Hochschule Konstanz
Department of Computer Science

Ubiquitous Computing WS 24

Exercises Documentation

Contents

List of Figures

List of Tables

Acronyms

1 Exercise Introduction

1.1 Exercise 0: Intro

The first or zero exercise is quite forward and can bee seen as a warm-up exercise. The goal is to get familiar with the tools and the environment and write a small program for a blinking LED (see following code).

```
1
2     const int ledPin =    LED_BUILTIN;  // The onboard LED pin
3     const int delayTime = 1000;         // Delay time in milliseconds (1 second)
4
5     void setup() {
6       pinMode(ledPin, OUTPUT);
7     }
8
9     void loop() {
10      digitalWrite(ledPin, HIGH);       // Turn the LED on
11      delay(delayTime);
12
13      digitalWrite(ledPin, LOW);        // Turn the LED off
14      delay(delayTime);
15    }
```

1.2 Exercise 1; Internal RGB

In the first exercise we are going to use the internal RGB LED to switch between the three colors red, green and blue. Cause there is also a template for this exercise, we will not go into detail here.

```
1
2     #include <WiFiNINA.h>
3
```

```
4   const int delayTime = 500; // Delay time in milliseconds (0.5 seconds)
5
6   void setup() {
7     pinMode(LEDR, OUTPUT);
8     pinMode(LEDG, OUTPUT);
9     pinMode(LEDB, OUTPUT);
10  }
11
12  void loop() {
13    blink_red();
14    delay(delayTime);
15    blink_blue();
16    delay(delayTime);
17    blink_green();
18    delay(delayTime);
19  }
20
21  void blink_red(){
22    Serial.print("Red\n");
23    digitalWrite(LEDR, HIGH);   // Red
24    digitalWrite(LEDB, LOW);    // Blue
25    digitalWrite(LEDG, LOW);    // Green
26  }
27
28  void blink_blue(){
29    Serial.print("Blue\n");
30    digitalWrite(LEDR, LOW);    // Red
31    digitalWrite(LEDB, HIGH);   // Blue
32    digitalWrite(LEDG, LOW);    // Green
33  }
34
35  void blink_green(){
36    Serial.print("Green\n");
37    digitalWrite(LEDR, LOW);    // Red
38    digitalWrite(LEDB, LOW);    // Blue
39    digitalWrite(LEDG, HIGH);   // Green
40  }
```

1.3 Exercise 2: Temperature Sensor

In the second exercise we are going to use the internal temperature sensor to measure the temperature of the Arduino board. Also we will use the LEDs from the task before to visualize the temperature like described in the task.

The following figures show the different states of the LEDs depending on the temperature. As it can be seen in the figures, the board was cooled down using a cooling pad and heated up using a hair dryer.
The code for this exercise can be seen on the github repository `https://github.com/Smokey95/AIN_Ubiquitous_Computing/blob/main/exercise/exercise2_temperature/exercise2_temperature.ino`.

Figure 1.1: Temperature Sensor between 20 and 36 degrees Celsius

Figure 1.2: Temperature Sensor under 20 degrees Celsius

Figure 1.3: Temperature Sensor above 36 degrees Celsius

1.4 Exercise 3: Microphone

In the third exercise we are going to use the internal microphone to measure the sound level. Therefore we used the link provided in the task description. Like seen in the figures below, the LED turn on/off when a certain sound level is reached.

Like before the code for this exercise can be seen on the github repository `https://github.com/Smokey95/AIN_Ubiquitous_Computing/blob/main/exercise/exercise3_microphone/exercise3_microphone.ino` It has to be noted that the `while-loop` for preventing the program from running till a serial monitor is connected will only work if all monitors are closed (even in other Arduino IDE instances).

