Experiment M3
Thermostat
Procedure

Deliverables: checked lab notebook, tech memo, demonstration of working device to Lab TA
Recommended Reading: Sections 6.1 and 14.2, Example 14.1 in textbook

Overview

A thermostat is the simplest way to control the temperature in an engineering system. A thermostat will actuate either a heater or a cooling system when the system temperature reaches a certain value known as the "set point". Cars, computers, refrigerators, ovens, clothes dryers, hot water heaters, HVAC systems and many other familiar consumer products all contain thermostats.

Thermostats typically use two set-points: a high set-point T_{HS} and a low set-point T_{LS}. For example, the thermostat in your house turns ON the heater when the temperature goes below the low set-point T_{LS}. It then turns OFF the heater when it gets above the high set-point T_{HS}. The difference between the two set-points is known as *hysteresis*, and it allows engineers to dictate how frequently the system cycles ON and OFF.

In this lab, you will create a simple thermostat. A sample will be heated, and its temperature will be measured by a thermistor. When the temperature rises above the high set-point T_{HS}, a blower fan will turn ON to cool the sample. Then, when the sample cools to a temperature below the low set-point T_{LS}, the fan will turn OFF. You will learn how to implement the thermistor and fan, and integrate them with an Arduino microcontroller. Then you will experimentally observe how hysteresis effects the frequency of ON/OFF fan cycles.

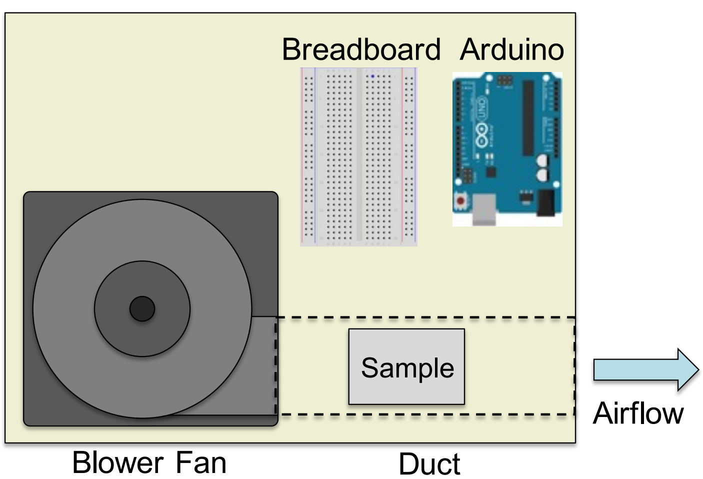

Figure 1 – A thermostat system consists of a heated sample in an air duct. The temperature of the sample is measured by a thermistor connected to an Arduino microcontroller. When the sample becomes too hot, the Arduino turns on a blower fan to cool the sample.

Subsystem A: Thermistor

A thermistor will be used to measure the temperature of the sample. Recall that a thermistor is a transducer whose electrical resistance changes with temperature. This particular thermistor has a negative temperature coefficient (NTC), which means that its resistance decreases as it gets hotter.

The thermistor will be wired up in the voltage divider circuit shown in Fig. 2. The Arduino microcontroller will use its analog-to-digital convertor (A/D) to read the voltage output V_{out} from this circuit.

