Lab 2 – Feasibility Model Phase 2

ECE 298 – S2021

Lab Section:	1215-ECE.298.041.8.LAB	Group:	156

DC Motor

Summary

Item	Description
Purpose	The motor will open and close the door.
Device physical domain and range	As described in the device properties window, the motor has a zero load rotations per minute (RPM) of 360 and it runs at 24 pulses per revolution. No maximum rotations per minute is given in the device data sheet, but on testing using a 12 Volt source with a motor driver integrated circuit (IC) as shown below, the motor turns at a maximum of 176 RPM. Note that this is slightly lower than the previously tested maximum (from Lab 1) of 180 RPM, when it was directly attached to a 12 V source instead of a driver.
Device type chosen	Brushed DC Motor
Proteus Library component name	ECE298_DCMOTOR_ENCODER
Device input / output properties	The motor has two terminal inputs, in which an analog voltage is applied across. The sign (positive or negative) of the potential difference across the motor determines its direction. Note that the motor outputs (i.e. pins Q1, Q2, and IDX) are described in the 'Motor Encoder' table. Additionally, a motor driver IC (DRV8871) will be used, which accepts two digital inputs (IN1 and IN2), a 12 V source at the VM pin, and outputs an analog signal (OUT1 and OUT2) to the motor.
Device input / output range	The motor accepts voltages from 0V to 12V. Note that if accepting a pulse-width modulated (PWM) signal, the motor driver accepts a maximum logic input PWM frequency of 200 kHz (Table 6.3 of the device datasheet), while the microcontroller unit (MCU) can output a maximum PWM frequency of 42 MHz (Table 58 of the MCU datasheet). Therefore, the PWM frequency in the sub-system has a maximum of 200 kHz.
MCU connectivity details	To control the motor speed, a digital PWM output from the MCU will be required.
Device/MCU interfacing details	The MCU digital PWM output signals will be connected to the input pins (IN1 and IN2) of the DRV8871 motor driver IC. This driver will be connected to a 12 V source at the VM pin that will power the motor, thus acting as a digital to analog converter.

Schematics and Simulations

Several aspects of the DC motor will be tested when driven by the interface as described above.

Figure 1 shows the schematic to test for the motor going in both directions using the motor driver interface. To simulate the outputs of the MCU digital pins, square waves with a low of 0V and high of 3.3V are outputted into IN1 and IN2. The waves have a pulse width of 50% and period of 3 seconds. It should be noted that IN1 begins at 1.5 seconds, shifting it by half a period with respect to IN2. This will simulate the MCU driving the motor to go in different directions (in this case, it will oscillate between clockwise and counterclockwise). Also note that the motor driver VM pin is connected to a 12V source; the OUT1 and OUT2 pins are connected to the motor's two terminals; the ILIM pin has a 10kOhm resistor to ground; and the PGND, PPAD and GND pins are all grounded.

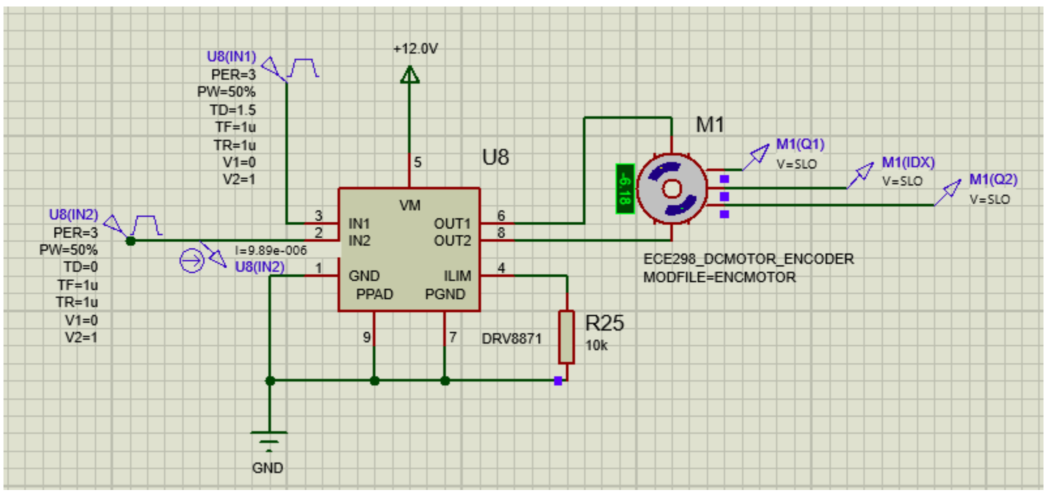

Fig. 1: Motor direction testing schematic

Figure 2 shows the outputs of Q1, Q2, and IDX of the motor. The motor is set to 24 pulses per revolution, indicated by the pulses created by IDX. Likewise, as with Lab 1, the direction of the motor can be detected by the rising edges of Q1 and Q2. At 1.5 seconds, the motor is at an inflection point in turning its direction. This is reflected by the longer pulses of Q1 and Q2, indicating a slowing down of motor revolutions. After 1.5 seconds, we can see that the rising edge relationship between Q1 and Q2 has flipped, thus indicating that the motor is now rotating in the opposite direction. This pattern is repeated at 3 seconds. Thus, given this interface can drive the motor in either direction, it will be suitable.

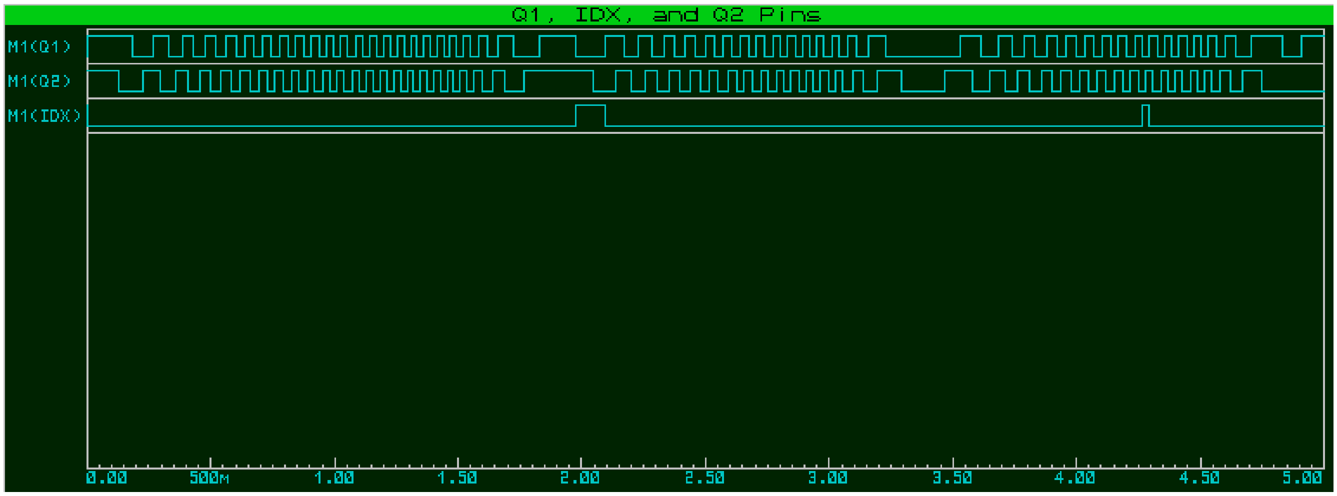

Fig. 2: Motor direction testing results

Figure 3 shows the schematic to test full-power in-rush current of the motor, as well as the current pulled from the MCU output at full-power. In this test, IN1 is given a constant 3.3V to represent an MCU digital output at a strong high, and IN2 is given a constant 0V to represent an MCU digital output at a strong low. Doing so provides the full extent of what the MCU can output, thus providing full power to the motor driver and therefore motor.

