

# Lab 2 – Feasibility Model Phase 2

ECE 298 – S2021

Lab Section: N/A

Group: 90

## DC Motor

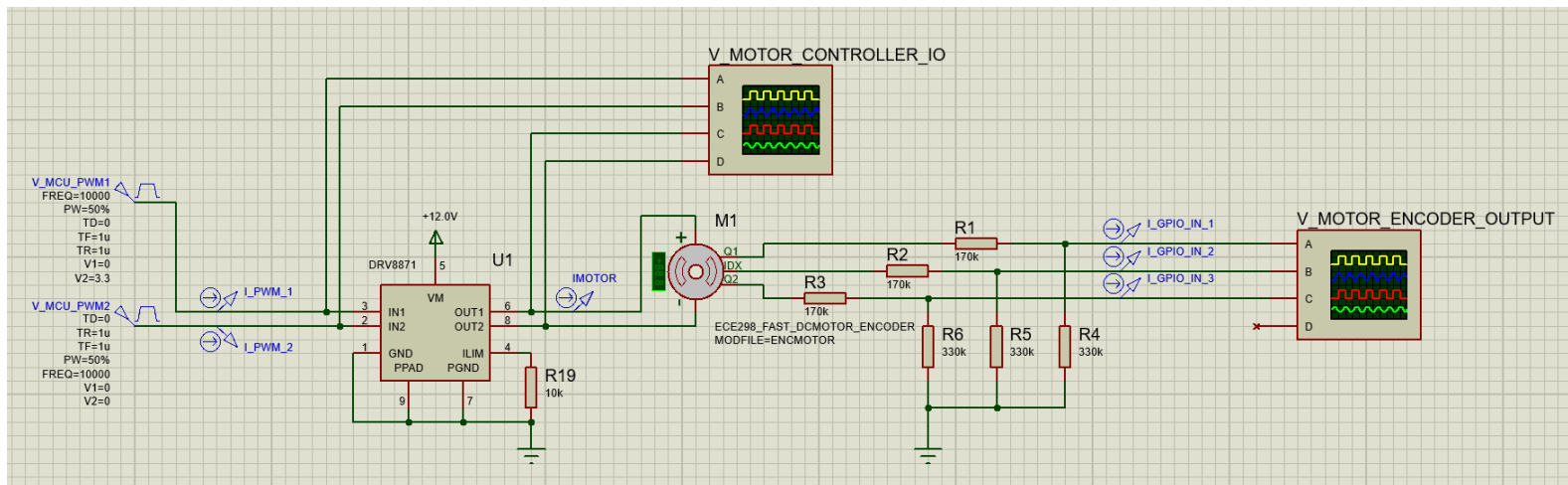
### Summary

Item	Description
Purpose	The purpose of the DC motor/motor encoder is to function as an actuator that controls the rotation speed of the wheels. In addition to controlling the rotation speed, the motor encoder will send information back to the controller to indicate its rotation speed. In this sense it is also a sensor.
Device physical domain and range	The speed of rotation of the motor no load speed of 800 RPM. Thus, the rotation speed ranges from 0-800 RPM. The coil inductance of the motor is 100 mH and the coil resistance of the motor chosen is 10 Ohms.
Device type chosen	The motor chosen for simplicity of design is a brushed DC motor. The brushed DC motor's rotation speed is proportional to the average value of the current applied to it.
Proteus Library component name	The component used to implement this function is the ECE298_FAST_DCMOTOR_ENCODER.
Device input / output properties	<p>The input of the motor is a current, where the rotation speed of the motor is proportional to the average value of the current through the motor. Due to the large current required by the motor, the motor will be connected to a motor controller that will be controlled by the MCU using a PWM signal from 0-3.3V. A PWM input applied to two inputs of the motor controller and will control the amount of average current applied to the motor.</p> <p>The output of the motor are three pins, Q1, Q2, and IDX. Each of these pins output a digital signal from 0-5V, which must be sent through a voltage divider before reaching the MCU input pins in a range of 0-3.3V. The IDX pin is pulsed once per rotation, indicating the speed of the motor. The Q1, Q2 pins are pulsed once every 24<sup>th</sup> of a rotation, indicating the absolute angle of the motor. Depending on which of Q1 or Q2 rises high when the other is low, this indicates if the motor is moving forwards or backwards. The outputs of the DC motor (encoder) is a push-pull output.</p>
Device input / output range	<p>Through the motor controller, the DC motor will be sent a 0-12 V PWM signal whose duty cycle depends on the PWM signal that is sent to the motor controller.</p> <p>The devices outputs (Q1, Q2, IDX) are square wave pulses whose range is 0-5V.</p>

Item	Description
MCU connectivity details	<p>The device will be connected to the two output pins of the motor controller where it will be supplied a PWM signal with varying duty cycle at 0-12V. The motor controller will be controlled by two PWM signals from the MCU's TIMx pins. These two timer pins will control the exposure of the motor to current using a PWM. The output of the motor encoder will be connected to the MCU GPIO input pins via a voltage divider to step down the maximum voltage from 5V to 3.3V for the MCU to read.</p> <p>The electrical properties of the MCU GPIO pins are:  <math>V_{IL}(\max) = 0.6\text{ V}</math>  <math>V_{IH}(\min) = 2.0\text{ V}</math></p>
Device/MCU interfacing details	<p>The device will be connected to the motor controller with is controlled with a digital signal (0-3.3V PWM) and outputs another digital signal (0-12V PWM). The output of the motor encoder is three digital signals (0-5V PWM). Thus, the MCU interfacing of the entire motor + motor controller + motor encoder will be digital to digital interfacing.</p>

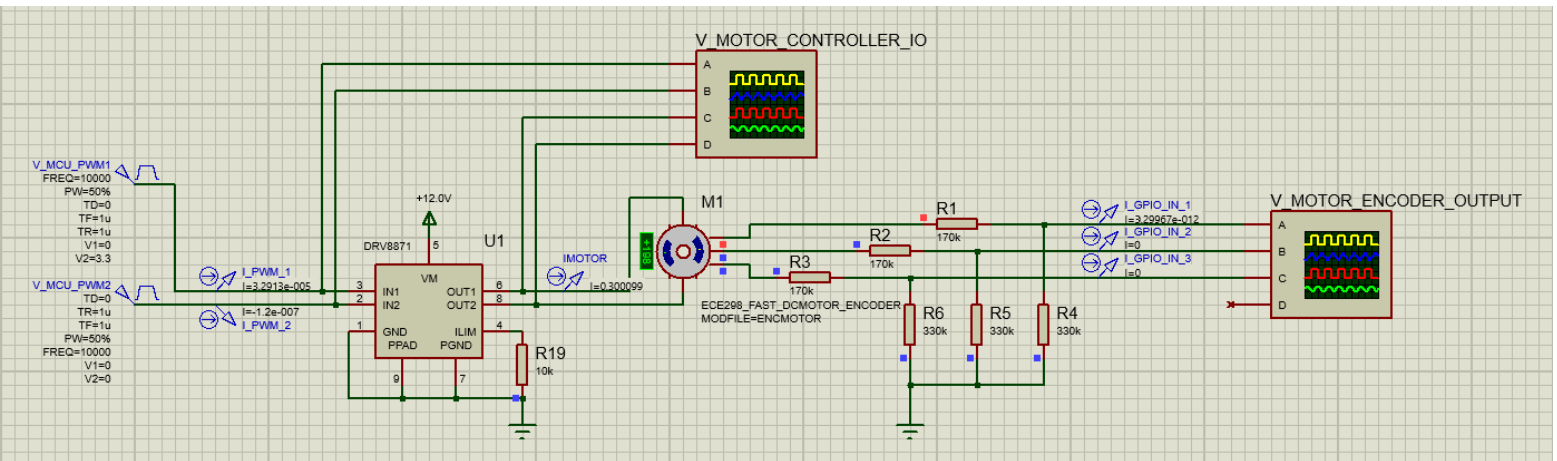
### Schematics and Simulations

Below is the schematic for interfacing with the DC motor, DC motor controller, and DC motor encoder (part of the DC motor):



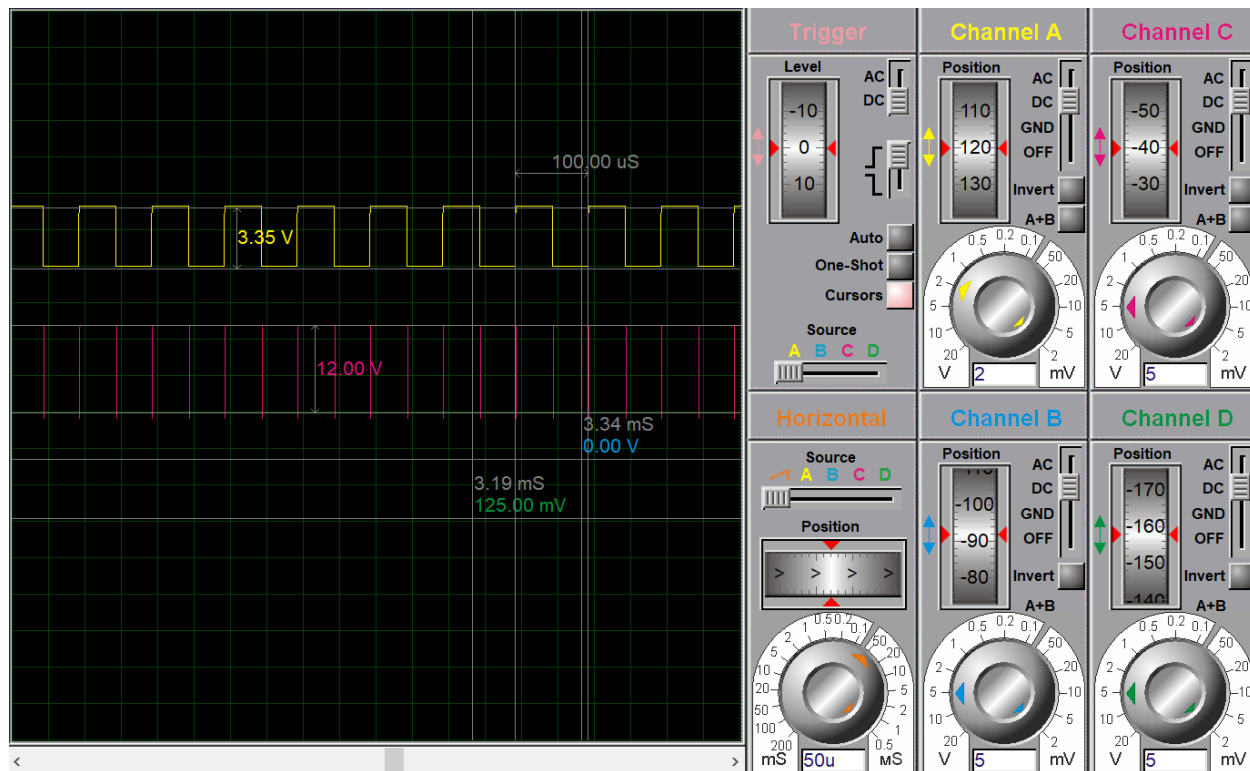
As shown, the DRV8871 GND, PPAD, PGND, and ILIM pins are all connected to ground. The two IN1, IN2 pins of the controller are connected to two PWM inputs that are meant to simulate the MCU digital output. The OUT1, OUT2 are connected in series with the DC motor. The VM pin is always connected to +12 V. The output of the motor (motor encoder) is connected to voltage translation circuits which take a 5 V signal to a 3.3 V signal that can be input to the MCU (which is simulated by an oscilloscope in this circuit).