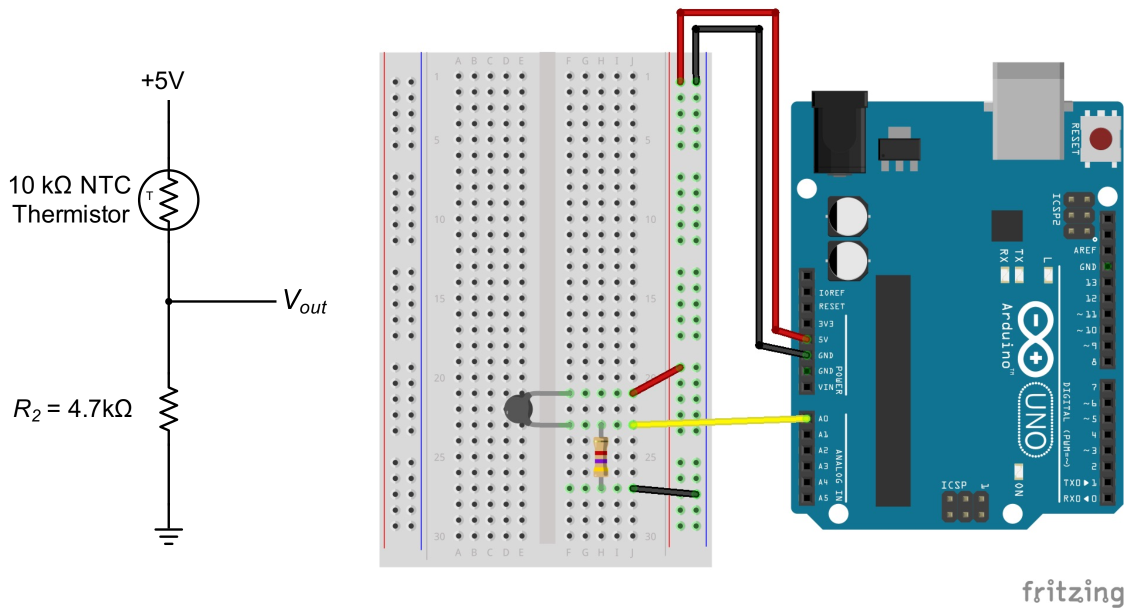

Figure 2 – A thermistor is wired up in a voltage divider circuit. Its output voltage V_{out} is read by the A0 analog input of the Arduino.

Procedure

1. Sketch the complete thermostat system in Fig. 1 in your lab notebook.
2. Locate the sample. It is an aluminum plate with black and red wires hanging off the side. Measure the dimension of the rectangular aluminum surface, and record the values in your lab notebook.
3. Sketch the circuit drawing on the left side of Fig. 2 in your lab notebook.
4. Connect the USB cable to the Arduino and lab computer. The green LED on the Arduino should light up.
5. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the right side of the breadboard, as shown in Fig. 2. Use the orange handheld DMM to verify that it is providing +5V of power.
6. Use the DMM to determine which resistor is the 4.7 kΩ resistor. Measure its resistance, and write the value down in your lab notebook.

7. Locate the sample. It is a thin rectangle of aluminum with 4 wires hanging off of it. The thermistor and heater are already taped to the sample. The black wires are the thermistor, and the red wires are the heater.
8. Measure the resistance of the thermistor. It should be around 10 kΩ at room temperature. Clasp the sample in the palms of your hands, and the resistance should gradually decrease.
9. Construct the thermistor voltage divider circuit pictured in Fig. 2, and connect the voltage output V_{out} to analog input pin A0 on the Arduino.
10. Download the B7 "Thermistor Template" code template from the lab webpage, read the comments, and modify it.
    a. Right-click the link and "Save as…" with an intelligent file name (i.e. "B7_thermistor_yourName.ino").
    b. Open the file in the Arduino IDE software.
    c. Replace the *** in the variable declaration with the correct analog input pin for the thermistor.
    d. Replace the other *** with the value of you measured for the 4.7 kΩ resistor.
    e. Save the code file.
11. Use the Arduino IDE software to compile the code and send it to the microcontroller.
    a. Go to "Tools" > "Board" and make sure either "Arduino/Genuino" or "Arduino UNO" is selected.
    b. Go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
    c. Click the check mark at the top of the Arduino program to check the code for errors.
    d. Press the arrow button to compile the program and send it to the Arduino.
    e. Go to "Tools" > "Serial Monitor" (or press "Ctrl + Shift + M") to view the output from the "Serial.print()" commands at the bottom of the screen. (Make sure the Baud rate is set to 9600.)
12. You should see two columns of data separated by a comma printed to the serial monitor.
    a. The first value is the time in seconds.
    b. The second value is the temperature in degrees Kelvin.
13. Clasp the sample between the palms of your hands. You should see the temperature slowly increase.
14. **Demonstrate the working subsystem to the TA, so you can be awarded points on your score sheet.**

Subsystem B: DC Power Supply and Heater

You will now connect the 12V DC power supply and heater.

