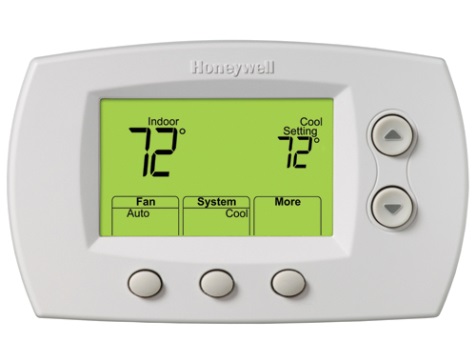
**Lab 9: Design of a Digital Thermostat**

In this lab, we will again be using the National Instruments myDAQ device and your knowledge of MATLAB to design a digital thermostat. Historically, thermostats worked based on the properties of different materials. A bar with two different metals was used, with each metal expanding at a different rate depending on temperature. As the temperature increased or decreased, the bar would bend in different directions to either engage the furnace for heating or the air conditioner (AC) for cooling.



**Figure 1: Bimetalic Thermostat**

With the increasing prevalence of small-scale computing systems and the design of electronic sensors for measure a variety of physical attributes, such as temperature, thermostats have moved away from the old bimetal systems to digital systems. In the digital system, the voltage produced by a temperature sensor is measured using a small computing system, which is then converted into a temperature, which is used to determine whether to turn on the furnace or the AC. The user of the thermostat can often program different profiles for different days and different temperatures for different times of the day. Many new digital thermostats are also equipped with WIFI, allowing the user to program the thermostat using his/her smart phone.



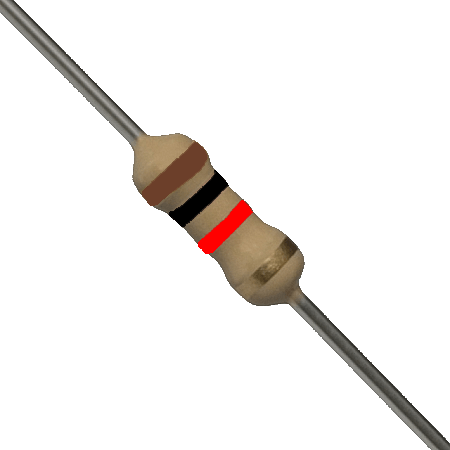
**Figure 2: Example of a Digital Thermostat**

In order to create your thermostat system, you will be required to use your knowledge of conditional structures and loops. You will work in groups of three to design and test your system.

1. **Building the Circuit**

The circuit you must build for this lab is shown below in figure 3. If you are not adept at reading a circuit diagram, follow the picture steps below to create the circuit. To construct it, you will need the following items:

* 2x Light Emitting Diode (LED)
* 3x 1kΩ resistor
* 1x LM335 temperature sensor
* 1x yellow connector wire
* 1x orange connector wire
* 2x red connector wire
* 1x blue connector wire
* 3x black connector wire



+5V

AO0

AO1

AI0+

AI0-

AI1+

AI1-

AGND

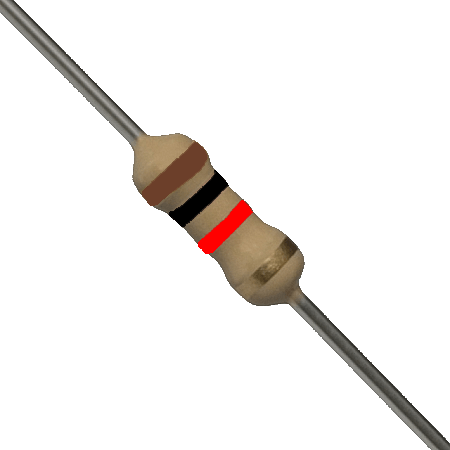
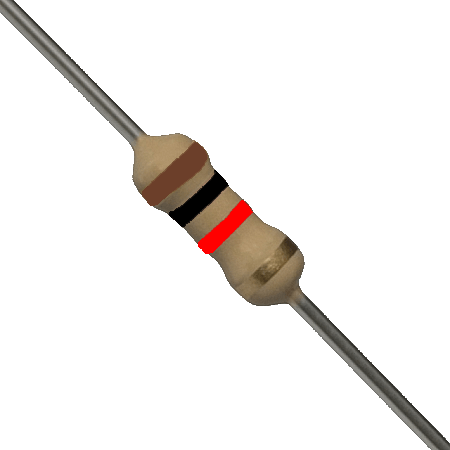
DGND

