Lab 3 – Prototype Phase 1

ECE 298 – S2021

Lab Section:	N/A	Group:	90

Part 1 – Pin Mapping

MCU Pin	Pin Mode	Functional Description
PA0	TIM2_CH1	Outputs a PWM signal to the left DC Motor
PA1	TIM5_CH2	Outputs a PWM signal to the right DC motor.
PA4	GPIO_Output	Outputs the first bit of 4 that is sent to the LCD in 4-bit mode.
PA5	GPIO_Output	Outputs the second bit of 4 that is sent to the LCD in 4-bit mode.
PA6	GPIO_Output	Outputs the third bit of 4 that is sent to the LCD in 4-bit mode.
PA7	GPIO_Output	Outputs the fourth bit of 4 that is sent to the LCD in 4-bit mode.
PA10	GPIO_EXTI10	Senses the state of the ON/OFF switch.
PB0	GPIO_Output	Outputs the E (enable) bit that is sent to the LCD in 4-bit mode.
PB1	GPIO_Output	Outputs the RS bit that is sent to the LCD in 4-bit mode.
PB2	GPIO_Output	Outputs the R/W (read/write) bit that is sent to the LCD in 4-bit mode.
PB4	GPIO_Output	Controls the red LED indicating battery voltage.
PB5	GPIO_Output	Controls the orange LED indicating battery voltage.
PB6	GPIO_Output	Controls the yellow LED indicating battery voltage.
PB7	GPIO_Output	Controls the green LED indicating battery voltage.
PC3	GPIO_Output	Controls the green LED indicating controller mode.
PB14	GPIO_Output	Selects the mux output controlling forward and backward rotation of the right DC motor.
PB15	GPIO_Output	Selects the mux output controlling forward and backward rotation of the left DC motor.
PC0	ADC1_IN10	Analog to digital converter input of battery voltage.
PC1	ADC1_IN11	Analog to digital converter input of speed control potentiometer.
PC2	ADC1_IN12	Analog to digital converter input of steer control potentiometer.
PC6	GPIO_EXTI6	Input of the left motor encoders's Q1 output to sense rotation speed.
PC7	GPIO_Input	Input of the left motor encoders's Q2 output to sense rotation direction.
PC8	GPIO_EXTI8	Input of the right motor encoders's Q1 output to sense rotation speed.
PC9	GPIO_Input	Input of the right motor encoders's Q2 output to sense rotation direction.

Part 2 – MCU Resources

MCU Resource	Functional Description
TIM1	Counts time until the last 1/24-th of a rotation occurred when sensing the Q1 voltage of the left DC motor encoder.
TIM2	Generate PWM signal whose duty cycle is varied dynamically that is sent to control the left DC motor controller.
TIM3	If the controller is in "Locked mode" and the battery is less than 60% of its maximum value, the timer counts upwards and blinks once it has reached its count value set to repeat every 0.5 s.
TIM4	Counts time until the last 1/24-th of a rotation occurred when sensing the Q1 voltage of the right DC motor encoder.

TIM5	Generate PWM signal whose duty cycle is varied dynamically that is sent to control the right DC motor controller.
ADC	Measures the analog voltage of the DC battery and potentiometer voltage division circuits on a scale of 0-3.3 V.
GPIO	Outputs digital signals to the LEDs, multiplexers, LED NFETs and LCD interfaces and receives input from the mode switch and DC motor encoder's outputs.
NVIC	Triggers interrupts based on TIM3 Output Compare, voltage from the digital switch, and the DC motor encoder's Q1 output to determine the rotation speed and direction of the motor.

Part 3 – Test Cases

DC Motor control and interface

Test Summary

The system involving the DC motor, DC motor encoder, DC motor controller, potentiometer steering and speed inputs has been tested with various inputs and the corresponding outputs have been observed in Proteus 8. The DC motor is controlled with a software-based PID controller that varies the duty cycle of the PWM signal sent to each of the DC motor controllers based on the PID controller set-point. The set point of the PID control is set by the steering/speed potentiometer voltage divider ouput that is sensed by the ADC. The output of the DC motor encoder is then sensed by the MCU and the corresponding adjustments to the duty cycle are made. The LCD displays the sensed rotation speed sensed from each of the Q1 and Q2 pins of the motor controller (RPM of the motor divided by 6). Due to the inevitable overshoot of the PID controller, there is an error of <5% of the set-point RPM that is observed on the DC motors.

The following is used to calculate the set point:

$$V_{P1} = voltage\ sensed\ from\ potentiometer\ 1$$

$$V_{P2} = voltage\ sensed\ from\ potentiometer\ 2$$

$$s_p = speed\_setting\ = (1.5 - V_{P1})\left(\frac{10}{1.5}\right)$$

$$s_t = \ steer\_setting\ = (1.5 - V_{P2})\left(\frac{10}{1.5}\right)$$

$$\alpha = \ speed\_factor = 4 * speed\_setting$$

$$n_{R_{set}} = \{\ \alpha\left(10 - s_p\right), s_t \geq 0; 10\alpha, s_t < 0\}$$

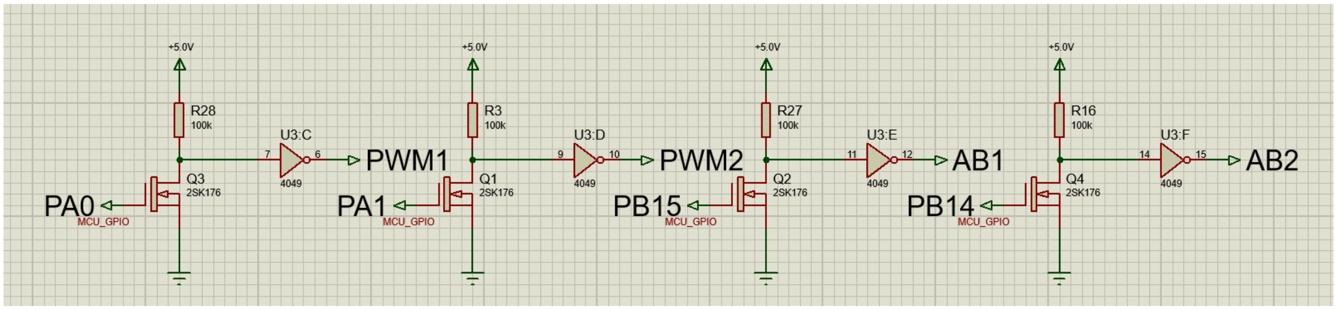
$$n_{L_{set}} = \{\ \alpha\left(10 - s_p\right), s_t < 0; 10\alpha, s_t \geq 0\}$$

Where n_R_set and n_L_set are the set-points of the left and right motor RPM respectively.

