Lab 1 – Feasibility Model Phase 1

ECE 298 – S2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Members** | Team Member 1 Barry Cimring  Team Member 2 Anweshi Avadya | **Team Number** | 90 |

# Part 1 – Project Design Requirements

1. **Functional Requirements**
   1. The speed and direction of both wheel must be controlled by two (2) potentiometers acting as speed and steering inputs. The result of the potentiometer movement must correspond to max/min speeds and max/min turning directions.
   2. The motorized wheelchair must be able to rotate its wheel at a maximum speed of 1.67 m/s and minimum speed of -1.67 m/s when moving forward, which is approximately the speed of walking human adults.
   3. …
2. **Non-functional Requirements**
   1. The ramp-up of the motor controlling the speed of the wheelchair must be pleasing to the user of the wheelchair.
   2. The bulkiness of the hardware components of the project must be small enough such that the user will not think it stands out on the wheelchair.
3. **Constraint Requirements**
   1. The form factor of the PCB on which the hardware components for the design is built must be \_\_\_.
   2. The total cost of the wheelchair control system must not add more than $50 to the total cost of the wheelchair.

# Part 2 – Project Considerations for I/O

## **Project Sensors and User Inputs**

**Potentiometer**

The potentiometer will act as a user input to the control system that determines the speed and direction that the wheelchair must go. The range of the potentiometer should be from 10 k to 100 k which will be connected to an analog input pin of the MCU in a voltage divider circuit (more described below in ‘Potentiometer Connection’).

**Potentiometer Connection**

Both potentiometers will be connected as the first resistor in a voltage divider circuit powered by the battery. At the node between the potentiometer and the second resistor, one of the ADC IN pins will be connected. This second resistor will be on the order of 100 k to ensure that the minimum sensed voltage of the ADC is 3.3/2 V = 1.65 V and the maximum voltage is 3.3 V \* 3 V. This will correspond respectfully to maximum/minimum speed and maximum left/maximum right turning for speed and steering. To identify whether or not a change in the potentiometer has been made, the potentiometer’s change in input will trigger an interrupt that the MCU will deal with my changing the speed and steering direction of the wheelchair.

**Battery-Level Sensor**

A circuit will be connected to the external battery to measure its voltage level over time. It is assumed that the maximum possible battery that this project requires is 20 V. The battery-level sensor range must be able to transform a 20 V input voltage into a voltage on the range of 0-3.3V as input to the MCU’s ADC.

**Batter-Level Sensor Connection**

The circuit that will sense the battery level, whose maximum possible value is assumed to be 20 V, will be connected to the ADC peripheral device of the MCU. The voltage of the battery will be sent through an op-amp buffer to isolate it from the ADC. The output of the voltage buffer will then be connected to a voltage divider whose output range (for an input range of 0-20 V) is 0-3.3 V, the maximum voltage of the ADC. This can be achieved with a R! and R@ ohm resistor.

**DC Motor Encoder**

The motor encoder that will be used in this project is also part of the motor. The motor encoder outputs 3 square waves whose frequency is dependant on the speed of rotation. The output labelled ‘IDX’ send a pulse whenever the zero value of the wheel is rotated around (i.e. a revolution has occurred). The two outputs labelled ‘Q1’ and ‘Q2’ will determine the angular speed (which can be forwards or backwards) and the absolute angular position.

**DC Motor Encoder Connection**

The outputs of the motor encoder (IDX, Q1, Q2) will be sent as inputs to three of the GPIO pins of the MCU. A counter will be used to time the rate at which IDX pin transmits a pulse (to determine rotational speed) and a SOMETHING IDK YET will be used to determine which of Q1 or Q2 is high while the other is low to determine the direction of rotation.

**Pushbutton**

**Pushbutton Connection**

## Project Actuators and User Outputs

**Motor + Motor Encoder**

An actuator required for the project is a DC motor that will control the speed of rotation of the wheelchair’s wheels. It is assumed that the number of revolutions of the motor to the number of revolutions of the wheelchair’s wheels is 6:1. Thus, as per the functional requirement, the maximum/minimum rotation speed of the motor must be (if we assume that the wheel is of 0.25 m radius): . Therefore, the max/min rpm of the wheel is .

**Motor + Motor Encoder Connection**

Because the DC motor requires a negative voltage to spin in reverse, the motor will be connected to an analog OUT pin of the MCU via a series of op-amps. The first op-amp will reverse the polarity and reduce the value by a factor of two of the 3.3 V signal from the MCU. The second op-amp will add the voltage from the previous op-amp to an analog output signal from the MCU from 0 to 3.3 V. This puts the voltage range of the input to the motor at -1.65 – 1.65 V. In Part 3, when testing the motor we see that this is enough range to achieve the maximum/minimum speeds desired.

