.

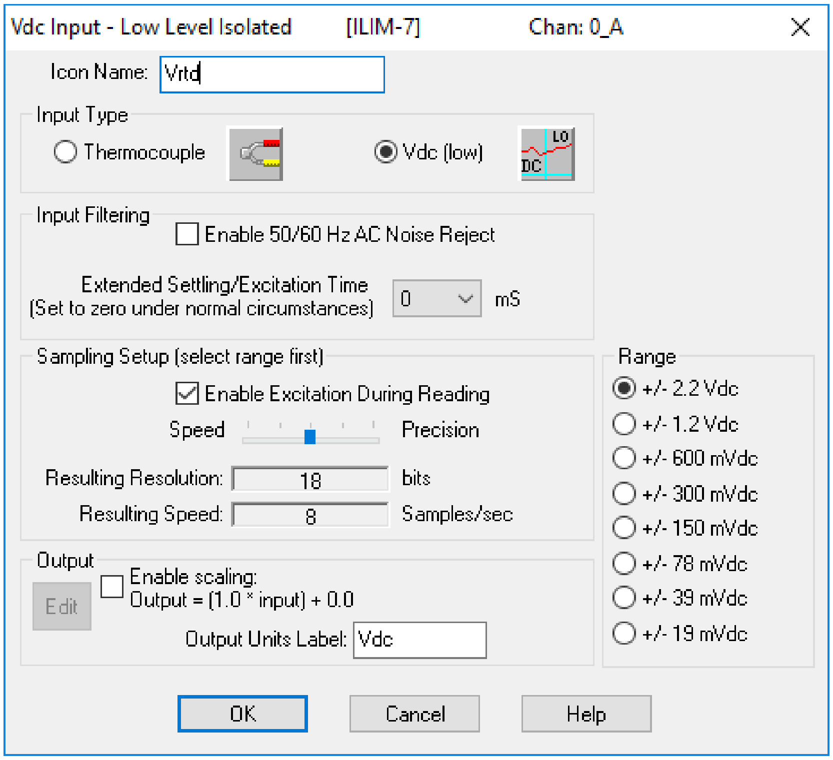


Figure 7: Vdc (Low Range) Input Icon Configuration Dialog

INTELLILOGGER PROGRAMMING

Once the external resistor value has been calculated (as above) and the IntelliLogger and RTD probe wired, the IntelliLogger needs to be programmed to sample the RTD current and voltage via the HyperWare-II software. Familiarity with use of HyperWare-II is assumed for the purposes of this Application Note.

INPUT CHANNEL RANGE SELECTION

The mAdc and Vdc input channels have software programmable signal input ranges. A quick calculation should be done to select the narrowest range that will accept the expected current and voltage input signals. Selecting the minimum width range that will accept the expected signal input maximizes the IntelliLogger resolution and accuracy.

mAdc Input Channel Range Selection:

The maximum current that will flow through the series of resistances making up the RTD sensing circuit will be V_{EXC} / R where R is the sum of the 1.58KΩ external limiting resistor, the RTD resistance and the 100Ω shunt resistor integral to the ILIM-7 measurement channel. The maximum current will flow with the minimum resistance in the series circuit so the RTD resistance at 0C which is 1000Ω. The V_{EXC} is a nominal 5Vdc.

The max current is then 5/(1580+100+1000) = 1.87mA. The lowest range that will accept 1.87mA in the mAdc Input Configuration dialog is 3mA which is chosen (Figure 8).

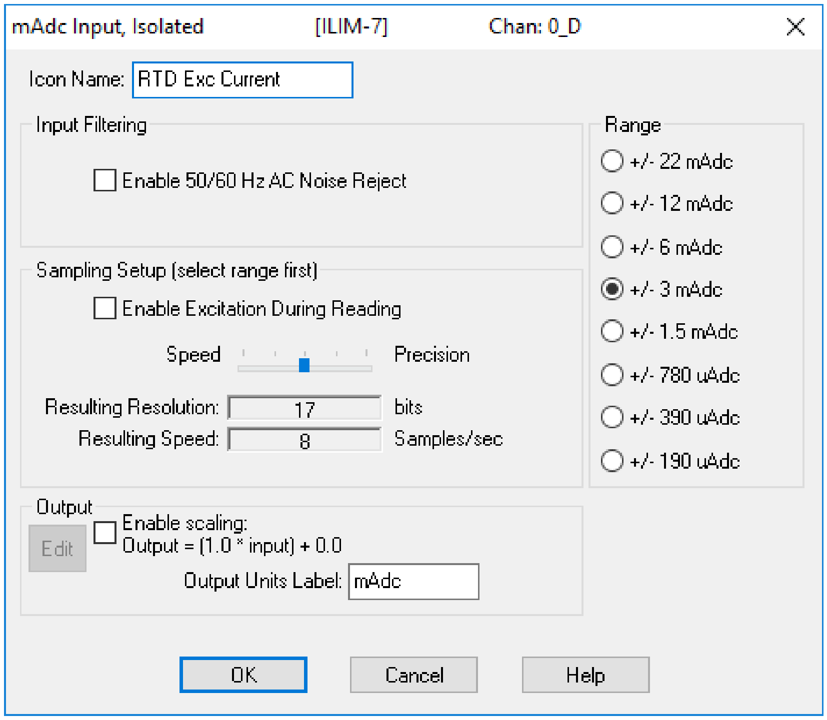


Figure 8: mAdc Input Configuration Dialog Range Selection

Vdc Input Channel Range Selection:

Based on a simple voltage divider calculation, the maximum voltage that will be developed across the RTD will be when the RTD resistance is at its maximum which is 1385.1Ω (i.e. @60C). The voltage across the RTD is calculated using a simple voltage divider calculation.

$$Vrtd = Vexc * \frac{Rrtd}{(Rrtd + Rshunt + Rext)}$$

Which in our application results in:

$$Vrtd = 5 * \frac{1385.1}{(1385.1 + 100 + 1580)} = 2.26Vdc$$

The Vdc Input Configuration Dialog (set as Vdc Low input per Figure 7) offers a maximum range of 2.2Vdc so it appears that the expected maximum signal will not quite fit the 2.2Vdc range. The next larger range offered by the ILIM-7 is +/-7.5Vdc which would result in use of only 1/3rd of the input range compromising a bit of accuracy and resolution. Since the maximum signal is so close to the fitting the 2.2Vdc range, let's looks at our options…

1. Change the IntelliLogger input to the Vdc High input via module configuration switches and use the wider +/-7.5Vdc input range that readilyk encompasses the expected 2.26V.