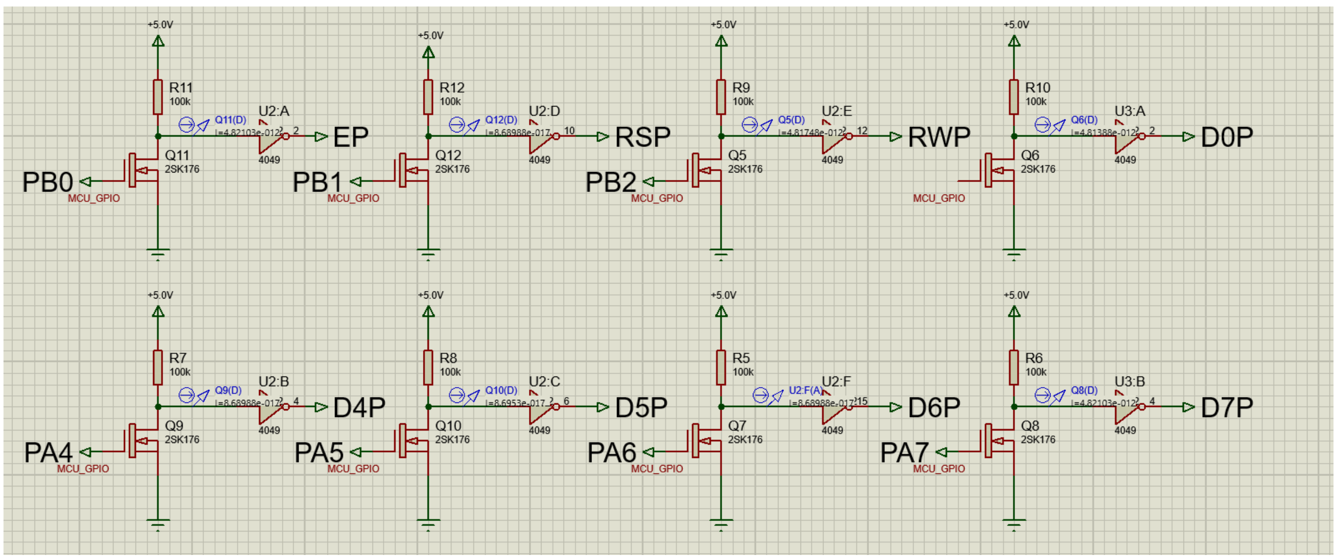
There has been a design change in the process of integration of the PWM signals and the motors. A 2-1 multiplexer has been placed between the PWM signals after they have been translated from 0-3.3 V to 0-5 V, which is used to select between ramping speed up and ramping speed down with a mux select GPIO output. The multiplexer is used to apply positive and effectively negative PWM voltages to the DC motor (the effect of the MUX is to reverse the polarity). The select pins, which are labelled AB1, and AB2 control the multiplxers in connected to the left and right motor respectively. The output of the multiplexers is then sent to the motor controllers, which control the current flow through the DC motors.

The output of each voltage divider circuit for the steering and speed setting potentiometers is connected to a channel of the ADC.

The connectivity of the motor system's PWM signals are translated from 0-3.3 V to a 0-5 V range using the following voltage translation circuit (an open drain connection driving a digital inverter):



These signals drive the 2-1 multiplexers. The connectivity of the GPIO Output signals to the LCD is similar, as the LCD requires a 0 V and 5 V range for digital '1' and '0' respectively:



Additionally, the output voltage from the DC Motor Encoder is translated from the 0-5 V range to the 0-3.3 V range using a voltage divider circuit, which is seen in the Schematics and Simulations section.

The following tests are displayed in this section:

<u>Test 1: (forward steering)</u>

1. The controller is set to ON mode.
2. The speed input potentiometer is set to position 7, which corresponds to a maximum speed of 160 RPM
3. The steering input potentiometer is set to position 4, which corresponds to decreasing the left motor speed by 40% (turning left).
4. The sensed DC motor input is displayed on the LCD.

<u>Test 2: (backward steering)</u>

1. The controller is set to ON mode.
2. The speed input potentiometer is set to position 1, which corresponds to a speed of -320 RPM
3. The steering input potentiometer is set to position 7, which corresponds to decreasing the left motor speed by 20% (turning right while reversing).

4. The sensed DC motor input is displayed on the LCD.

Test 3: (maximum speed)

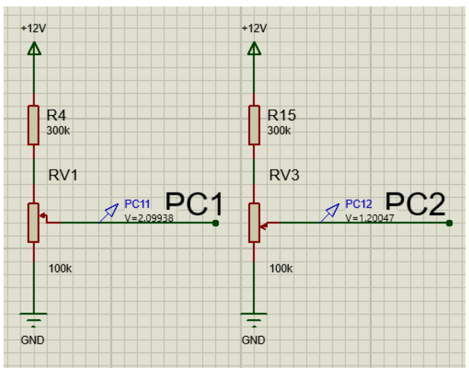
1. The controller is set to ON mode.
2. The speed input potentiometer is set to position 10, which corresponds to a maximum speed of 400 RPM
3. The steering input potentiometer is set to position 5, which corresponds to setting the speed on both the left and right motors to be the same.
4. The sensed DC motor input is displayed on the LCD.

Schematics and Simulations

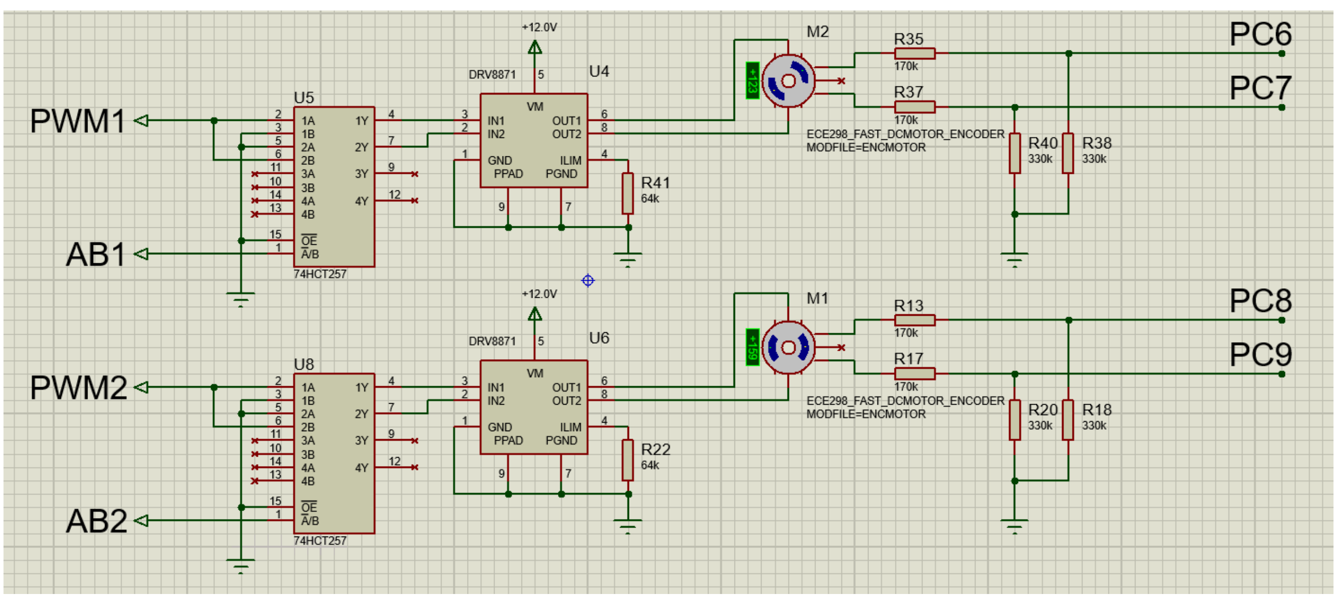
To run these simulations one need only set the ON/OFF switch to the closed position, and adjust the value of the potentiometers.