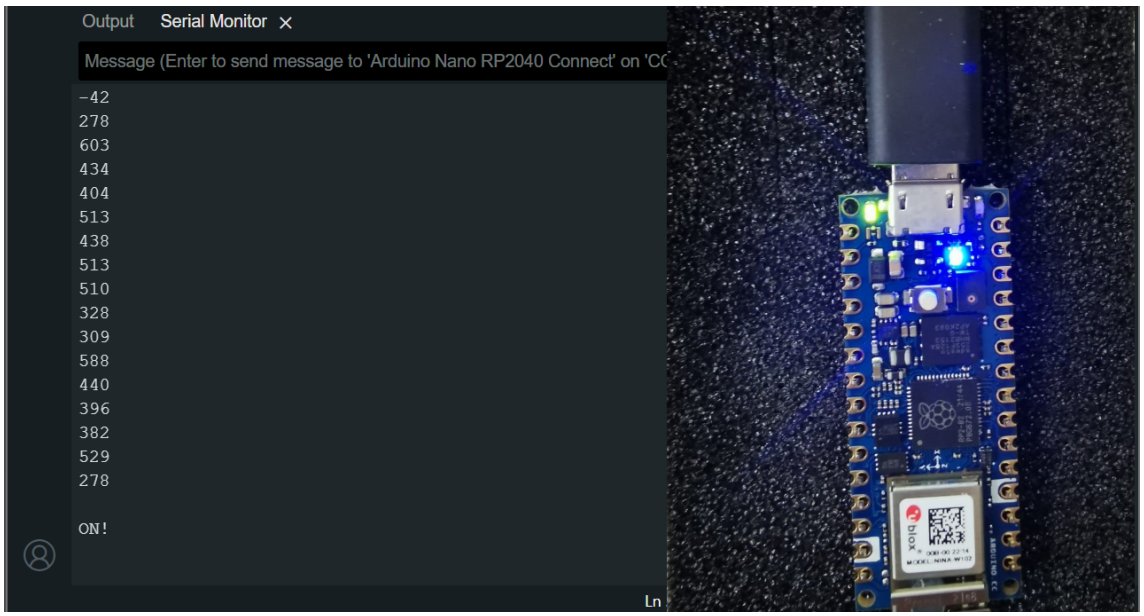

Figure 1.4: Sound detected, LED on

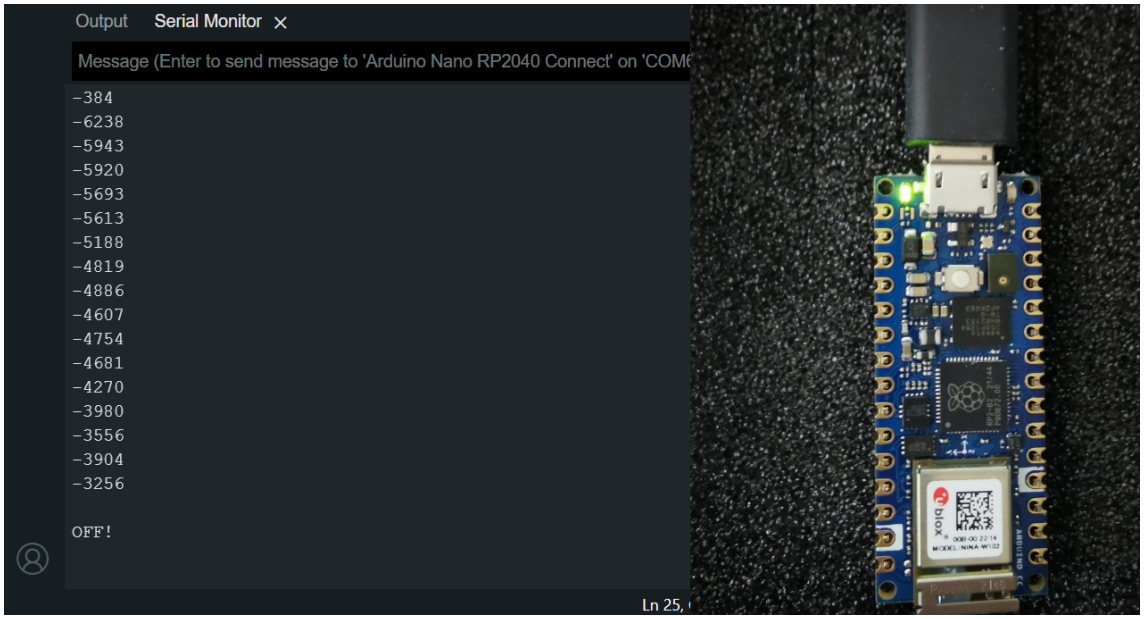

Figure 1.5: Sound detected, LED off

1.5 Exercise 4: Posture Detector. Accelerometer

In the last exercise we are going to use the internal accelerometer to detect the posture of the Arduino board. Therefore we used the link provided