2. Use the 2.2Vdc range. Since there is typically a bit of over-range "headroom" on input ranges[3] by design (to ensure that the IntelliLogger can fully handle the ranges specified in the Configuration Dialog) it is known that the 2.26Vdc maximum reading will be readily measured by the 2.2Vdc range.

3. If we reduce the excitation current slightly by increasing the excitation current limiting resistor slightly, the voltage developed across the RTD will similarly decrease and fit within the standard 2.2Vdc range. A quick calculation of the required percentage reduction in current:

   0.06Vdc/2.26Vdc = 2.7%

   Calculating 2.7% of the total resistance string (R_{RTD}, R_{SHUNT} and $R_{EXT)}$ yields:

   (1385Ω + 100Ω + 1580Ω) x 0.027 = 82.8Ω resistance increase needed.

   And adding 82.8Ω to the current External resistor value of 1580Ω yields a new resistor value of 1663Ω. Per a standard resistor look up chart, the nearest standard 1% resistor ≥ 1663Ω is 1690Ω.

Option 3 seems the best in that it *very minimally* compromises accuracy and resolution, doesn't require any insider information and as a bonus will actually slightly decrease the current through, hence the power dissipation of the RTD even further.

A quick recheck of the calculations ensures the standard 2.2Vdc range will suffice:

$$Vrtd = 5 * \frac{1385.1}{1385.1 + 100 + 1690} = 2.18Vdc$$

[3] Logic Beach inside information….

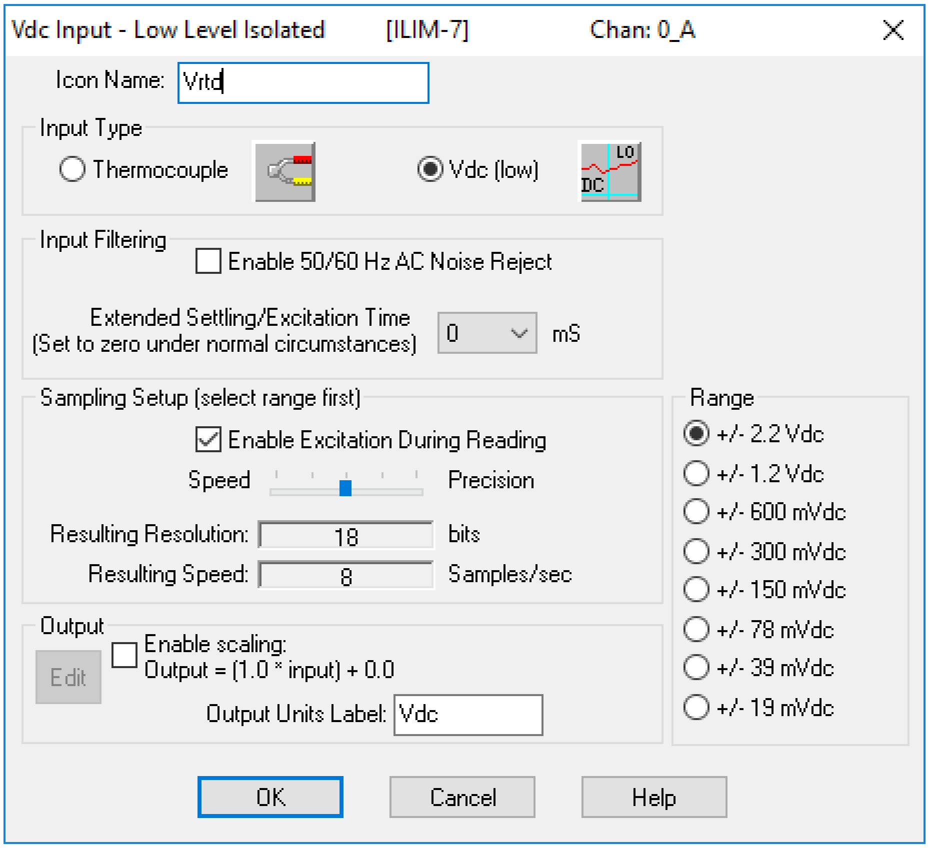


Figure 9: Vdc Input Icon Dialog Range Selection

PROGRAMMING THE INTELLILOGGER FOR RTD RESISTANCE TO TEMPERATURE CONVERSION USING LINEAR APPROXIMATION

Although experience indicates that over the fairly wide desired range of °C to 100°C, a linear approximation method of calculation of RTD temperature from a measured RTD resistance will not meet the desired accuracy requirement (+/-0.1C) for our example application … it is a good exercise so we will demonstrate the implementation.

In prior steps, we configured the wiring and set up the IntelliLogger input channel ranges for current and voltage measurements and follow-on calculation to determine the current RTD resistance. This resistance is then to be input to a Math Icon within HyperWare-II to calculate the corresponding RTD temperature.

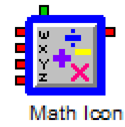
Linear Approximation Equation Parameters

To generate a straight line with endpoints on the RTD curve we can initially start with simply using the resistance/temperature pairings at each end of the RTD range which were specified above and repeated here for convenience:

1000Ω @ 0C

1385.1Ω @ 100C

Linear Equation Generation



The standard y=mX+b equations shown in Equation 1 (earlier in this App Note) can be used to calculate the equation to program into the IntelliLogger Program Net Math icon, however a much simpler method is to simply use the power of the Equation Generator that is integral to the Math Icon.

Math Icon Linear (mX+b) Equation Generator

Open the Math Icon dialog, select *Generate Equation* and the Linear Output Scaling Equation Generator displays. Enter the Input Signal (in this case the RTD resistance) and corresponding "Real World" temperature values into the dialog and hit *Calculate*. The mX+b format equation is generated and displayed in the *Resulting Equation* pane.