Test 1: (forward steering)

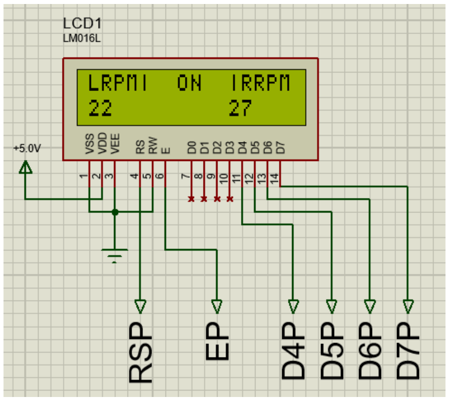
The voltage of the steering/speed potentiometers is set to position 4/7 respectively:



The motor is fed a PWM which ramps up its speed to what is shown below:



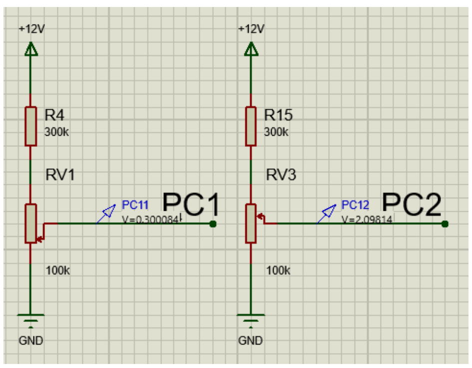
And the PC[6..9] pins are used to sense the rotation speed of the motors and set the duty cycle of the PWM pins in closed loop control. The LCD displays the corresponding mode and rotation speed:



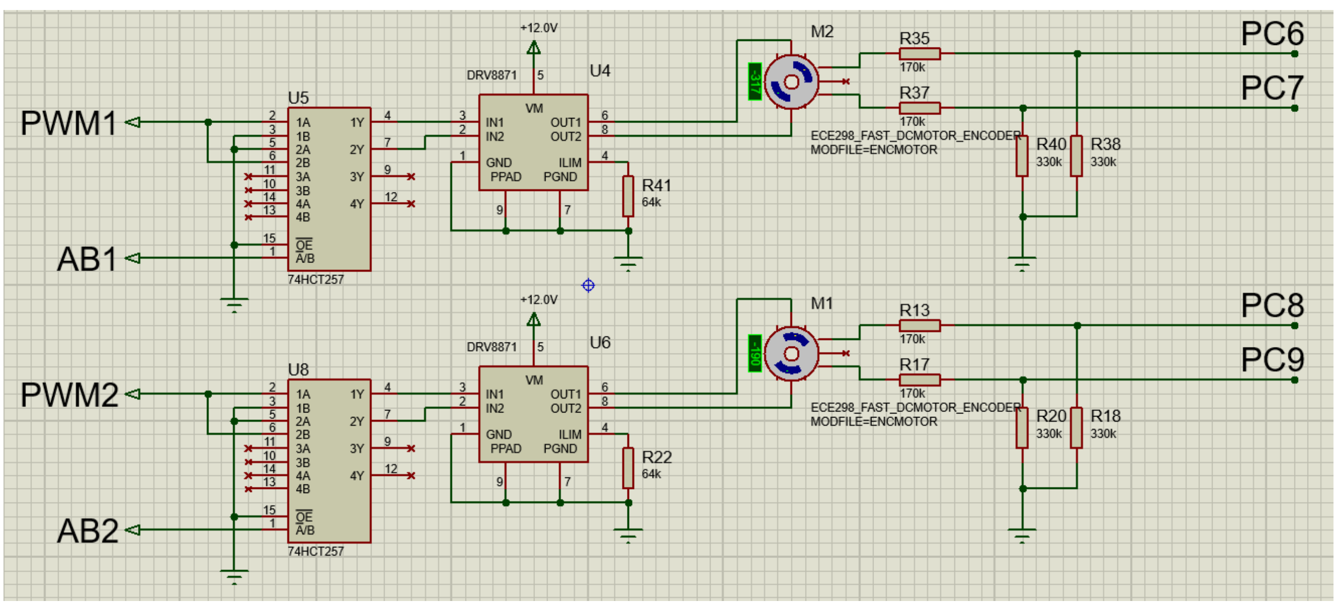
Which corresponds to roughly 132 and 162 RPM for the left and right motors respectively. The test confirms the forward rotation and steering control is as designed, implemented with PID control of the PWM duty cycle fed to the DC motors.

Test 2: (backward steering)

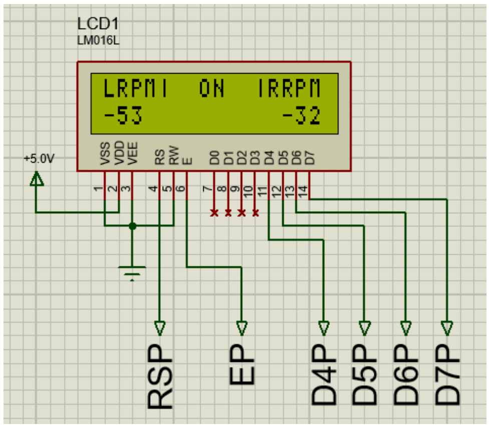
The voltage of the steering/speed potentiometers is set:



The motor is fed a PWM which ramps up its speed to what is shown below:



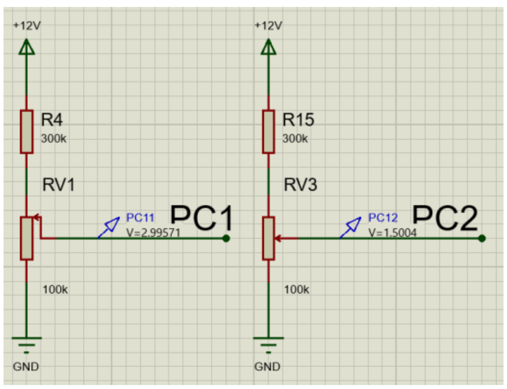
And the PC[6..9] pins are used to sense the rotation speed of the motors and set the duty cycle of the PWM pins in closed loop control. The LCD displays the corresponding mode and rotation speed:



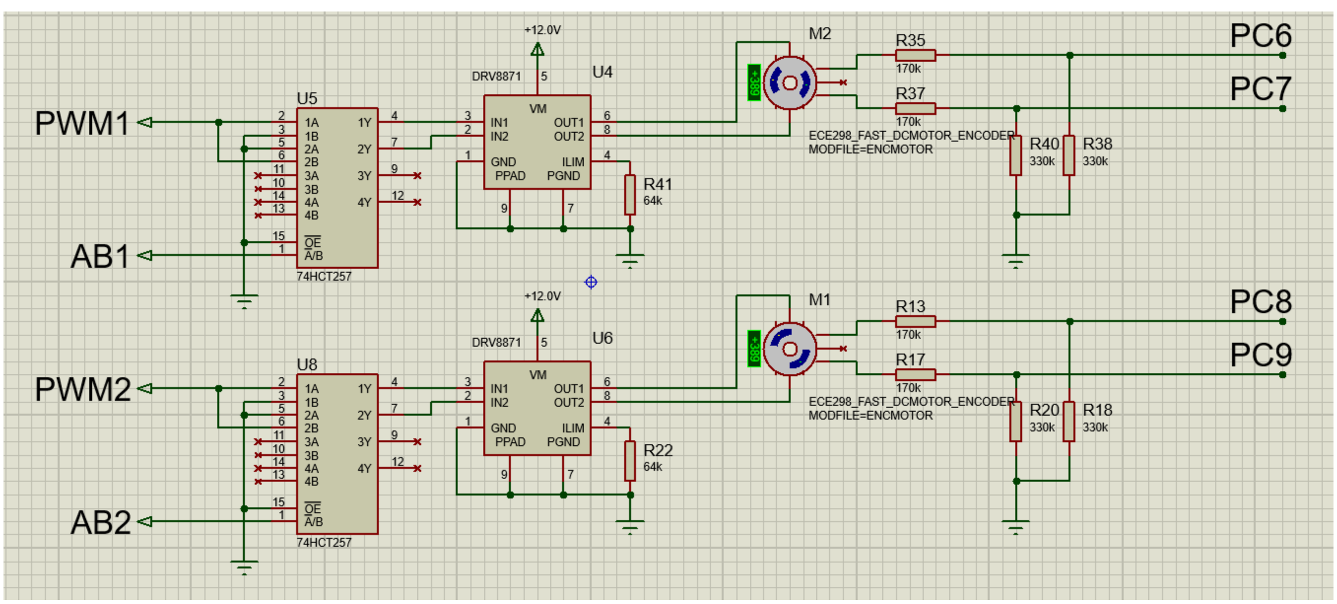
The test confirms the backward rotation and steering control is as designed, implemented with PID control of the PWM duty cycle fed to the DC motors.

Test 3: (maximum speed)

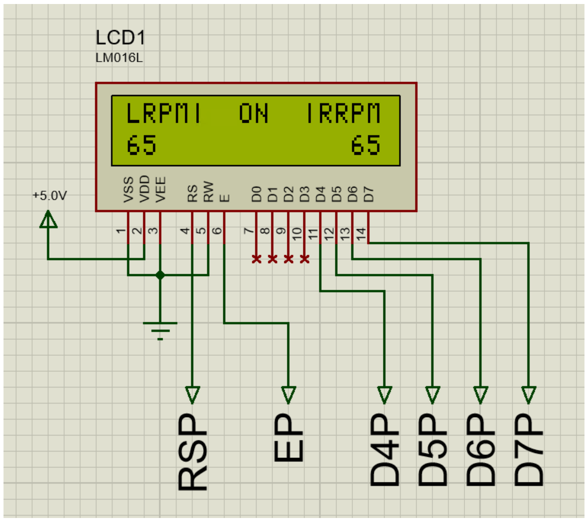
The voltage of the steering/speed potentiometers is set:



The motor is fed a PWM which ramps up its speed to what is shown below:



And the PC[6..9] pins are used to sense the rotation speed of the motors and set the duty cycle of the PWM pins in closed loop control. The LCD displays the corresponding mode and rotation speed:



The test confirms the maximum forward rotation and straight speed control is as designed, implemented with PID control of the PWM duty cycle fed to the DC motors.

Weaknesses

The following weaknesses have been identified in the motor control system. Due to time constraints these weaknesses have not been mitigated entirely.

1. Slow speed motor control
   - When the motor speed is low, the rate at which the Q1 signal is pulsed to indicate the rotation speed of the motor to the MCU is low. When the pulse rate is low, the rotation speed can change significantly before the rotation speed is sensed by the MCU, thus the PID controller experiences a higher overshoot for an extended period of time when adjusting to a motor speed which is below 80 RPM.
2. Overshoot/undershoot
   - Due to the PID motor control performed in software, when ramping the motor speed up/down there is overshoot/undershoot of the motor that dissipates exponentially in time. This is why the rotation speeds in the schematics shown above are not precisely as is set by the steering/speed potentiometers.
3. Smaller maximum speed than expected
   - As shown in the schematics, the maximum rotation speed shown is +- 389 RPM. This is different than 400 RPM, as was shown and tested to be the maximum speed of the motor in the Lab 2 report. This effect is likely due to the output of the DC motor controller being unable to produce the exact voltage that is provided from the MCU.

Battery sensor and LED indicators

Test Summary

The tests performed to ensure the LED indicators diplaying the battery level to the user exhaust the operational range of the system. For all tests, the precharge of the capacitor that simulates the battery voltage was set to a specific value, and the LED ON/OFF states were observed that were set by the MCU.

Each LED is connected in series with power, a resistor, and a transistor. To control the LED the gate voltage of the transistor is connected to a GPIO Output pin of the MCU, which, when set to 3.3 V, will turn the LED to the ON state. The output of the Battery sensor circuit is connected to the ADC.

The TIM3 Output Compare channel cycles every 0.5 s. After each cycle the ADC samples the voltage level of the battery and updates the display of the indicator LEDs.

The tests that were conducted are as follows:

