Monitor and Control of Temperature in a Closed Environment

Software Design Document

Jameson Clark

December 18, 2015

Table of Contents

1) **Program Outline**

a) *Purpose*

This software design document defines the application, hierarchical structure, and usability of monitoring and controlling the temperature inside a given environment.

b) *Scope*

This program is designed to control and maintain a given temperature in a system by allowing the user to set the desired temperature, in either Celsius or Fahrenheit, and then walk away. The program will also allow for a custom tolerance range and will display the current temperature and plot the temperature vs time. The user also can opt to have the temperature and time data output to a data file.

c) *Useful Definitions*

User interface: The set of controls and indicators present on the computer monitor which the user inputs and monitors their desired set temperature and current temperature.

Thermistor: A resistor whose resistance is variable with temperature surrounding it.

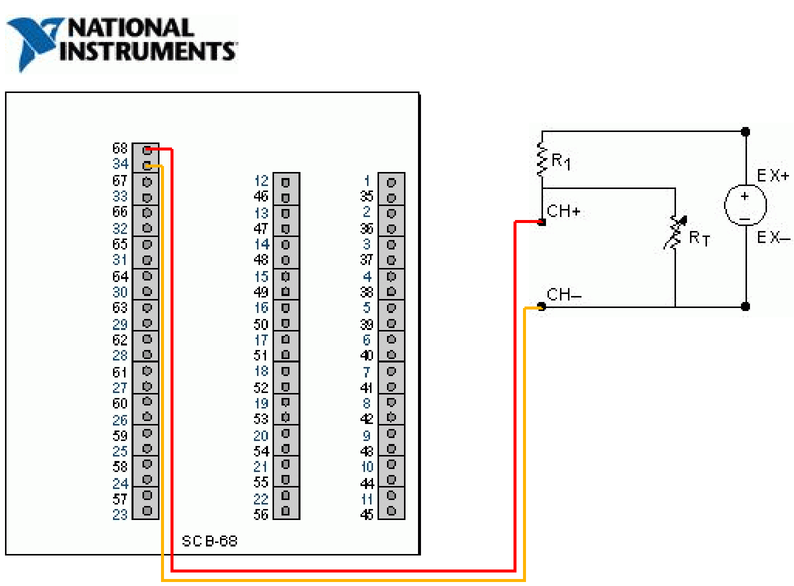
Peltier Cooler: A Peltier Cooler is a type of thermoelectric cooler (TEC) which draws heat from one of its sides to the other when an electric current is applied to it.

DAQ: Data Acquisition device. Used in this instance to read voltage and output two voltages, all on different channels and in separate circuits.

2) **System Overview**

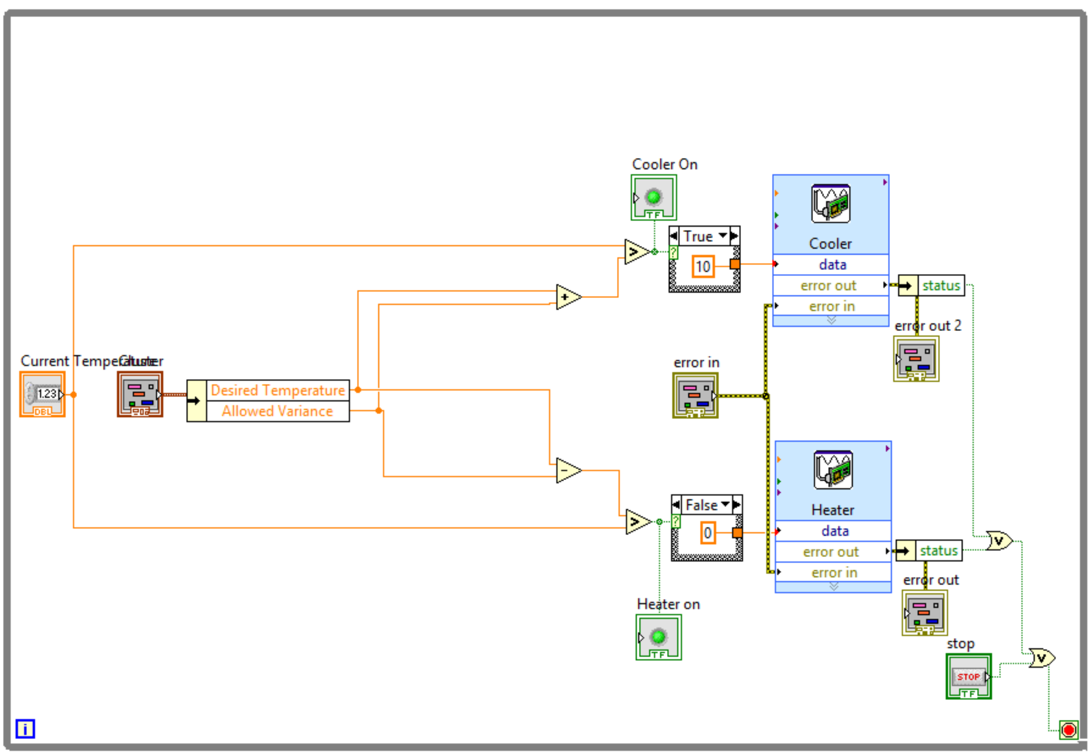
The program works in two different ways for the two different aspects, monitoring and controlling the temperature.