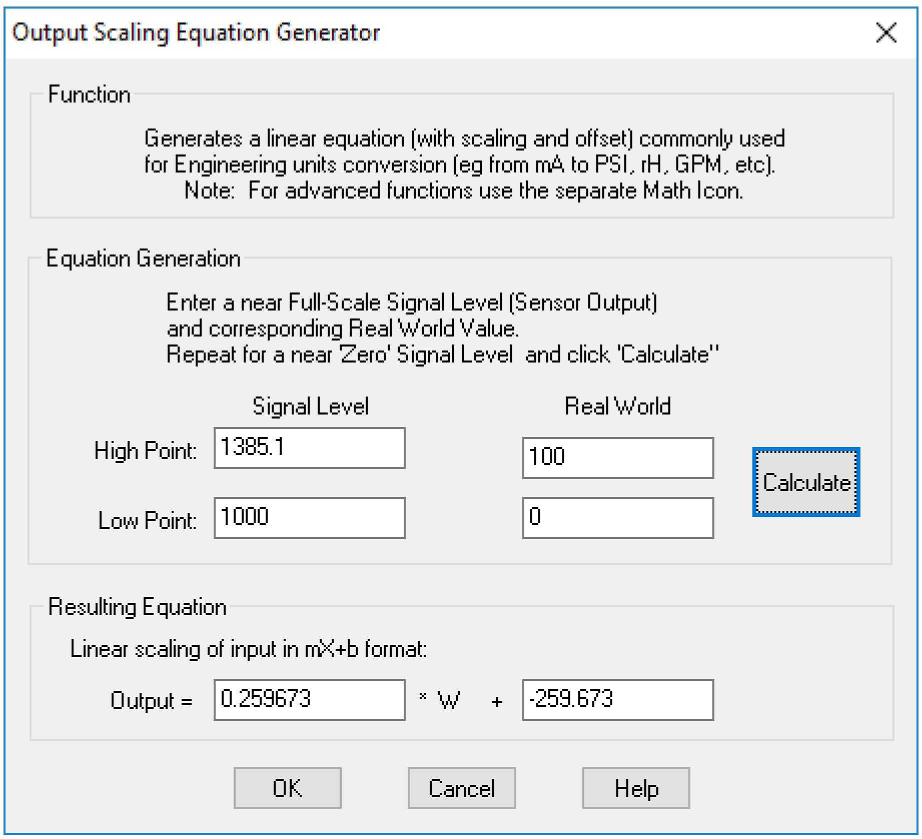


Figure 10: Math Icon Linear Equation Generator dialog

Click OK and the Generator dialog closes and the equation is automatically pasted into the Math Icon dialog Function field.

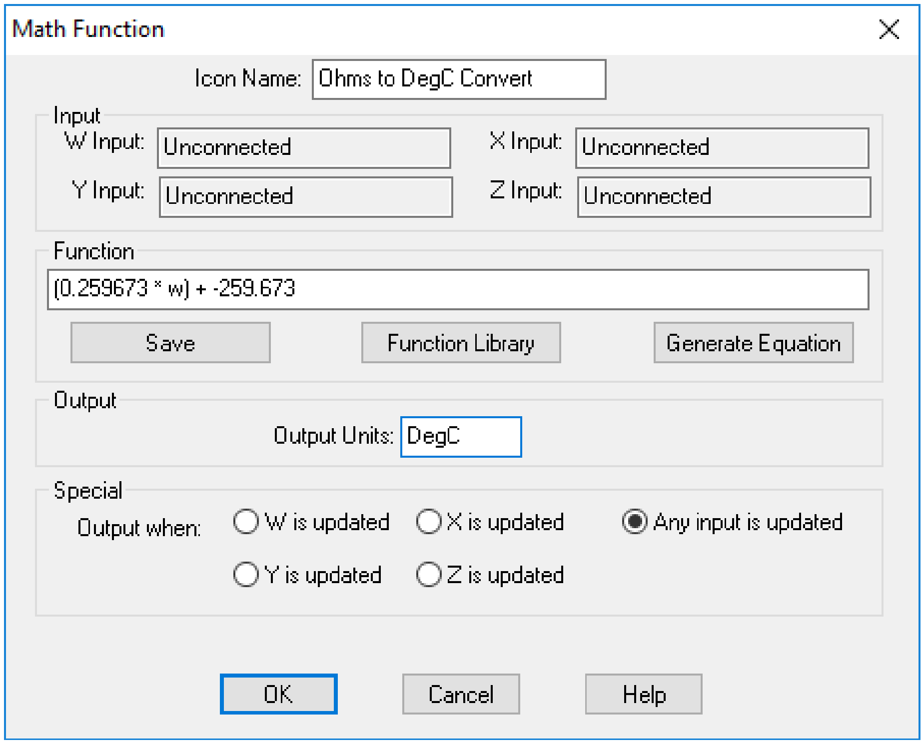


Figure 11: Math Icon with Linear Equation programmed as the Function

Accuracy Testing of Linear Approximation

To test the equation, a simple Program Net could be built using a Programmable Constant Icon as an input to the Math Icon W terminal and various simulated RTD resistance values could then be entered via the Programmable Constant icon while the Program Net is executing. The Math Icon output could then be observed and compared to a reference RTD Resistance to Temperature table to determine the resultant accuracy.

Intuitively, if one looks at the straight line approximation of the RTD curve as illustrated in Figure 3 it is obvious that the least error will manifest at the end points and the most error will manifest at near mid-range. For this application, mid-range is 50C and per the reference table, the resistance at 50C is 1194.0Ω.

Instead of building a test net to fully characterize the accuracy, lets simply crunch through the Math Icon Function on a calculator. Using 1194.0Ω as the input (W) results in a calculated temperature of 50.38C for a resulting error of 0.38C (i.e. 50.38 – 50.0 = 0.38)… considerably off from the desired +/-0.1C desired in this demonstration application.

TUNING THE LINEAR APPROXIMATION EQUATION…

In the case of the RTD over this 0 to 100C range, the curve approximates a simple arc. Use of the end point values for the mX+b conversion maximizes the error at the middle of the curve where it deviates furthest form the straight line approximation.