<u>Test 1 (battery capacity $\geq$ 90%):</u>

1. The precharge of the capacitor CBAT was set to 11.5 V.
2. The resulting LED indicators set by the MCU were observed.

<u>Test 2 (80% $\leq$ battery capacity $<$ 90%):</u>

1. The precharge of the capacitor CBAT was set to 10.7 V.
2. The resulting LED indicators set by the MCU were observed.

<u>Test 3 (60% $\leq$ battery capacity $<$ 80%):</u>

1. The precharge of the capacitor CBAT was set to 9.4 V.
2. The resulting LED indicators set by the MCU were observed.

<u>Test 4 (battery capacity $<$ 60%, near top of range):</u>

1. The precharge of the capacitor CBAT was set to 7.1 V.
2. The resulting LED indicators set by the MCU were observed.

<u>Test 5 (battery capacity $<$ 60%, near bottom of range):</u>

1. The precharge of the capacitor CBAT was set to 0.5 V.
2. The resulting LED indicators set by the MCU were observed.

<u>Test 6 (All states observed):</u>

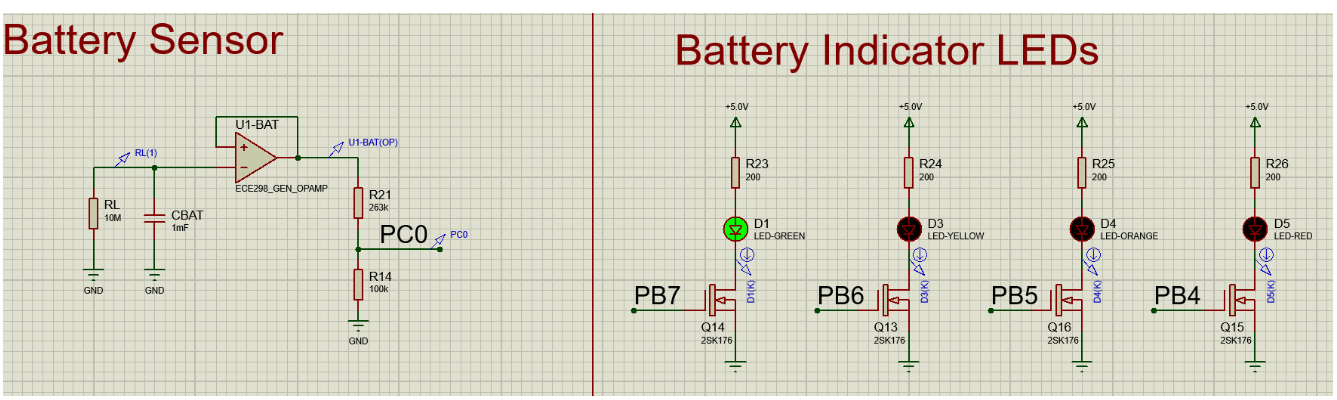
1. The precharge of the capacitor CBAT was set to 12.0 V.
2. The value of R_L was lowered to 200 Ω to reduce the decay time of the capacitor voltage.
3. The resulting LED indicators set by the MCU were observed as the voltage decayed.

Schematics and Simulations

To run these simulations one need only set the ON/OFF switch to the closed position, and adjust the value of the precharge of the battery/capacitor.

<u>Test 1 (battery capacity $\geq$ 90%):</u>

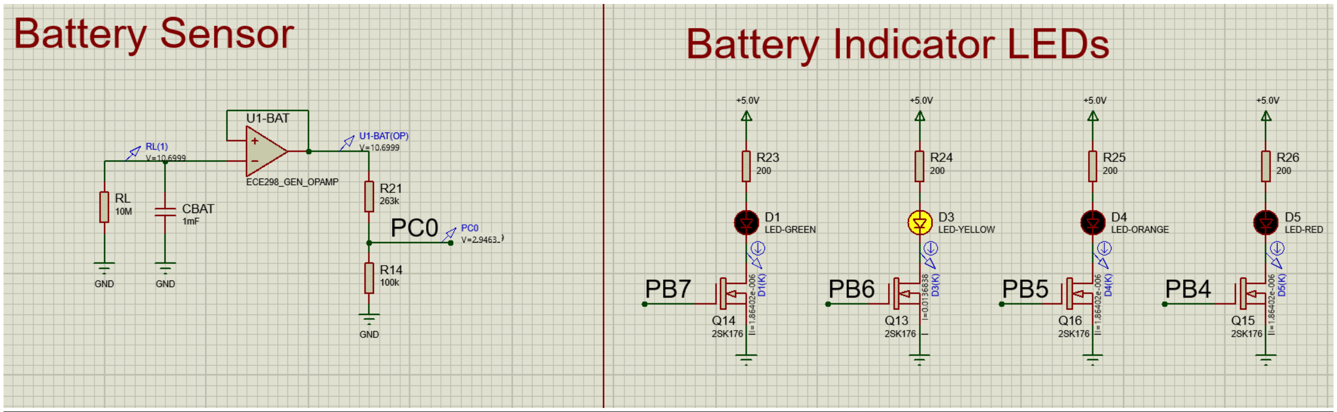
When the battery voltage is in the range [10.8 V, 12 V] the only LED that should be ON is the GREEN_LED. When the precharge of the battery was set to 11.5 V, the following was observed:



This confirms the correct LED indication in this battery range.

<u>Test 2 (80% $\leq$ battery capacity $<$ 90%):</u>

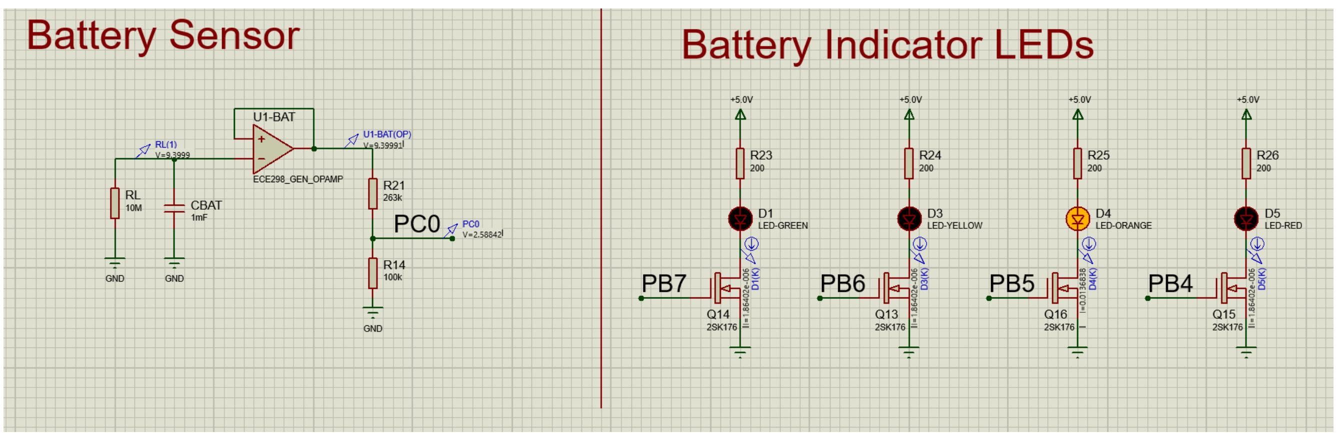
When the battery voltage is in the range [9.6 V, 10.8 V) the only LED that should be ON is the YELLOW_LED. When the precharge of the battery was set to 10.7 V, the following was observed:



This confirms the correct LED indication in this battery range.