In order for the software to monitor and provide feedback on the current state of the system a thermistor is placed inside of the desired environment and connected to a constant voltage source at 5V. Since the thermistor's resistance is variable with temperature it can be calibrated so that we know the resistance at certain temperatures. This is done by using the calibration feature of the DAQ express controller and then applying known temperatures to current readings of the system. The more known points the better but at least two are needed. This calibration just needs to be saved and applied to be complete. Once the relationship between temperature and resistance is known we can simply convert the measured resistance of the environment into its current temperature. The resistance of the thermistor is determined by measuring the voltage drop across it. The circuit for this system, has three parts, the voltage source, a standard resistor, and the thermistor. The resistor value must be known and it must be in series with the thermistor. The voltage source is constant and the voltage drop across resistors in series is proportional to the value of each resistance over the total resistance so the voltage drop will change across each resistor as temperature changes the thermistor's resistance. This is how the temperature is measured. A lead is placed on each end of the thermistor and is read by the DAQ, although the leads could also be placed on each end of the static resistor and we could get the same results. Then every iteration of the program measures the voltage across the thermistor and with the calibration made previously the temperature is known. Below is a diagram of the proper setup taken directly from the National Instruments DAQ assistant page.

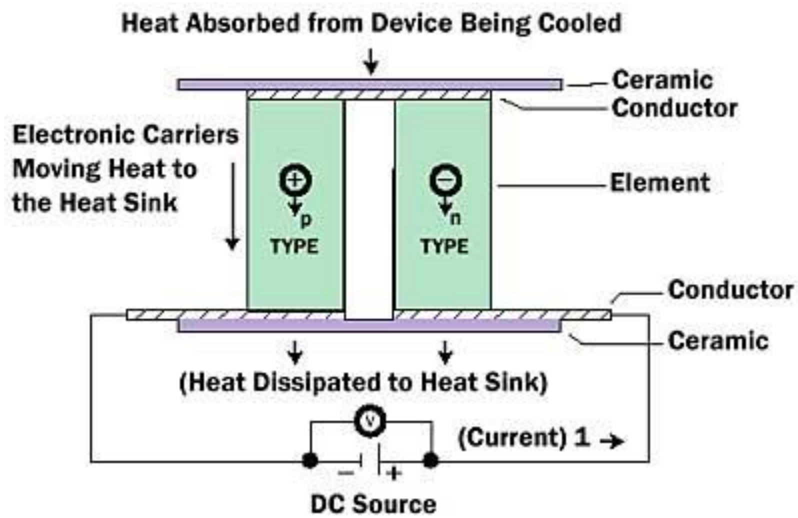


Once the resistance is determined it is output to the UI as a value, as a visual temperature gauge, sent to the input of the control program, and finally sent to the temperature vs. time graph.

The other important aspect is the control of temperature in the system. Once enabled by the detection of a difference between the real world temperature and the desired temperature the program will output an on/off command to the appropriate thermoelectric coolers to either heat up, or cool down. This is done by enabling a voltage output when heating/cooling is needed and providing 0 voltage when it isn't. The module will then be disabled as the temperature measured is equal to the desired. The decision to enable or disable one of the two Peltier coolers is made by taking the maximum and minimum value allowed by the range of desired temperature +/- the allowed variance and then running a check to see if the current temperature is below the minimum, or above the maximum. If one of these returns true then a voltage is applied to the appropriate circuit and the Peltier cooler is activated. The same true/false value activates a light on the UI to let the user know the control has been activated. This process is expressed logically as the program sees below.



