Experiment A8
Microcontrollers
Procedure

Deliverables: checked lab notebook, demonstration of working device to Lab TA

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab.

Overview

A microcontroller is a rudimentary computer packed into a small IC that can be programmed to automate various tasks. In this lab, you will use an Arduino UNO microcontroller to read the voltage output from the thermistor voltage divider circuit used in A4, calculate the temperature, and light up different colored LEDs depending on the temperature.

There is no tech memo for this lab. Rather, you will construct and program two different electronic systems and demonstrate them to the TA, who will award you points.

Part I: Measuring Temperature

Background

The Arduino UNO is an inexpensive microcontroller commonly used by hobbyists and engineering students. Shown in Figure 1, the UNO is connected to a breadboard where various sensors and circuits can be implemented (i.e. a thermistor in a voltage divider circuit). Here are a few of the more salient features of the UNO.

- **Analog inputs** - The UNO has a 10-bit analog-to-digital converter (A/D) that can read up to six different analog voltages into memory. Voltages between 0V and 5V are mapped to integer values between 0 and 1023, respectively.
- **Digital outputs** - The UNO has 14 different digital outputs. These pins always output a voltage of 0V (LOW) or +5V (HIGH). However, the pins marked with a '~' can output a 500Hz *pulse width modulation* (PWM) square wave. This square wave oscillates between 0V and +5V. Varying the *duty cycle* (% of time the +5V is ON) creates an average voltage that can be treated as an analog output.
- **+5V DC Power** – This pin outputs a constant +5V, which can be connected to a breadboard to power various sensors and peripherals. (It is only capable of producing a very small amount of current, so beware of voltage droop!)
- **USB Connection** – Code and commands are sent to the UNO and data can be received from the UNO via a USB cable. The USB cable also provides the +5V power to the UNO. (In the absence of a USB connection, the UNO can be powered via the "Vin" pin or the 7 – 12V DC 2.1mm barrel connector.)

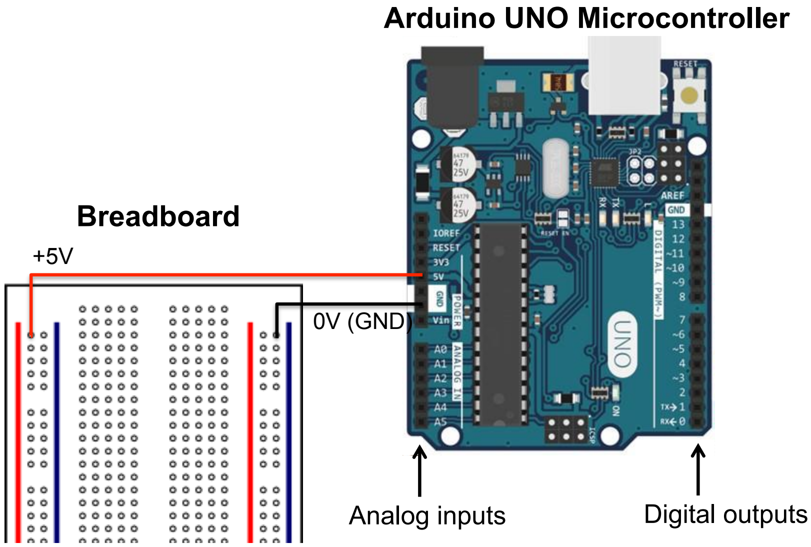

Figure 1 – The Arduino UNO Microcontroller works in tandem with breadboard.

Procedure

You will now construct the thermistor voltage divider circuit from the A4 calibration lab. The Arduino will read the analog voltage V_{out} from the transducer and use it to calculate the temperature via the voltage divider and Steinhart equations.

Important: You will use the Arduino to power the circuit on the breadboard. Do NOT turn on the breadboard.

1. Connect the Arduino UNO to the lab computer via the USB cable. You should see a green LED light up on the Arduino.
2. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the breadboard, as shown in Fig. 1. Use the orange handheld DMM to verify that it is providing +5V of power.
3. Use the handheld DMM to verify that the thermistor works. It should have a resistance around 4.7kΩ at room temperature, and the resistance should decrease when it is warmed up in your hand.
4. Take a 4.7k resistor out of its bin. Measure its resistance with the orange handheld DMM. Record the measured value for R_2 in your lab notebook.