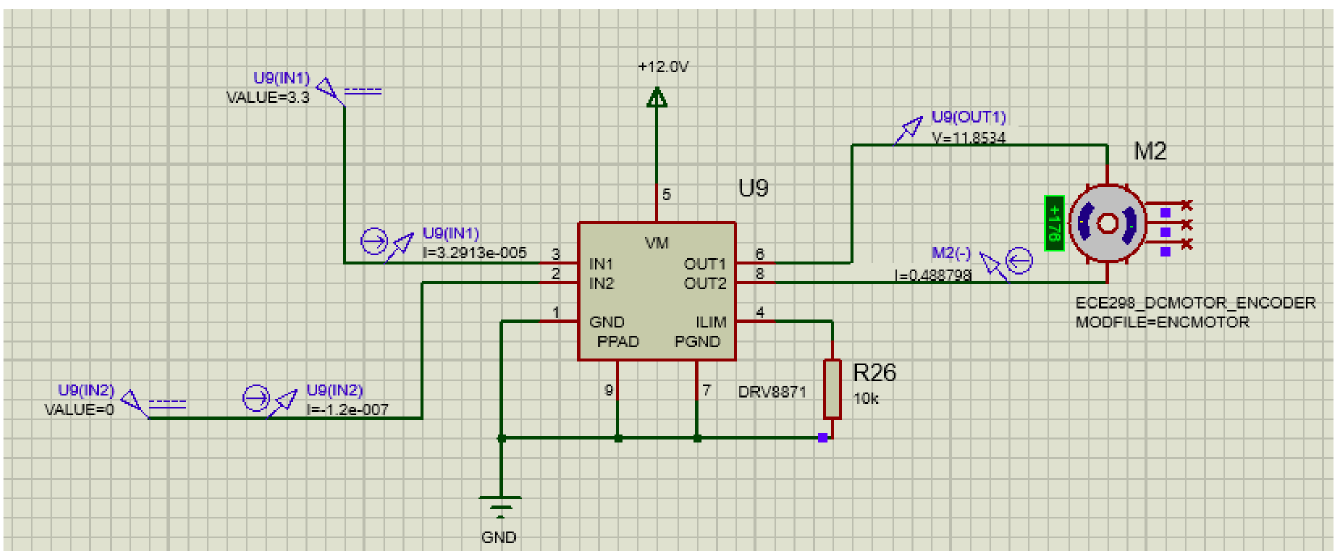

Fig. 3: Full power in-rush testing schematic

First, it would be useful to know if the interface draws a current within the limits of the MCU pins. Figure 4 shows the current drawn from the MCU outputs at IN1 and IN2. For U9(IN1), for which the maximum digital output from the MCU is supplied, the current remains constant at around 33µA. For U9(IN2), for which the digital output is a strong low, the current remains constant at near 0A. Therefore, the current drawn from the MCU digital output pins can be anywhere from 0A to 33µA when connected to this motor driver interface. This is much lower than the limit of 25mA per pin, making it a suitable interface.

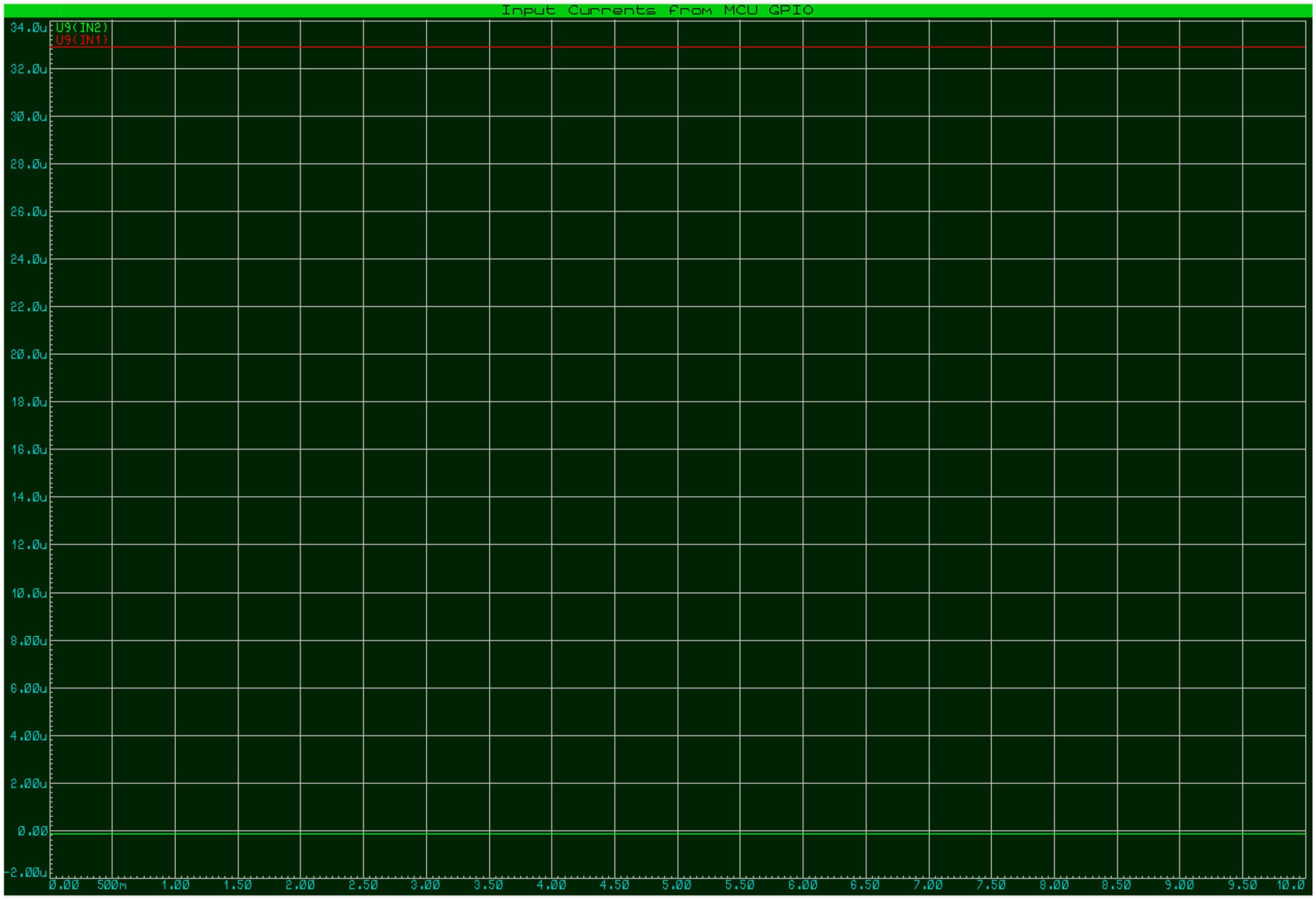

Fig. 4: Full power in-rush input currents to the motor driver

Figure 5 shows the full-power in-rush motor current and voltage. The voltage remains at a constant 12V, as the motor draws it from the source connected to the VM pin. The current spikes to around 952mA and stabilizes to around 500mA. Table 6.1 of the motor driver IC datasheet shows that the absolute maximum rating for the output current (at 100% duty cycle) is 3.5A. As such, the 952mA full-power in-rush current is tolerable and will not pose any major danger to damaging the IC. Therefore, this interface will be suitable in terms of handling the full power in-rush current.

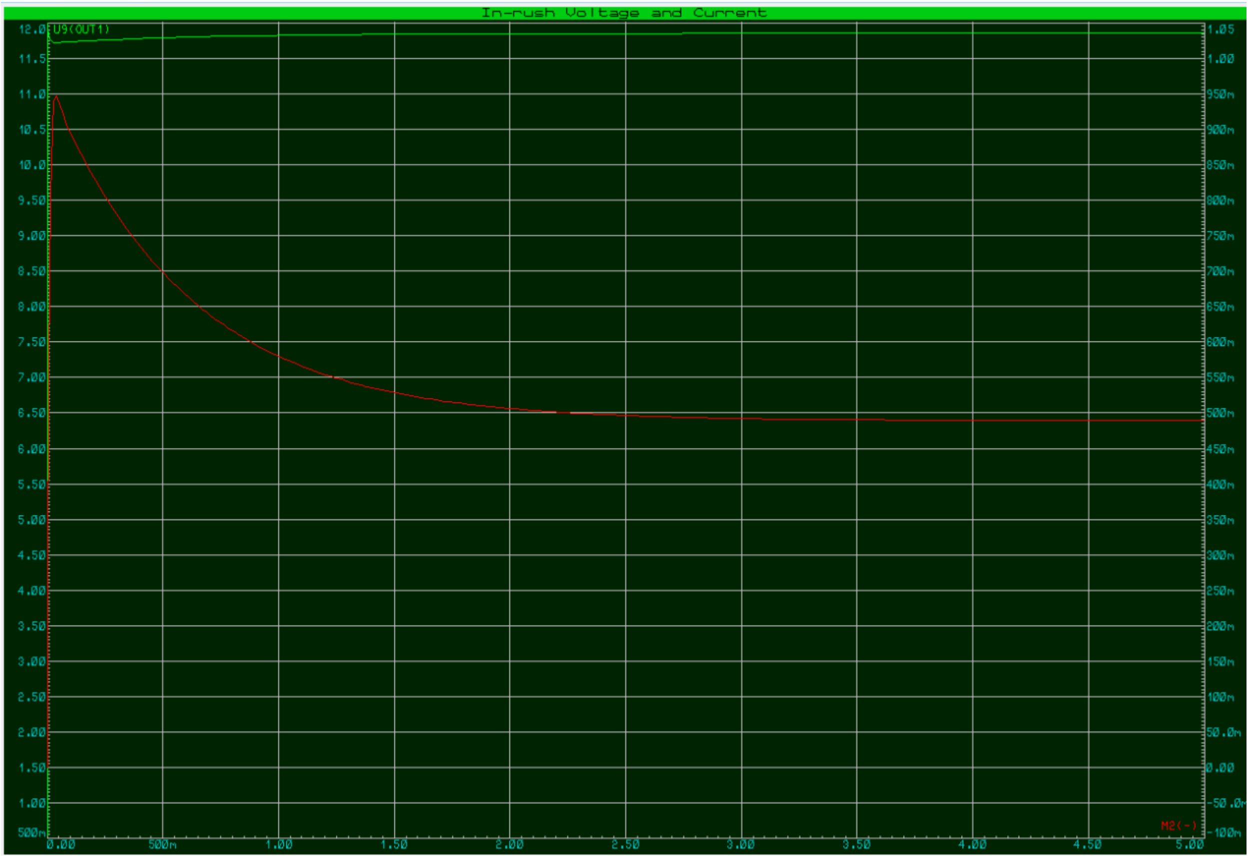

Fig. 5: Full power in-rush motor current and voltage

Figure 6 shows the schematic to test the motor voltage and current at varying inputs. As with the previous test, IN2 is set to 0V. However, this test sets IN1 to varying inputs of voltages (0V, 1V, 2V, 3V) to see what the interface response would be if the MCU pin output a weak high or weak low. Note that the IN1 voltages go from 0V to 3V and then down to 0V again. This is to test what the IC response would be in both directions.