<u>Test 3 (60% $\leq$ battery capacity $<$ 80%):</u>

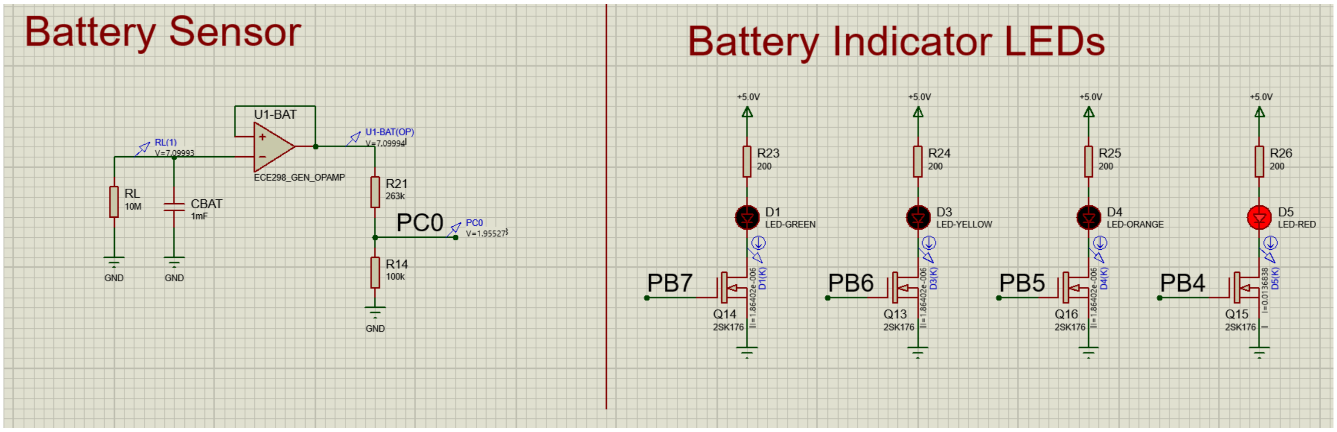
When the battery voltage is in the range [7.2 V, 9.6 V) the only LED that should be ON is the ORANGE_LED. When the precharge of the battery was set to 9.4 V, the following was observed:



This confirms the correct LED indication in this battery range.

<u>Test 4 (battery capacity $<$ 60%, near top threshold):</u>

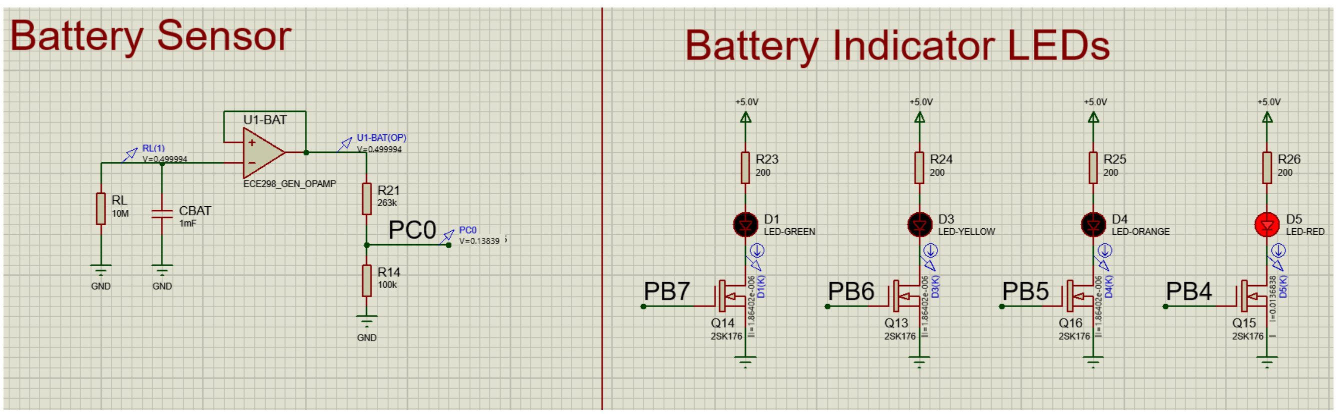
When the battery voltage is in the range [0 V, 7.2 V) the only LED that should be ON is the RED_LED. When the precharge of the battery was set to 7.1 V, the following was observed:



It cannot be shown in image form, but the red led blinks on and off every 0.5 s. This confirms the correct LED indication in this battery range, when the battery voltage is close to the upper limit of the voltage range.

Test 5 (battery capacity < 60%, near bottom threshold):

Since the range of battery voltage is large when the red led must be on, a voltage near the lower limit of the range was also tested. When the precharge of the battery was set to 0.5 V, the following was observed:



It cannot be shown in image form, but the red led blinks on and off every 0.5 s. This confirms the correct LED indication in this battery range, when the battery voltage is close to the lower limit of the voltage range.

Test 6 (All states observed):

It was observed that the state of the LEDs changes as the battery voltage decays. This is not presentable here as it was observed in animation mode in Proteus 8.

Switch Indication and response

Test Summary

The user input of the switch changes the mode of operation of the wheelchair controller. The tests conducted to ensure the user input of the switch correctly adjusts the motor output and the display indicators of the controller.

The switch is connected to power and a GPIO Input pin of the MCU whih is pulled down to ground when the switch is in the open position.

The tests conducted are the following:

Test 1 (Mode = "Locked")

1. The switch input is in the open position.
2. The state of the indicator LED, LCD, Motor RPM, and battery indicators are observed.

Test 2 (Mode = "Run")

1. The switch input is in the closed position.
2. The steering potentiometer is set to position 7 and the speed potentiometer is set to position 8.
3. The state of the indicator LED, LCD, Motor RPM, and battery indicators are observed.

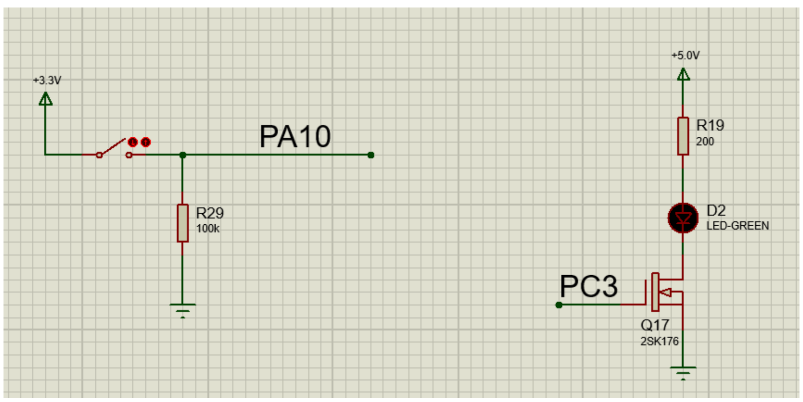
Test 3 (Switching modes)

1. The switch input is in the closed position.
2. In Proteus 8 Animation Mode, the switch is toggled from the open position to the closed position every 30 seconds for 4 cycles.

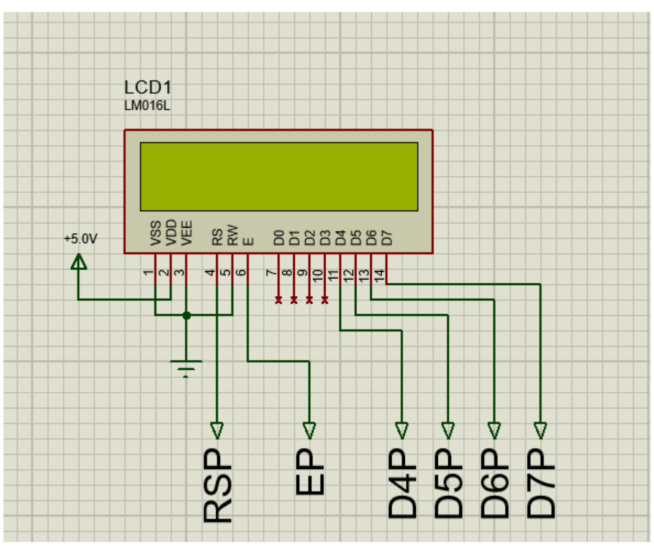
Schematics and Simulations

To run these simulations one need only toggle the ON/OFF switch from the open to the closed position.