**Multicoloured LEDs**

**Multicoloured LED Connection**

**LCD – Liquid Crystal Display**

The liquid crystal display (LCD) will be used to display the that the wheelchair is in and the RPM of both wheels on the wheelchair. The display will show 32x2 characters. The top row will display the mode that the wheelchair will be in (Locked or Run mode), and the bottom row will display the RPM of the left and right wheels.

**LCD – Liquid Crystal Display Connection**

The LCD has \_\_ inputs. The inputs and their connections are below:



The LCD must be controlled via a read/write operation as described in the \_\_\_ datasheet. To achieve this, the \_\_\_ pin of the MCU will be connected to \_\_\_ respectfully. This will send data to the MCU continuously to display the rpm value of the wheel and modes.

## Project MCU Internal Resources

**ADC – Analog to Digital Converter**

**Internal MCU timer**

**GPIO**

**Interrupts**

**Software Parameters**

The software parameters that must be kept track of are:

1. The voltage level of the battery.
2. The angular speed of the two motors attached to the wheels.
3. The angular direction of the two motors attached to the wheels.
4. The desired user input (after processed from the potentiometer input) of the speed of the wheelchair.
5. ANYTHING ELSE

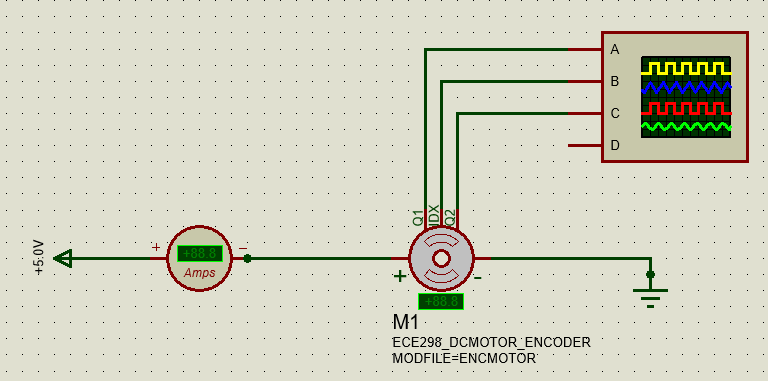
# Part 3 – Device Testing Methodology

## Device 1 – ECE298\_DCMOTOR\_ENCODER

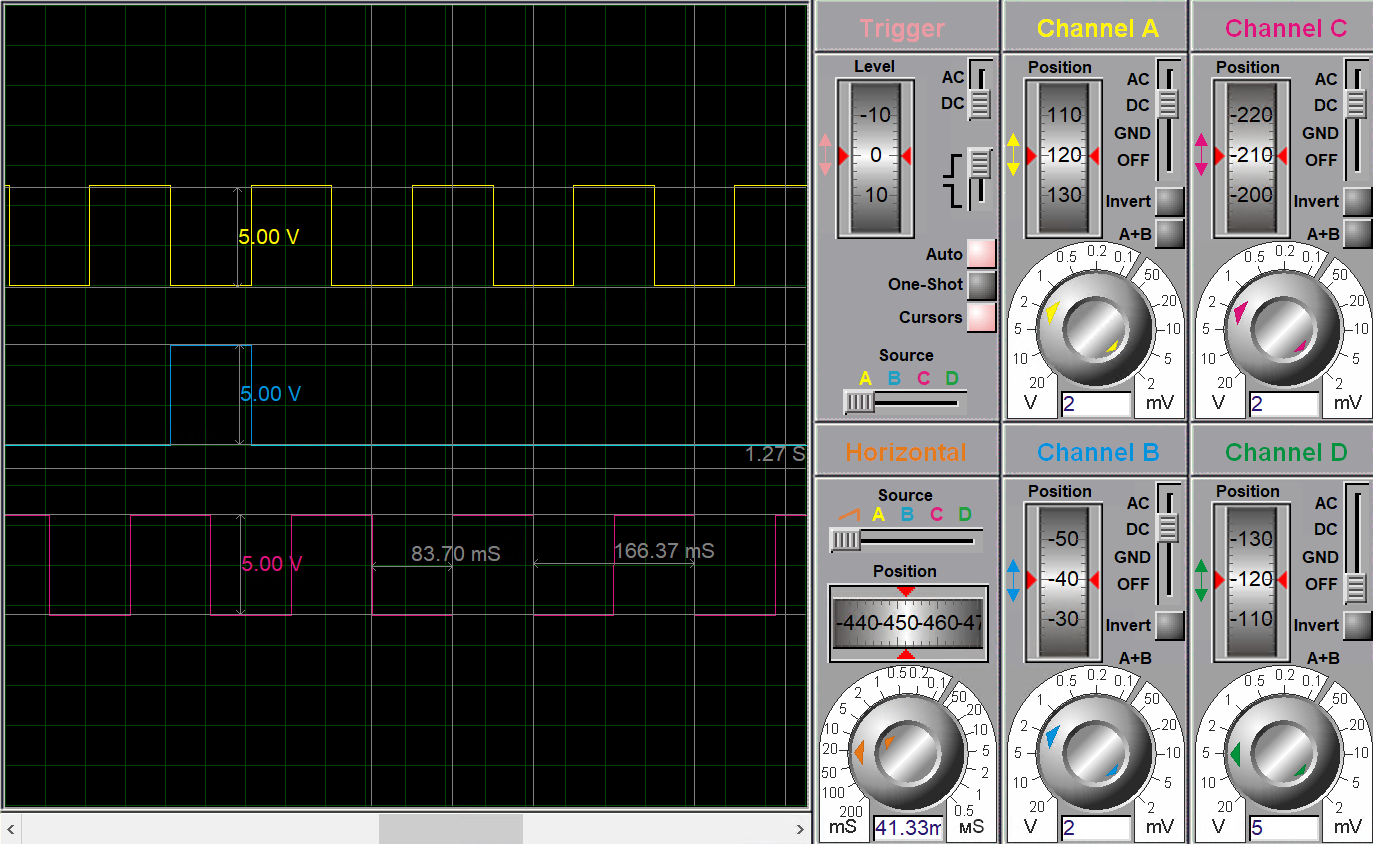
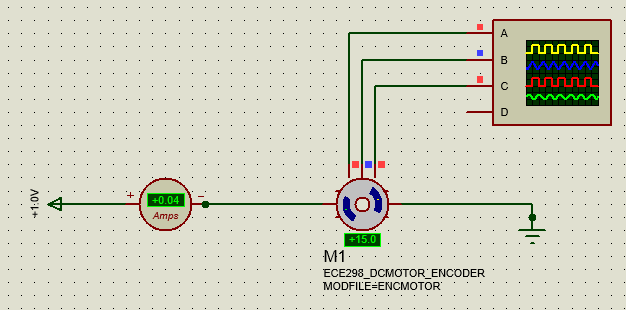
The ECE298\_DCMOTOR\_ENCODER is the DC motor that will be used to control the speed of revolution of the wheels on the wheelchair. This motor’s rotation speed is controlled by a DC current, and the current that the motor draws from a power source is proportional to the rotation speed of the motor.