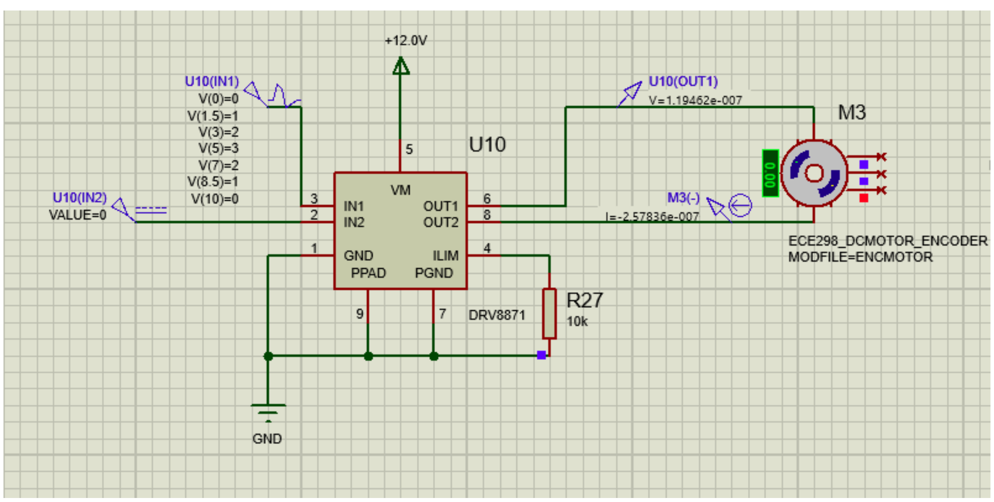

Fig. 6: Motor voltage and current testing schematic

Figure 7 shows the voltage and current of the motor. The voltages across the motor respond in a step-like manner, where it jumps from one value to the next. This is because the motor is no longer attached directly to the input, but rather it is being sourced by the 12V at VM. IN1 and IN2 only determines the response of the H-bridge as shown in section 7.2 of the device datasheet, which in turn determines the voltage supplied to the motor. One can see that the voltage at its maximum reaches 12V. With every step up of voltage, the current responds as though it is an in-rush current, shooting upwards and then decaying. With every step down of voltage, the current responds as though it is an out-rush current, shooting downwards and then exponentially growing. These are determined by the time constants made up of the internal resistance of the motor and any parasitic capacitances that exist. Again, the peak current only reaches around 800mA, which is far below the motor's maximum limit (3.5A) as described in Table 6.1 of the datasheet. Therefore, the performance of this interface is within limits and is suitable.

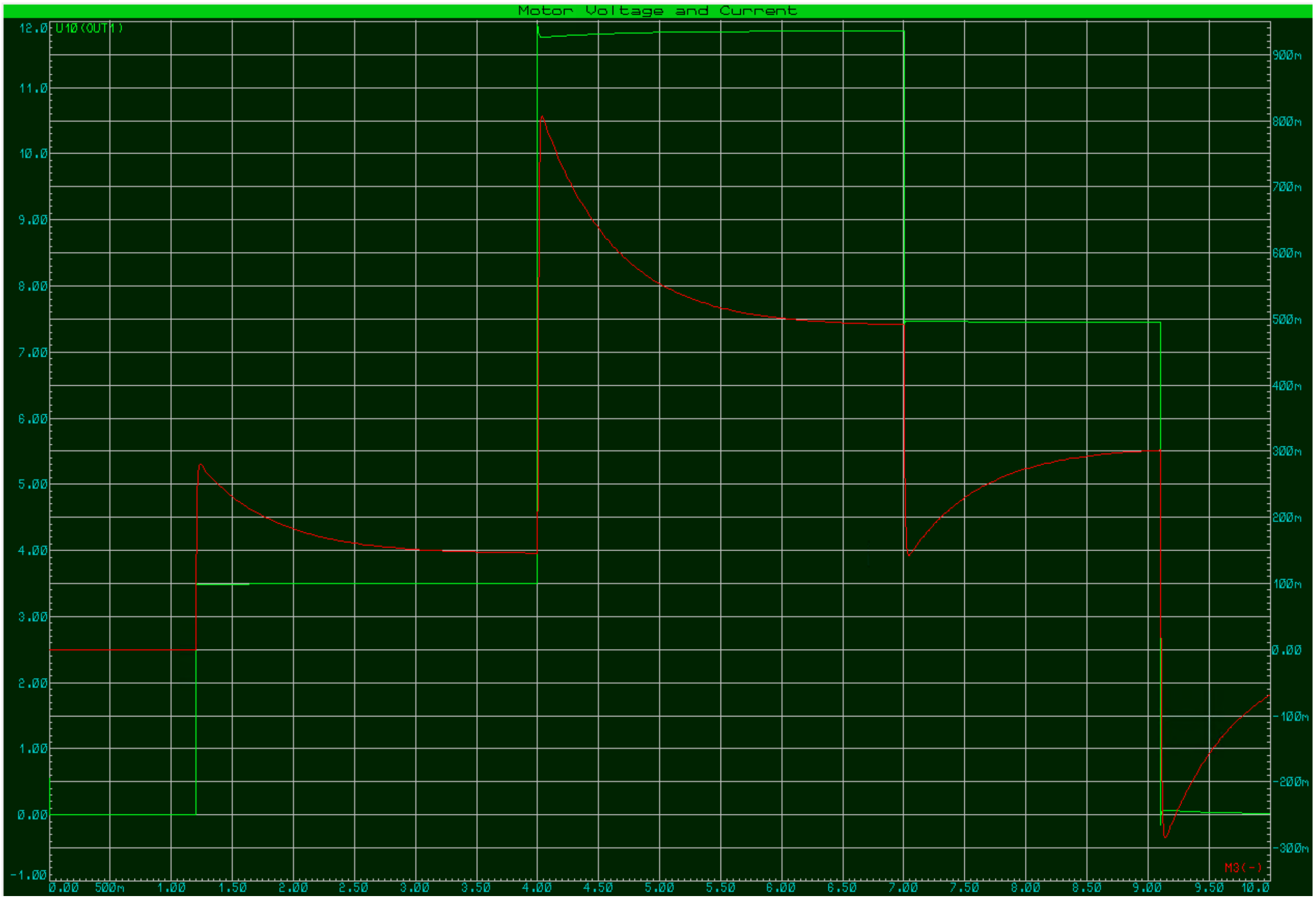

Fig. 7: Motor voltage and current testing results (note: y-axis scales for voltage and current do not align)

Figure 8 shows the schematic to test the motor interface under a PWM input. The PWM input is given by the PWM_input_testing.txt file which has a PWM signal with a low of 0V and high of 3.3V. As described in the device datasheet, the motor driver can take in a PWM input through IN1 and IN2, and the driver will be able to output another PWM signal to the motor. For this test, IN2 was set to 0V and IN1 had the PWM input. This corresponds to the description in the data sheet on how to use the motor driver to drive a PWM signal to the motor.

