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\_Input | Controls the full system by turning it On/Off. |
| 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. |
| PB3 | GPIO\_Output | Controls the red LED indicating battery voltage. |
| PB4 | GPIO\_Output | Controls the orange LED indicating battery voltage. |
| PB5 | GPIO\_Output | Controls the yellow LED indicating battery voltage. |
| PB6 | GPIO\_Output | Controls the green LED indicating battery voltage. |
| PB7 | 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 that is sent to control the left DC motor controller. |
| 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 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 DC motor encoder’s outputs. |
| NVIC | Receives voltage from 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 input/output test cases. The DC motor is controlled with a software-based PID controller that varies the duty cycle of the PWM signal that is input to each of the DC motor controllers. 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 voltage. Due to the overshoot of the PID controller, there is an error of ~5% of the set-point RPM that is observed on the DC motors.

It should be noted that a 2-1 multiplexer is used to apply a backward rotation to the DC motor, the select pins of which are labelled AB1, and AB2 for the left and right motor respectively.

The following tests are displayed in the following section:

**Test 1: (forward steering)**

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 2, which corresponds to decreasing the left motor speed by 60% (turning left).
4. The sensed DC motor input is displayed on the LCD.

**Test 2: (backward steering)**

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 9, which corresponds to decreasing the left motor speed by 80% (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

**Test 1: (forward steering)**

The voltage of the steering/speed potentiometers is set:

Chart

Description automatically generated with medium confidence

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

Diagram, schematic

Description automatically generated

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:

Diagram, schematic

Description automatically generated

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:

Chart

Description automatically generated

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

Diagram, schematic

Description automatically generated

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:

Diagram, schematic

Description automatically generated

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 1: (forward steering)**

The voltage of the steering/speed potentiometers is set:

Diagram, schematic

Description automatically generated

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

Diagram, schematic

Description automatically generated

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:

Diagram, schematic

Description automatically generated

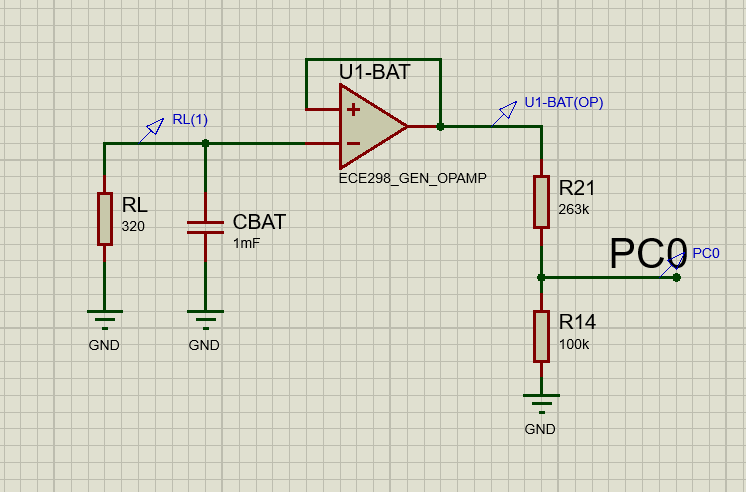
The test confirms the maximum forward rotation and straight speed control is as designed, implemented with PID control of the PWM duty cycle fed to the DC motors.

