Introduction to Robotics: Homework #6

Pseudo-Smart Environment Monitor and Logger

Andrei Dumitriu andrei.dumitriu@fmi.unibuc.ro

Last Updated: November 13th, 2023, 19:27

Due Date

Submit your completed assignment to your respective lab session during the week of November 20th - November 26th, 2023. Ensure your Git repository is ready and the assignment is submitted before the lab begins.

1 Objective

Develop a "Smart Environment Monitor and Logger" using Arduino. This system will utilize various sensors to gather environmental data, log this data into EEPROM, and provide both visual feedback via an RGB LED and user interaction through a Serial Menu. The project focuses on integrating sensor readings, memory management, Serial Communication and the general objective of building a menu. **See the partial example video (only of menu parsing) here:** `https://www.youtube.com/watch?v=mh0KYdul1Sk`

2 Components Required

- Arduino Uno Board
- Ultrasonic Sensor (HC-SR04)
- LDR (Light-Dependent Resistor)
- RGB LED
- Resistors as needed
- Breadboard and connecting wires
- (Optional) Additional sensors / compoennts for extended functionality

2.1 Menu Structure

1. **Sensor Settings** // Go to submenu

   1.1 Ultrasonic Sensor Sampling Interval // Parameter: Seconds ($0 < n < 10$)

   1.2 Ultrasonic Sensor Alert Threshold // Parameter: Max distance value

   1.3 LDR Sampling Interval // Parameter: Seconds ($0 < n < 10$)

   1.4 LDR Sensor Sampling Interval // Parameter: Seconds ($0 < n < 10$)

   1.5 Back // Return to main menu

2. **Reset Logger Data** // Manage and reset stored data

   2.1 Reset All Logged Data // Clear all data in EEPROM

   2.2 Reset Individual Sensor Data // Go to submenu

      2.2.1 Reset Ultrasonic // Clear Ultrasonic sensor data

      2.2.2 Reset LDR // Clear LDR sensor data

      2.2.3 Back // Return to upper or main menu, your choice.

   2.3 Back // Return to main menu

3. **System Status** // Check current status and health

   3.1 Current Sensor Readings // Display real-time readings. Use a keypress to exit.

   3.2 Display Logged Data // Display last 10 sensor readings.

   3.3 System Health // Show EEPROM usage (%), last reset moment for each sensor.

   3.4 Back // Behavior: Return to main menu

4. **RGB LED Control** // Go to submenu

   4.1 Manual Color Control // Set RGB LED color manually

   4.2 Toggle LED: Use Manual Values // Change LED based on manually input values. Green if in scope, red if outside the scope.

   4.3 Toggle LED: Use Sensor Data // Change LED based on Sensor Data and set thresholds

   4.4 Back // Return to main menu

2.2 Be Careful:

1. **EEPROM Write Cycles:** Avoid excessive writing to EEPROM to prevent wear
2. **Sensor Calibration:** Ensure sensors are correctly calibrated for accurate readings. (aka you know their interval values)
3. **Serial Communication Errors:** Implement error handling for Serial Communication to manage incorrect inputs.

3 Submission Guidelines

Upload your code to GitHub and update the README with at least:

1. Task requirements. Include the menu structure in the description.
2. A photo of your setup
3. A link to a video demonstrating the functionality (preferred: YouTube)
4. Ensure the video is correctly oriented.

Submit your homework through MS Teams once your Git repository reflects the latest changes.

Coding Standards

Clean and readable code is essential for full credit. Prioritize using `millis()` or `micros()` over `delay()` and avoid loops like `while()`. Use `for()` loops to streamline repetitive tasks like setting segment pins. Focus on the project's primary functionality before attending to minor details.

4 Bonus Opportunities

1. **Advanced Data Analysis:** Include features like calculating averages or detecting trends in the sensor data.
2. **Additional Sensor Integration:** Incorporate more sensors (like temperature or humidity) for a more comprehensive monitoring system.
3. **Creative LED Feedback:** Innovate with the RGB LED to create intricate feedback patterns or color schemes based on environmental changes.