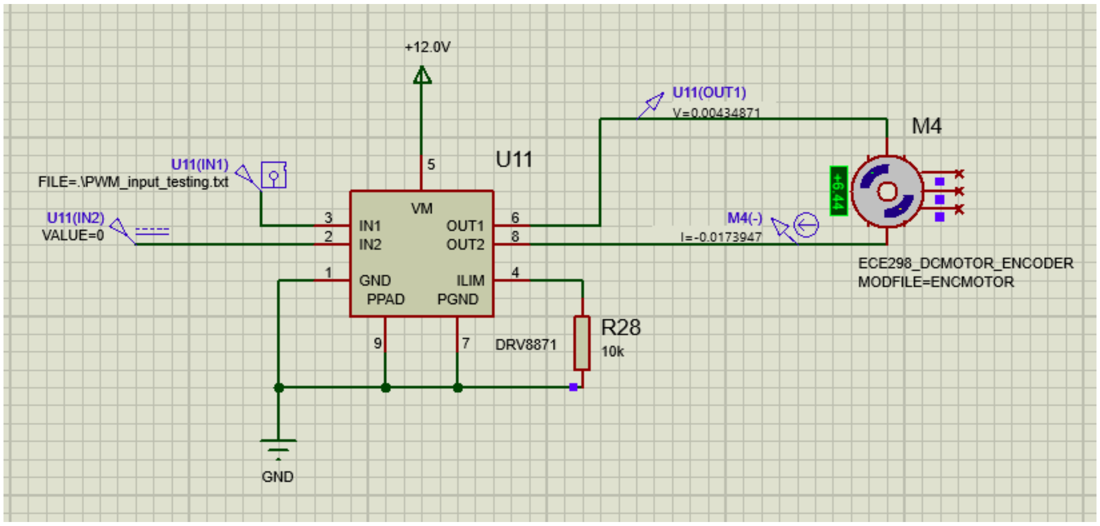

Fig. 8: PWM input testing schematic

Figure 9 shows the current and voltage across the motor as the PWM signal is being input. The voltage oscillates between 0V and 12V, as is expected (given that the VM pin is connected to a 12V source), and the current responds as an in-rush current, shooting up then decaying. During the PWM signal with a higher frequency, the current shoots up and down in several 'steps'. The rather quick time constant means that the current decays very quickly, resulting in larger oscillations. This can be seen by the first 'step' of current from 500ms to 700ms, where the current fluctuates between 200mA to 500mA. It should be noted that 7.4.1 and 7.4.2 of the device datasheet indicates that the motor driver can operate a PWM signal with or without current regulation. As such, it would be possible to be safer and more precise in terms of current behavior. Also note that Table 6.3 of the datasheet shows the maximum PWM frequency to be 200kHz. This is the bottleneck if also considering the MCU's maximum PWM frequency of 42 MHz. Either way, the RPM measurements of the PWM signal from the text file (which has a frequency much less than 200kHz) is high enough to provide for our needs. As such, this interface will be suitable with a PWM input. We just need to decide the proper duty cycles and frequencies of the PWM signal for our purposes.

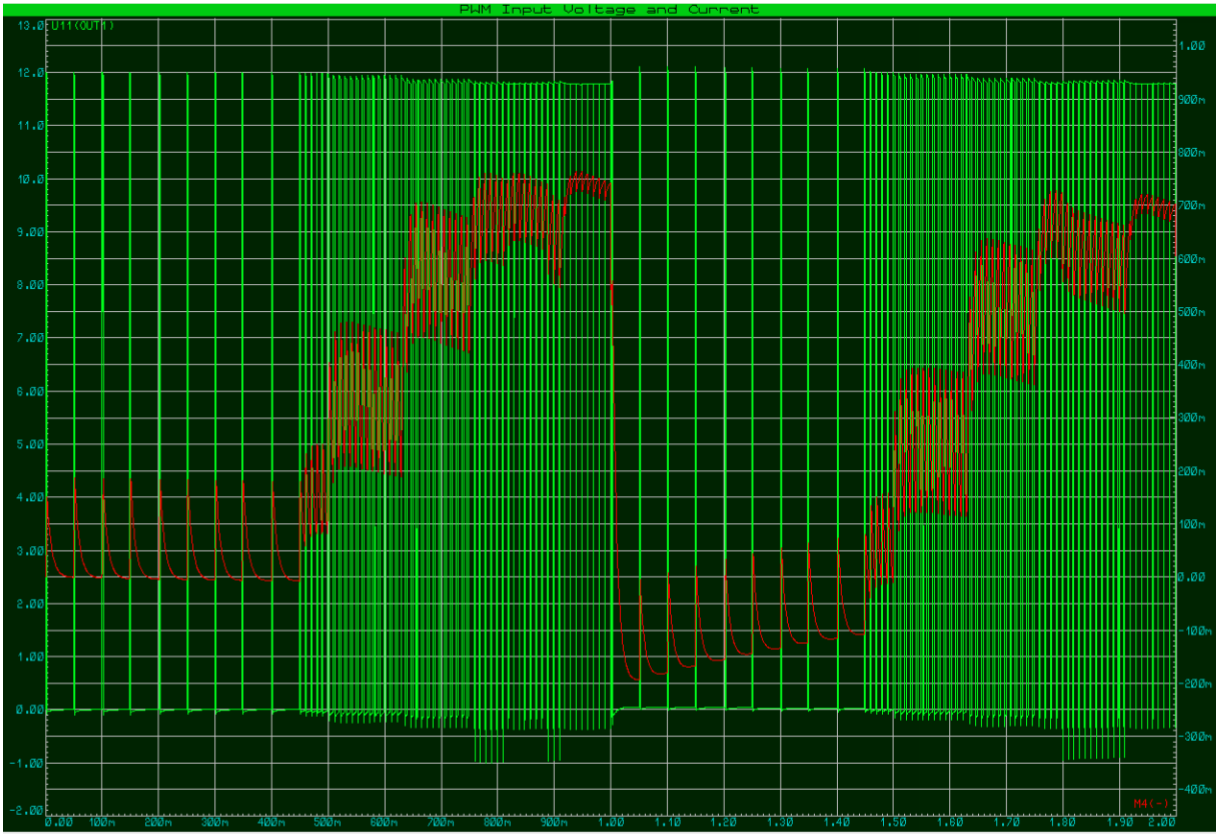

Fig. 9: PWM input testing results

Light Emitting Diode (LED)

Summary

Item	Description
Purpose	To indicate to the user the current state of the automatic door system. A red LED indicates the door in motion, a yellow LED indicates the door has collided with an object, and a green LED indicates an object is within the inside and outside proximity ranges.
Device physical domain and range	The light intensity of the LED (determined when the specific LED is chosen later) and the color of the LED (green, red, and yellow).
Device type chosen	LED
Proteus Library component name	LED-RED, LED-GREEN, LED-YELLOW
Device input / output properties	The LED has two terminals, which when an analog is applied across, will produce a current, lighting the LED. For all three components, the forward voltage at 20 mA is 2.2 V, the full drive current is 10 mA, and the minimum on-time to light the LED is 10ms.
Device input / output range	The breakdown voltage of the LEDs is 4 V, meaning that there should not be a voltage applied in the reverse direction of the diode more than 4 V. Otherwise, the LED will be damaged.

MCU connectivity details	The MCU will output a digital signal through the correct pin that is connected to the corresponding LED. Note that it is assumed that only one LED color can be turned on at a time.
Device/MCU interfacing details	The MCU pin will output a digital signal to the gate terminal of an n-channel MOSFET (2N7002). The MOSFET will act as a switch for the LED, which is connected via the drain terminal to a 5 V analog source with a pull-up resistor of 145 Ohms. The source terminal will be grounded. A digital high signal (3.3 V) applied by the MCU digital output is enough to allow current to 'turn on' the transistor, resulting in an analog input into the LED. As such, the interface acts as a digital to analog converter.

Schematics and Simulations

Figure 10 shows the schematic for LED testing. To simulate an MCU digital output, a square wave with a low of 0V and high of 3.3V are outputted into the gate terminal of the n-channel MOSFET. The wave has a frequency of 0.5Hz, and a pulse-width of 50%. The LED is attached to the drain terminal of the MOSFET with a pull-up resistor (145 Ohm) and 5V source, and the source terminal of the MOSFET is grounded. Note that the square waves begin at 0s, 0.3s, and 0.6s for the green, red, and yellow LEDs respectively. This is to make the current measurement easier to read. But also note, that this would mean the LEDs do not turn on at the same time in the simulation.

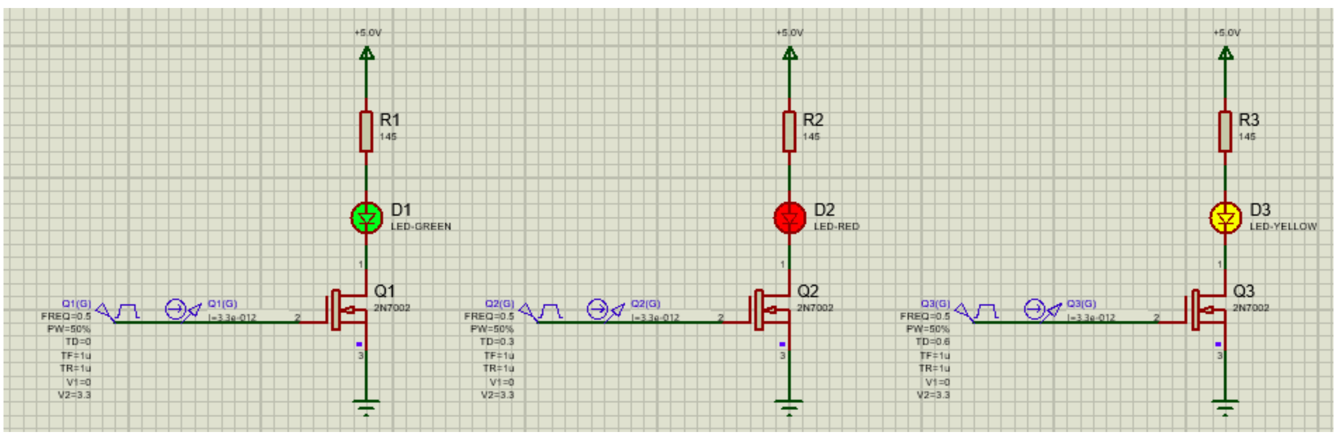

Fig. 10: LED interface testing schematic

Figure 11 shows the currents being drawn from the MCU pins into the interface. The maximum current being pulled from any of the three LEDs at Q1, Q2, and Q3 respectively, is around 560µA. Likewise, the minimum current being pulled from any of the three LEDS does not go below -250µA. As such, the range of operation for this interface is from -250µA to 560µA. This is far lower than the limit given in the MCU datasheet of +/-25mA per pin. Given there are 4 total LEDs in this project, the worst-case scenario would give an aggregate current of 2.24mA (i.e. all LEDs are on), which is far lower than the aggregate limit of 120mA. Also, most importantly, the LEDs turn on when the signal is high, and off when the signal is low. As such, given all these considerations, this interface will be suitable.