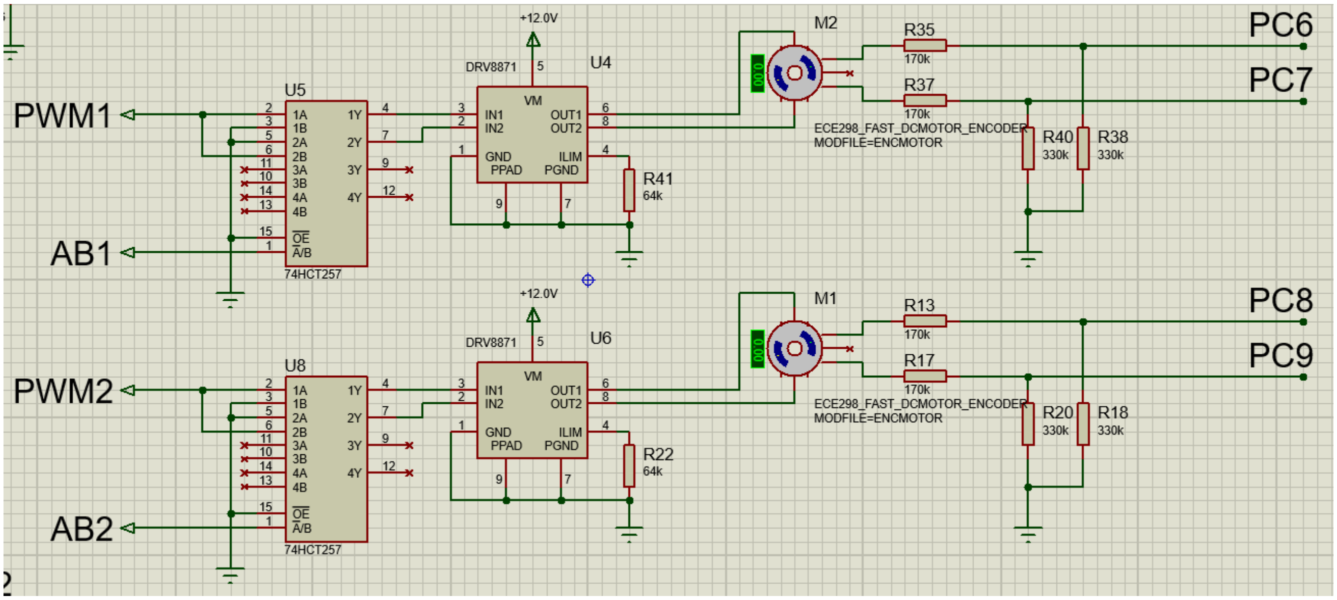
As shown below the switch is in the open position and the indicator LED is OFF to indicate to the user that the controller is in the Locked Mode.



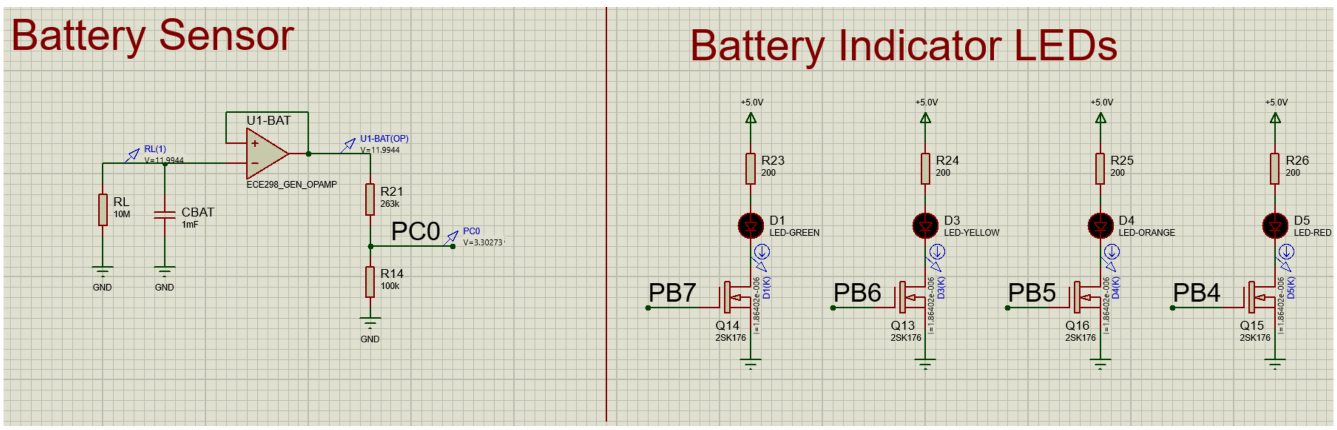
It is also observed LCD is correctly cleared in this mode.



The Motors are provided with no power, and their rotation speeds are also 0 RPM.



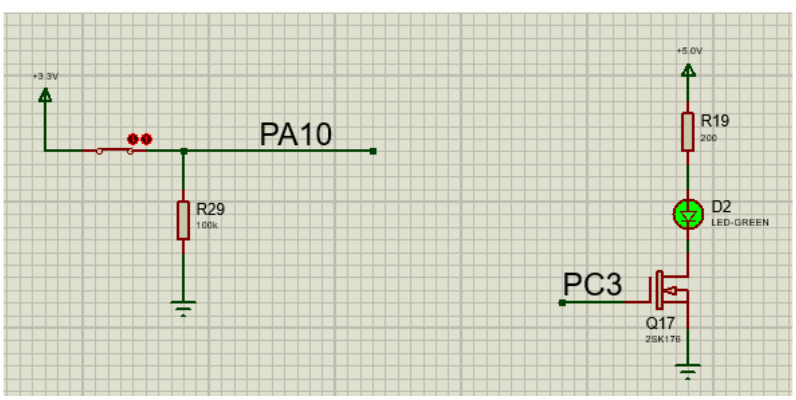
The battery indicator LEDs are also correctly set to the off state.



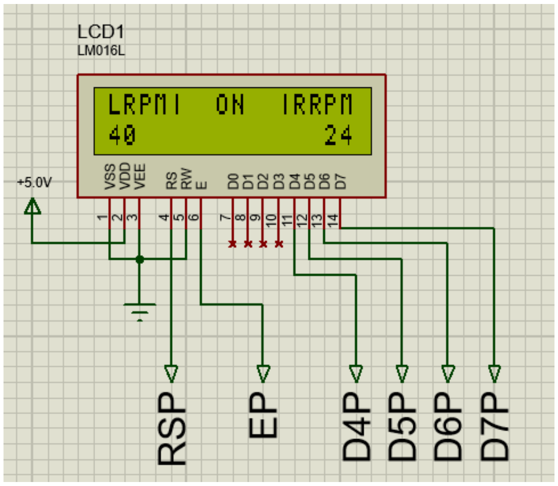
This confirms the correct operation of the controller in "Locked Mode".