Below shows the schematic when IN2 is connected to 0V and IN1 is sent a 50% duty cycle PWM signal at 10 kHz:



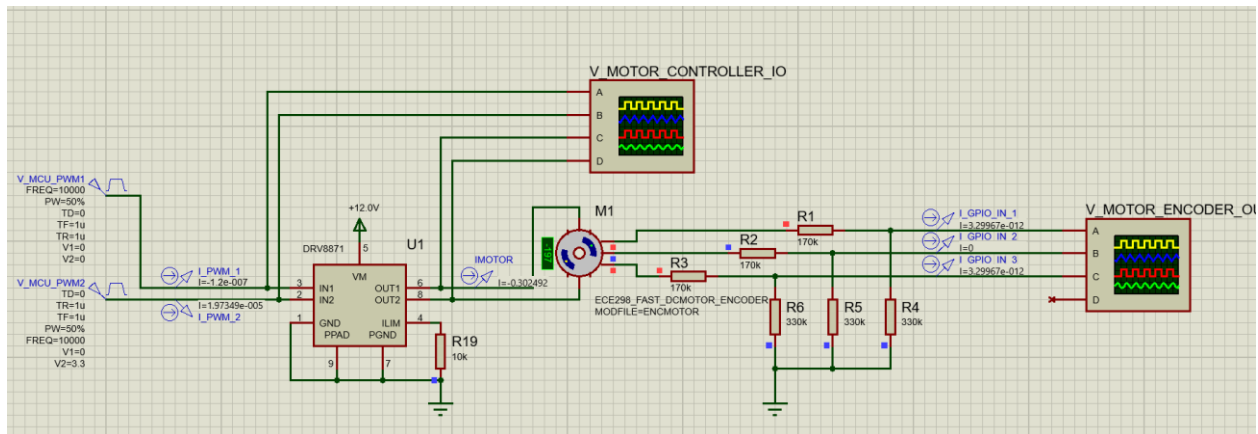
The motor rotates at an asymptotically approaching 200 RPM.

Below shows oscilloscope captures of V\_MOTOR\_CONTROLLER\_IO:

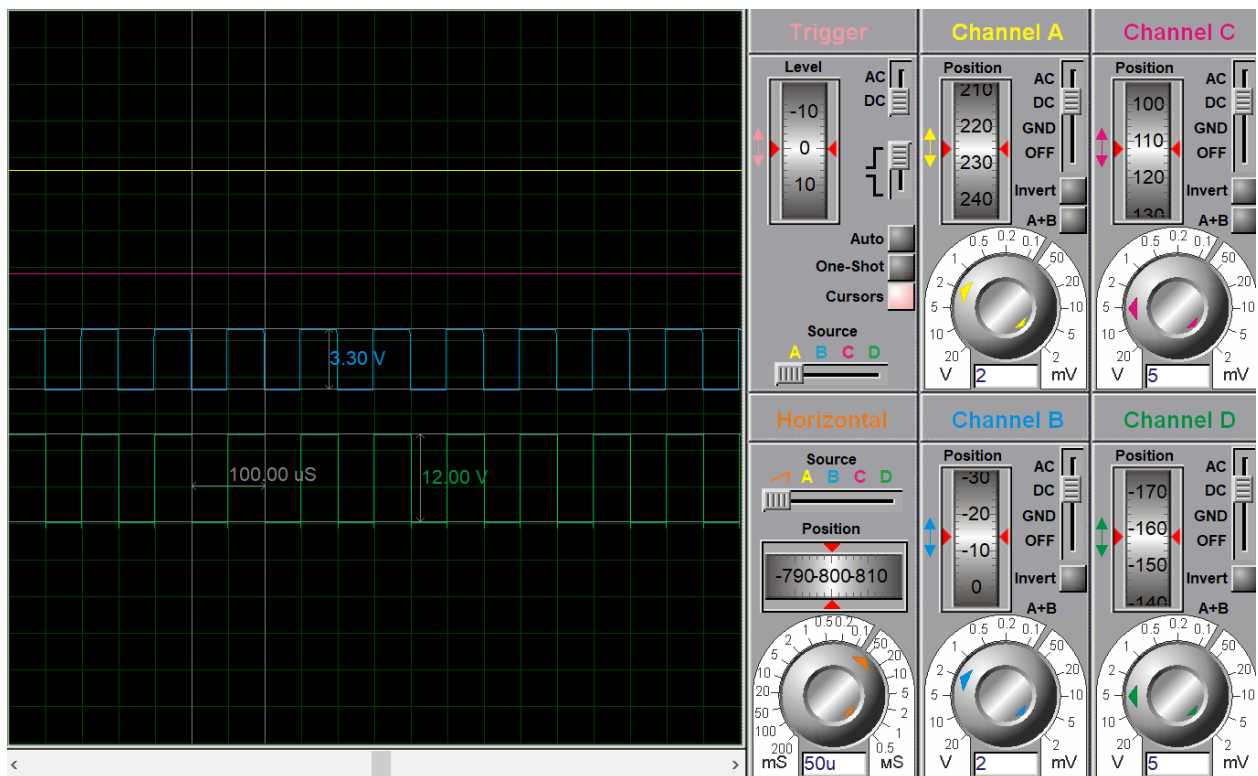


As expected, the output (OUT1) is a PWM signal at 12 V amplitude with duty corresponding to the IN1 duty cycle (a PWM signal at 3.3V).

When the IN1 is fed a signal of 0V and the IN2 signal is a 3.3V amplitude, 50% duty cycle PWM signal the motor rotates in the opposite direction (is passed the opposite current) and the rest of the circuit behaviour is the same (voltage translation and OUT2 corresponding to the IN2 at 12 V):



V\_MOTOR\_CONTROLLER\_IO:



These simulations confirm the interfacing with the MCU's GPIO pins is as required.

## DC Motor Controller

### Summary

Item	Description
Purpose	The purpose of the DC motor controller is to control the flow of current through the brushed DC motor.
Device physical domain and range	The DC motor controller can supply the DC motor anywhere between -3.5 to +3.5 A. This depends on the duty cycle of the PWM signals that it is input into the controller.

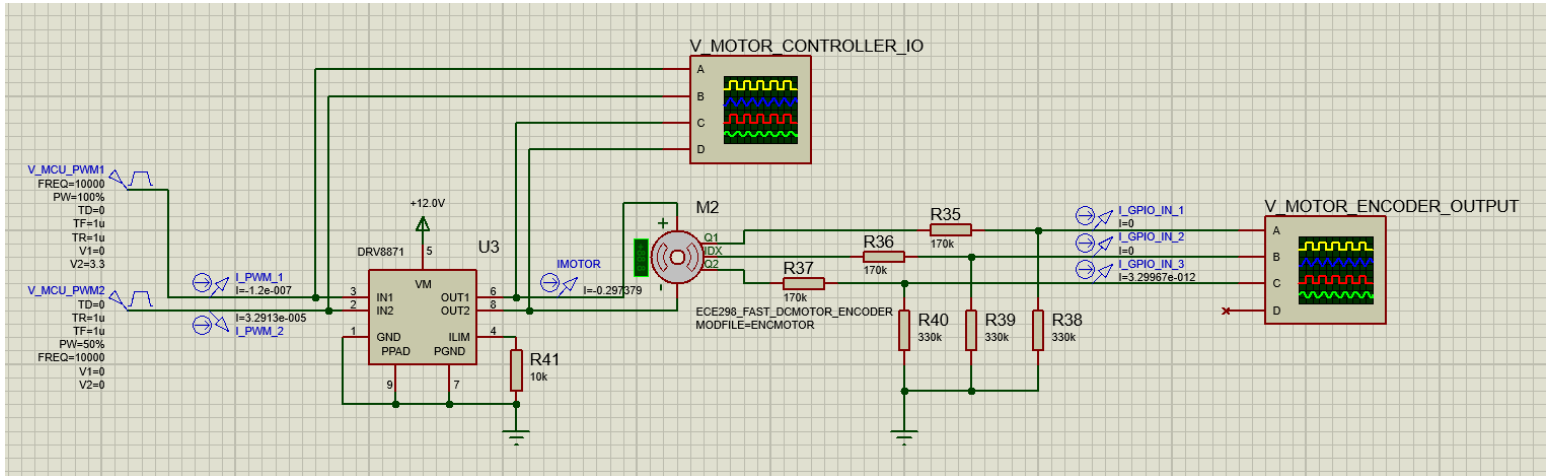
Item	Description
Device type chosen	The type of device chosen is a digital motor controller that can be controlled using PWM inputs that must be connected to a 12V source.
Proteus Library component name	The component used to implement this function is the DRV887.
Device input / output properties	<p>The device's input is two channel PWM signal that controls an H-controller circuit which powers the motor. The maximum frequency that the device can be driven with is a 200 kHz PWM signal. To ensure that the controller works, the device will be driven with a 100 kHz PWM signal from the MCU (which has a maximum PWM frequency of 42 MHz). The device's output is a 12V PWM signal from two output pins whose RMS depends on the duty cycle of the PWM supplied at the input. The current output will power the DC motor.</p> <p>Since the device is driven with a PWM signal the device is a push-pull with strong highs and lows.</p>
Device input / output range	<p>The device must be powered by a maximum 50 V supply.</p> <p>The logical inputs (driven by the PWM signal) has a <math>V_{IL}</math> of maximum 0.5 V, a <math>V_{IH}</math> of minimum 1.5 V, a max/min current of <math>\pm 1</math> A.</p> <p>The maximum output current of the controller is 3.6 A.</p>
MCU connectivity details	The two inputs IN1, IN2 will be connected to two PWM channels of the MCU (TIMx). The duty cycle of these channels will determine the current output, and whichever channel is high when the other is low will determine which way the motor is spinning.
Device/MCU interfacing details	The communication between the DC motor driver and the MCU will be unidirectional digital communication. The DC motor driver input pins will be connected to PWM inputs from the MCU, and the outputs of the DC motor will be sensed directly from the DC motor device (DC motor encoder).