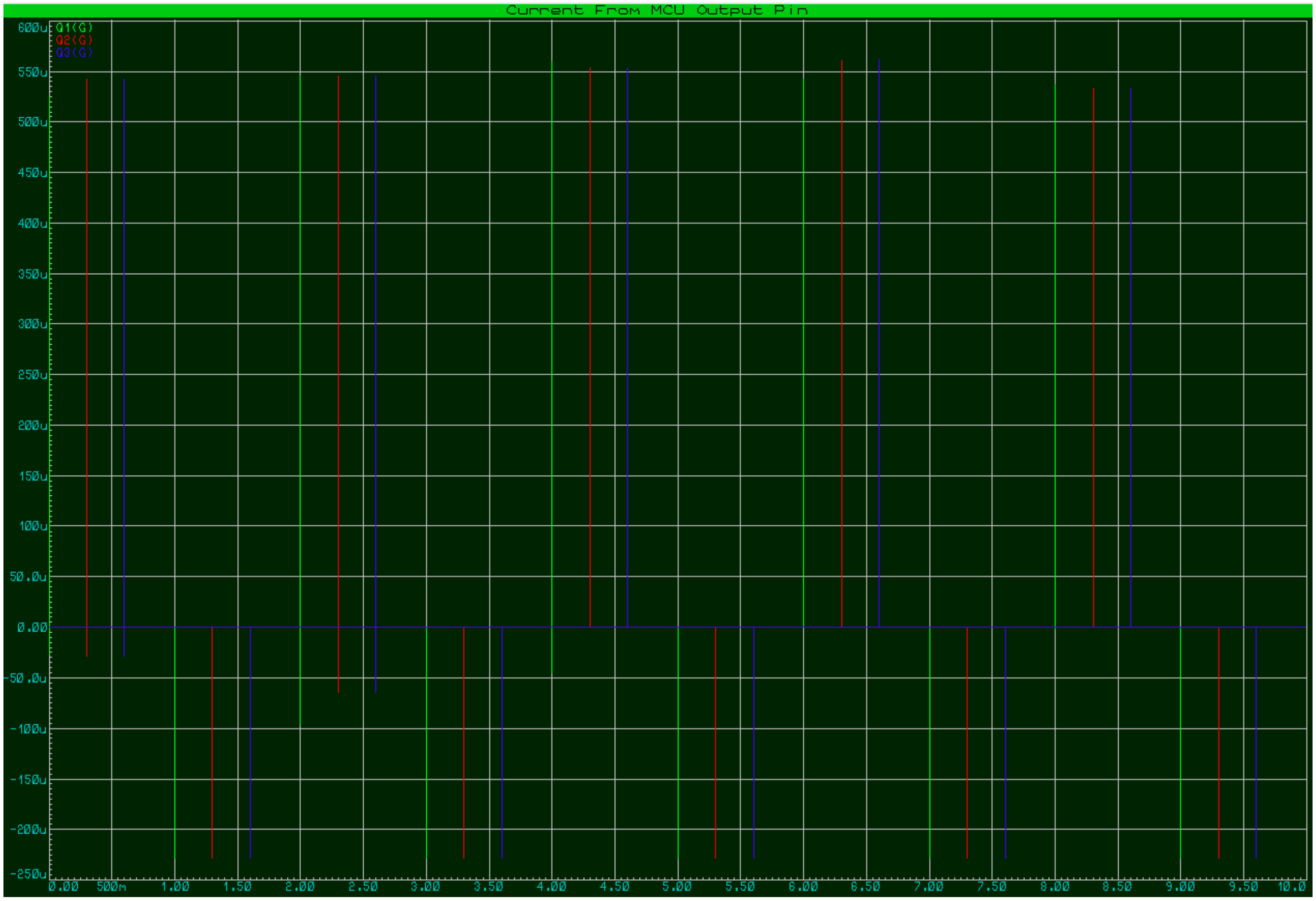

Fig. 11: LED interface testing results

Liquid Crystal Display (LCD)

Summary

Item	Description
Purpose	The LCD displays an alphanumeric string indicating the current mode of operation of the door.
Device physical domain and range	The alphanumeric string to be output to the LCD.
Device type chosen	LCD
Proteus Library component name	LM016L
Device input / output properties	The LCD has eight parallel data bus lines (D0 to D7) that correspond to the byte mapping of the ASCII characters to output. There is also an enable signal pin (E), a read-write pin (R/W), a register selector pin (RS), and the power supply pins (V_{DD}, V_{SS}, V_{EE}).
Device input / output range	The LCD takes +5V digital inputs at 1 to 10 mA (section II.A of the device datasheet). The datasheet in Proteus does not show a minimum input high or a maximum input low, but the Hitachi datasheet (found online) shows a minimum input high voltage of 2.2 V and maximum input low voltage of 0.6 V (Hitachi).

MCU connectivity details	The MCU will output digital signals for the following pins on the LCD: enable (E), register select (RS), read/write (R/W), data (for now it is D7 to D4, but in the future this may include all eight pins).
Device/MCU interfacing details	The MCU digital output will be 3.3V, while the LCD accepts only 5V digital inputs. As such, a digital-to-digital unidirectional step-up voltage interface will be used to bring up the voltage to 5V. This involves inputting the MCU signal into the gate of an n-channel MOSFET, with the source grounded, and the drain connected to a pull-up resistor of 100kOhm with a 5V source. The output will be taken from the drain. But this output will be inverted. As such, an extra inverter is added to re-invert the signal to match the input signal. This step-up voltage interface will be implemented for each LCD pin being used (E, RS, RW, Data pins)..

Schematics and Simulations

Figure 11 shows the schematic for testing the LCD using the interface. It should be noted that a pattern generator was used. However, this pattern generator only output 5V digital signals. As such, that output first needed to be stepped down to 3.3V to mimic the digital outputs coming from the MCU pins. This circuitry is shown in Figure 12, where all the needed pins from the pattern generator are pulled down using a voltage divider with two resistors of 47kOhm and 91kOhm. The pattern generator and the step down digital to digital voltage dividers can be thought of as the MCU internals, which would then output the appropriate 3.3V digital signals to the external interface, shown in Figure 13. The LCD also takes in only 5V digital signals. As such, the 3.3V outputs from the MCU need to be stepped-up using an n-channel MOSFET with a pull up resistor of 100kOhm and a 5V source. The MCU output is connected to the gate of the MOSFET, the pull up resistor to the drain, and ground to the source. However, this stepped-up voltage will be inverted due to the n-channel MOSFET. So, an extra inverter is required to re-invert the output signal, so it matches the input signal. Note that the E, R/W, and RS pins are connected, alongside the data bus line (which for now are just D7 to D4). As with Lab 1, the pattern is set to output the team number '156'.