## Battery sensor and LED indicators

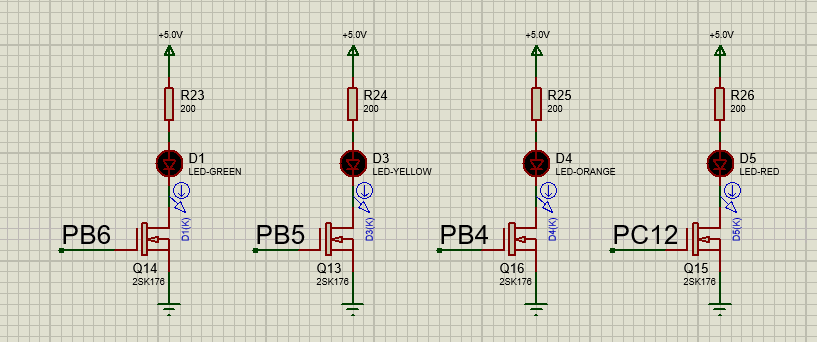
### Test Summary

This test involved the testing of the battery level sensor and its integration with the LEDs that are used to provide user output. The battery circuit was edited so a faster battery dissipation could be simulated which would help us in seeing the the LED output for its test. This setup will display the LEDs changing (to show battery level) over time as the battery dissipates. The green LED when on, displays a battery level of 90% or greater. The yellow LED when on, displays a battery level that ranges from 80% - 89%. The orange LED when on, displays a battery level that ranges from 60% - 79%. The red LED when on is always flashing (as per lab requirements) as it displays a battery level that is less that 60%. When the system is in the locked mode then all the LEDs are off, they only turn on when it is in the run mode. The green LED is connected to pin PB6 on the MCU. The yellow LED is connected to pin PB5 on the MCU. The orange LED is connected to pin PB4 on the MCU. The flashing red LED is connected to pin PC12 on the MCU.

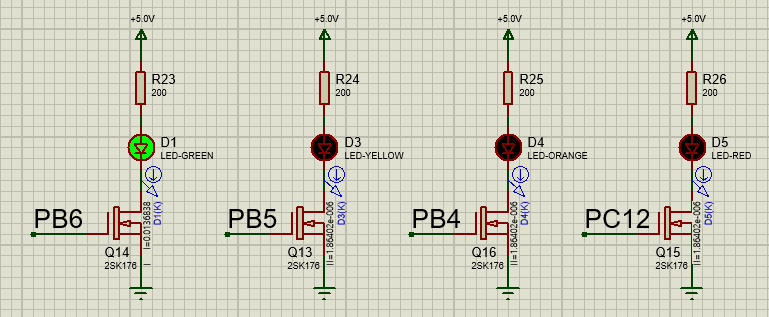
### Schematics and Simulations

This is the battery setup this is used to test the LED indicators for the battery level:

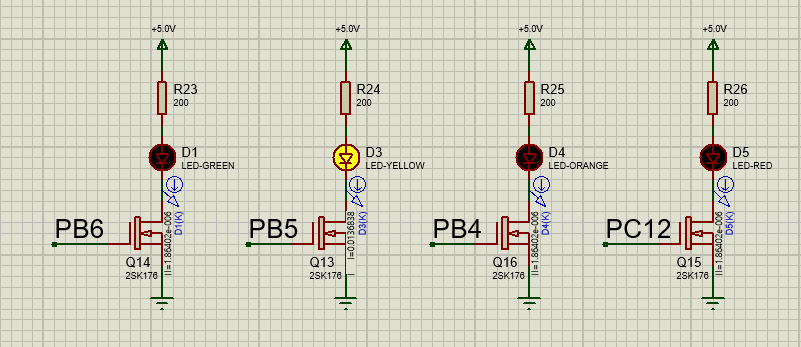
The setup below shows the four LEDs placed in a line (from left to right: green, yellow, orange, red) in the locked mode, that’s why all of them are off:



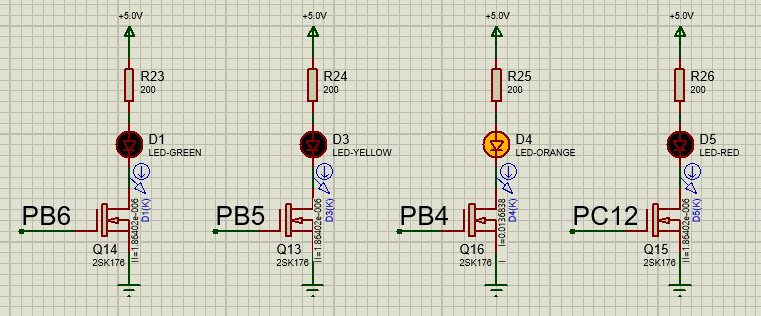
Now that the system is in the run mode and the battery is being used, you can see that the leftmost LED (green) turns on to depict a battery level that is greater than 90%. This is depicted in the picture below.