### Schematics and Simulations

The other schematics and simulations for the “DC motor controller” can be found in the “DC motor” section. As part of the same system, the two parts are tested together.

## Lab 2 – Feasibility Model Phase 2

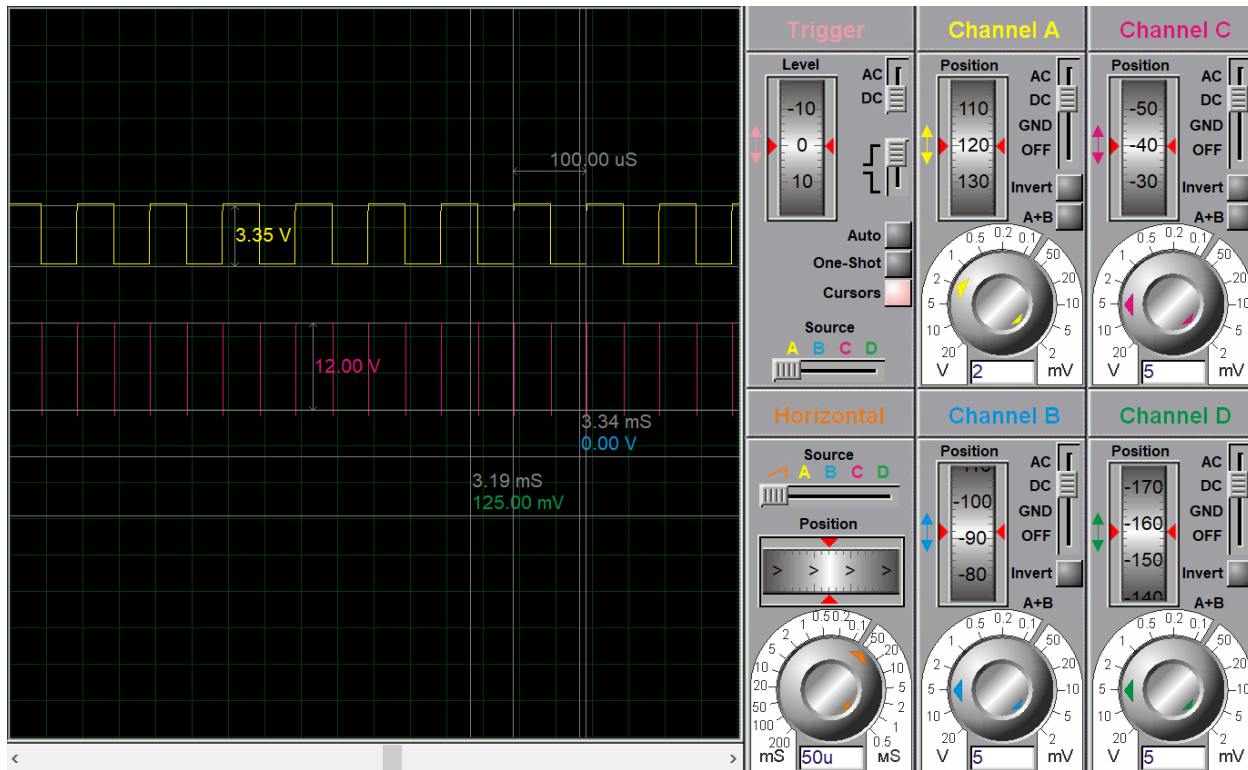
To test the amount of current required by the motor upon maximum start-up, the following circuit is used to test the inrush current. The IN1 signal is sent a 3.3V DC signal (100% PWM duty cycle) and the IN2 signal is 0V. the current flowing through the motor is measured as a function of time



The maximum current that the motor can output is 3.5 A. This confirms that the maximum current that the motor requires upon startup is smaller than the maximum current that the motor controller can supply.

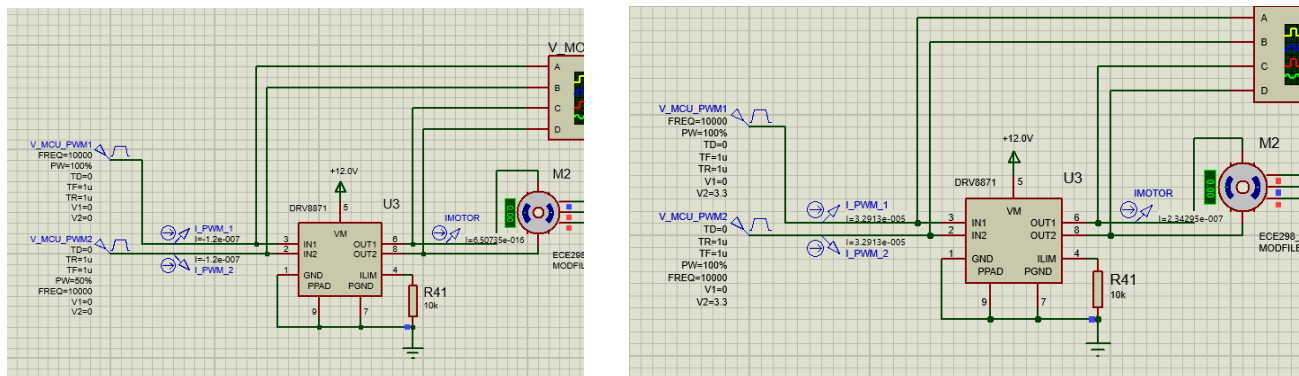
When fed an input of IN1 as a PWM signal from 0-3.3V at 50% duty cycle, the V\_MOTOR\_CONTROLLER\_IO shows that the outputs of the controller are OUT2 at 0V and OUT1 as a 12V PWM cycle at 50% duty cycle:

Below shows oscilloscope captures of V\_MOTOR\_CONTROLLER\_IO:



As expected, the output (OUT1) is a PWM signal at 12 V amplitude with duty corresponding to the IN1 duty cycle (a PWM signal at 3.3V), as per the motor controller datasheet.

Additionally, it is confirmed that when IN1 and IN2 are both low (left) and both high (right) there is no current through the motor (the outputs will be in the high-z and low respectively):



These simulations confirm that the controller can supply the correct amount of current and be driven by the correct PWM input, and that the operation according to its datasheet is correct.

## DC Motor Encoder

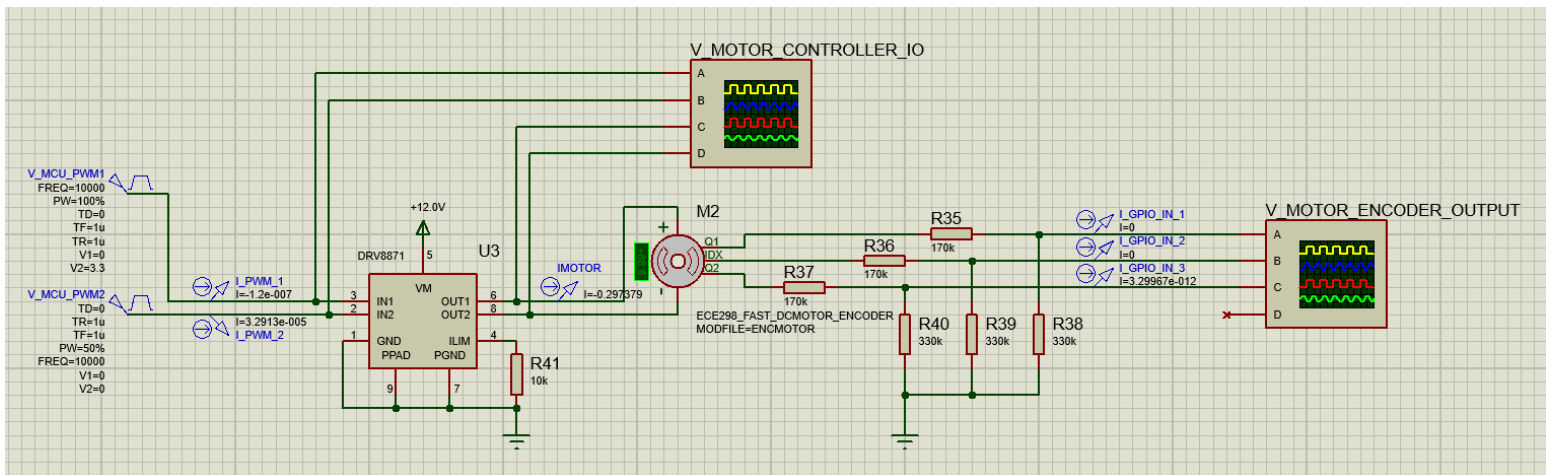
### Summary

Item	Description
Purpose	The purpose of the DC motor encoder to send information back to the controller to indicate its rotation speed. It is a sensor.