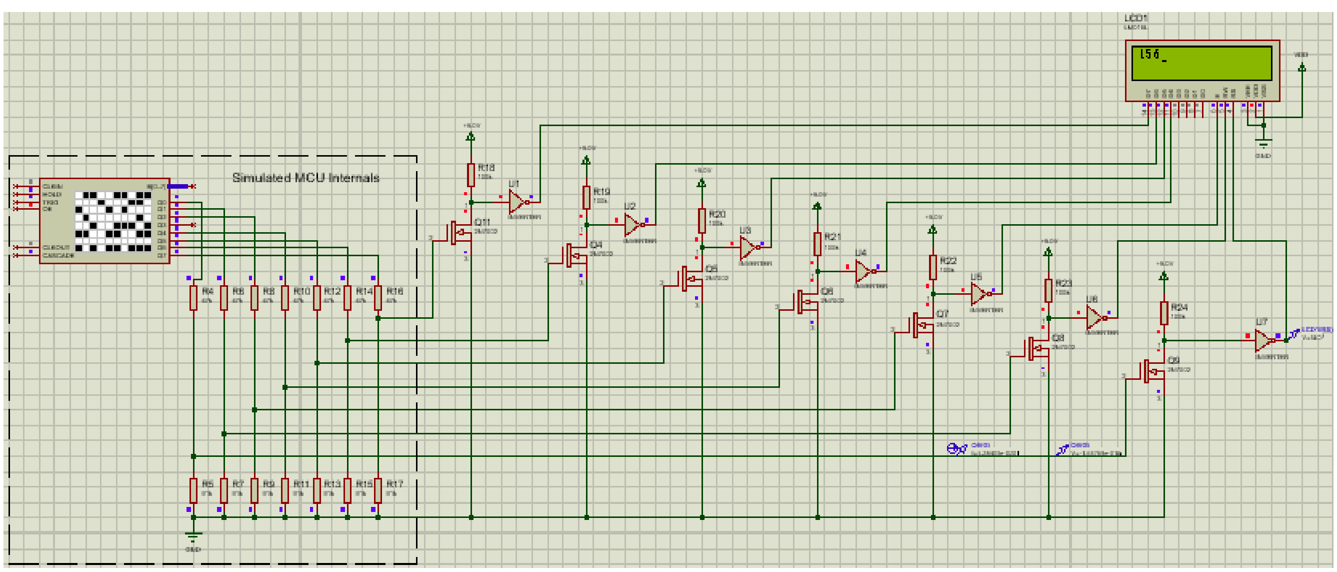

Fig. 11: LCD interface testing schematic

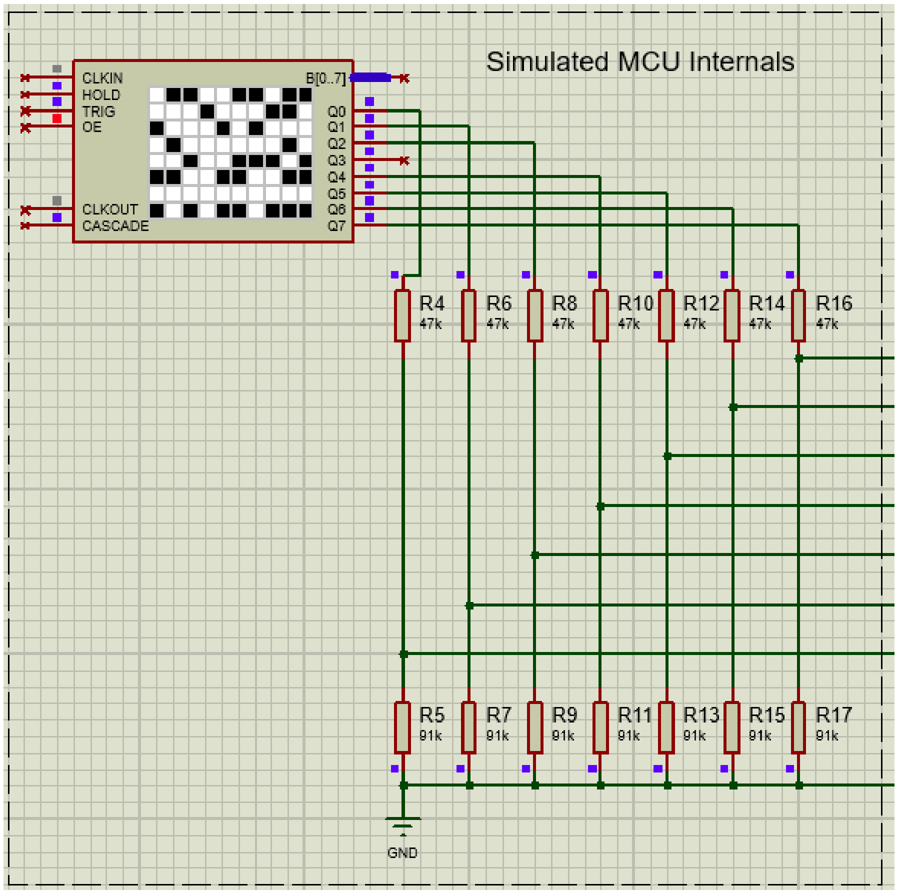

Fig. 12: Simulated MCU internals schematic

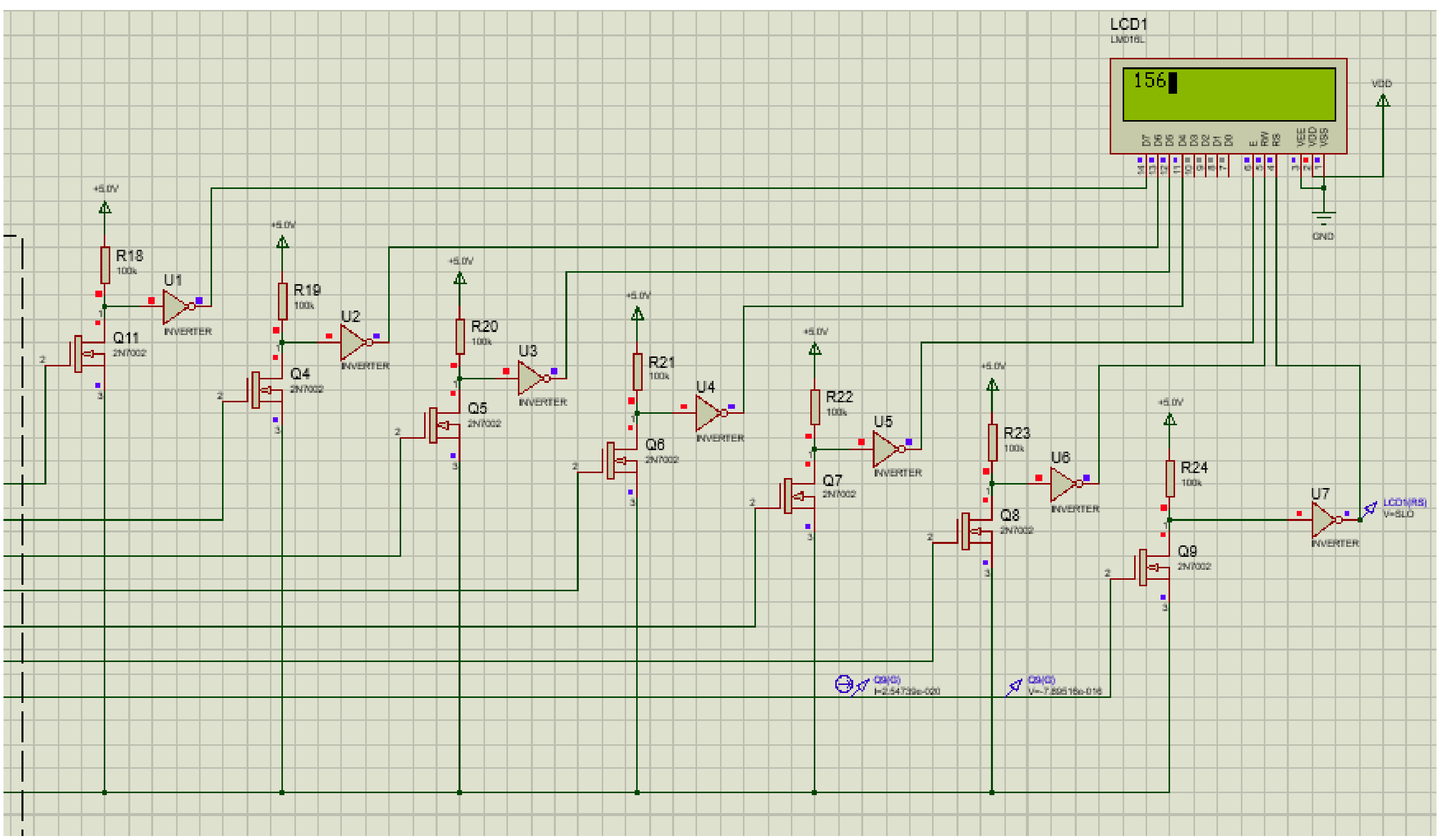

Fig. 13: LCD external interface schematic

Figure 14 shows the input current, input voltage at Q9 (i.e. one of the MCU digital outputs). The figure also includes the output of the interface at LCD1(RS). First, the input current to the interface reaches a maximum of around 110μA (at 4.5 seconds), and a minimum of around -110μA (at 8.5 seconds). These current spikes coincide with transitions from a strong low to a strong high and vice versa. Either way, this tells us that the interface will draw a range of current from -110μA to 110μA, which is far below the stated MCU limit per pin of +/- 25mA. Assuming that all LCD pins are being outputted to at the same time, the aggregate current drawn from the MCU pins will be 110μA x 11, which is 1.21mA. This is also far below the stated aggregate limit on the MCU pins of 120mA. Therefore, this interface will be appropriate for the current limits. Also, it can be seen that as the input voltage is a strong low (i.e. 0V), so is the output voltage. When the input voltage Q9(G) is a strong high (3.3V from the MCU), the output voltage LCD1(RS) is a strong high of 5V for the LCD. This indicates that the step-up interface works, and this is further shown by the LCD correctly displaying '156'. Therefore, this interface will be suitable.