Improvements in accuracy can be made by simply choosing Temperature/Resistance pairs for the mX+b calculation that are a distance in from the end points… such as +25C and 75C (as illustrated in

Figure 3). This will serve to reduce the maximum error considerably and experimentation with this method may result in sufficiently accurate results for many applications. For example, using 25°C and 75°C temperature/resistance pairs results in a maximum error at the midpoint (50°C) of 0.103°C… effectively meeting the error goal over this appllication's fairly narrow temperature span. It should not be expected to maintain this error over even a double span (i.e. 0 to 200°C) using the linear approximation method. The Callendar-Van Dusen equation below should be used instead.

HyperWare Program Net Implementation of the Linear Approximation Calculation

Using the above calculations, a fairly simple Program Net (Figure 12) can be built for reading RTD temperatures using a current and a voltage analog input channel on the IntelliLogger ILIM-7 module (or isolated channels of an IL-80).

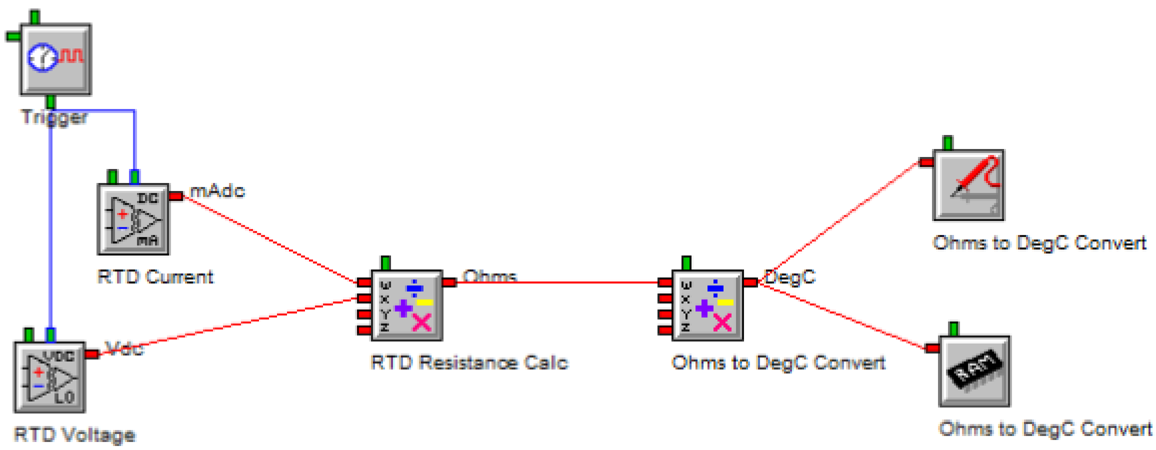


Figure 12: Program Net for Linear Approximation RTD Temperature Reading

To further simplify, both the RTD resistance calculation and the down-stream resistance to temperature calculation could be combined into a single Math Icon.

PROGRAMMING THE INTELLILOGGER FOR RTD RESISTANCE TO TEMPERATURE CONVERSION USING THE CALLENDAR-VAN DUSEN EQUATION

To achieve high accuracy over wider temperature ranges, the Callendar-Van Dusen equation excels over the Linear Approximation method described previously. The conversion equation is a bit more complex but still readily handled within HyperWare-II and the IntelliLogger instrument. Two cases will be presented:

- Positive measurement range (as in our example above of 0 to 100C)
- Ranges encompassing negative temperatures (eg -100 to +100C)

For positive measurements only, a single calculation can be performed for the conversion. For ranges involving negative temperatures, iterative calculations are required[4].

Using the Callendar-Van Dusen equation for Positive RTD Temperature Calculations knowing Resistance

As described above, the resistance of the RTD is calculated using two analog inputs channels and performing basic math. Once this resistance is known, it is sent into a Math Icon to execute the Callendar-Van Dusen equation and output the temperature.

Following is a form of the Callendar-Van Dusen equation derived from the Callendar-Van Dusen equation mathematically massaged into a form where it solves for temperature knowing resistance… which better suits our application needs. This form can be used for accurate conversions of measured resistance to temperature in the mainly positive range of -50 to 800C:

$$T_R = \frac{-R_0 A + \sqrt{{R_0}^2 A^2 - 4 R_0 B (R_0 - R_T)}}{2 R_0 B}$$

Figure 13: Callendar-Van Dusen equation converted to solve for Temperature knowing Resistance

CALLENDAR-VAN DUSEN EQUATION CONSTANTS:

At initial glance the Callendar-Van Dusen equation with its multitude of constants can seem foreboding however as will be shown, the equation can be readily simplified for use in an IntelliLogger Program Net.

The constants are derived from manufacturer measured resistances at three key temperatures, 0, 100 and 260°C. Following is a listing of the constants and variables used in the Callendar-Van Dusen equation.

R_0 = RTD resistance at 0°C (1000Ω for this example application)

R_T = Measured RTD resistance (which we will be converting to temperature)

T_R = the temperature result for the input measured resistance R_T

A, B and C are constants that are derived indirectly from the specific RTD's manufacturer measured resistances at 0° (R_0), 100° (R_{100}) and 260°C (R_{260}):

$$A = \alpha + \frac{\alpha \cdot \delta}{100}$$

$$B = \frac{-\alpha \cdot \delta}{100^2}$$

[4] The Callendar-Van Dusen equation in its original form solves for resistance knowing temperature. Conversion of the equation to its inverse form to solve for temperature knowing resistance requires mathematical manipulation of the equation. For positive values, this conversion is relatively simple. For negative temperature values, the conversion of the Callendar-Van Dusen equation does not simply convert.. hence the need for iterative calculations discussed later.

$$C = \frac{-\alpha * \beta}{100^4}$$

Where…

$$\alpha = \frac{R_{100} - R_0}{100 \cdot R_0}$$

$$\delta = \frac{R_0 \cdot (1 + \alpha \cdot 260) - R_{260}}{4.16 \cdot R_0 \cdot \alpha}$$

β = a manufacturer supplied constant for T < 0°C else it is 0 if T ≥ 0°C..

Note that Constant "C" above is a function of β and becomes 0 if T ≥0.