LED - AC

1kΩ resistor



LED - Furnace



LM335

**Figure 3: Circuit Diagram**

***Notes:***

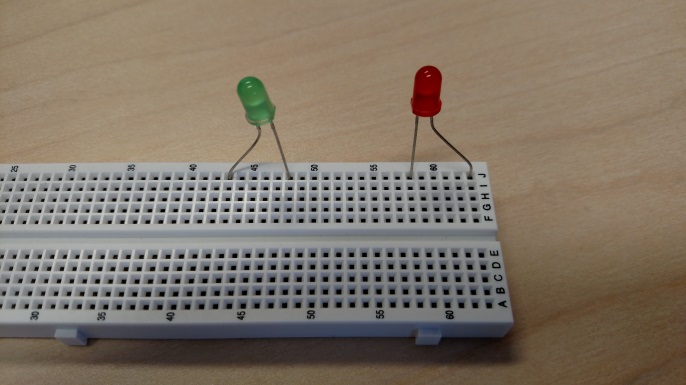
* Just like the LED that was used in Lab 4, LEDs have a polarity. Connect the longer lead from the LED to the Positive (AO) connections and the shorter lead to the connections with the 1kΩ resistors.

**Step-by-Step Circuit Construction:**

1. Connect your cables to the National Instruments myDAQ. Note that you will use a wire to connect the AI0- and AI1- terminals to the AGND terminals using a black wire. This is done because we will be using the analog input channels to measure the voltage with respect to ground.

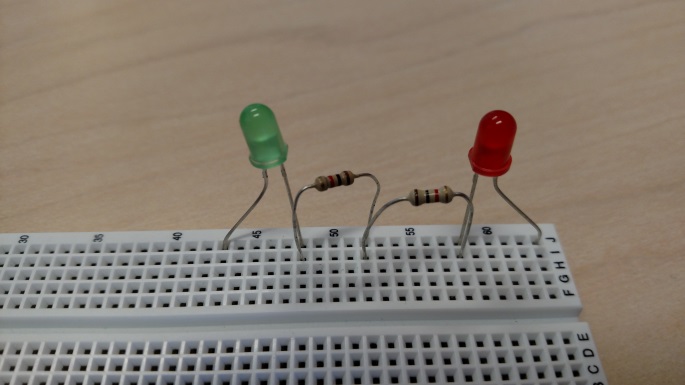
|  |  |  |
| --- | --- | --- |
| **NI myDAQ Connection** | **Cable Color** |  |
| AGND ↔ AI0- | **Black** |
| AO0 | **Blue** |
| AO1 | **Red** |
| AGND ↔ AI1- | **Black** |
| AI0+ | **Yellow** |
| AI1+ | **Orange** |
| 5V | **Red** |
| DGND | **Black** |

1. Begin constructing the circuit by placing the LEDs on the breadboard. Make sure you know which side is the longer side, and place that lead farthest away from the other LED.



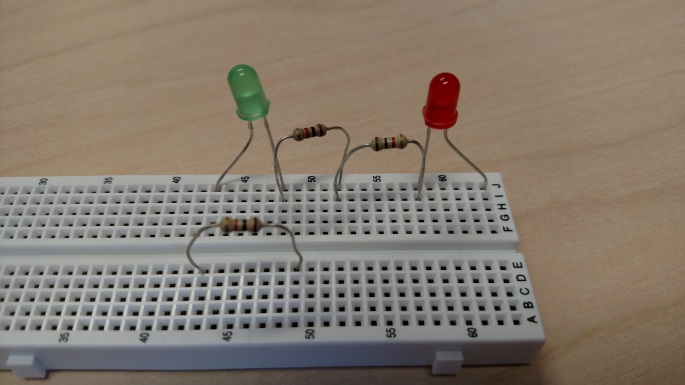
Longer lead

1. Place two 1kΩ resistors on the board, connecting one side of each resistor to the short lead on the LED and the other side of the resistors to the same row on the breadboard.