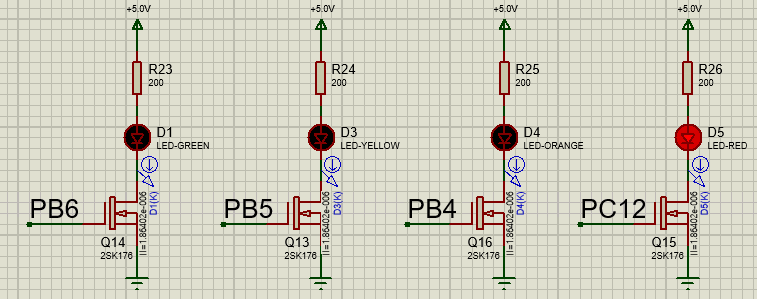
Now over time as the battery dissipates you see the green LED turn off and the yellow LED turn on. Now we can see that the battery level has dropped to a range that is between 80% - 89%. This is depicted in the picture below.



As time goes on even more, the battery dissipates further and the yellow LED turns off and the orange LED turns on. This shows that the battery level is now in the range of 60% - 79%. This is depicted in the picture below.



After a while when the battery level drops even further then the yellow LED turns off and the red LED turns on. One difference between the red LED and the other LEDs is that the red LED doesn’t just stay on, instead it is constantly flashing as this is a requirement as per the lab instructions. This shows that the battery level has dropped below 60%. This is depicted in the picture below.



These tests show us that the desired response is retrieved from the machine and it is working just as expected.

