Introduction to Robotics: Homework

Andrei Dumitriu andrei.dumitriu@fmi.unibuc.ro

Last Updated: November 8th, 2023, 16:30

1 Homework #5 - Stopwatch Timer

Deadline (hard): Your respective lab in the week of November 13th - November 19th, 2023. You must have the Git ready and the assignment turned in before the start of lab.

- **General description:** Using the 4 digit 7 segment display and 3 buttons, you should implement a stopwatch timer that counts in 10ths of a second and has a save lap functionality (similar to most basic stopwatch functions on most phones).
- **I don't have an example to show you, but it should work similar to any phone stopwatch (but with the lap viewing functionality separated):** `https://www.youtube.com/watch?v=rd7SzV7t888`
- **Components**: 1 7-segment display, 3 buttons, resistors and wires (per logic)
- **Requirement:** The starting value of the 4 digit 7 segment display should be "000.0". Your buttons should have the following functionalities:
  - Button 1: Start / pause.
  - Button 2: Reset (if in pause mode). Reset saved laps (if in lap viewing mode).
  - Button 3: Save lap (if in counting mode), cycle through last saved laps (up to 4 laps).

  Workflow:

  1. Display shows "000.0". When pressing the **Start** button, the timer should start.
  2. During timer counter, each time you press the lap button, you should save that timer's value in memory (not persistent, it is OK to be deleted upon reset), up to 4 laps (or more if you want); pressing the 5th time should override the 1st saved one. If you press the reset button while timer works, nothing happens. If you press the pause button, the timer stops.

3. In **Pause** Mode, the lap flag button doesn't work anymore. Pressing the reset button resets you to 000.0.
4. After reset, you can now press the flag buttons to cycle through the lap times. Each time you press the flag button, it takes you to the next saved lap. Pressing it continuously should cycle you through it continuously. Pressing the reset button while in this state resets all your flags and takes the timer back to "000.0".

- **Be careful:**
  1. Make sure you put the dot "." on the display to separate the seconds from the 10ths of a second
  2. You should display all the digits. For example, if you have "20.4" seconds, you should display "020.4".
  3. For CTI, interrupts are mandatory for the **save lap and pause button**. This is to increase precision of counting. For Computer Science, these are bonus.
- **Publishing task:** You must add the code to the Github repo and continue updating the readme with at least the following details (but feel free to be more creative). I recommend using dropdowns in the readme, for each project, if you feel the readme is too cluttered:

  `https://gist.github.com/citrusui/07978f14b11adada364ff901e27c7f61`

  1. Task Requirements
  2. Picture of the setup
  3. Link to video showcasing functionality (I recommend youtube, but anything I can access is fine)
  4. Remember to publish the video in the correct orientation. Don't do this: `https://youtu.be/Y8H0PlUtcto`
  5. **Hand in the homework on MS teams when done - aka when Git is up to date**
- **Coding task:** Coding style is of utmost importance. You must have a perfectly clean code in order to receive the maximum grade. An important coding challenge present in this task is using **millis() or micros()** and **interrupts** instead of **delay()**. Do not use repetitive structures such as **while()**. You can use **for()** in certain cases where it removes repeated code (such as addressing all the segments, changing the pinMode() etc.).

  However, remember that the bulk of the grade is if the projects **works**. Do not tend to small details unless you have the entire functionality done.

- **Possible bonus points:**
  - Add other components, such as sounds, other LEDs etc. Be creative. For example, you can add a LED to each button, signaling if you can press it or not.
  - You are also welcomed to use a different setup, for example a joystick instead of a button (and cycle through lap times using the joystick, instead if button press). You are not allowed to change the basic requirements, though (especially the states order). Just to approach them differently.