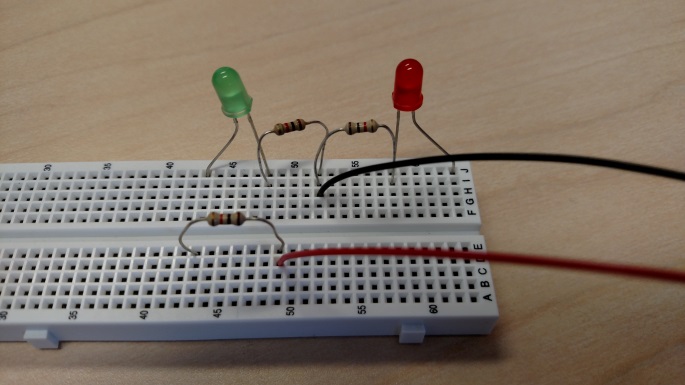


Same row

1. Place the last 1kΩ resistor on the opposite side of the breadboard from the LEDs.



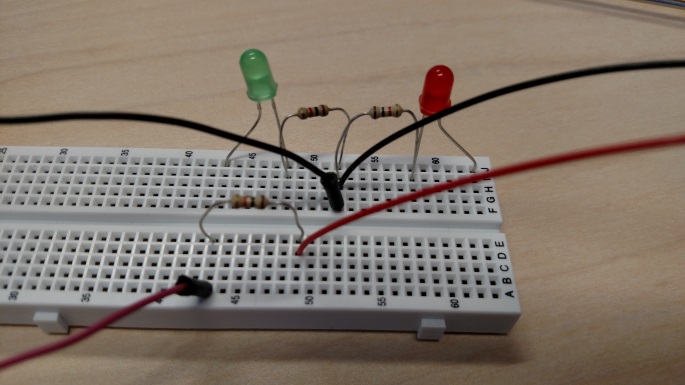
1. Connect the temperature sensor by placing the red wire into the same row as the 1kΩ resistor placed in the previous step and the black wire into the same row as the connection of the two 1kΩ resistors placed in step 3.



Black wire to temp sensor

Red wire to temp sensor

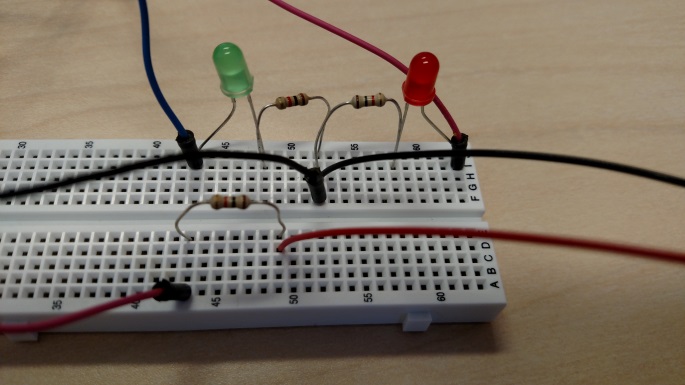
1. Connect the 5V wire from the myDAQ to the opposite side of the 1kΩ resistor from the temperature sensor and the DGND wire from the myDAQ to the row of the point of connection with the two 1kΩ resistors and the black wire of the temperature sensor.



DGND wire

5V wire

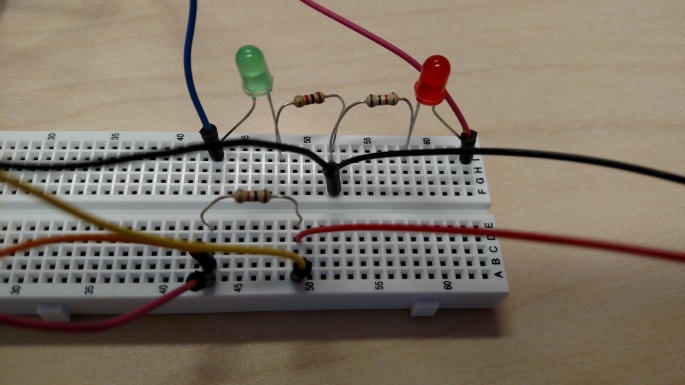
1. Connect the AO0 wire from the myDAQ to the long side of one of the LEDs and the AO1 wire from the myDAQ to the long side of the other LED.