The Peltier cooler is a series of P and N type semiconductors, each with one end applied to the "cold" side and one attached to the "hot" side. The current flows "up" the N type semiconductor, so the free holes oppose the current and flow "down." On the other hand, the orientation of the materials is such that the current flows "down" the P type semiconductor and so the free electrons also move "down". It is the actions of both the free electrons and the free holes moving in the same direction that draws energy and so heat from the cold "up" side to the hot "down" side. The following diagram represents the structure of these coolers and is from electronics-cooling.com"s November 1, 2005 article "Advances in High Performance Cooling for Electronics."



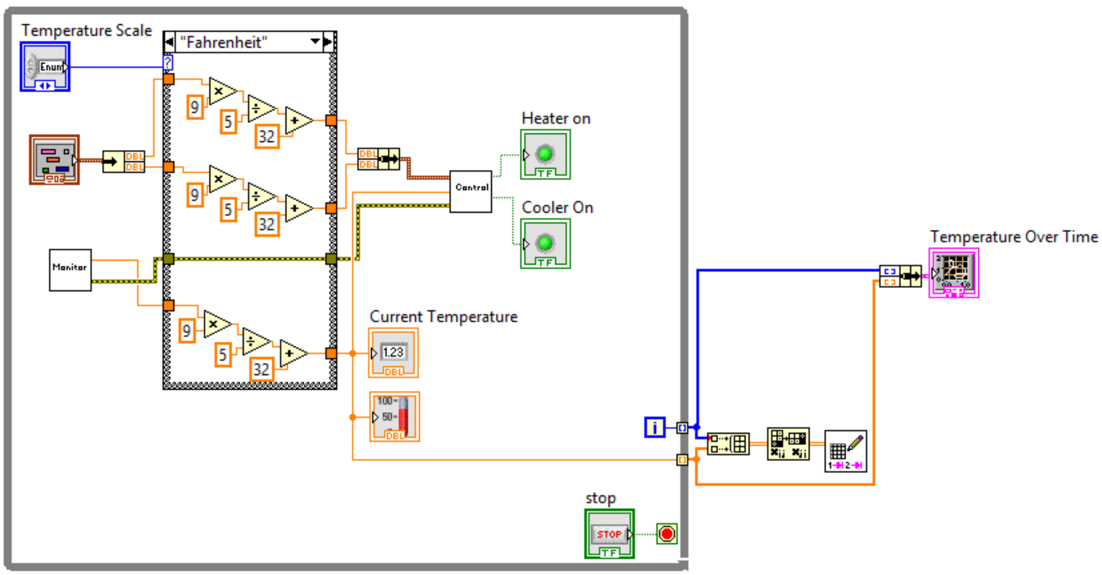
Because there is current flowing through the Peltier coolers there will be extra heat generated which will cause the cooler to become less and less efficient as time goes on, to the point where if left on long enough there will be no cooling effect and both sides will end up radiating the excess heat. This effect is proportional to the amount of current being applied to it and the time that the current has been flowing. This affects the range of temperatures that this program can control because the Peltier cooler designated to keep the system at a lower temperature cannot be activated indefinitely, whereas the warming Peltier cooler can be used for long periods of time and will still input heat into the system as desired.

The Coolers are placed in the system in opposite orientations. The one used to cool the system when necessary is oriented such that the cold side is inside of the system and the hot side is outside of it. On the other hand the one used to heat the system is oriented so that the warm side is within, and the cold on the outside.

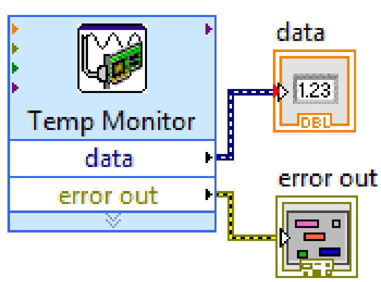
3) **Program Structures**

a) Hierarchy and Sub Structures

The initial step in the program is the user interface input. This is where the user will input their desired temperature in their desired scale which will be available in either Celsius or Fahrenheit. On the back end the necessary conversions will be made to the desired scale to allow input to the temperature control and proper readings on the output. A value of tolerance can be input if constant corrections are not necessary. Finally the user can decide whether or not to record the temperature data and output to a desired file name.