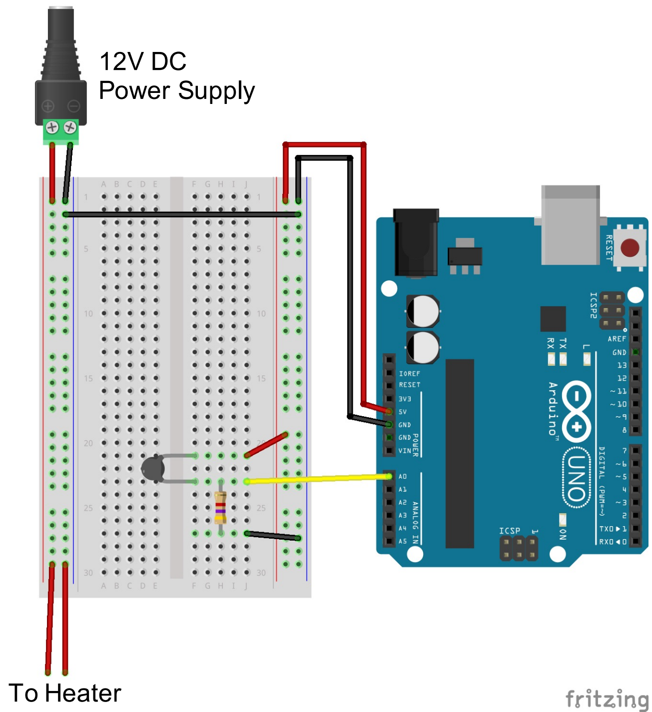

Figure 3 – The vertical bus lines on the left side of the breadboard are powered by the 12V DC power supply.

Caution: Hot! Be careful handling the sample. It will become very HOT if the heater is left on.

Procedure

1. Locate the 12V DC power supply. (It looks like a laptop charger.)
   a. Connect the 5.5mm barrel connector on the 12V DC power supply to the inline kill switch. Make sure the kill switch is in the OFF position.
   b. Connect the other end of the kill switch to the screw terminal adapter.
   c. Connect red and black male pin jumper wires to the + and – screw terminals.
2. Use the handheld DMM to verify that the power supply works by measuring its voltage output. Plug in the DC power supply. Flip the kill switch ON and OFF.
3. Switch off the DC power supply, and make the following connections illustrated in Fig. 3.
   a. Connect the DC power supply to the vertical bus lines on the left side of the breadboard.
   b. Connect the negative bus line on the right side to the “common ground” bus line on the right side, as shown in Fig. 3. That is, both negative vertical bus lines should be connected to GND on the Arduino.

4. Use the handheld DMM to measure the resistance of the heater. Record the value in your lab notebook.
5. Use the appropriate formula to calculate the power that will be dissipated in the heater (in units of Watts) when it is connected to 12V DC.
6. Calculate the surface area of the sample (units of m^2). Then calculate the total heat flux through the sample (units of W/m^2)
7. Run the thermistor program on the Arduino, and pull up the serial monitor.
8. The red wires coming out of the sample are the heater wires. Connect the heater wires to the 12V bus lines, as illustrated in Fig. 3.
9. Use the inline kill switch to turn on the DC power supply and power the heater. You should see the temperature values begin to increase.
10. Switch off the DC power supply and the temperature should decrease.
11. Make sure the 12V DC power supply is turned OFF, and disconnect the heater wires from the breadboard.

Subsystem C: Blower Fan and Solid-state Relay

The blower fan will be switch ON and OFF via a MOSFET solid state relay. Recall that a MOSFET is a transistor that only allows current to flow between the source and drain when a sufficient voltage is applied to the gate. Shown in Fig. 4, the MOSFET controls the flow of current through the blower fan motor. When the Arduino sends 5V (digital HIGH) to the gate, the transistor switches ON the blower fan. If the Arduino sends 0V (digital LOW) to the gate, the transistor switches OFF the blower fan.

