Introduction to Robotics: Homework

Andrei Dumitriu andrei.dumitriu@fmi.unibuc.ro

Last Updated: October 19th, 2024, 17:00

1 Homework #2

Deadline (hard): Your respective lab in the week of October 23rd-29th, 2023.

This assignment focuses on controlling each channel (Red, Green, and Blue) of an RGB LED using individual potentiometers. By the end of this task, you will gain experience in interfacing potentiometers with the Arduino and controlling RGB LEDs based on the analog readings.

You must finish the work before your laboratory session and present it there. This includes everything: hardware, coding and github documentation.

Example: `https://www.youtube.com/watch?v=tbynfti3OEo`

1.1 Components

- RGB LED (At least 1)
- Potentiometers (At least 3)
- Resistors and wires as needed

1.2 Technical Task

Use a separate potentiometer for controlling each color of the RGB LED: **R**ed, **G**reen, and **B**lue. This control must leverage **digital electronics**. Specifically, you need to read the potentiometer's value with Arduino and then write a mapped value to the LED pins.

1.3 Publishing Task

1. Add the code to the Git repository.
2. Update the repository's README with:
    - Task Requirements and/or Description
    - A photo of your setup

- A link to a video showcasing functionality (YouTube is recommended, but any accessible platform is acceptable). Ensure the video is in the correct orientation. Avoid this: `https://youtu.be/Y8H0PlUtcto`

3. Once your Git repository is up to date, submit your homework on MS Teams.

1.4 Coding Task

- Clean, well-structured code is crucial. Adhere to best practices to achieve the highest grade.
- Avoid using "magic numbers". Use constants with meaningful names instead.
- Review the style guide and provided style document to ensure uniformity and clarity in your code.

1.5 Common Mistakes to Avoid

1. Using "potValue / 4" which lacks precision. Opt for the **map() function** instead.
2. Calculating the value inside the *analogWrite* function.
3. Using non-descriptive variable names like **potValue** or **potPin**. Consider future component changes, e.g., replacing a potentiometer with a slider.
4. Using magic numbers such as **4**, **5.0**, and **1023.0**. Replace these with well-named constants.
5. Neglecting to use the appropriate number of components.
6. Failing to use filters when encountering value flickering.