Project 2: Motorized Wheelchair Controller

ECE 298 S2021

Problem Statement

People with mobility issues face challenges every day. Design a motorized wheelchair with simple user interfaces and energy-efficient motor drives to maximize the range of operation between battery recharge cycles.

Functional Interfaces to be Considered

Sensors

- The **Right Wheel Rotation Encoder** and **Left Wheel Rotation Encoder** are used to sense the rotational direction and speed of the right and left wheels.
- The controller uses the **Battery Energy Level Detector** to determine the battery's state of charge.

Actuators

- The **Right Wheel Motor** and **Left Wheel Motor** are used to drive the right and left wheels.
    - The ratio of motor revolutions to wheel rotations can be assumed to be 6 to 1.
    - The motor must activate with a speed ramp-up and ramp-down when starting and stopping, saving energy and improving the user experience.
    - The software must monitor the number of motor revolutions to control the speed ramp-up and ramp-down properly.

User Inputs

- **User Inputs** (switches, pushbuttons, potentiometers, a keypad, etc.) must be available to:
    - Select the mode of operation (Locked or Run Mode)
    - In **Run Mode**:
        - Control the forward and reverse motion from zero to some maximum
        - Control the left and right direction from some left maximum, through the centre, to some right maximum
        - To input a speed or direction, you may use a potentiometer-based circuit that provides an analog voltage between 0 V and 3.3 V to an MCU Analog Input channel pin. This voltage could represent a speed (0 V = 0 m/s, 3.3 V = max m/s), with a direction provided by a switch. It could also represent the left-right axis on a 2-axis joystick (0 V = drive Right Wheel Motor only to turn left, 1.65 V = both wheels the same speed, 3.3 V = drive Left Wheel Motor only to turn right).
        - Use whatever combination of User Inputs you need to achieve these requirements.

User Outputs (Indicators)

- **LCD Display** to indicate the Left Wheel Motor and Right Wheel Motor RPMs in Run Mode.
- **Green LED1** showing that the Wheelchair Controller is in Run Mode
- **Green LED2** showing that the battery state of charge is over 90%
- **Yellow LED** to indicate that the battery state of charge is between 80%-90%
- **Orange LED** to indicate that the battery state of charge is between 60%-80%
- **Red LED** (flashing) to indicate that the battery state of charge is less than 60%.

Operating Modes

- There are two modes of operation: **Locked Mode** and **Run Mode**.
    - In **Locked Mode**, all Power is disabled.
    - In **Run Mode**:
        - The software displays the Left Wheel Motor and Right Wheel Motor RPMs on the LCD Display.
        - Each wheel rotates at a rate of the Motor RPM/6.
        - The Right Wheel Rotation Encoder and Left Wheel Rotation Encoder monitor each wheel's rotation cycles for RPM and rotation direction.
        - An MCU timer is to be employed in determining RPMs. Note that the MCU has a real-time clock (see RTC in the API reference).
        - The MCU has a 16-channel Analog to Digital converter. These channels may monitor the analog voltage levels related to the User Input circuits for direction and speed.
- **Note:** The Battery can be "faked" for your simulations by using a massive capacitor with an initial voltage equal to the battery voltage.

Constraints

1. Only parts from the Proteus Libraries may be used in this project (none from the web).
2. The Sensors, Actuators, User Inputs, and User Outputs (Indicators) will be connected to the Prototype Adapter Board through wiring harnesses to connectors on that board.
3. The PCB shape (Sometimes called the "form factor") must be as per the PCB layout template in ECE 298 Lab 4.
4. The Prototype Adapter Board PCB has receptacle connectors to mate with the development board. They must not be changed in type or physical location.
5. The final schematic and PCB design must include suitable decoupling capacitors for power supply voltage filtering.
6. This project must employ voltage level translations (3.3 V → 5 V, 5 V → 3.3 V) for signals between the STM32 MCU (3.3 V) and 5 V devices **where appropriate** (refer to the part datasheets).

Responsibilities

Team Member 1: Motor Wheel Rotation Encoders, MCU Timers, LCD Display, and Software

Team Member 2: Motor Drive Controls, User Inputs, Battery Measurement, LEDs, and Software