

Physics 524

Survey of Instrumentation and Laboratory Techniques 2025, 2026

University of Illinois at Urbana-Champaign

Unit 4c: Thermometry

Goal for this unit....2
About thermometry, once over lightly....2
Breadboarding....3
Taking data and an initial check of data integrity....4
Homework assignment for this unit....4

Goal for this unit

- Learn about thermometry, in general, and the behavior of the BME680 in particular.

About thermometry, once over lightly

There are many, many ways to determine a temperature! Some considerations in picking the technique to be used:

- How high/low are the temperatures to be measured?
- With what precision will you want to measure the temperature?
- How rapidly will you want to obtain measurements? This can be a subtle question: will the device need time to come to thermal equilibrium with the thing it's measuring?
- Can your sensor be in physical contact with whatever it's measuring?

Lots of the devices we have on hand are able to measure temperatures. Besides the BME680, the DPS310 super-precise barometric pressure sensor can tell us the temperature to about a half-degree.

We have a number of (analog) TMP36 three-wire devices; you attach the output pin to one of your Adalogger's ADC inputs. The device makes use of the known temperature dependence of a semiconductor's current-voltage relationship.

For really high temperatures we can use a thermocouple. (This is handy for calibrating a home oven, for example: I've used this to figure out how my ovens behave.) From Adafruit:

> "Unlike semiconductor temperature sensors such as the TMP36, thermocouples have no electronics inside them, they are simply made by welding together two metal wires. Because of a physical effect of two joined metals, there is a slight but measurable voltage across the wires that increases with temperature. The type of metals used affect the voltage range, cost and sensitivity, which is why we have a few different kinds of thermocouples. The main improvement of using a thermocouple over a semiconductor sensor or thermistor is that the temperature range is very much increased. For example, the TMP36 can go from -50 to 150°C, after that the chip itself can be damaged. Common thermocouples on the other hand, can go from -200°C to 1350°C (K type) and there are ones that can go above 2300°C!

 ©George Gollin, 2024

> Thermocouples are often used in HVAC systems, heaters and boilers, kilns, etc. There are a few different kinds… [the Adafruit tutorial discusses] K type [thermocouples], which are very common and easier to interface with.
>
> One difficulty in using them is that the voltage to be measured is very small, with changes of about 50 μV per °C (a μV is 1/1000000 Volts). While it is possible to read these voltages using a clean power supply and nice op-amps, there are other complications such as a non-linear response (it's not always 50μV/°C) and cold-temperature compensation (the effect measured is only a differential and there must be a reference, just as ground is a reference for voltage).

For non-contact measurements, there's the IR MLX90614, an I2C device that's also good to about half a degree. Note that there are two different kinds, meant to be powered by 3V or 5V. I believe you have the 5V version in your parts kit. (If you feed 5V to the 3V version you're probably going to fry it, so be careful.)

When we discuss matters of accuracy, we'll need to be sure that we are distinguishing between manufacturing-induced offsets (which tend to be about half a degree between individual BME680s taken straight out of a parts bin) and intrinsic, noise-driven imprecisions. These are smaller, typically a quarter of a degree or less.

Breadboarding

There'll be a mix of breadboarding, programming, and data analysis tasks, some of which can probably be done directly on your Adalogger. Do this:

- Disconnect the power from all devices on your breadboard so that they are incapable of generating electrical noise.
- Wire up two BME680s to an Adalogger: have one use SPI and the other I2C. Be sure to place bypass capacitors between power and ground for each BME680.
- Code up a loop that will, each pass through the loop, read the temperature from both devices (using the usual default parameters for the temperature sensors) and store the values obtained in an array. Add whatever delay() is necessary to make the loop take about 10 seconds to record 1,000 values from each sensor.

 ©George Gollin, 2024

Taking data and an initial check of data integrity

Write some files of data inside Loomis 262, out in the hall, and outside the building. Open your files with a plain text editor to check that what you're recording makes sense.

Homework assignment for this unit

- Write these values to a microSD memory card, either as plain text or as comma-separated value files, making sure to include something that will let you distinguish among various data sets that you'll store on your laptops, either when opening an SD file or else when transferring the data to a laptop.
- Have your Adalogger calculate the following for each run of 1,000 recordings: <TI2C>, <TSPI>, <TI2C - TSPI>, and the RMS deviations from the means of TI2C, TSPI, and TI2C - TSPI. Write these to the end of each file of data.
- Write a python program that will read a data file and generate graphs of the values of T_{I2C}, T_{SPI}, and T_{I2C} - T_{SPI} as functions of sample number. Also have your program generate histograms of the values of these three quantities for the data file. Do this for several different files you've stored.



 ©George Gollin, 2024