Item	Description
Device physical domain and range	The motor encoder is able to sense the maximum rotation speed of the DC motor, which is 800 RPM.
Device type chosen	The motor chosen for simplicity of design is a digital motor encoder that is part of a DC brushed motor.
Proteus Library component name	The component used to implement this function is the ECE298_FAST_DCMOTOR_ENCODER.
Device input / output properties	The output of the motor encoder are three pins, Q1, Q2, and IDX. Each of these pins output a digital signal from 0-5V, which must be sent through a voltage divider before reaching the MCU input pins in a range of 0-3.3V. The IDX pin is pulsed once per rotation, indicating the speed of the motor. The Q1, Q2 pins are pulsed once every 24 <sup>th</sup> of a rotation, indicating the absolute angle of the motor. Depending on which of Q1 or Q2 rises high when the other is low, this indicates if the motor is moving forwards or backwards. The outputs of the DC motor (encoder) is a push-pull output.
Device input / output range	The devices outputs (Q1, Q2, IDX) are square wave pulses whose range is 0-5V.
MCU connectivity details	<p>The output of the motor encoder will be connected to the MCU GPIO input pins via a voltage divider to step down the maximum voltage from 5V to 3.3V for the MCU to read.</p> <p>The electrical properties of the MCU GPIO pins are:  <math>V_{IL}(\text{max}) = 0.6 \text{ V}</math>  <math>V_{IH}(\text{min}) = 2.0 \text{ V}</math></p>
Device/MCU interfacing details	The output of the motor encoder is three digital signals (0-5V PWM). Thus, the MCU interfacing between the motor encoder and MCU is digital to digital interfacing.

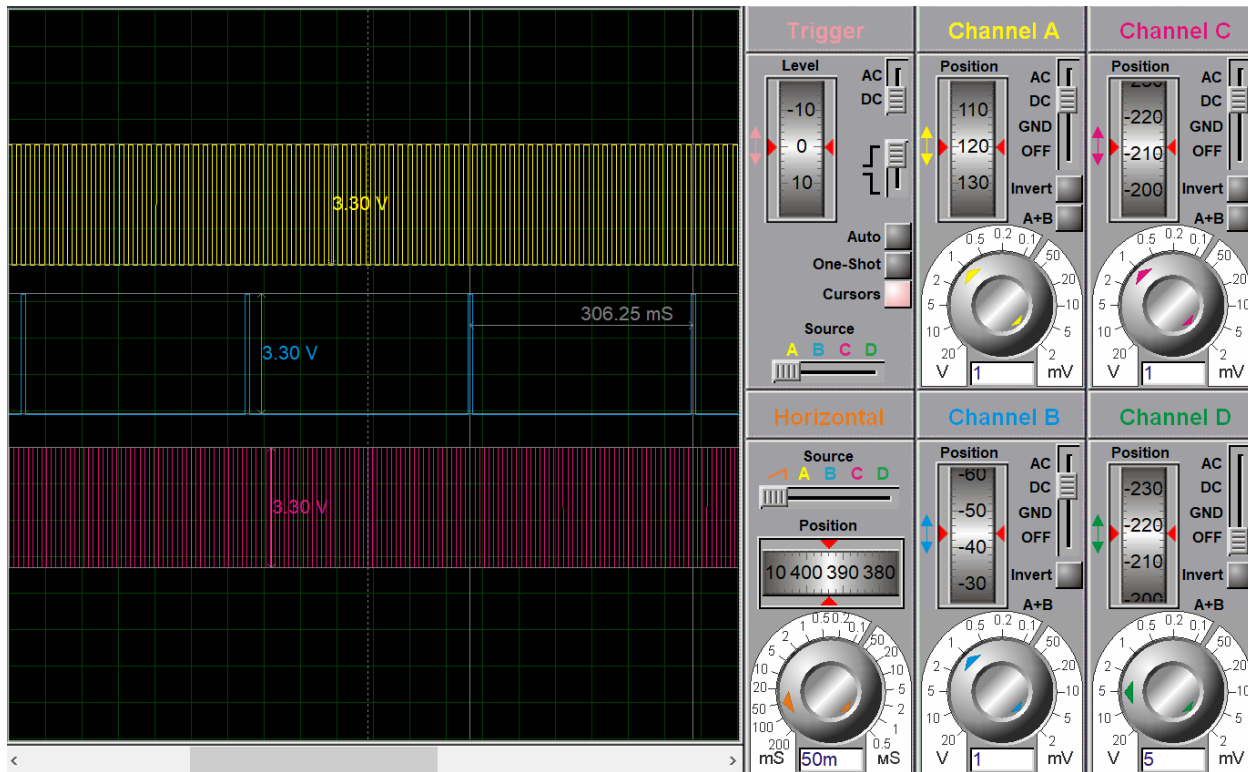
### Schematics and Simulations

The following tests are in addition to the tests performed in the DC motor section. The same circuit is used to test the DC motor encoder as the DC motor and controller:



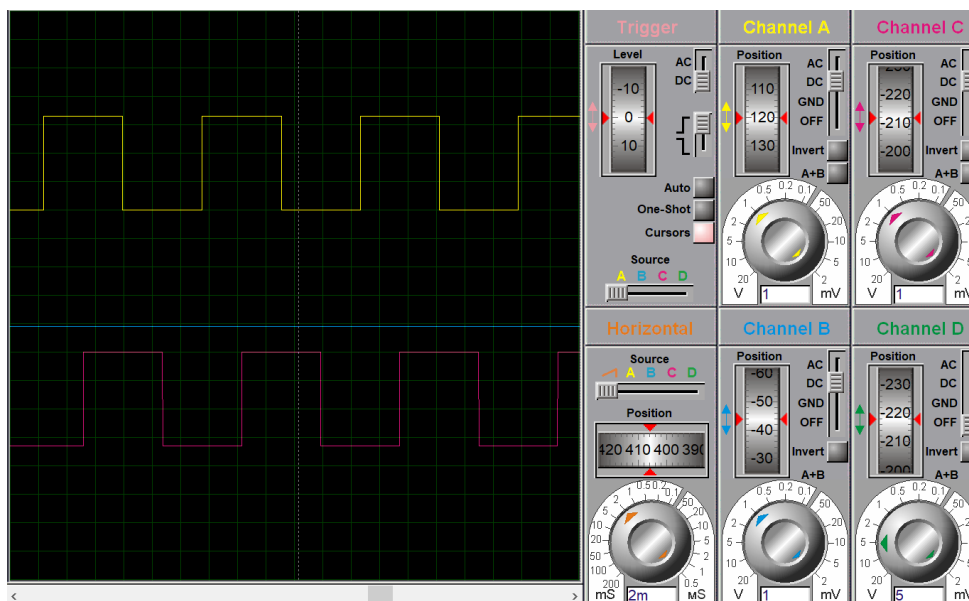


Below shows the screen capture of the V\_MOTOR\_ENCODER\_OUTPUT when provided with the PWM input in the image above:

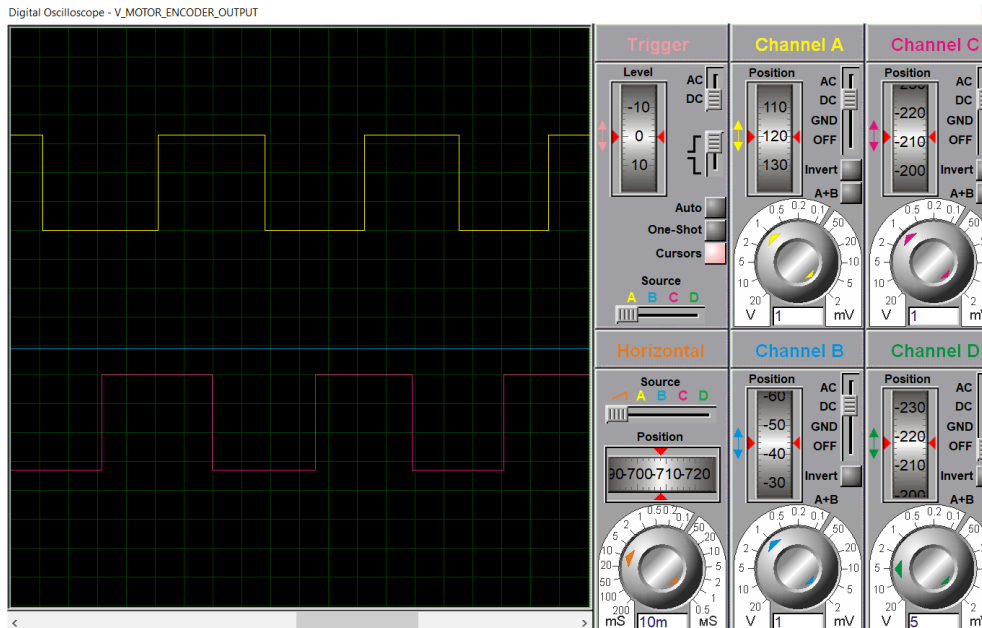


As expected, the original 5 V output of the motor encoder has now been translated down to a 3.3V signal for the MCU to be able to process. Additionally, the input to the MCU (also seen in the DC motor section) is negligible (on the order of 10 uA).

It is also seen correctly that the rising edge of Q1 occurs when Q2 is low when spinning with a positive valued RPM:



When fed an opposite input of  $IN1 = 0\text{ V}$ ,  $IN2 = \text{PWM at } 0\text{--}3.3\text{ V}$  (100% duty cycle, just a 3.3V DC signal), the reverse ordering of Q1 rising edge when Q2 is high is observed 24 times per cycle:



This confirms the DC motor encoder's operation and interfacing with the MCU GPIO pins will be within the requirements.