in the task description to realize the following code as well as the serial monitor output seen in figure 1.6. Like before the code can be seen on the github repository `https://github.com/Smokey95/AIN_Ubiquitous_Computing/blob/main/exercise/exercise4_posture/exercise4_posture.ino`.

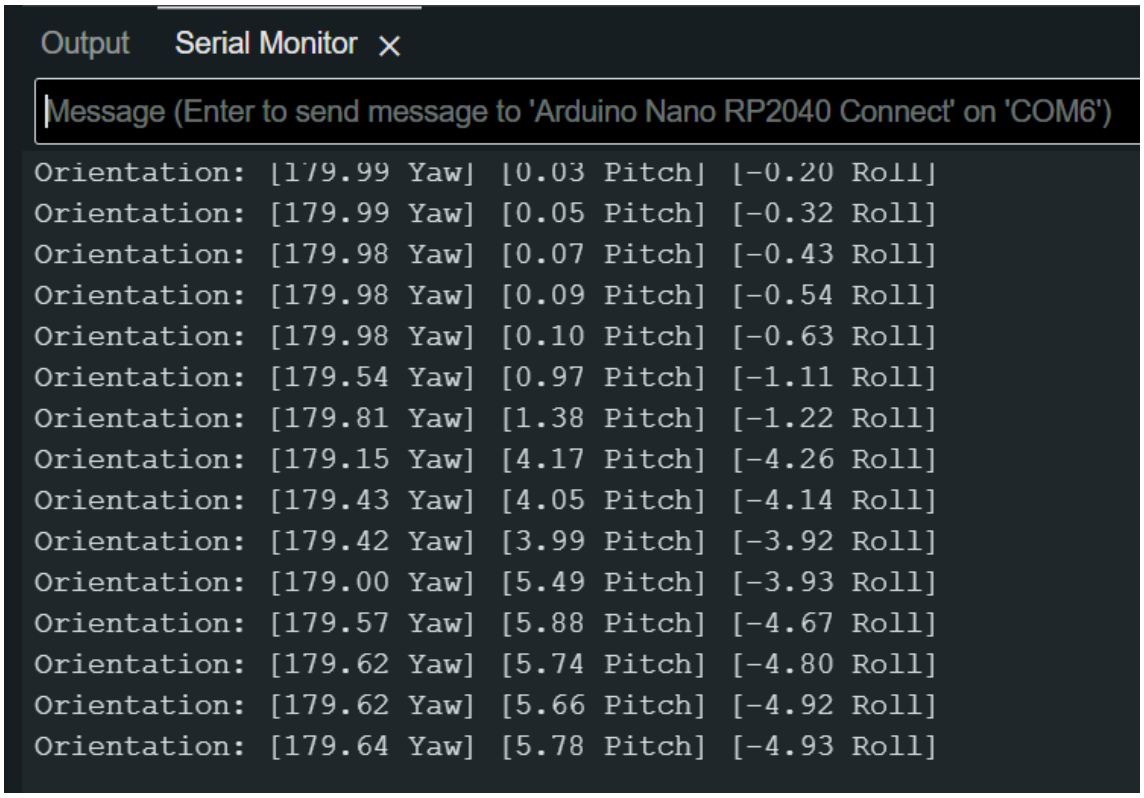

Figure 1.6: Posture Detector