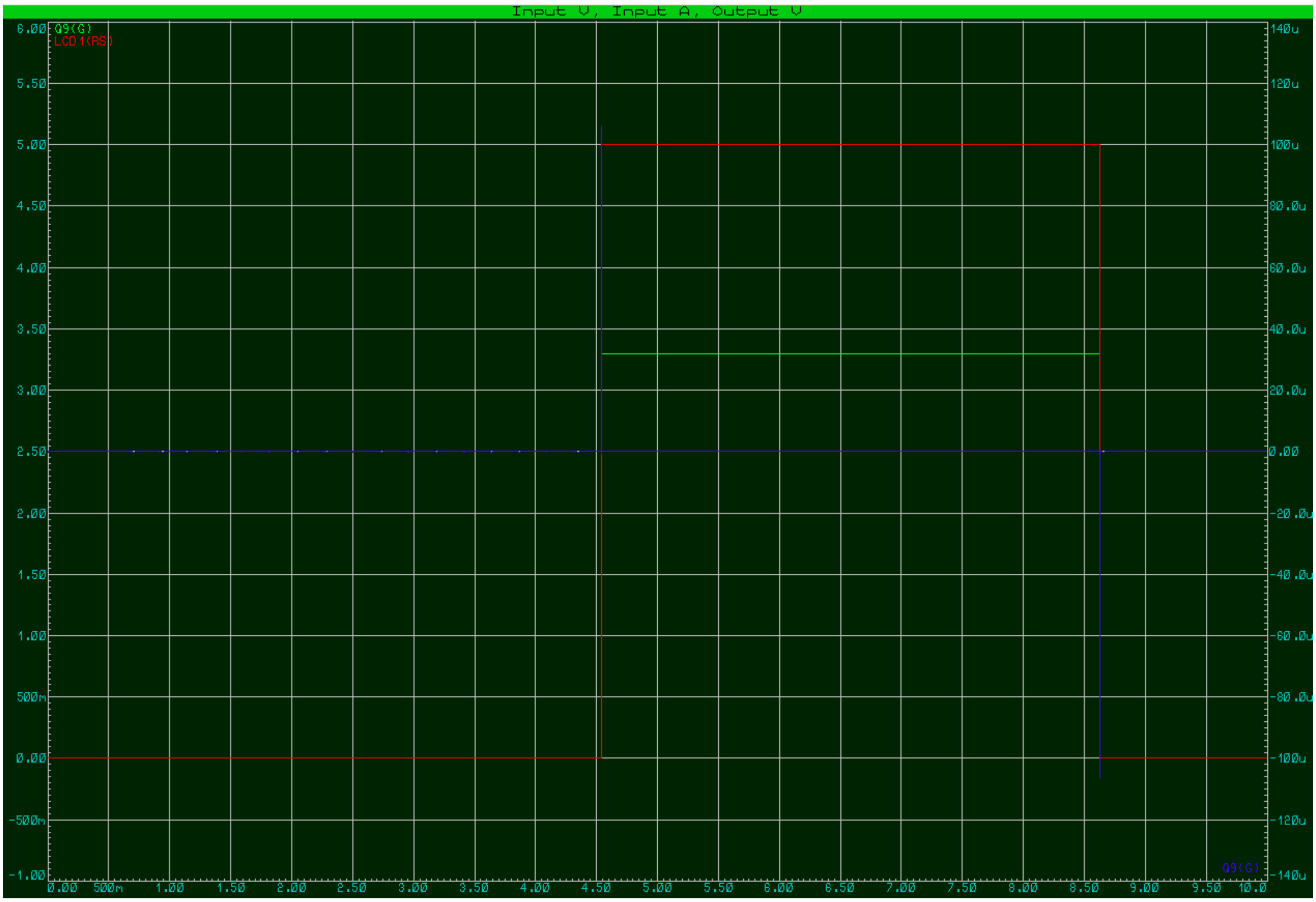

Fig. 14: LCD interface testing results

Ultrasonic Ranger Module

Summary

Item	Description
Purpose	Detects when an object is within a certain distance of the sensor so the door can automatically open.
Device physical domain and range	The module senses objects within 2cm to 4m in a range of 15 degrees.
Device type chosen	The chosen sensor is an ultrasonic proximity sensor
Proteus Library component name	HCSR04
Device input / output properties	The device has four digital input/output pins. One for source voltage and ground. The other two correspond to the input trigger TTL pulse and the time-varying echo pulse (depends time taken by pulse to leave and return). The input and output are push-pull since they can only be two values, strong 0 or strong 1.
Device input / output range	The device has a working voltage of 5V. VOH, min = 2.7V. VOL, max = 0.4V. VIH, max = 2V. VIL, min = 0.8V. The input trigger pulse must have a width of 10us and cannot be transmitted again until the echo has returned or timed out (60ms). The output signal will have a 8* 25us delay since it must wait for the sonic burst to output from the module. The output signal corresponds to the distance measured to an object and could range from around 150 us to 25ms or up to 38ms if nothing is detected.
MCU connectivity details	A digital input pin and a digital output pin will be needed for the trigger and echo pins.
Device/MCU interfacing details	Input: (D/D) The input will need an N-channel MOSFET with a pullup up resistor with an inverted signal to increase the signal. Output: (D/D) The output will need a voltage divider to scale down the signal.

Schematics and Simulations

Figure 15 shows the schematic for testing MCU input and output to the ultrasonic ranger module. The Q1 signal generator simulates the MCU output and the Pi terminal represents the input to the MCU. Since the N-channel MOSFET inverts the signal, Q1 must start at a constant high signal of 3.3V and generate a 10us pulse of 0V. In Figure 16, you can see that's what it does and the MOSFET correctly inverts and adjusts the signal to a 5V pulse of 10us into the trigger.