To control the device, a voltage source will be connected to the + pin and the – pin of the motor will be connected to ground. For testing this is shown as a voltage source but in reality, this will be an analog output pin from the MCU.

The schematic used to test the DC brushless motor is below:

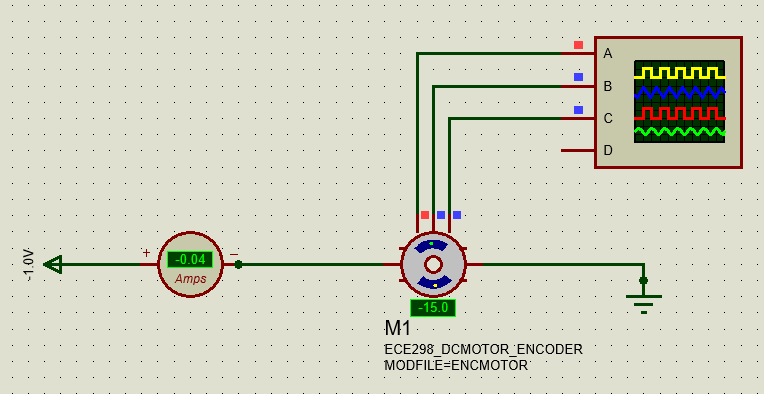


The DC motor’s + pin is connected to a voltage source, which draws a current. The output Q1, Q2, and IDX pins are connected to an oscilloscope to monitor the output of the motor. It is seen that the rotation speed (in RPM) of the motor is directly proportional to the voltage supplied: 0.04 A, the motor’s rotation speed is 15 RPM, and at 0.08 A the motor rotates at 30 RPM. Below is an oscilloscope capture of the output pins for the same schematic:



As shown, for 0.04 A input (where the voltage signal is 1.0 V) the output of Q1, Q2 is a square wave from 0-5 V with a period of 166.37 ms. This period is inversely proportional to the rotation speed. The duty cycle of the square wave is 50%. When the motor is rotating in the forward direction Q1 turns from low to high when Q2 is low.

We see that, in the same way, the motor will rotate in the opposite direction with the same magnitude of RPM if a negative voltage is applied:

INSET IMAGE

The same angular frequency occurs in the negative direction when the negative current is applied, and the rising edge of Q1 occurs when Q2 is high.