AO1 wire

AO0 wire

1. Finally, connect the AI0+ wire from the myDAQ to the row with the red wire from the temperature sensor and connect the AI1+ wire from the myDAQ to the same row with the 5V wire.



AI0+ wire

AI1+ wire

1. **Software Check for National Instruments myDAQ**

***Note: if you used your computer on Lab 4 and you have not uninstalled any software, you can skip to step 4 to check MATLAB communication with the DAQ***

1. Form teams of 3 students at your recitation tables making sure that at least one team member does not have a Mac because the hardware will not work with a Mac (unless you have a Windows partition with the drivers installed).
2. Ask your T.A. to give you one of the National Instruments myDAQs.
3. Hopefully, at least one of your team members installed the National Instruments Driver file last week during recitation or in Engineering Foundations. If not, have one team member (without a Mac computer) go to the Recitation Folder on the Blackboard metasite and follow the instructions for the MATLAB Update.
4. Software Check: Type the following command in MATLAB:

>> d = daq.getDevices

If you get an output like this then you are good to go:

d =

ni: National Instruments NI myDAQ (Device ID: 'Dev1')

Analog input subsystem supports:

-2.0 to +2.0 Volts,-10 to +10 Volts ranges

Rates from 0.1 to 200000.0 scans/sec

2 channels ('ai0','ai1')

'Voltage' measurement type

**⁞**

**If you get the message that No Data Acquisition Devices Are Available, then you have not installed the National Instruments Drivers file correctly.** Try reinstalling the National Instruments Drivers. Once the drivers are installed, try this MATLAB command once again:

>> clear all; d = daq.getDevices

***NOTE: If you lose connectivity with the DAQ at any point, type*** clear all ***at the command prompt. This will clear out the old session and allow you to start a new one.***

1. **Simple DAQ Input/Output Commands**
2. Make sure that you have the circuit wired and connected to the National Instruments myDAQ following the diagrams shown in Part A. Run the following commands by copying them into a script file and running it:  
     
   devices\_found = daq.getDevices;

DAQ\_ID = devices\_found.ID ;

DAQ = daq.createSession('ni');

measurement1 = DAQ.addAnalogInputChannel(DAQ\_ID,0,'Voltage');

Measurement2 = DAQ.addAnalogInputChannel(DAQ\_ID,1,'Voltage');

AC = DAQ.addAnalogOutputChannel(DAQ\_ID,0,'Voltage');

FURNACE = DAQ.addAnalogOutputChannel(DAQ\_ID,1,'Voltage');

These commands will set up a new communication session with the myDAQ, adds two input channels to allow you to read voltage measurements, and adds two output channels to control the LEDs for the AC and Furnace.

1. Execute the following line of code at the command prompt:  
     
   >> DAQ.outputSingleScan([10,0]);  
     
   Record what happens to the LEDs – check you circuit:

|  |
| --- |
| The LED- AC lit up and the LED-Furnace stayed unlit. |

1. Execute the following line of code at the command prompt:  
     
   >> DAQ.outputSingleScan([0,10]);  
     
   Record what happens to the LEDs:

|  |
| --- |
| The LED- Furnace lit up and the LED-Furnace stayed unlit. |

1. Execute the following line of code at the command prompt:  
     
   >> DAQ.outputSingleScan([10,10]);  
     
   Record what happens to the LEDs:

|  |
| --- |
| Both the LED-AC and LED-Furnace lit up. |

1. Execute the following line of code at the command prompt:  
     
   >> DAQ.outputSingleScan([0,0]);  
     
   Record what happens to the LEDs:

|  |
| --- |
| Neither the LED-AC nor the LED-Furnace lit up. |

1. Execute the following line of code at the command prompt:  
     
   >> Status = DAQ.inputSingleScan  
     
   Record the value of Status

|  |
| --- |
| Status = 2.9674 4.7803 |

1. Move your orange AI1+ wire to the row with your DGND cable. Execute the following line of code at the command prompt:  
     
   >> Status = DAQ.inputSingleScan  
     
   Record the value of Status

|  |
| --- |
| Status = 2.9671 0.0001 |

1. Move your orange AI1+ wire back to the row with the red 5V wire
2. Check with your T.A. to see if all of your answers for this part are OK!