## Liquid Crystal Display

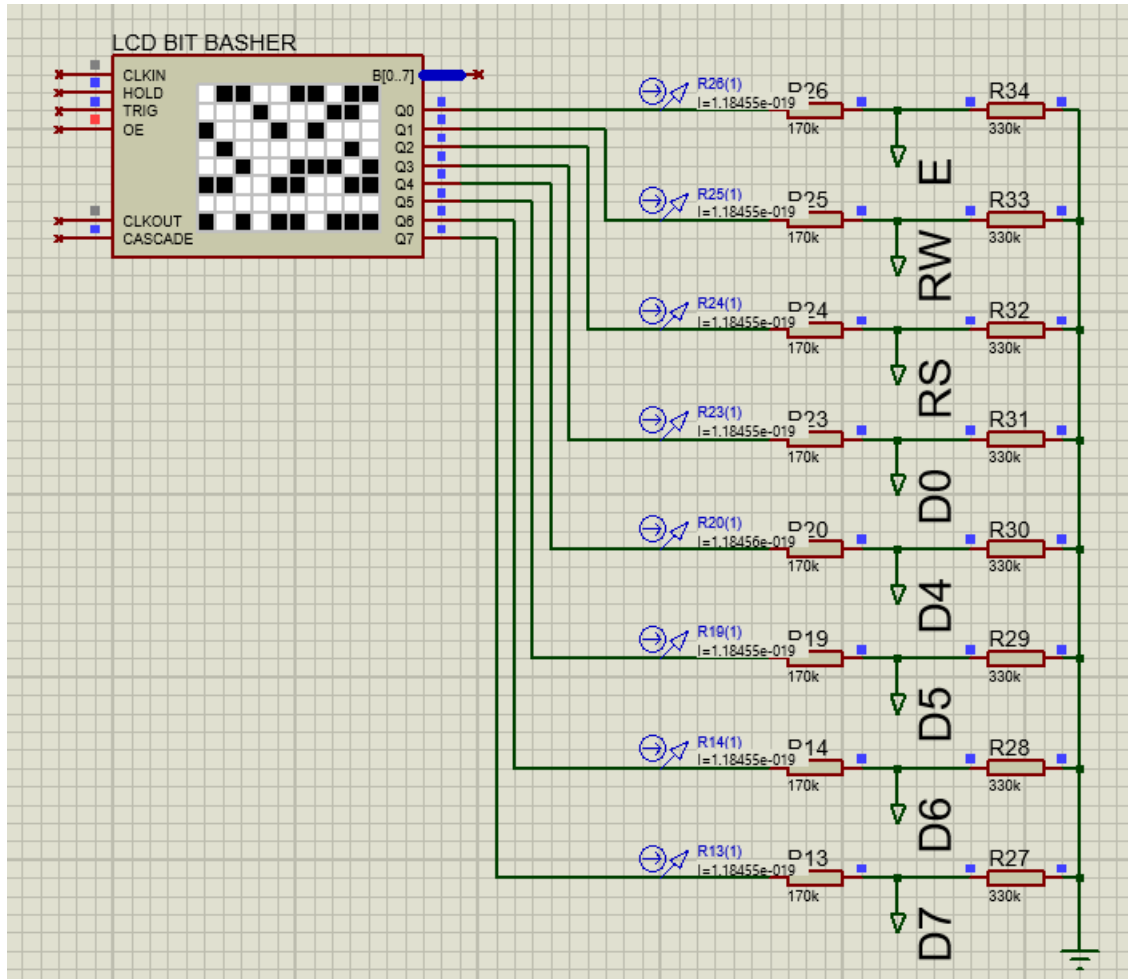
### Summary

Item	Description
Purpose	The purpose of the Liquid Crystal Display (LCD) is to display the mode that the wheelchair controller is in and the speed in RPM of the wheels to the user.
Device physical domain and range	The range of the device, in terms of number characters displayed, is at least a 2x16 grid of characters (i.e., 2 rows of 16 characters). This is enough to display all of the information required.
Device type chosen	The device chosen is an alphanumeric LCD display that can display 32 digits in 2 rows (a 2x16 LCD display)
Proteus Library component name	The proteus library component name is the LM016L.
Device input / output properties	The device has 11 input channels, power, and ground. Only 8 of the 11 input channels will be used to communicate with the device in 4-bit mode. 4 of the 8 channels are input data lines, 1 of the channels is a data line that will be used for reading the busy flag, and the other 3 channels are Enable, Read/Write, and RS. The device's only output is along channel D0, which will indicate whether or not the LCD is busy with the Busy Flag (BF).

Item	Description
	<p>As shown in the simulations below, each digital output of the MCU will be connected to an NMOS transistor and inverter to drive the LCD.</p> <p>The electrical characteristics of the LCD logic is:  Maximum power supply (<math>V_{DD} - V_{SS}</math>) = 7.0 V  <math>V_{IH}</math> = 2.2 V min.  <math>V_{IL}</math> = 0.6 V max.  <math>V_{OH}</math> = 2.4 V min.  <math>V_{OL}</math> = 0.4 V max.S</p> <p>The electrical characteristics of the inverter when supplied a 5V VDD (4049) logic is:  <math>V_{OL}</math> = 0.05 V  <math>V_{OH}</math> = 4.95 V  <math>V_{IL}</math> = 1.0 V  <math>V_{IH}</math> = 4.0 V  <math>I_{OL}</math> = 4.0 mA  <math>I_{OH}</math> = 4.6 mA  <math>I_{IN}</math> = -1.0 mA</p> <p>The electrical characteristics of the NMOS transistor (2SK176) is:  Drain-source voltage rating = 200 V  Gate-source voltage rating = +- 20 V</p> <p>The maximum frequency the LCD can be written to is 270 kHz, so the device will be fed signals at 10 kHz.</p>
Device input / output range	The device must be driven with 0-5V logic (0 = '0', 5V = '1'). The output range is also from 0-5 V.
MCU connectivity details	<p>Since the device requires a digital input, the 8 digital input lines will be connected to 8 GPIO pins of the MCU – 7 of which will be set to output and 1 of which will be set to input. Since the device runs on 5 V logic, the output line will be connected to an NMOS transistor that will behave like a switch, whose drain input is pulled up to VCC. The result of this stage will be a 0 or 5V signal that will be sent through an inverter, which will drive the LCD display.</p> <p>The CMOS transistors that behave like a switch are open drain that are pulled up to VDD when not active. This drives the inverter which is a push-pull (strong high and low) device.</p>
Device/MCU interfacing details	The communication between the LCD and the MCU will be digital to digital (D/D) communication. The LCD requires digital inputs from 0-5V and outputs the busy flag between the same 0-3.3V. The MCU can read and output digital inputs between 0-3.3V, so voltage translation is required.

## Schematics and Simulations

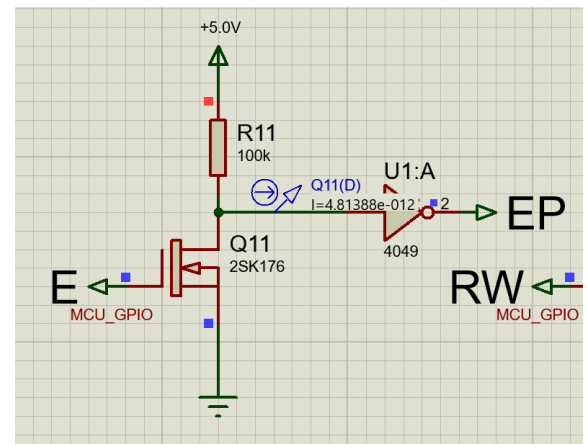
To simulate the LCD interface a digital pattern generator is used to simulate the digital signal that the MCU would output:



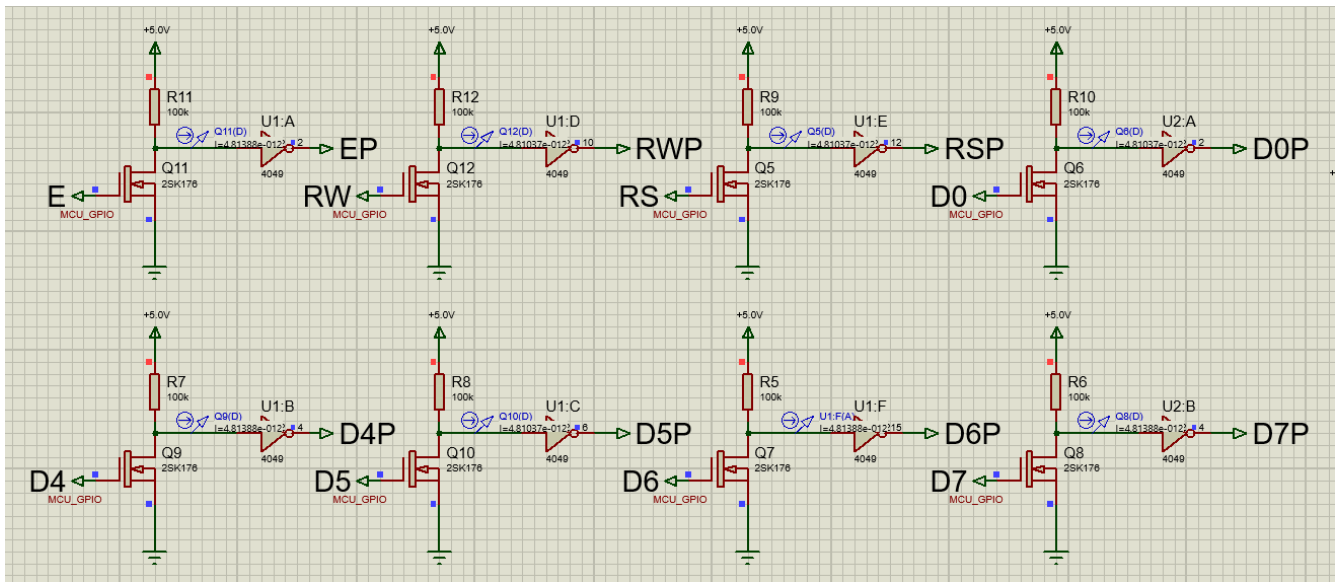
In Proteus, the output of the pattern generator is 0 or 5 V, so translation circuits that step the voltage down to 0-3.3V are implemented above. The outputs of the MCU are taken to be E, RW, ..., etc.