If extremely high accuracy measurement is desired, many RTD manufacturers can provide the resistances measured during manufacturing for 0°, 100° and 260°C and these values can be used in the above equations to determine the constants and ultimately the final equation form. However for many applications (such as our example) using standard published values for the constants that are available from RTD manufacturers will suffice to provide reasonably high accuracy. The following table contains standard values for 0.00375, 0.00385 and 0.00392 alpha RTDs:

Alpha (α)	**0.00392**	**0.003750**	**0.003850**
Delta (δ)	1.49627	1.605	1.4999
Beta (β)	0.11	0.16	0.10863
A	3.978×10^{-3}	3.81×10^{-3}	3.908×10^{-3}
B	-5.841×10^{-7}	-6.02×10^{-7}	-5.775×10^{-7}
C	4.351×10^{-12}	-6.0×10^{-12}	-4.183×10^{-12}

Table 1: Callendar-Van Dusen equation Constants for some common RTD's

Substituting the known R_0 (1000Ω), and A and B values from the table for the 0.00385 Alpha curve part (specified in our application example) into the Callendar-Van Dusen equation form shown in Figure 13 allows the Callendar-Van Dusen equation to be simplified to the following:

$$T = \frac{-3.908 + \sqrt{15.2725 - 57750 * (1000 - R_T)}}{-0.001155}$$

Equation 3: Callendar-Van Dusen equation with constants included

And this equation can then be written inline[5] and entered into a Math Icon Configuration Dialog (Figure 14) to accept the RTD resistance (R_T) and output the RTD temperature as follows[6]:

(-3.908+SQRT(15.2725+0.00231*(1000-W)))/-0.001155

Where “W” is the RTD resistance calculated in an upstream Math icon which connects to this Math Icon terminal “W”.

Note: The above equation can simply be copied and pasted from this document into a Math Icon Formula field within HyperWare-II if the Math icon input terminals are connected as described above and a 1000Ω, 0.00385 RTD is employed.

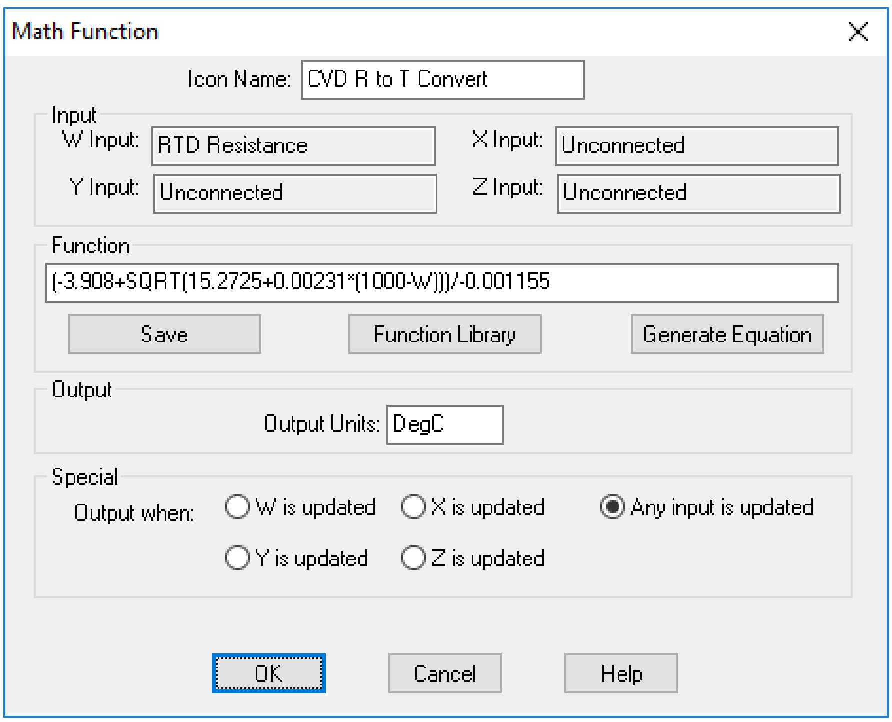


Figure 14: Math Icon configuration dialog with Callendar-Van Dusen equation programmed

HyperWare Program Net Implementation of the Callendar-Van Dusen equation Calculation

The Program Net built to use the Callendar-Van Dusen equation (Figure 15) will appear identical to the Program Net using the linear approximation calculation (Figure 12) however the Math Icon named *Ohms to DegC Convert* would utilize the Callendar-Van Dusen equation as shown in Figure 14.

[5] For quick reference while entering equations, the Math Icon *Help* button describes equation syntax

[6] Note that once entered, this equation can be saved to the Math Icon Function Library for future re-use by selecting the *SAVE* button in the Math Icon dialog.

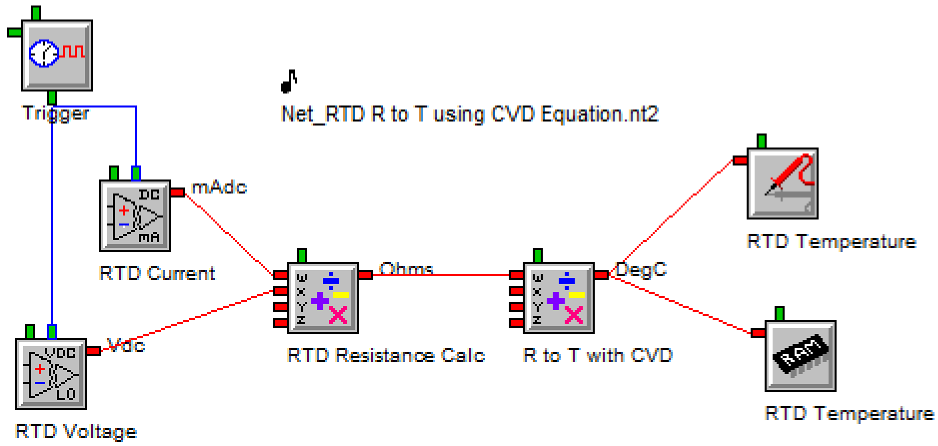


Figure 15: RTD Resistance Calculation and conversion to Temperature using the Callendar-Van Dusen equation

Conversion Accuracy Testing with the IntelliLogger

Rather than crunching manually through the Callendar-Van Dusen equation iteratively to test its conversion accuracy (and that the equation was entered correctly into the Math Icon) it is much easier to simply program the IntelliLogger with the equation and feed constant values into the Math Icon via the Remote Control icon. A Program Net as shown in Figure 16 can be quickly built and used. Note that the CJC temperature input is connected to the Math Icon input Z simply to force the Math Icon to update… reading a new value from the Remote Control Constant named *RTD Resistance.* Math icon input Z is not used in the Math icon equation so it has not effect on the calculation.

