Homework 5 - Sensor node iteration 1

The objective of this assignment is simple: record and transmit temperature data continuously for 48 hours. However, getting there might take some work!

Temperature IC

Write whatever code is necessary to get temperature measurements from your sensor via SPI or I2C. You are welcome to collaborate with your classmates who have the same sensor, provided each person writes their own code. That is, you can share ideas, pointers to datasheet pages, links to library functions, and so on, but you should not be sharing sections of code.

Thermistor

I suggest you tackle the thermistor in three steps:

1. Get the ADC working with a potentiometer. There is starter code to sample an ADC pin on the course website.
2. Test your ADC code with your thermistor. Do something to make it get warmer or colder (putting your finger on it should suffice) and confirm that the value changes.
3. Figure out the conversion from ADC value to temperature. The thermistor temperature-resistance curve is well documented in the datasheet, so converting voltage to temperature should be straightforward (although the equation isn't necessarily simple).
   The ESP32-C3 ADCs are not linear around the endpoints, so you will almost certainly need to do some calibration to convert the raw readings into meaningful voltage measurements. There are ways to do this using a calibration procedure and fuse bits built into the ESP32, but you may find it easier for now to just hard-code the calibration constants in your code.

Data transmission

Your device must publish data to the following topics on the `en1-pi.eecs.tufts.edu` broker at least once every thirty minutes.

- `UTLN/hw5/ic_temp`: Temperature in Celsius reported by your IC, as a text string
- `UTLN/hw5/thermistor_temp`: Temperature in Celsius measured by your thermistor, as a text string

The server will attach a timestamp to each message, and will display a dashboard page plotting your temperature readings.

For any other `UTLN/hw5/` subtopics that you publish to, the server will display a table of the topic and the most recent payload. For example, you could publish the current WiFi signal strength to `UTLN/hw5/rssi`, or the raw thermistor reading.

Packaging and power

I'll bring in an assortment of tupperware containers that you're free to have, or devise your own enclosure. Make sure to include a sticker or piece of paper that includes your name, the class name, and your contact information, just in case someone discovers it!

For power, it's easiest to use a USB battery recharging pack — 3000mAh should last at least a couple days even without power management. If you don't have one, I have four 10Ah power packs that I'm happy to lend out.

Capturing some data

Set up your development board somewhere outside where it has a WiFi signal, and let it run for at least 48 hours. You should be able to track it via the web dashboard at en1-pi.eecs.tufts.edu.

If something goes wrong, just retrieve it, debug, and try again!

What to turn in

Turn in a brief writeup containing the following:

- A description of your packaging and placement.
- A plot of your measurements (it's fine to just screenshot the graph from the web dashboard, or you can download the raw data and make your own plot)
- How accurately you think your reported values matched the outside air temperature? What sources of error are there?

As usual, upload your writeup to the `hw5` folder in Google Drive.