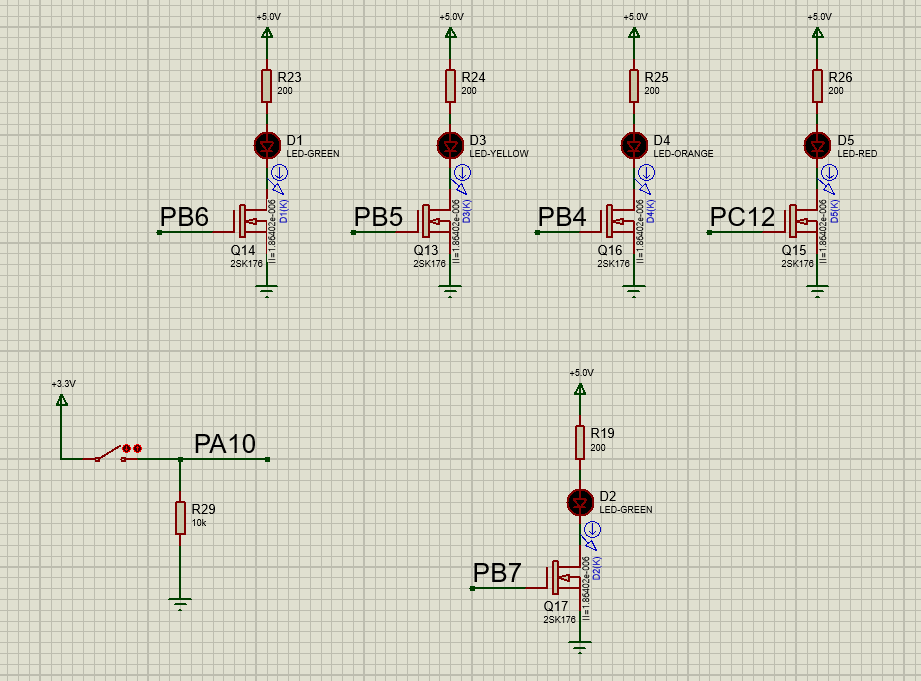
## Button Indication and response

### Test Summary

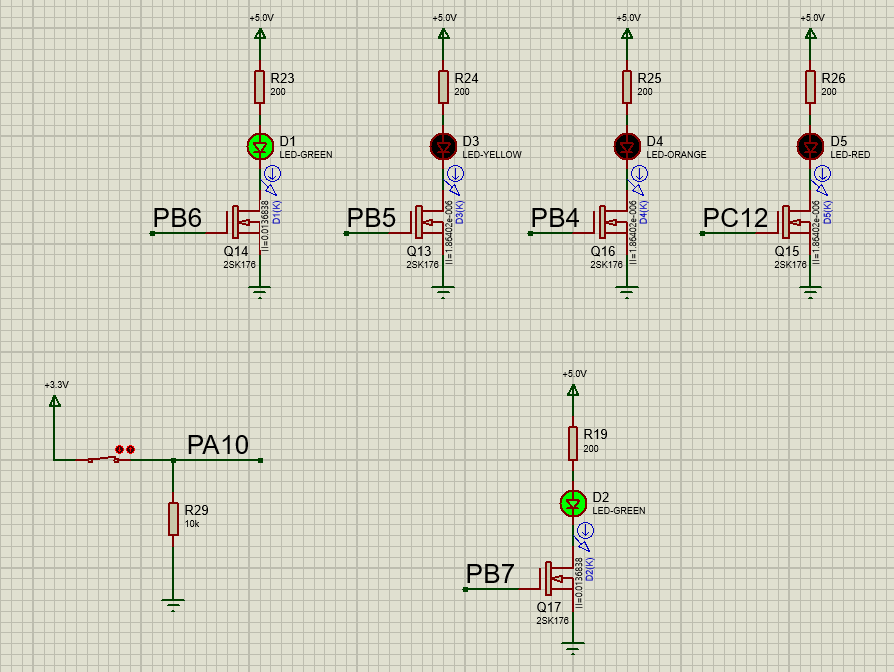
In this system a single switch is implemented to switch between the two modes that are offered by the system. The switch when set to the On position sets the system to the run position while the switch when set to the Off position sets the system to the locked position. There is also a green LED beside that is used to output the mode the system is currently in. If the switch is set to the off position then the LED remains off to depict that it is in the locked position and when the switch is set to the On position then the LED is turned on to depict that that the system is in the run position. The switch is connected to the MCU as a GPIO Interrupt on the PA10 pin. The LED that shows us the mode of the system is connected to pin PB7 on the MCU.

### Schematics and Simulations

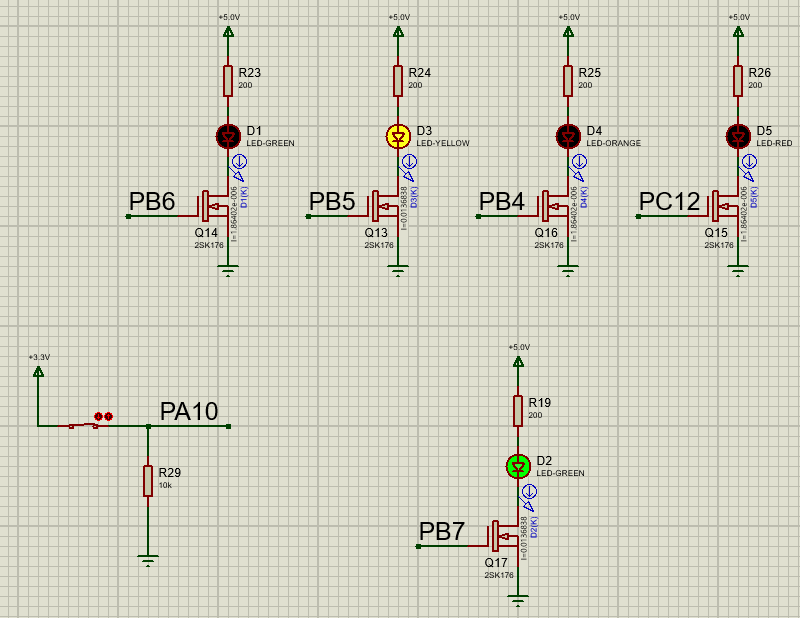
When the simulation is run the switch is in the off position to depict that it is in the locked mode and so the LED (to the right of the switch) is also off. The four LED lights in a row on the top depict the battery level and they are also all off as the switch is in the off position. This is depicted in the picture below:



The switch is now turned On to put the system in to the run mode. This turns on the green LED (to the right of the switch) which is used to show that the system is in the run mode. The green LED (leftmost of the set of four LEDs on top) is also now On as it depicts the battery level (90% and above) of the system. This is shown in the picture below:



The picture below shows the same thing as the picture above but now the battery level has dropped, that’s why now the green LED isn’t On but the yellow LED is (battery level between 80% - 89%).



These tests above help us conclude that the system is behaving as expected and passes all our requirements.