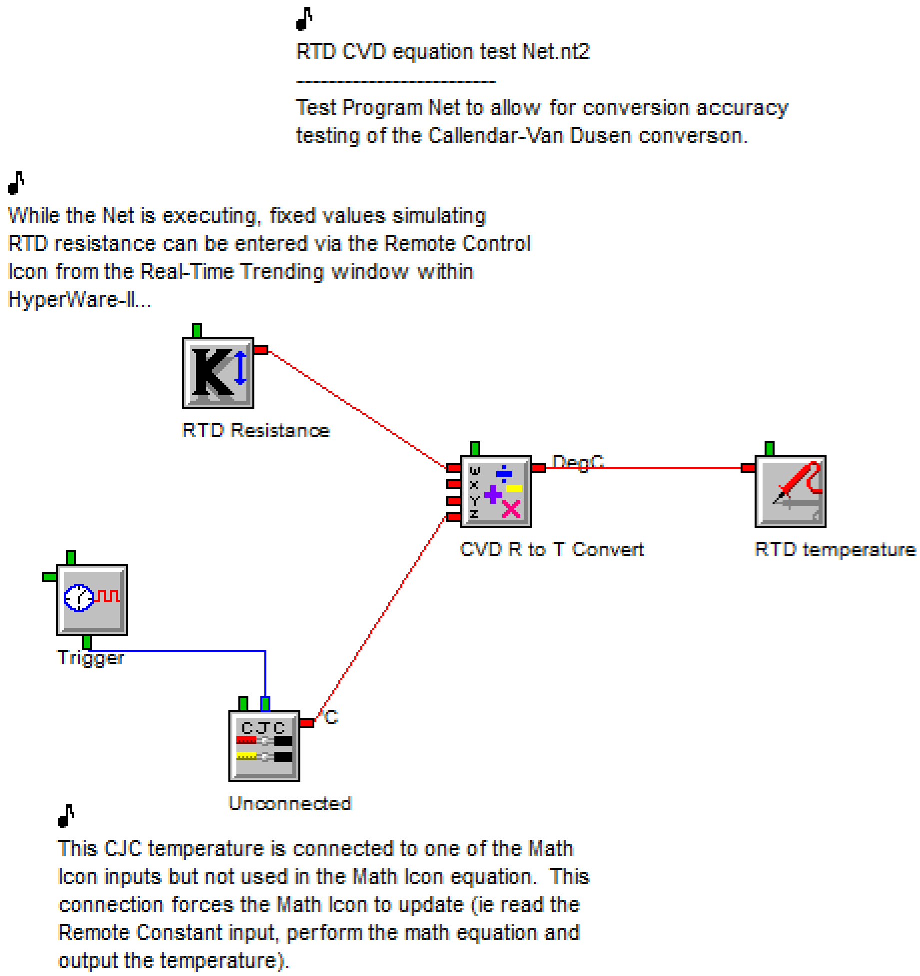


Figure 16: Callendar-Van Dusen Conversion Accuracy Test Program Net

Conversion Accuracy Test Results

Figure 17 shows IntelliLogger calculated reference vs Callendar-Van Dusen equation error over a temperature range of -200 to 800C. Per the IntelliLogger Program Net above, RTD resistance values from a DIN 43 760 reference table were input to a Math Icon executing the Callendar-Van Dusen equation. As explained above, the Callendar-Van Dusen equation does a highly accurate conversion for temperatures down to approximately -50C. Below that the error increases quickly.

In our example application requiring accuracy of +/-0.1°C over the temperature range of 0 to 100C, the accuracy could actually be maintained using the Callendar-Van Dusen equation down to -50°C.

RTD Resistance (Ω) per Reference Tables	Corresponding RTD Temperature (DegC)	Math Icon CVD Equation Output	Temperature Error (DegC)
185.2	-200	-202.443	-2.443
397.2	-150	-150.887	-0.887
602.6	-100	-100.209	-0.209
803.1	-50	-50.0181	-0.018
901.9	-25	-25.0139	-0.014
1000.0	0	-0.0039	-0.004
1194.0	50	50.0074	0.007
1385.1	100	100.016	0.016
1758.6	200	200.023	0.023
2120.5	300	300.017	0.017
2470.9	400	400.024	0.024
2809.8	500	500.048	0.048
3137.1	600	600.057	0.057
3452.8	700	700.051	0.051
3757.0	800	800.062	0.062

Figure 17: Reference RTD data vs Callendar-Van Dusen calculated data

Resistance to Temperature Conversion for Negative RTD Temperatures

With the goal of extending our application example temperature range beyond -50C.. down to -200C one might initially think that for that fairly narrow 150C range, a linear approximation method could be employed. A bit of testing will quickly demonstrate that the RTD curve is sufficiently non-linear that the desired +/-0.1C can't be met using linear approximation method.

ITERATIVE MATHEMATICAL REFINING OF THE CALCULATED TEMPERATURE

An alternative method to achieve higher accuracy conversion utilizes successive approximation calculations to achieve desired accuracy in the negative temperature range. An initial "positive temperature" Callendar-Van Dusen equation calculation (Figure 14) is performed and then iterative corrective calculations follow to tune the result for the negative temperature range. This iteration is performed by taking the result of a prior calculation and using it in the following equation and can be done by linking several Math Icons which will be sequentially executed during Program Net execution.