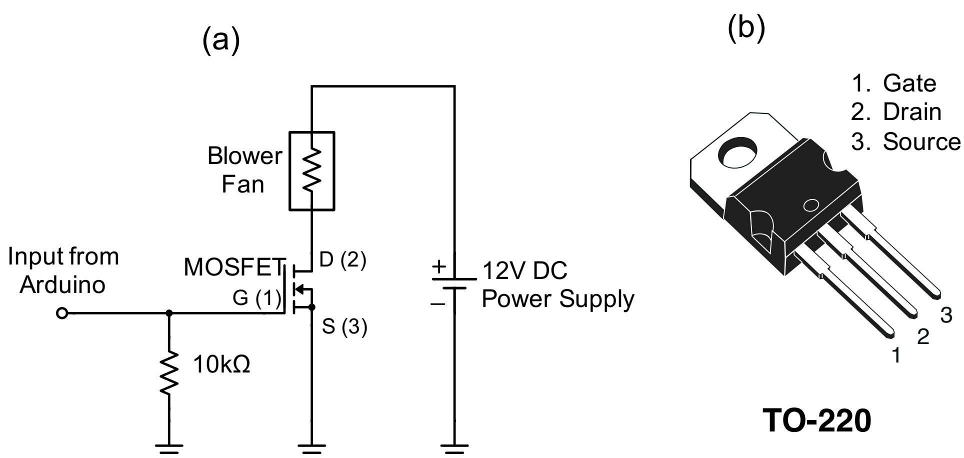

Figure 4 – An N-channel MOSFET is used to switch the blower fan ON and OFF. A digital voltage from the Arduino is used drive the gate of the MOSFET and control the flow of current through the blower fan.

Procedure

1. Make sure the heater is disconnected from the breadboard.
2. Test that the blower fan works by connecting it directly to 12V.
    a. Make sure there are no loose papers that will be blown off the table.
    b. Use a jumper cable with male pins to connect the black wire to GND on the breadboard.
    c. Use another male pin jumper cable to connect the red wire to +12V on the breadboard.
    d. Turn on the DC power supply, and the blower fan should begin spinning.
    e. Turn off the DC power supply and disconnect the jumper wires from the breadboard.
3. Sketch the pinout for the TO-220 N-channel MOSFET shown in Fig. 4b in your lab notebook.
4. Use the breadboard to construct the circuit shown in Fig. 4a.
    a. The blue vertical bus lines on the breadboard are common ground. Anything with a ground symbol in the circuit should be connected to the common ground.
    b. Use a jumper wire with male pins to connect the red wire of the blower fan to +12V on the breadboard
    c. Use another jumper wire with male pins to connect the black wire of the blower fan to the drain of the MOSFET.
    d. Connect a 10 kΩ "pull-down" resistor from the gate to the common ground.
    e. Connect the source of the MOSFET to ground.
5. Turn on the 12V DC power supply. The fan should NOT run. If it runs, you did not construct the circuit correctly.
6. Use a jumper wire to connect the gate of the MOSFET to 5V DC. This will charge the gate and allow current to flow from the source to drain, thus turning ON the fan.
7. Disconnect the gate from 5V. The fan should turn OFF.
8. Connect the gate to digital I/O pin 7 on the Arduino.
9. Place the sample in the duct area on the wood board with the exposed aluminum side facing up. Cover it with the duct, such that the wires pass through the notch in the duct.

15. Download the B7 "Fan Template" code template from the lab webpage, read the comments, and modify it.
    f. Right-click the link and "Save as…" with an intelligent file name (i.e. "B7_Fan_yourName.ino").
    g. Open the file in the Arduino IDE software.
    h. Replace the *** in the beginning with the correct digital I/O input pin for the MOSFET gate.
    i. Replace the ***s in the main loop either HIGH or LOW depending on what the comments indicate.
    j. Save the code file.