Since the LCD must be driven with 0-5V logic, voltage translation circuits using a transistor and inverter has been implemented for each LCD driver signal:

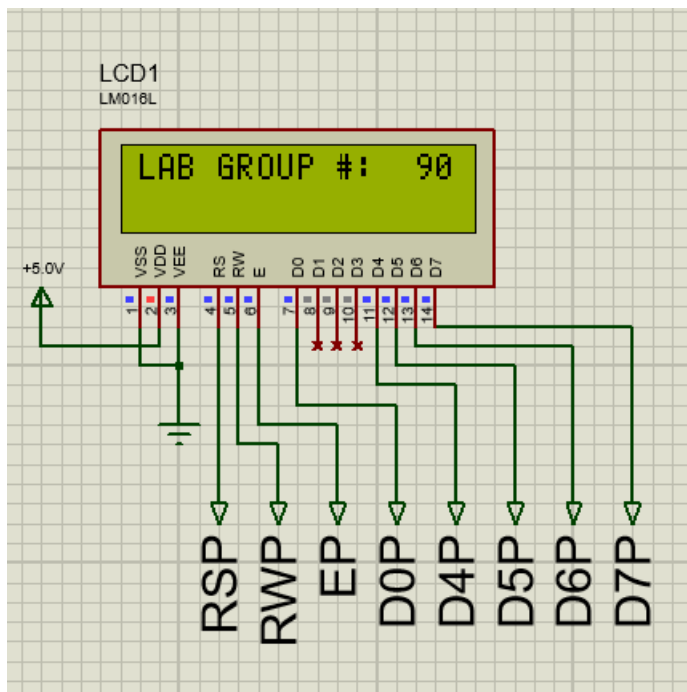
If the signal E is high, the transistor switch ON and is conducting, connecting the inverter (which must also be driven with 5 V logic) to ground, and the output of the inverter is high. If E is low, the inverter is pull up to the 5V rail and the output of the inverter is 0V. Hence, is E is high EP (which stand for E-processed) is high, and the same is true if E is low.



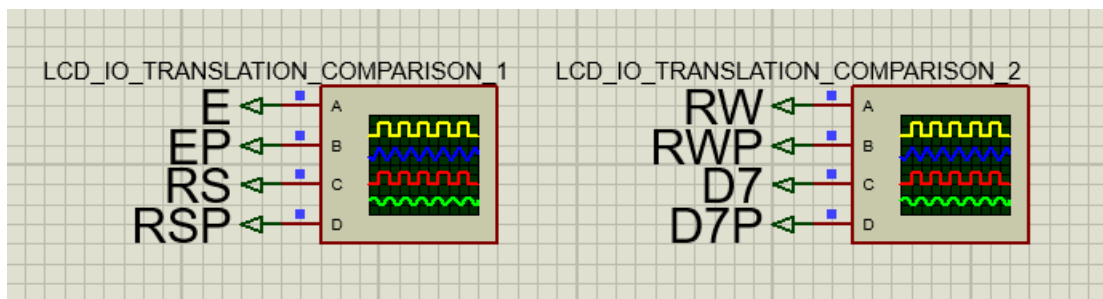
Below shows the same configuration for each of the LCD outputs.

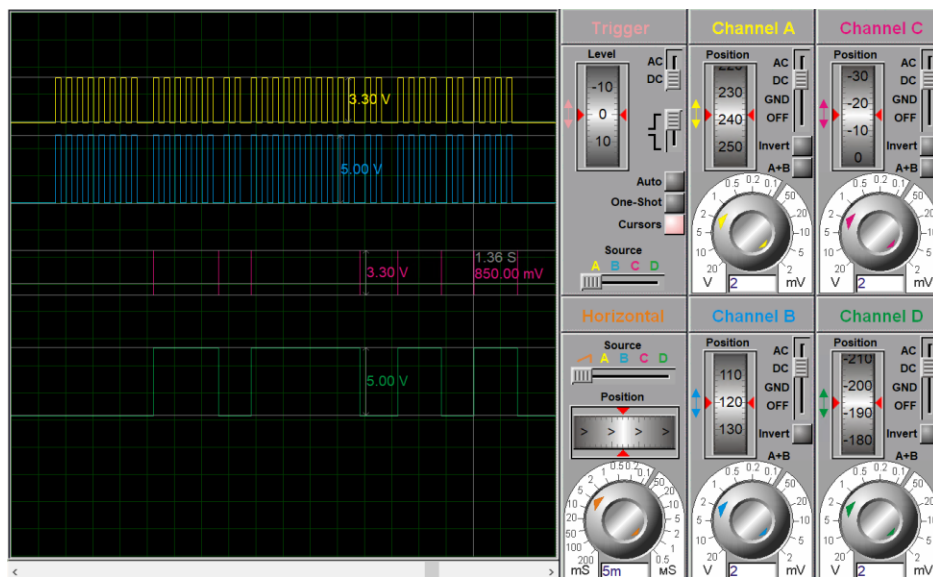


Each of these processed signals then drive the LCD in the correct manner corresponding to what was implemented in lab 1:



Below is an oscilloscope capture of the digital signals shown which display that the correct 3.3V logic to 5V logic conversion occurs in the circuits:





This confirms that the voltage translation circuits that process the outputs from the MCU GPIO pins are translated to the correct 0-5V logic that is able to drive the LCD.

## Coloured LEDs

### Summary

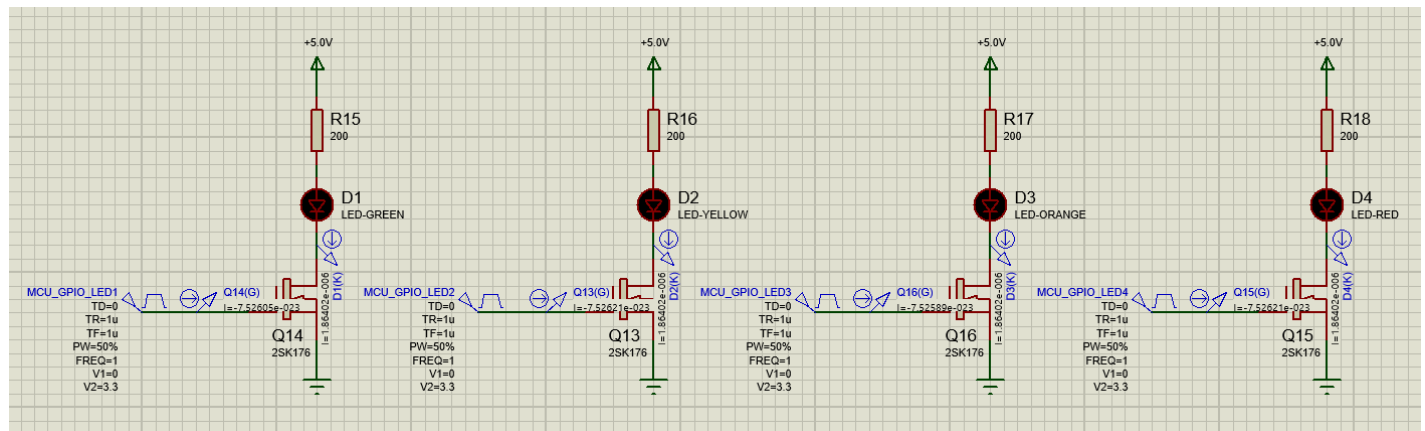
Item	Description
Purpose	The purpose of the light emitting diodes (LEDs) is to convey information to the user regarding the controller mode and the battery levels.
Device physical domain and range	The current flowing through the LEDs during testing was 13.68 mA, which is satisfactory for us as the Proteus models, the forward current of the diodes is 10 mA. We also validate that the gate voltage of 0-3.3V is enough to bring the current flowing through the LED.
Device type chosen	The devices chosen for prototyping are several multicoloured LEDs. Specifically, the colours of green, yellow, red, and orange are used.
Proteus Library component name	The proteus library component names are: LED-GREEN LED-YELLOW LED-ORANGE LED-RED
Device input / output properties	Each LED will be powered by a 5 V source. Each LED will be connected in series with a resistor and transistor. The transistor will be controlled by a digital GPIO output pin of the MCU which will control the current flow through the LED. There is not output besides emitting light required from these devices.
Device input / output range	The device must be provided with at least a 10 mA current to successfully emit its maximum light. The diode voltage at 10 mA is 2.2 V. The minimum supplied voltage to the gate of the NMOS

Item	Description
MCU connectivity details	transistor must be 1.45 V to allow current to flow through the transistor and LED.  The transistor that is connected in series with the resistor will be connected to a GPIO pin of the MCU that is set to digital output. The transistor is in the open-drain configuration as it is connected to VDD via a resistor and is controlled by its gate voltage. This will control the current flow through the resistor by switching between 0-3.3V. There is no set frequency of switching between ON and OFF for the LED.
Device/MCU interfacing details	The GPIO pin will control the gate voltage of a transistor that is connected in series to the LED. This transistor will act as a switch to turn on the LED, which will be powered by the battery in series with a resistor to control the current sent through the LED. Since the device's ultimate output is an analog output (the intensity of light shining from it) the communication is digital to analog communication.