The iterative calculation general form equation is as follows:

$$\frac{R_T - R_0}{\alpha * R_0} + \delta * \left(\frac{T_1}{100} - 1\right) * \left(\frac{T_1}{100}\right) + \beta * \left(\frac{T_1}{100} - 1\right) * \left(\frac{T_1}{100}\right)^3 = T$$

Where:

T_1 = Prior resistance to temperature solution (to be refined through iterations)

Other constants are as defined previously in this Application Note

Substituting constants from our application using standard values for constants from Table 1 and changing the equation to an inline format results in the following equation:

(R_T-1000)/(0.00385*1000)+1.4999*((T_1/100)-1)*(T_1/100)+0.10863*((T_1/100)-1)*(T_1/100)^3)

IMPLEMENTING ITERATIVE CALCULATIONS WITHIN A PROGRAM NET

Before entering the above equation into a Math Icon, the above equation needs two inputs to perform the calculations, the calculated RTD resistance (R_T) from an upstream Math icon and the prior calculated RTD temperature (R_T) from an upstream Math Icon. In construction of the Program Net, we will connect the calculated RTD (R_T) resistance to the Math Icon terminal "W" and the previous Math Icon's calculated RTD temperature ($T_{1)}$ to terminal "X". After the above substitutions the resulting equation below is then entered into the Math Icon Function field in the Math Icon Configuration Dialog

(W-1000)/(0.00385*1000)+1.4999*((X/100)-1)*(X/100)+0.10863*((X/100)-1)*(X/100)^3

> *Note: This equation can simply be copied and pasted from this document into a Math Icon Formula field within HyperWare-II if the Math icon input terminals are connected as described above and assuming a 1000Ω, 0.00385 RTD is used.*

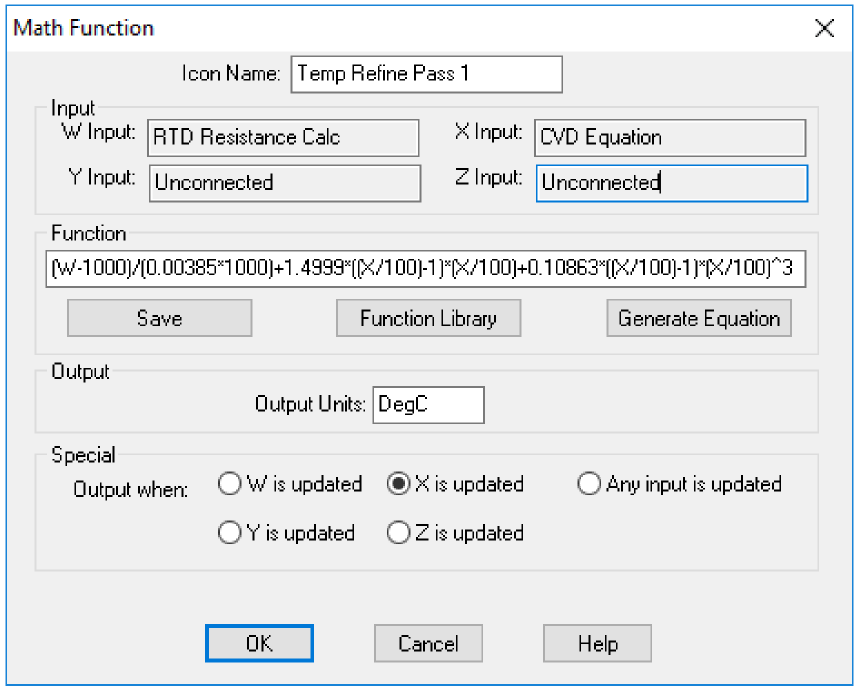


Figure 18: Math icon with Iterative Temperature correction equation entered

The Program Net is then built with multiple iterative Math icons as shown in

Figure 19. The RTD resistance is calculated from the measured RTD voltage and currents in the Math icon *RTD Resistance Calc* and fed into another Math icon *CVD Equation* where the Callendar-Van Dusen equation is applied to make a first pass

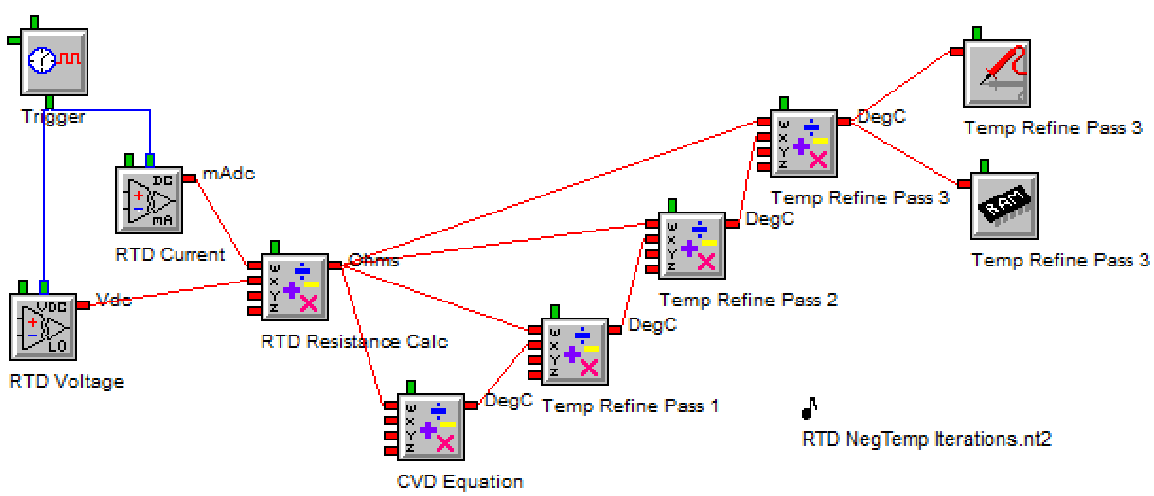


Figure 19: Program Net for Negative Temperature RTD input with iterative corrections

temperature calculation. This temperature output is then fed into ensuing Math icons along with the RTD resistance and the iterative accuracy refining calculations are performed. Increasing the number of iterative

Math icons generally improves the accuracy. To quantify the number of iterations required to meet desired accuracy, Probe Point icons can be applied on the output of each iterative Math icon and with a known simulated RTD input, the level of correction from each iteration can be observed while the IntelliLogger is executing.

Hybrid High Accuracy RTD Measurements Encompassing Positive and Negative Temperature Range

Note that this Program Net (Figure 19) is ONLY applicable for negative temperatures and use for positive temperatures will result in excessive conversion errors. If the RTD operational range is going to encompass both positive and negative temperatures (below approximately -50C where the basic Callendar-Van Dusen equation accuracy falls off) then a "hybrid" Program Net can be employed. This hybrid Program Net automatically transitions between a simple Callendar-Van Dusen equation calculation (for positive temperatures) and an iterative Callendar-Van Dusen equation calculation (for negative temperatures).

