

Arduino Thermistor Lab Part 2

We will program the Arduino to print resistance to the serial monitor, and then use Python to capture those values, write them to a file, and convert to temperature and plot. **NOTE: We will do the connection in Python together in class.**

Install pyserial and open a connection between the serial data feed from the Arduino and your computer. (We will do this part in class together).

Collect a temperature time series. In this part of the exercise, each person will collect a night's worth of temperature data at their house. This will require some coordination as each group member will need to use the Arduino and thermistor board for a night, so handing off will be necessary.

In class:

1. Set up the computer, with the Arduino plugged into the USB.
2. Use [therm_plotter_polled.ipynb](#) and instructions to make a script for loading, plotting and saving.
3. Open the serial connection with the Arduino.
4. Grab the string output using the pyserial modules (<https://pythonhosted.org/pyserial/pyserial.html>)
5. Parse the strings from pyserial to find the resistance values being printed to the screen (Note. Python doesn't know that the resistances are numbers, so it treats them like strings. You have to recast each value as a float.
6. Append each resistance value to a numeric array of floats for plotting. This is the y-data in plotting.
7. Convert resistance to temperature using the Steinhart-Hart equation. Note, this gives temperature in degrees kelvin. Will need to convert to degrees celsius before plotting.
8. Record the time of the measurement using the Python 'datetime' library and append to a time array that we will use as the x-data in plotting.

At home:

9. Use a while loop to execute repeatedly throughout the night. (Note you can use Jupyter's kernel interrupt menu to interrupt the loop.
10. Add a delay to the while loop. This will determine how frequently you take a temperature measurement.
11. Plot the output using Matplotlib.pyplot as we have done in previous homeworks. Your x-axis should reflect date and time.
12. Use the plt.savefig() command to save a .jpg a .png or similar to submit with your homework write up.

13. Save the output using Pandas `.to_csv()` function. This creates a file that can be loaded at a later date.
14. Try to collect at least 8 hours of temperature data. Make meta data observations of the conditions where the measurements were taken. What are the heat sources and sinks? Where was the thermistor in the room? Was it near any doors?

What to turn in?

1. Upload your `.ipynb` script. Feel free to work in a group on the coding, but everyone should individually upload the script.
2. CSV file containing overnight temperature data.
3. Short PDF containing
 - a. Image as jpeg or png of your overnight temperature data. This should be a unique plot for everyone in the class.
 - b. Include any metadata notes, plus any other documentation (e.g. photos) that you made of the data collected.