Lab 1: Introduction

The exercises we completed in the first lab were primarily designed to familiarize us with the Arduino environment and basic coding. All the code for these exercises can be found at github.com/Sauttets/Ubiquitous-Computing

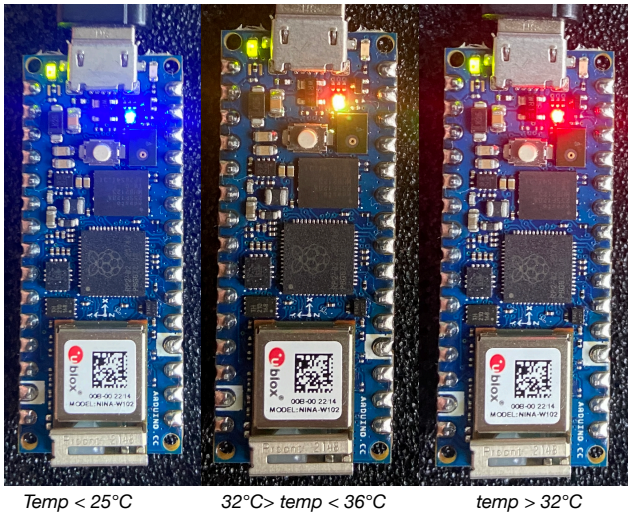
Exercise 0: Blink

The introductory exercise was straightforward, as the instructions were highly detailed. After installing the Arduino IDE and setting up the board, it was as simple as opening the example sketch provided by Arduino.

Exercise 1: RGB

For this exercise, the task was to make the RGB LED blink every half second. We used the delay( ) function to ensure the LED changed color every 500ms. If more functionality had been required, we would have opted for an interrupt-based timer, as the delay( ) function blocks other code from executing.

Exercise 2: Temperature



Temp < 25°C *32°C> temp < 36°C* *temp > 32°C*

In this exercise, we reused the RGB LED from the previous task in combination with the temperature sensor built into the RP2040. The LED was programmed to change color based on the current temperature. To achieve this, we used the Arduino_LSM6DSOX library as suggested. We encountered an issue when testing the temperature sensor: it is located very close to the CPU, causing it to consistently show values much higher than the actual room temperature.
To address this, we used a radiator to heat the Arduino to above 36°C and placed it in a fridge to lower the temperature below 20°C for more accurate readings.

Exercise 3: Microphone

The goal of this task was to familiarize ourselves with the microphone on the controller. As described, the code toggled the microphone on or off when the noise level exceeded a predefined threshold.

Exercise 4: Sitting posture

In this exercise, we were tasked with modifying existing code to use a different IMU library than the one originally implemented. This process was straightforward, as the two libraries were quite similar.
However, we encountered an issue we couldn't resolve: setting the filter frequency to 104 Hz resulted in either incorrect or severely delayed readings. As a result, we decided to stick with a frequency of 25 Hz, using the millis( ) function to read 25 times per second instead of 104 times.
Since we were only interested in tracking sitting posture, we simplified the code to focus solely on the pitch value, as this corresponds to leaning forward, which indicates poor sitting posture.