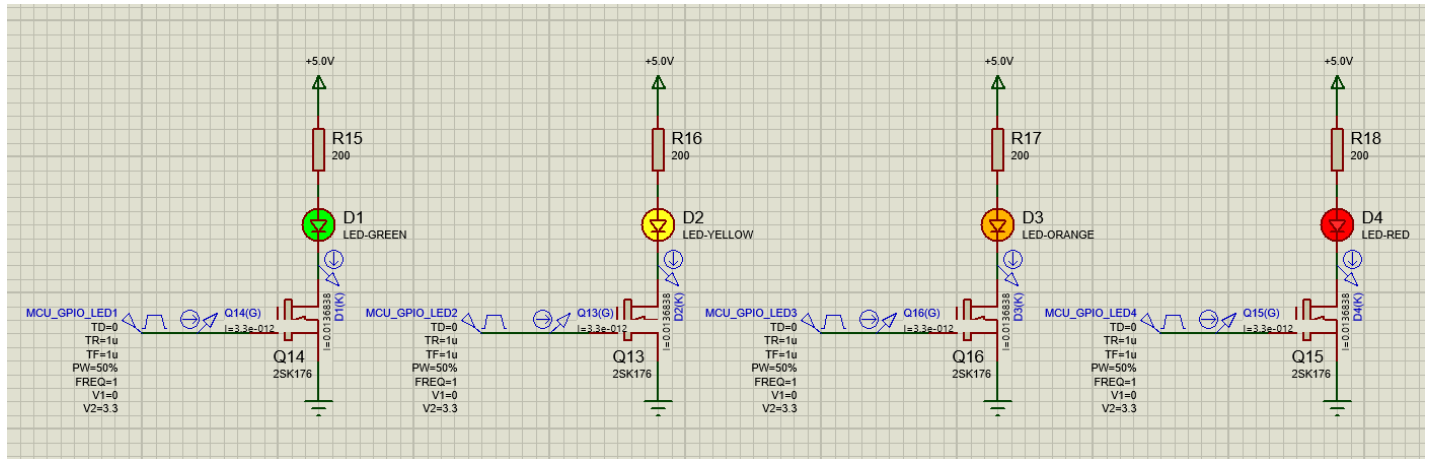
### Schematics and Simulations

To limit the drainage of current from the MCU, digital output pins from the MCU will drive transistors that behave like switches to turn ON and OFF an LED.

The circuit below shows a green, yellow, orange, and red LED connected via a 200  $\Omega$  resistor to a 5 V power source, where a transistor is used as a switch controlling whether current flows through it (i.e., it is observed to shine).



Each LED is connected in series to a 5V rail which supplies it current when the transistor is turned on. For visualization, each transistor is blinked using a PWM signal from 0-3.3V with a 50% duty cycle. As shown, when ON, there is virtually no current flowing through the LED and from the MCU digital output pin.



The picture above displays all of the LEDs in the ON position, as the 'GPIO pin' supplies the gate voltage of each transistor with 3.3 V. It is seen that current now flows through each of the LEDs, and they are shining their respective colours. The current through each LED is the same 13.68 mA. This is satisfactory since, for the Proteus models, the forward current of the diodes is 10 mA.

These simulations confirm that the GPIO pin of the MCU is capable of driving the LEDs in this configuration.

## Battery Sensor Circuit

### Summary

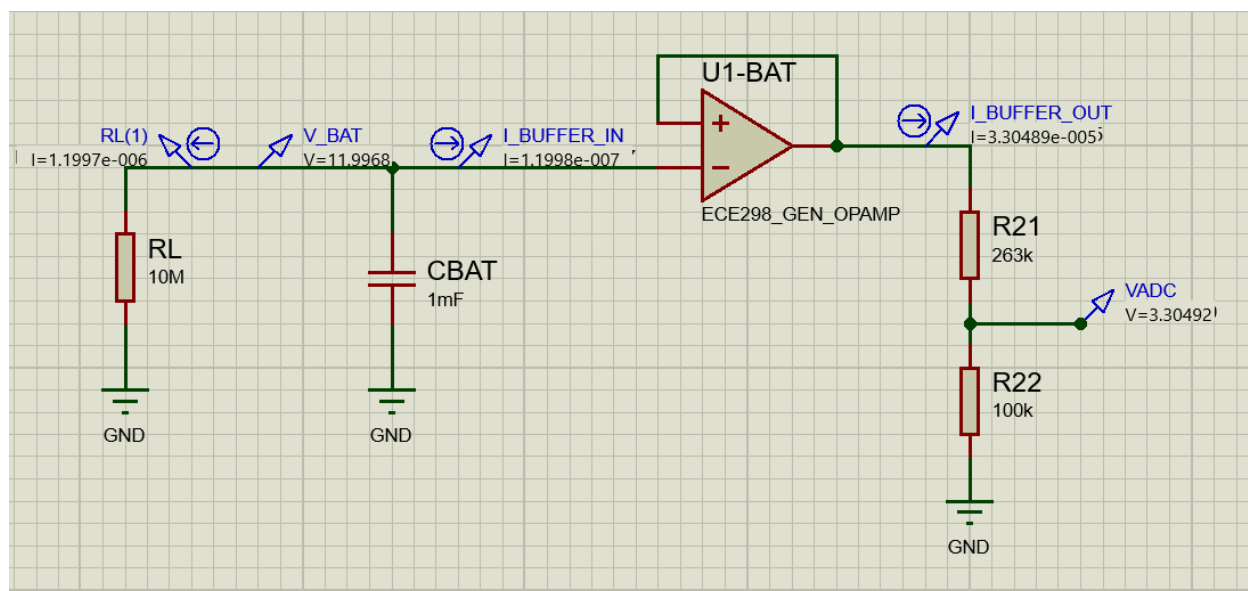
Item	Description
Purpose	The purpose of the battery sensor circuit is to periodically measure the voltage level of the battery to indicate to the user how much charge is left in the battery.
Device physical domain and range	The battery sensor circuit is able to take in a voltage on the level of 0-12V and transform it to a level of 0-3.3V – a voltage that can be read by the ADC of the MCU.
Device type chosen	The battery sensor is not a singular device, rather it is made up of a voltage divider and an op-amp buffer.
Proteus Library component name	The op-amp chosen for the battery sensor is the ECE298-GEN-OPAMP
Device input / output properties	The battery sensor's input is a wire connected to the battery that indicates the voltage of the battery. The output of the sensor is the voltage whose input is stepped down to a range of 0-3.3V.
Device input / output range	To minimize the amount of current/power drawn from the battery and injected into the MCU, and op-amp buffer is used to isolate the voltage in the sensor. The input range – to work accurately with the MCU chosen – is 0-12 V and the output range is 0-3.3 V. The output current is on the order of 50 uA and the input is even lower due to the very high input resistance of the op-amp. The maximum input current to the ADC is 500 uA, so this is well below the threshold.
MCU connectivity details	The output of the device will be connected to the MCU's ADC. The injection current of the ADC is a maximum of 20 mA, which is more than satisfied by the low output current of the device.



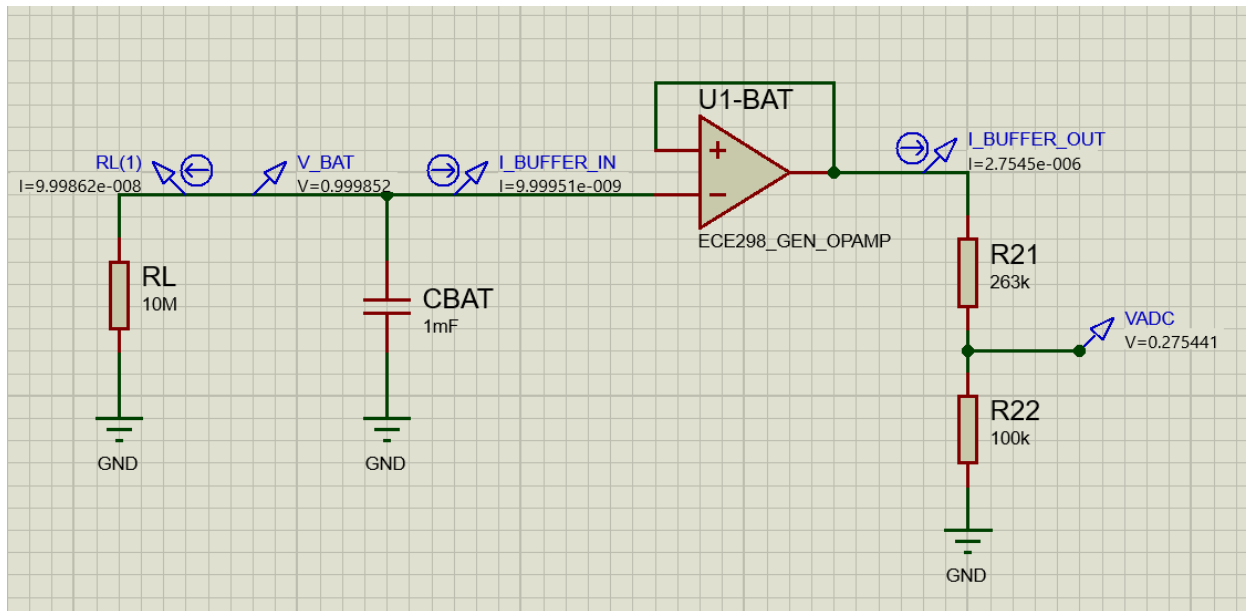
Item	Description
	The MC's ADC can take in a maximum voltage of 0-3.3V, which is satisfied using a voltage divider.
Device/MCU interfacing details	The communication with the MCU will be analog to digital (A/D). The output of the battery sensor is an analog voltage from 0-3.3V which will be sent the MCU's ADC to be converted to a digital signal which is stored in memory.

### Schematics and Simulations

The output of the battery sensor will be sent to the ADC. As was done in lab 1, the battery is simulated using a large capacitor with an initial charge of 12 V. The output of the circuit below shows that only 0.1  $\mu$ A is drained from the battery to the sensor circuit, which itself only drains 33  $\mu$ A. The output of the circuit when the battery is fully charged is confirmed to be 3.3V, which is the maximum rated voltage that the ADC of the MCU can handle:



If the battery began with a 1 V charge, the same behaviour (with the appropriate output voltage) is observed:



These simulations confirm that for a maximum of a 12 V battery, the battery sensor circuit will output a 0-3.3V signal to the MCU's ADC IN pin with current on the order of 10 uA.