|  |
| --- |
| **Explanation:**  The inputSingleScan and outputSingleScan commands allow you to read data from your input lines or send signals out to your output lines, respectively. If you have multiple input lines, the inputSingleScan command will return the values in a vector for each input line (channel 0 first, channel 1 second). If you have multiple output lines, you will need to provide values for each line in order to use outputSingleScan. However, as you saw, the voltage will remain at the level set until you change the value again. |

1. **Setting up your Temperature Sensor**

As was mentioned in the background information, the temperature sensor will output a voltage depending on the temperature it is experiencing. Follow the steps below to correctly convert your voltage reading into a temperature in Fahrenheit.

1. Make sure that your temperature sensor is at room temperature (i.e. you haven’t been holding it or giving it an icy glare), then start by running an input scan to get the current voltage reading of your sensor. Find the thermostat in the room and record the temperature. The temperature at your specific location may be a bit different, but we’ll assume the room is the same temperature throughout.

|  |  |
| --- | --- |
| **Temperature Sensor Voltage:** | 2.9694 |
| **Room Temperature (in oF):** | 69 °F |

1. The voltage of the temperature sensor changes by 10mV per degree Kelvin. Since Celsius and Kelvin change at the same rate, we can use Celsius instead of Kelvin to set up our system. Convert the room temperature you read off of the thermostat in the room to Celsius.

|  |  |
| --- | --- |
| **Room Temperature (in oC):** | 20.6 °C |

1. The voltage output of the temperature sensor can be expressed using the following equation:

where T is the current temperature (in oC), To is the reference temperature (measured in step 1, in oC), Vo is the reference voltage for To (in V), and Vsensor is expected voltage measurement for the current temperature (in V).

While knowing what the voltage will be based on the current temperature is nice, we actually need to go in the opposite direction: we will measure the voltage and we need to turn that value into a temperature. Solve the Vsensor equation for T and plug in your values for the reference voltage and temperature:

|  |  |
| --- | --- |
| **Expression for current temperature based on voltage measurement:** |  |

1. **Creating the Thermostat System**

Download the script Thermostat\_Template.m from Blackboard and complete the required sections using the information from parts C and D and the descriptions below.

1. In the section marked PART E.1, write commands that will ask the user (1) for the desired temperature to maintain and (2) for the acceptable range around the desired temperature.
2. In the section marked PART E.2, you will write the majority of the commands to implement your thermostat system. The code will need to perform the following operations:
   1. Continue to perform the following steps for the thermostat system until the AI1+ channel drops below 2.5, indicating that the thermostat has been turned off.
   2. Measure the voltage on the temperature sensor and convert it to a temperature in Fahrenheit (remember that the temperature in the equation you created before is in Celsius!).
   3. Display the current temperature to the command window (the **clc** command may be useful here).
   4. Based on your temperature, do the following:
      1. If the temperature rises above your acceptable range, turn on the AC by illuminating the LED on line AO0 and displaying a message to the user that the AC is on.
      2. If the temperature falls below your acceptable range, turn on the Furnace by illuminating the LED on line AO1 and displaying a message to the user that the furnace is on.
      3. Once the AC or the Furnace is on, you should only turn them off once the temperature gets back to desired temperature. This means that to turn of the AC, the temperature must be at or below the desired temperature. Similarly, for the furnace, the temperature must be at or above the desired temperature.
   5. Include the following command to pause the program for 1 second between voltage measurements:

**pause(1);**

1. In the section marked PART E.3, write a command that will send out a 0V value to both LEDs and will also display a message to the user that the thermostat is no longer running.
2. To test your script, you will need a way to both heat up and cool down the temperature sensor. To heat up the sensor, you can use your fingers or hand (unless you’re a White Walker) and to cool it down, you will need to fill up a cup of water from the local drinking fountain in which you can stick the sensor.