Test 2 ("Run" Mode):

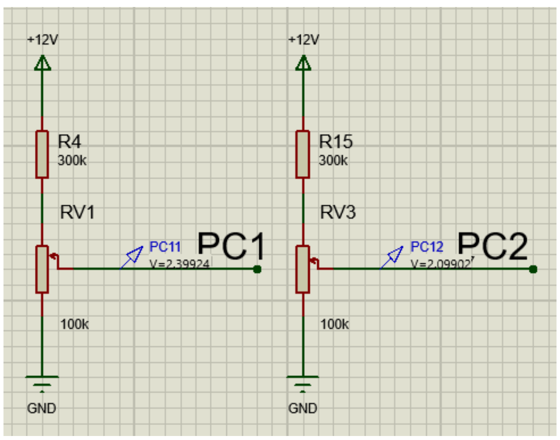
As shown below the switch is in the closed position and the indicator LED is ON to indicate to the user that the controller is in the Run Mode.

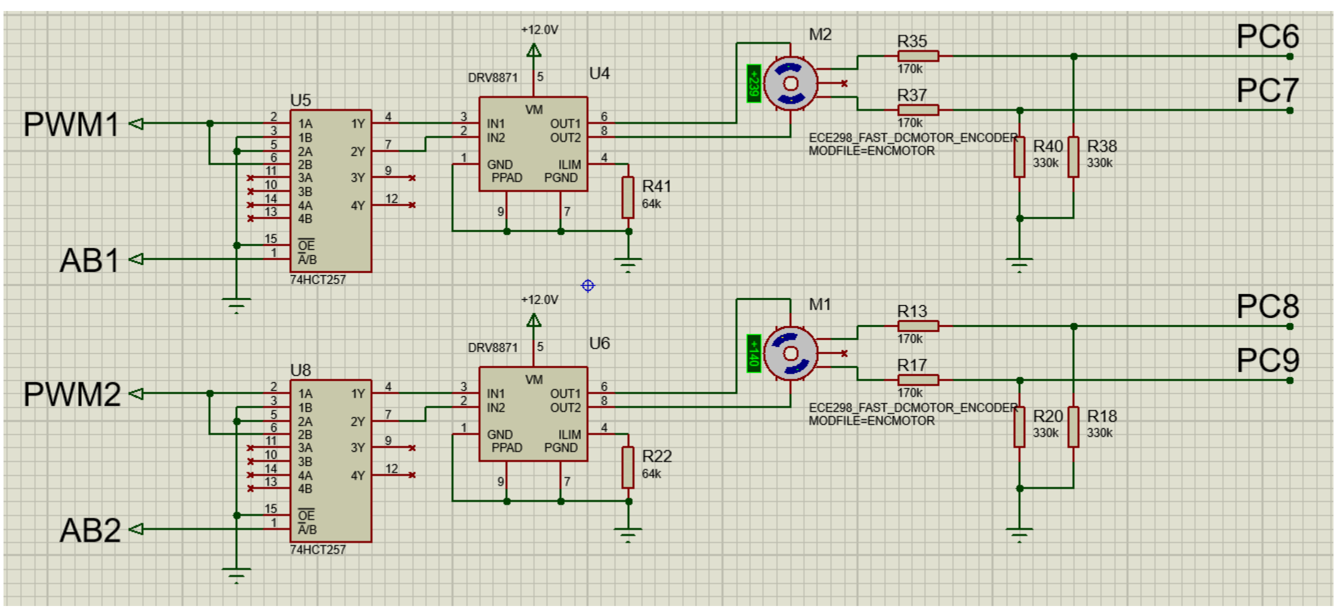


It is also observed on the LCD that the mode (encoded as ON) and the wheelchair RPM is displayed correctly on the LCD.

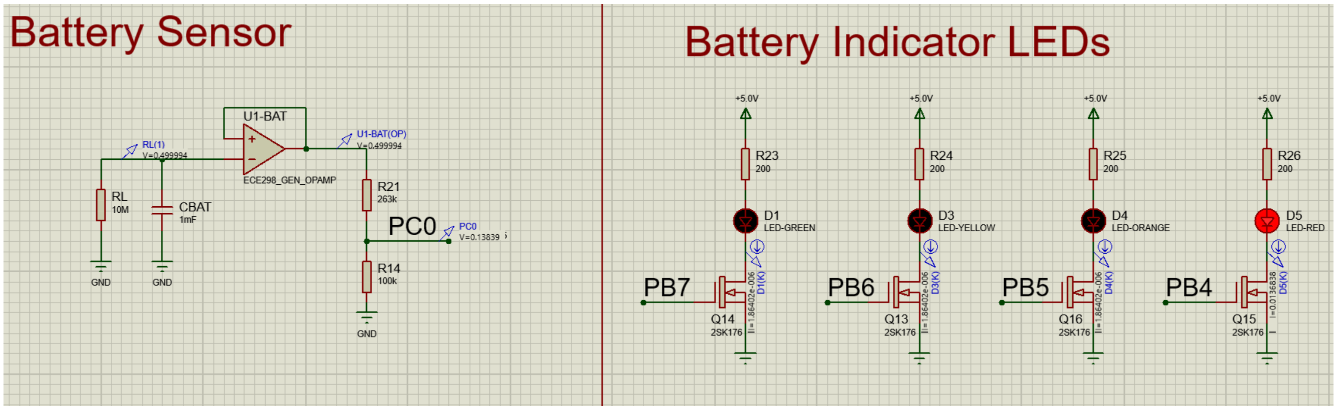


The Motors are provided with PWM signals, and their rotation speeds are also set to the correct user input as provided by the steering/speed potentiometers.





The battery sensor circuit also works as required (it is blinking in animation).



This confirms the correct operation of the controller in "Run" mode.

Test 3 (Switching Modes):

Since this test was conducted in Proteus 8 Animation Mode, it cannot be displayed here. It was observed that the correct user indicators turned to their "Locked" mode displays when the switch was in the closed position, and the motor began to ramp down from its speed when the switch was toggled. When the switch was toggled to the closed position, it was observed that the user indicator displays showed the correct displays for "Run" mode, and the motor was ramped up to the set user input speed from their original speed when the switch was toggled.

This test ensures that the state of the controller can dynamically be switched from ON to OFF while the controller is provided with power.

Weaknesses

The following weaknesses have been identified when the system is switch from ON to OFF mode. Due to time constraints these weaknesses have not been mitigated entirely.

1. Motor control turn off speed:
    a. When the controller is switched from "Run mode" to "Locked mode" the set-point of the motor is set to 0 RPM for both the left and right motors. The same problem is encountered when the motor speed is set to a small RPM; the MCU cannot accurately sense the rotation speed of the motor. To mitigate this, when the MCU senses that the rotation speed of the motor is less than 30 RPM, the PWM signals that are supplied to the motor controller are turned off.

Code Attribution

The code module that is used to write to the LCD has been taken (with permission from the course instructors) from MYaqoobEmbedded STM32 Tutorial Github Repo at:

https://github.com/MYaqoobEmbedded/STM32-Tutorials/tree/master/Tutorial%2011%20-%20LCD16x2