16. Use the Arduino IDE software to compile the code and send it to the microcontroller.
    a. Go to "Tools" > "Board" and make sure either "Arduino/Genuino" or "Arduino UNO" is selected.
    b. Go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
    c. Click the check mark at the top of the Arduino program to check the code for errors.
    d. Press the arrow button to compile the program and send it to the Arduino.
10. If you did everything correctly, the fan should repeatedly switch ON for 5 seconds, then switch OFF for 5 seconds.
11. **Demonstrate the working system to the TA, so you can be awarded points on your score sheet.**
12. Leave the circuit intact for the next part of the lab.

Complete Thermostat System

You will now integrate Subsystems A, B, and C to create the complete thermostat system.

Procedure

1. Turn OFF the 12V DC power supply.
2. Merge the thermistor code with the fan code to create a new code that can both measure temperature and switch the fan ON and OFF. Give this new code an intelligent file name (i.e. "B7_Thermostat_yourName.ino").
    a. Include all of the original variable declarations from both codes.
    b. The "setup" portion should set digital I/O pin for the fan to be an output and the "serial.begin()" statement.
    c. The main loop should include the print statements from the thermistor code that print the time and measured temperature in columns.
    d. Include only one delay() function, so it samples the temperature once per second.
3. Declare new float type variables for the high set-point T_{HS} and low set-point T_{LS}. Give them reasonable values above room temperature.
4. Modify the print statements to print two more columns for the high set-point T_{HS} and low set-point T_{LS}.
5. Add conditional statements to the main loop to do the following:
    a. Turn ON the fan if $T > T_{HS}$.
    b. Turn OFF the fan if $T < T_{LS}$.

6. Test the program with a high set-point T_{LS} = 320 K and low set-point T_{HS} = 322 K.
   a. Connect the heater to the breadboard.
   b. Send the code to the Arduino.
   c. Turn on the 12V DC power supply.

Caution: If the thermostat does not work properly, do NOT allow the sample to become too hot. Unplug the heater or switch OFF the 12V DC power supply and allow it to cool.

7. Turn off the 12V DC power supply when you are finished with the test
8. After you have confirmed that the thermostat works, modify the way the set-point is entered. Declare a new float point variable for a single set-point T_S and a hysteresis value ΔT.
9. In the setup portion of the code, use the single set-point T_S and hysteresis ΔT to calculate T_{HS} and T_{LS}, such that $T_{HS} = T_S + \Delta T/2$, and $T_{LS} = T_S - \Delta T/2$.
10. Test the program again with T_S = 322 K and ΔT = 2 K.
11. **Demonstrate the working system to the TA, so you can be awarded points on your score sheet.**
12. Leave the circuit intact for the next part of the lab.

Design Challenge

Redesign the system so the fan is always ON and the heater is switched ON and OFF to regulate the temperature.

- Use the exact same circuits as before, and simply swap the heater and fan.
- You will need to modify the conditional statement in the code. Note that turning the fan ON caused the temperature to decrease—turning the heater ON will have the opposite effect.
- Test the system with a setpoint T_S = 306 K and ΔT = 2 K. Demonstrate it to the TA or lab instructor to receive credit.

Clean-up

To receive full credit, you must return the lab bench to its initial state:

- Unplug the DC power supply. Remove the kill switch and screw terminal adapter.
- Disassemble the circuit. Disconnect the wires from the relays, Arduino, and breadboard.
- Disconnect the USB cable from the computer.

Appendix A

Equipment

- 12V DC power supply w/ inline kill switch and screw terminal adapter
- Arduino UNO Microcontroller
- 6ft USB cable
- Blower fan mounted to small plywood sheet – Amazon Part # B08P1S5DBN
- Breadboard mounted to small plywood sheet
- 2” × 2.5” × 1/16” aluminum sample w/ polyimide film heater and thermistor
    - Polyimide film heater – Amazon Part # B09X16XCVS
    - 10k NTC Thermistor, adhesive mount, Amphenol – Digikey Part # 235-1457-ND
- N-Channel MOSFET TO-220AB (Digikey Part # 497-2765-5-ND)
- Male-Male DuPont pin jumper wires
- 3M Velcro command strips