***Notes:***

* When running and testing your system, you will always want to make sure you have reset your AI1+ 🡪 DGND. Otherwise, your code might be right but your thermostat will never start up because you have it turned off initially!
* Use the DAQ.inputSingleScan and DAQ.outputSingleScan commands to read your voltage measurements and to set the voltage level of the LEDs.

***When you have your thermostat system working,***

***demonstrate it to your TA.***

**Paste your script below:**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lab\_09\_Thermostat\_2.m - <Team Members> - <Date>

% Description: This script will simulate the operation of a thermostat,

% utilizing the NI myDAQ and a simple LM335 temperature sensor. The

% voltage from the sensor will be read in and converted to a temperature

% value. The script will then determine, based on the temperature, whether

% to turn on the air conditioning (if it is too hot) or the furnace (if it

% is too cold), which are indicated by LED's.

% Usage: >> Lab\_09\_Thermostat;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Circuit Connections:

% 5V used to power temperature sensor

% AO0 connected to Red LED

% AO1 connected to Green LED

% AI0+ connected to point between reistor and temperature sensor

% AI0- connected to AGND

% AGND used for circuit grounding

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% move to command window

commandwindow;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% clean up

clear all;

close all;

clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% DAQ setup

fprintf('Setting up connection to DAQ...\n');

% check for connected device, if no device found, inform user and terminate

devices\_found = daq.getDevices;

if isempty(devices\_found)

error('No devices connected: please connect a DAQ device and rerun.');

end

% set up DAQ

DAQ\_Vendor = devices\_found.Vendor.ID; % determine type of device connected

DAQ\_ID = devices\_found.ID; % determine ID for connected device

DAQ = daq.createSession(DAQ\_Vendor); % create DAQ session

if strcmp(DAQ\_Vendor,'digilent')

DAQ.addAnalogOutputChannel(DAQ\_ID,1,'Voltage'); % add output voltage channel to turn on furnace light

DAQ.addAnalogOutputChannel(DAQ\_ID,2,'Voltage'); % add output voltage channel to turn on AC light

DAQ.addAnalogInputChannel(DAQ\_ID,1,'Voltage'); % add input voltage channel to check temperature value

DAQ.addAnalogInputChannel(DAQ\_ID,2,'Voltage'); % add input voltage channel to check thermostat off

else

DAQ.addAnalogOutputChannel(DAQ\_ID,0,'Voltage'); % add output voltage channel to turn on furnace light

DAQ.addAnalogOutputChannel(DAQ\_ID,1,'Voltage'); % add output voltage channel to turn on AC light

DAQ.addAnalogInputChannel(DAQ\_ID,0,'Voltage'); % add input voltage channel to check temperature value

DAQ.addAnalogInputChannel(DAQ\_ID,1,'Voltage'); % add input voltage channel to check thermostat off

end

DAQ.outputSingleScan([0 0]);

fprintf('Connection established...\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SECTION E.1

DesiredTemp = input('What temperature (°F) would you like to maintain? ');

RangeMax = input('What is the max temperature (°F) you would like? ');

RangeMin = input('What is the min temperature (°F) you would like? ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SECTION E.2

Status = DAQ.inputSingleScan;

TempC = Status(1)\*100-276.38;

while Status(2) > 2.5

Status = DAQ.inputSingleScan;

TempC = Status(1)\*100-276.38;

TempF = ((9/5)\*TempC) + 32;

clc;

fprintf('The Temperature is %0.2f (°F)\n',TempF);

if TempF > RangeMax

DAQ.outputSingleScan([10,0]);

elseif TempF < RangeMin

DAQ.outputSingleScan([0,10]);

else

DAQ.outputSingleScan([0,0]);

end

pause(1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SECTION E.3

DAQ.outputSingleScan([0,0]);

fprintf('The Thermostat is no longer running!!!\n');

**To be turned in:**

* **Everyone** should submit the m-file your group created during this lab activity, Thermostat\_Template.m, and a copy of this lab report with the all calculations, measurements, and observations entered. The team member who has this information should e-mail these documents to the other members of his/her team.
* Please place the names of your team members in the header of your script file, at the beginning of your Word document, and in the submission box.