The hybrid Program Net can be implemented as shown in Figure 20.

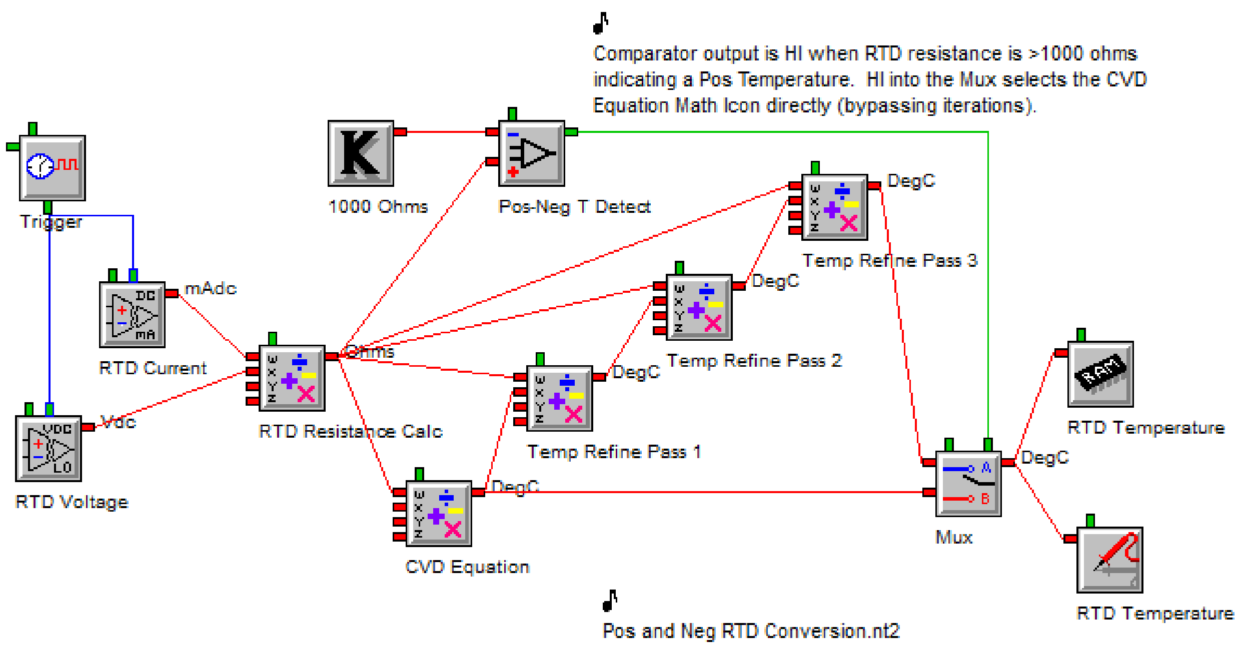


Figure 20: Positive and Negative RTD Range Calculation Program Net

In this hybrid Program Net, the Mux icon is used to route either a simple Callendar-Van Dusen equation output (for positive temperatures) or an iterative Callendar-Van Dusen equation branch output (for negative temperatures) to the RTD Temperature Memory icon. A Comparator icon *Pos-Neg T Detect* compares the calculated RTD resistance to the 0°C (R_0) resistance. If the resistance is >1000Ω (indicating a positive temperature), the Mux selects the *CVD Equation* Math icon output… if <1000Ω (a negative temperature), the *Temp Refine Pass 3* Math icon output is selected and routed to the *RTD Temperature* Memory and Probe Point icons.

SUMMARY AND CONCLUSIONS

Several methods of acquiring RTD temperature measurements with the IntelliLogger are possible. The following summary can provide user direction on the method to employ for their RTD application.

All methods require the accurate determination of the RTD resistance which as explained above and involves measuring the voltage across and the current through the RTD then using basic math to calculate the RTD resistance. Once the RTD resistance is known, one of the following resistance to temperature conversion methods (as detailed above) can be used with the IntelliLogger:

LINEAR APPROXIMATION

Use of a straight line calculation to model the RTD resistance to temperature conversion.

- Good for positive and negative temperatures as well as crossover ranges containing both positive and negative temperatures.
- Limited temperature band use. Width of temperature band is determined by the acceptable accuracy.
- Conversion accuracy decreases as the conversion temperature band expands. Over a 100°C band (eg 50° to 150°C), +/-0.1°C accuracy can be approached.
- Conversion error will be non-linear over temperature band. Calibration points defining the line will have 0°C error and the midpoint will have maximum error.
- Simple math to implement

SINGLE PASS CALLENDAR-VAN DUSEN EQUATION APPROXIMATION

- Good for positive temperature ranges and ranges extending to approximately -70C
- Excellent consistent accuracy over range
- Conversion error will be fairly linear over temperature range
- Fairly simple math to implement

ITERATIVE CALLENDAR-VAN DUSEN EQUATION APPROXIMATION

- Good for negative temperature ranges
- Excellent consistent accuracy over range
- Conversion error will be fairly linear over temperature range
- More complex math and Program Net implementation

HYBRID CALLENDAR-VAN DUSEN EQUATION APPROXIMATION

- Good for positive and negative temperatures as well as crossover ranges containing both positive and negative temperatures.

- Excellent consistent accuracy over range
- Conversion error will be fairly linear over temperature range
- More complex math and Program Net implementation

ABOUT LOGIC BEACH INCORPORATED…

Since 1986, Logic Beach Incorporated has been inventing, designing and manufacturing unique data acquisition, alarming and reporting instruments. With the unique programmability provided by our HyperWare-II™ icon-based programming software, our instruments readily handle complex data acquisition, alarming and reporting requirements yet are intuitive and quick to program and deploy. With flexible I/O, our instruments can easily interface to the array of industrial transducers and sensors …. including Modbus.

Our instruments are employed in environmental, industrial, process, R&D, energy, water/wastewater and many other more exotic data acquisition applications.

Please contact Logic Beach about your next data acquisition application.

sales@logicbeach.com

www.logicbeach.com

Document Revision History (AN-401 RTD Interface to the IL)

10-25-2017: First release