The second step in the program is to monitor and output the current temperature readings. This step will simply measure temperature in Celsius and send that information to the main structure for conversion and display.



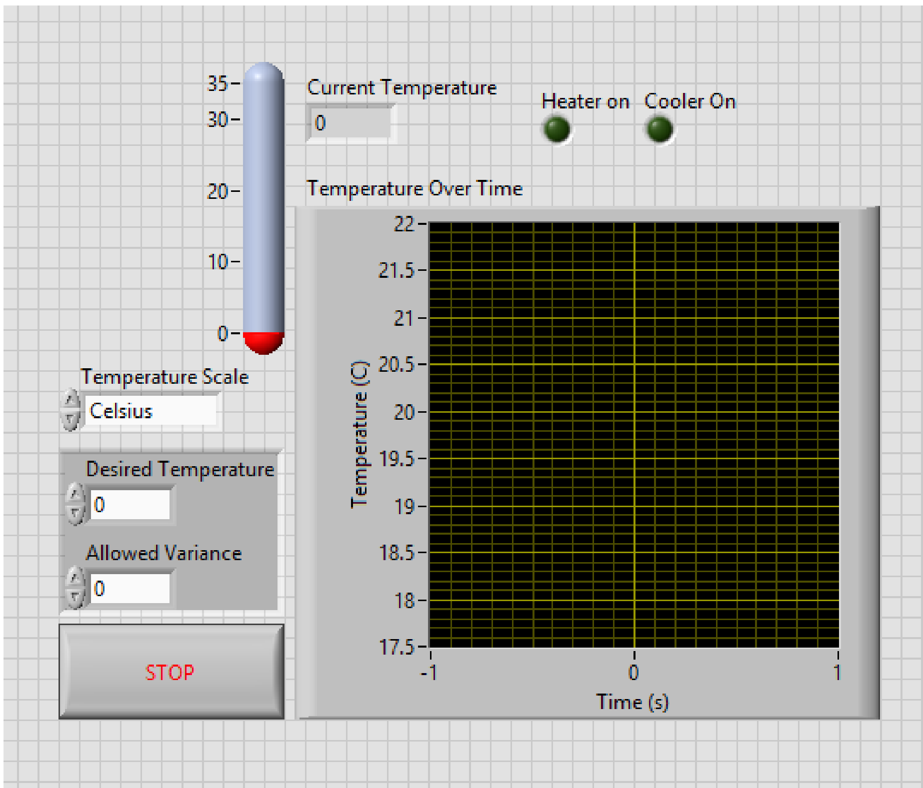
The final step is to alter the temperature. This is done by taking the temperature reading from step two and will either heat up or cool down depending on if the current value is above or below the desired temperature. If an acceptable threshold was selected on the user interface in step one then the temperature will not be corrected until it falls outside the plus or minus value at which point the temperature will be changed until the desired temperature is achieved and then the manipulation will cease until it falls outside of the range again. The program structure for this step was presented previously in the system overview.

b) Structure Motivation

The structure outlined above was chosen out of necessity. In order to control an object the current state and desired state must first be known so the control module must come last. The first and second steps would be interchangeable if it weren't for the inclusion of the user's option to save and output the results of the temperature measurements and to change between scales. Since this option is included then necessarily the user's input must come before the measurement. Hence the structure outlined above is determined.

4) **User Interface**

The User interface has many features which first and foremost are the controls for desired temperature, allowed variance, and scale. Along with the stop button. Since the program is useless without these initial inputs they are placed prominently as the first items on the left hand side. The desired temperature and allowed variance can either be typed into their respective boxes or the values can be changed by single degree amounts by pressing the up or down arrows net to the appropriate box. The temperature scale options are chosen by selecting up or down on the arrows to go between the two options.



It is of note that should the user wish to only monitor temperatures but not influence them that the program can be manipulated by intelligent selection of both the desired temperature and tolerance controls. If the desired temperature is set to a reasonable value for the system and the tolerance is set to a very large value such that variation in actual temperature would never approach it, then the control module will never engage.

The final aspects the user will notice are the current temperature value display and the plot of values for the temperature since the program was initiated.