## Variable Resistor (Potentiometer)

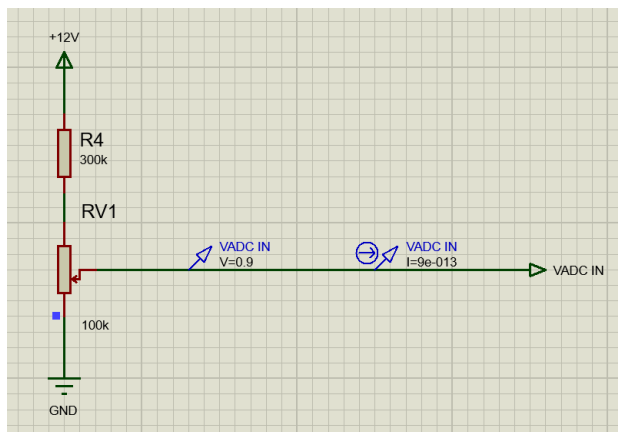
### Summary

Item	Description
Purpose	The purpose of the variable resistor will be to act as a user input to the control system that will determine the speed and direction that the left and right motors will turn
Device physical domain and range	The range of the potentiometers should be from 1 kΩ to 100 kΩ.
Device type chosen	The device chosen to achieve the requirements is an analog sliding potentiometer (i.e. a variable resistor).
Proteus Library component name	The Proteus library component name is POT.
Device input / output properties	The device's input is a voltage provided by a power source. By adjusting the resistor's resistance, one can adjust the voltage that is measured across it. The output of this voltage divider is an analog voltage that can be sensed by an Analog to Digital Converter (ADC)
Device input / output range	This second resistor will be 300 kΩ to ensure that the minimum sensed voltage, when provided with an input of 12 V, is $12\text{ V} * \frac{10^3}{10^3 + 3 \times 10^5} = 0.03987\text{ V}$ and the maximum voltage is $12\text{ V} * \frac{10^5}{10^5 + 3 \times 10^5} = 3\text{ V}$ . This will correspond respectfully to maximum/minimum speed and maximum left/maximum right turning for speed and steering. The power consumed by one of these circuits is a maximum of 480 uW.
MCU connectivity details	The voltage divider circuit will be connected to a power source of 12V. This will power the circuit whose output will be connected to an input of the ADC of the MCU. The ADC will

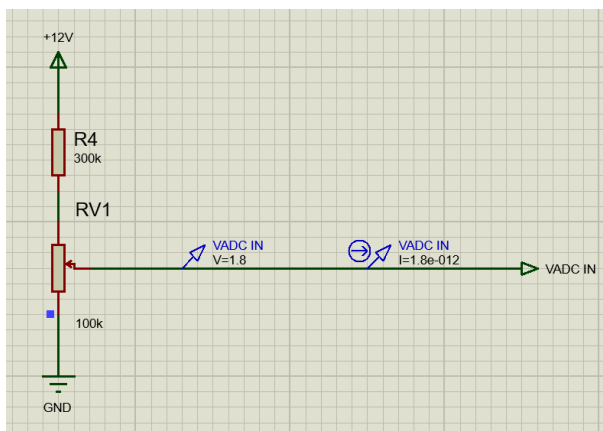
Item	Description
	measure the voltage across the static resistor to determine what resistance the user has applied. The input current will be on the order of 1 pA, below the maximum input current of the ADC (500 uA).
Device/MCU interfacing details	The communication with the MCU will be analog to digital (A/D). The output of the potentiometer voltage divider circuit is an analog voltage from 0.03987-3V which will be sent the MCU's ADC to be converted to a digital signal which is stored in the MCU's memory.

### Schematics and Simulations

The variable resistor will be connected in series to a 300 k $\Omega$  resistor and its adjacent pin will be connected to the ADC. In the circuit above the variable resistor is set to is 3<sup>rd</sup> position:

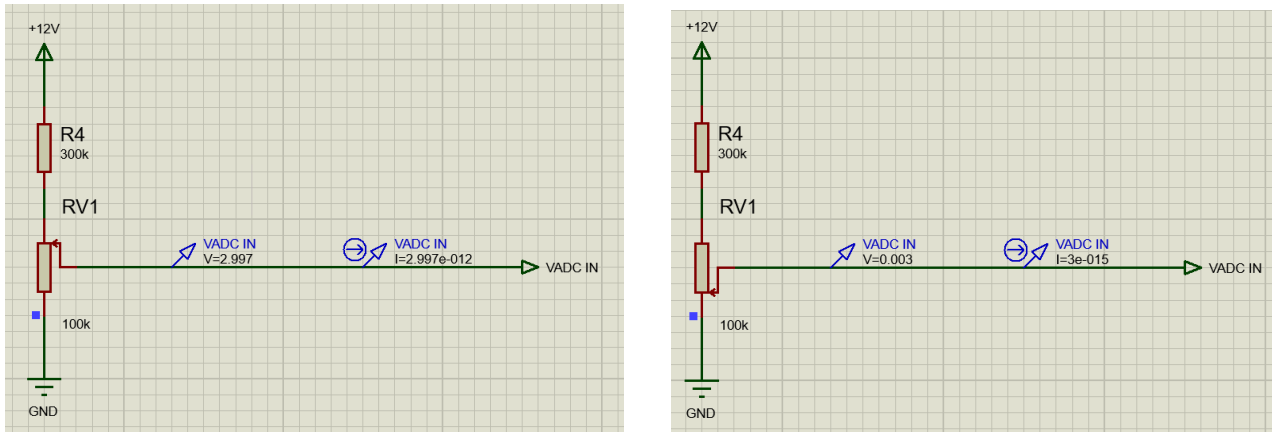


It is seen that the voltage sensed by the ADC would be 0.9 V, exactly 3 tenths the maximum voltage sensed for the 100 k $\Omega$  variable resistor.



In the circuit above the variable resistor is in the 6<sup>th</sup> position, and it is seen that the ADC would sense the correct voltage of 1.8 V.

Additionally, in the 10<sup>th</sup> (left) and 0<sup>th</sup> (right) positions respectfully:



The voltage sensed is 3 V (10<sup>th</sup> position) and close to 0 V (0<sup>th</sup> position). These simulations confirm that this variable resistor is linear, and that the current injected into the ADC using this circuit is negligible and will not contribute significantly to the input current that is maximum 120 mA for the STM32F302 MCU.

## ECE298\_GEN\_SWITCH

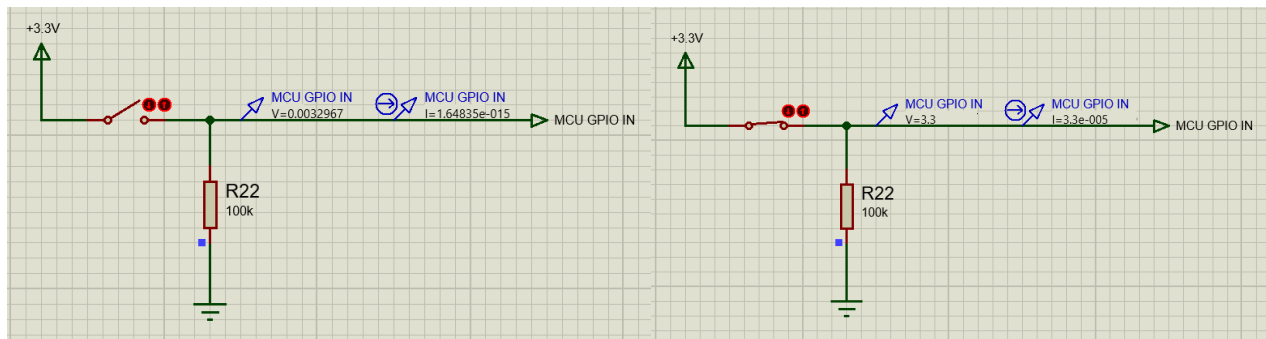
### Summary

Item	Description
Purpose	The purpose of the switches is to provide the user with an input to change the mode of the controller system. The first switch will be used as an ON/OFF switch for the controller system and the other switch will be used to switch between the Locked mode and the Run mode.
Device physical domain and range	The switch will be exposed to a range 3.3V signal which, when past through the switch will be a '1' signal while closed (at 3.3V) and will be a '0' signal when open (where the output is at 0V).
Device type chosen	The device chosen is a simple switch that toggles between the open and closed position.
Proteus Library component name	The Proteus library component name is ECE298_GEN_SWITCH.
Device input / output properties	The device does not have an input from the MCU GPIO pin, rather it is connected to a 3.3V source. The output of the device is a digital '0' or '1', depending on the state of the switch.
Device input / output range	The device's input range is 3.3 V and its output range is either 0 or 3.3V – which is a digital 0 or 1 respectively.
MCU connectivity details	The switch's output will be connected to an input pin (one of the GPIOs) of the MCU. The input will be pulled low via a pull-down resistor when the switch is in the open position. When in the closed position, the 3.3V source will be connected to the MCU pin as a digital 1. Since the MCU GPIO input pin has an input impedance of close to 10 MΩ, the input current into the MCU will be smaller than 1 uA, enough for the MCU GPIO pin to handle.

Item	Description
Device/MCU interfacing details	The communication between the switch and the MCU will be digital to digital (D/D) communication. The switch will be powered by a 3.3V input (a constant '1') and outputs 0-3.3V depending on if switched to the closed position. The MCU can read and output digital inputs between 0-3.3V, so no voltage translation is required.

### Schematics and Simulations

The interfacing circuit for the switch is shown below:



The switch is connected on one end to a 3.3V source, and the output of the switch is connected to where the MCU GPIO input pin would be placed. When in the closed position the MCU input is shorted to the 3.3V source. In the closed position the MCU input is pulled down to ground. The current drawn is minimal (33  $\mu$ A) and the voltage levels appropriate for the MCU is confirmed with the probes shown in animation mode in Proteus.