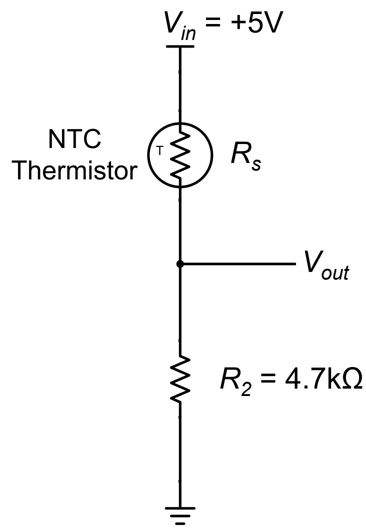

Figure 2 – A thermistor is wired up in a voltage divider circuit.

5. Use the breadboard to construct the thermistor voltage divider shown in Fig. 2. Test the circuit by measuring V_{out} relative to ground. The voltage V_{out} should increase when you warm up the thermistor in your hand.
6. Use a long jumper wire to connect V_{out} to the A0 analog input on the Arduino. The A0 analog input will read the voltage into the Arduino's memory as a 10-bit integer.
7. Open the Arduino IDE software on the lab computer.
8. Download the A8 Part I code template from the lab webpage. Read the comments and fill in the missing values for the calibration constants A and B and the measured resistance R_2. (Use the calibration constants you determined from the 2-point calibration in A4.)
9. Save the code with an intelligent file name (i.e. "A8_thermistor_yourName.ino").
10. In the Arduino IDE software, go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
11. Click the check mark at the top of the Arduino program to check the code for errors.
12. Press the arrow button to compile the program and send it to the Arduino.
13. 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. You should see the measured temperature printed. Hold the thermistor tip in your hand. Does the printed temperature seem reasonable?
14. **Demonstrate the working system to the TA, so you can be awarded points on your score sheet.**

Part II: LED Indicators

You will now use red and green LEDs to indicate whether the probe tip is hot or cold. Specifically, the micro controller will light up the Red LED if $T > 302$K, and light up the Green LED if $T < 302$K.

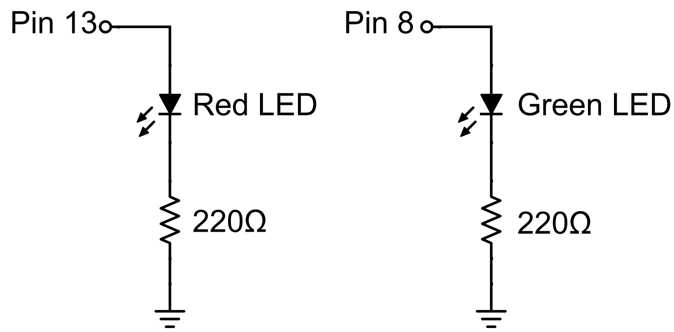

Figure 3 – Digital outputs from the Arduino are used to drive two different LEDs. The 220Ω resistors are used to limit the current and prevent voltage droop.

Procedure

1. Using the breadboard, connect the Red LED in series with a 220Ω resistor. Use a long jumper wire to connect digital output pin 13 to the LED, as shown in Fig. 3. The resistor should be connected to GND via the vertical bus line as shown in Figure 1.
2. Construct the Green LED circuit in a similar fashion. Use another long jumper wire to connect digital output pin 8 to the LED, as shown in Fig. 3.
3. Download the A8 Part II code template from the lab webpage. Save it as "A8_ Part2_yourName.ino".
4. Look through the code. It is similar to the previous code, except there is an if-else conditional statement that lights up different LEDs depending on the temperature.
5. Replace the "***" in the code with appropriate information.
    a. In the variable declaration, enter the pin numbers that the Red and Green LEDs are connected to.
    b. Fill in the missing values for the calibration constants A and B and the measured resistance R_2. (Use the calibration constants you determined from the 2-point calibration in A4.)
    c. In the if-else conditional, enter the correct temperature threshold.
    d. In the if-else conditional, enter appropriate values of "HIGH" or "LOW" that will light up the Red LED if $T > 302$K, and light up the Green LED if $T < 302$K.
6. Test the program. Note that the LEDs must be connected in forward bias. If you connect one backwards, it will not light up!

7. **Demonstrate the working system to the TA, so you can be awarded points on your score sheet.** Only the green LED must be lit at ambient temperature, and only the red LED should light up when warmed up by your hand.
8. Return the lab bench to its initial state. Disconnect all of the wires, sensors, and LEDs. Close all your code windows and exit the Arduino IDE software.
9. Wipe down the lab bench and sanitize the equipment.

Data Analysis and Deliverables

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab.