In a more zoomed out oscilloscope diagram (in the time axis) it is seen that at 0.04 A a pulse is sent by IDX every \_\_ s, and at 0.08 A a pulse is sent every \_\_ s:

INSERT HERE

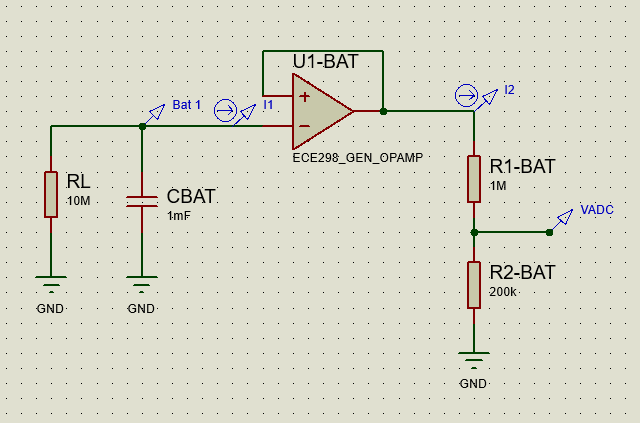
This motor has the output built-in to it so the MCU can receive input directly from the same component. Both the motor (actuator) and the motor encoder (sensor) has been tested in this section.

## Device 2 – Battery-Level Sensor

The battery-level sensor circuit is designed to output a voltage between 0-3.3 V given an input range of 0-20 V, which is the assumed maximum voltage of an external battery for this application.

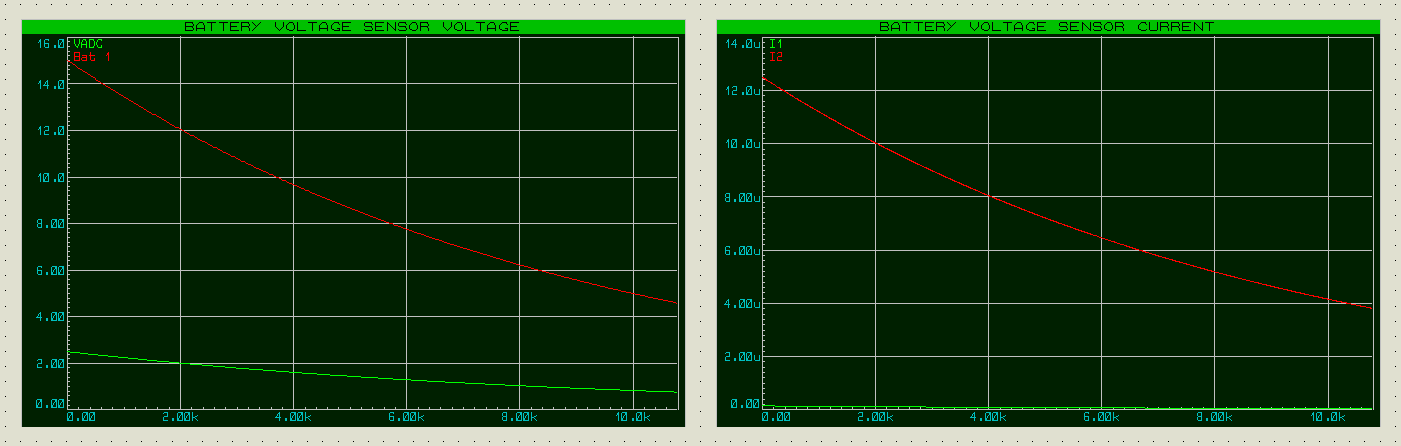
The output of this circuit will feed into the ADC peripheral of the STMicro…, which will be converted to a digital signal that the firmware of the MCU can read. Depending on the battery level, the MCU will determine which LEDs (indicating battery percentage) will be turned on.

A schematic for the battery-level sensor is shown below:



The battery is modelled by a large capacitor – CBAT – whose initial voltage is set to 15 V. The battery voltage is sent through a buffer to isolate the signal from the rest of the circuit (RL, representing the circuit load), which is the input of a voltage divider providing the correct maximum output rated for the MCU’s ADC. VADC will feed into the ADC of the MCU in the final implementation.

Due to the modelling of the external circuit as a large circuit, the battery’s voltage will drop exponentially over time (since the current drawn by the op-amp buffer is negligible). The lifetime of this battery is (rounded down). This is shown below in the time-domain voltage and current of VADC, Bat 1, I1, I2:



As shown the battery, in this configuration, retains at least 50% of its charge for 5.6k s = 1.5 h, which is exponentially decrease with the lifetime as stated above. The current drawn from the battery is negligible, and the current drawn in the voltage divider circuit is of the order of uA, which is also satisfactory. In this example the initial voltage read by the ADC would be 2.5 V – indicating to the MCU the battery is at 15 V charge.

## Device 3– ECE298\_

pushbutton

## Device 4– ECE298\_

potentiometer

## Device 5– ECE298\_

led

# Part 4 – System-Level Design

Insert block diagram and description