Appendix A

Equipment

- 10k Vishay NTC thermistor NTCLE413E213F102L (Digikey part # BC2647-ND) or 4.7k Vishay NTC thermistor NTCLE400E3472H (Digikey part # BC2466-ND)
- Arduino UNO Microcontroller
- 12" jumper wires with male pins
- 6ft USB cable
- Breadboard
- Red and Green LEDs
- Extech Handheld Digital Multimeter
    - One red banana-to-banana cable
    - One black banana-to-grabber cable

Appendix B

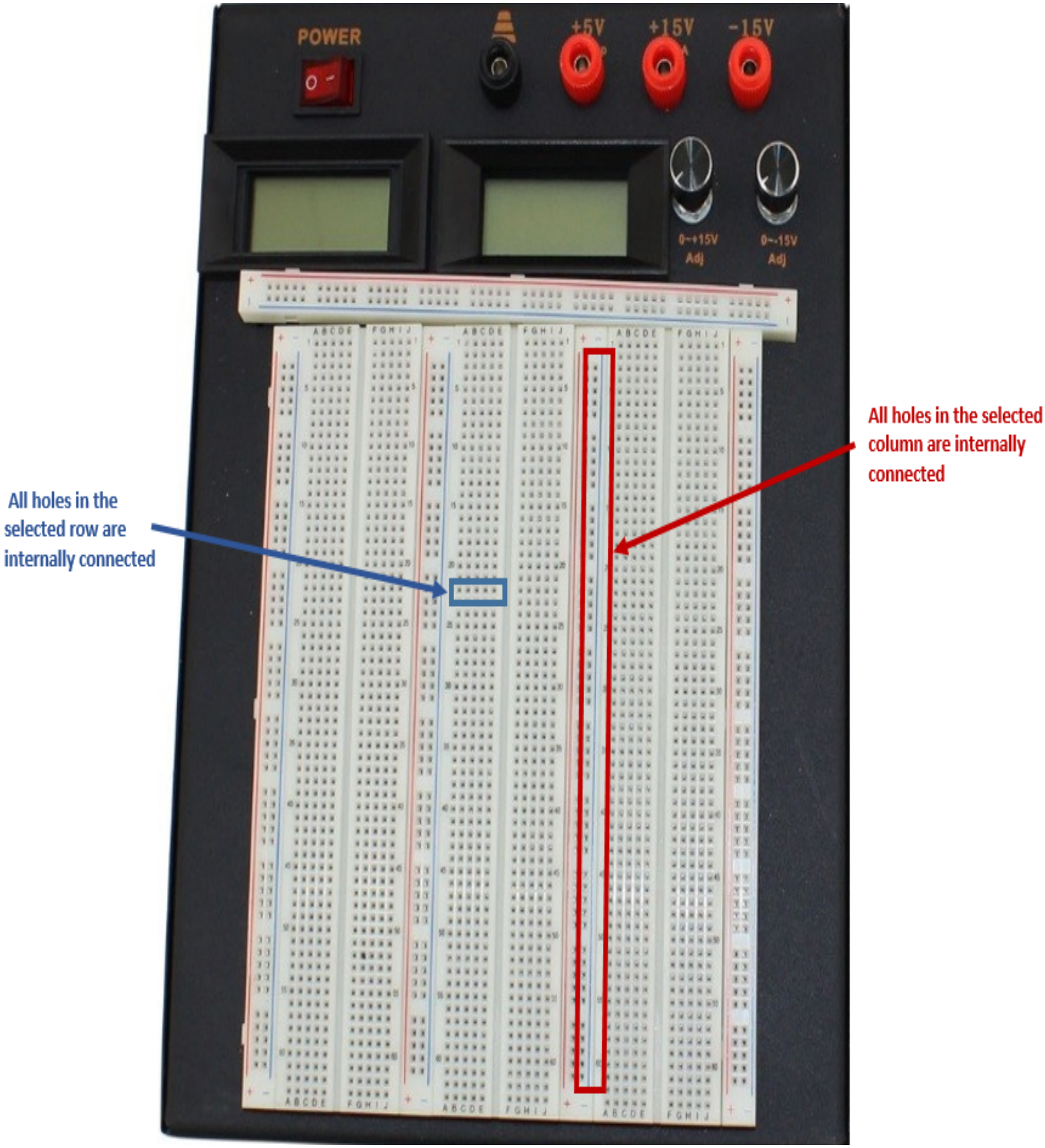

Figure 4 – Shown in blue, any 5 holes in a horizontal row are electrically connected, but they are NOT connected to the adjacent row of 5. Shown in red, all 50 holes in any vertical column or "bus bar" are electrically connected.