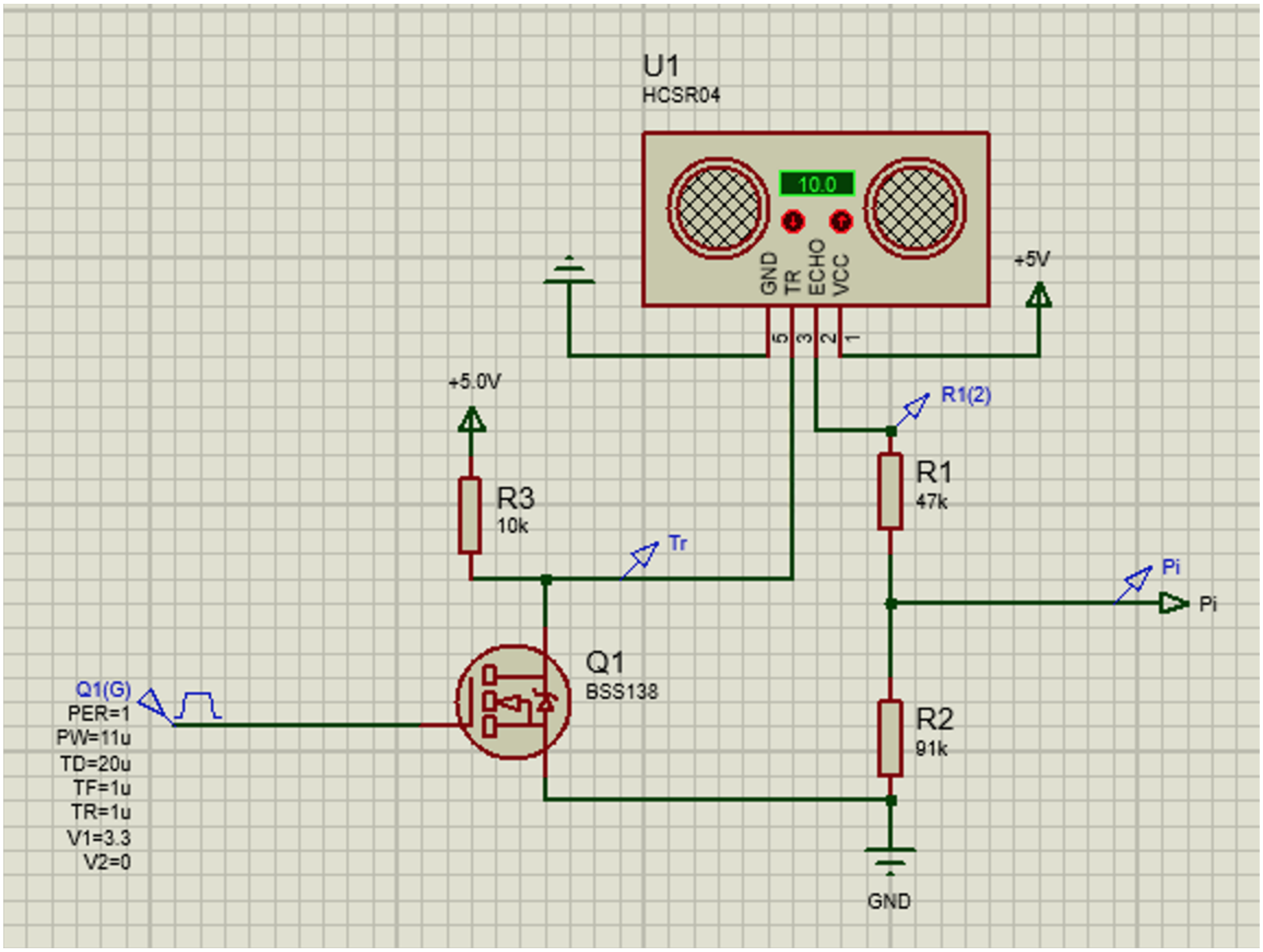

Fig. 15: Ultrasonic Ranger Module Interface Circuit

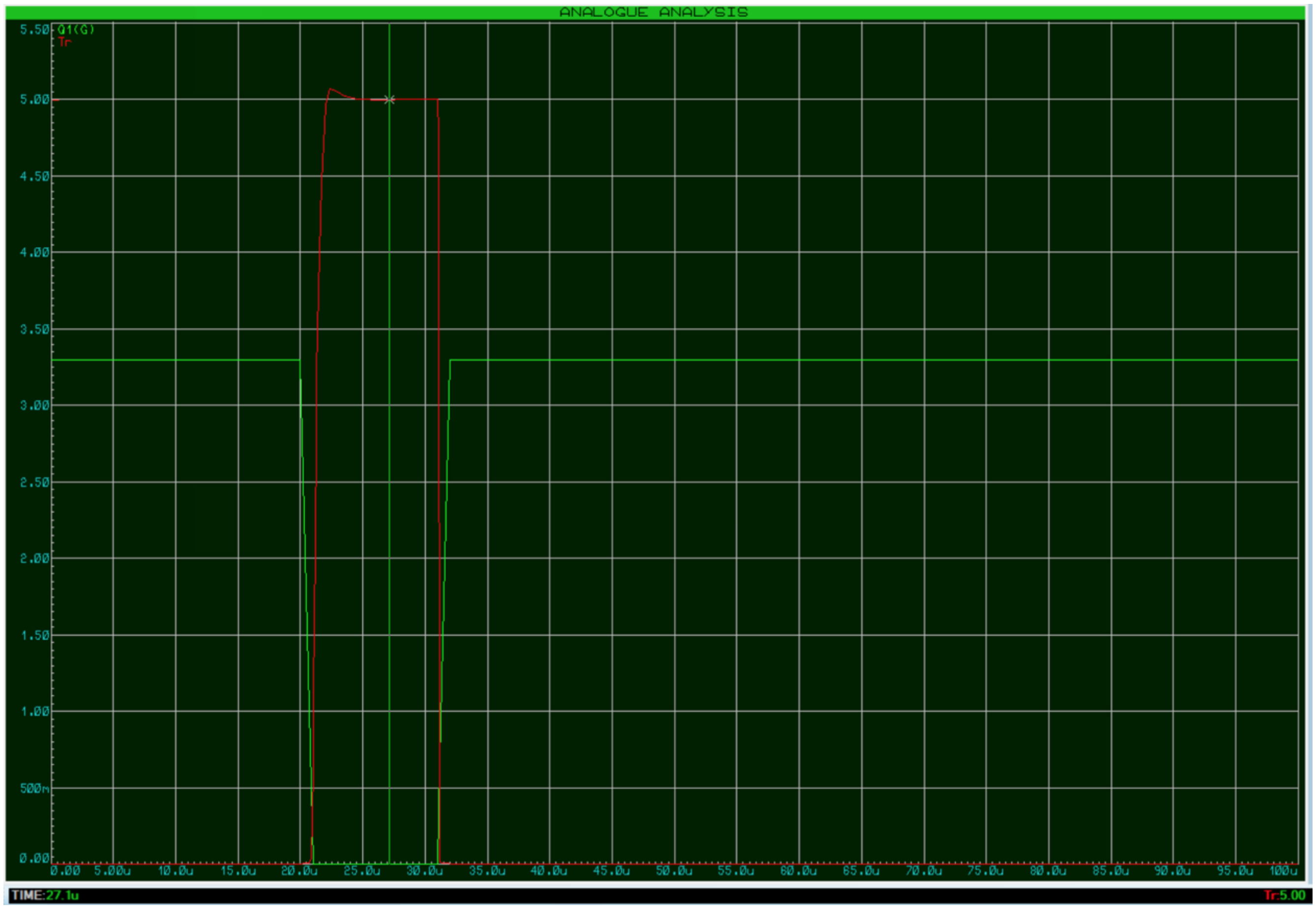

Fig. 16: Analog Analysis of the simulated MCU output (Q1) and the adjusted trigger input (Tr).

In Figure 17, the voltages on the ECHO line of the ranger are shown in the analog analysis as well. The signal produced straight out of the ranger module gives a varying pulse of 5V. The duration of the pulse depends on the distance of the object sensed and is not currently relevant. After adding a voltage divider, the signal is correctly reduced to 3.3 V without any change other than the magnitude, asserting that the MCU will obtain an accurate reading of the device.

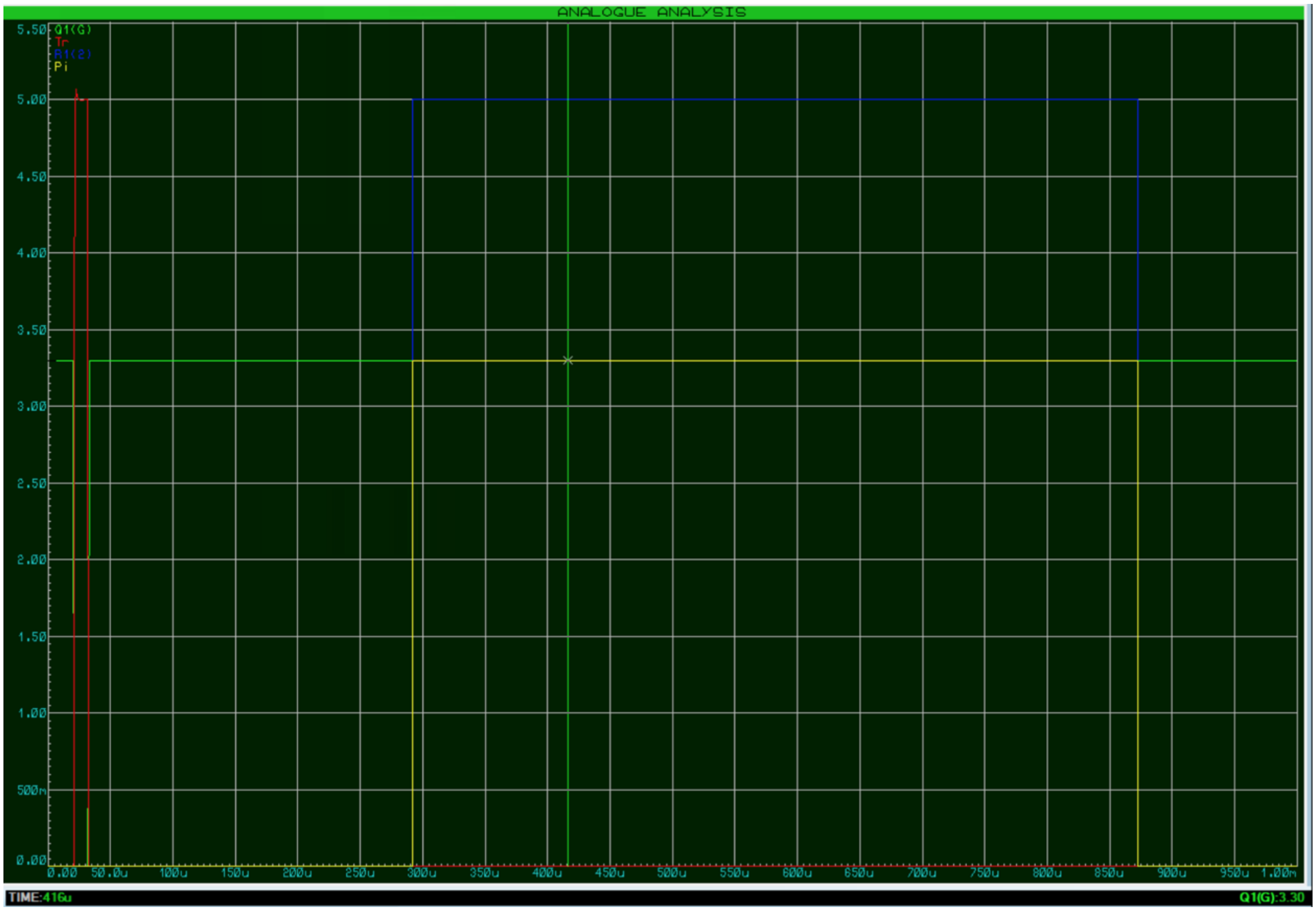

Fig. 17: Analog Analysis of all the voltages

DC Motor Encoder

Summary

Item	Description
Purpose	Senses and monitors speed and rotational direction of the door motor.
Device physical domain and range	The encoder will detect up to the motor's maximum speed and whether it is turning clockwise or anticlockwise
Device type chosen	A position encoder attached to the shaft of a DC motor.
Proteus Library component name	ECE298_DCMOTOR_ENCODER
Device input / output properties	The encoder outputs three digital signals that are all push-pull. Two outputs must be read together to determine the rotational direction and speed of the motor. The third signal generated can also be read to determine the speed of the motor. The frequencies of the signals depend on the speed of the motor.
Device input / output range	The device has a working voltage of 5V. VOH, min = 2.7V. VOL, max = 0.4 V.
MCU connectivity details	Three digital input pins will be needed to read the signals of the encoder.
Device/MCU interfacing details	Output: (D/D), The outputs will need voltage dividers to scale down the signal.

Schematics and Simulations

Figure 18 shows the schematic for testing the DC motor encoder. The probes labelled R7(2), R8(1), and R6(2) represents the input pins into the MCU. Since the max voltage of the output from the encoder is greater than what is accepted into the MCU, three voltage dividers will be needed to reduce the signal. A generic 1V DC generator is inputted into the motor to run it, since the encoder depends on the motor to work.

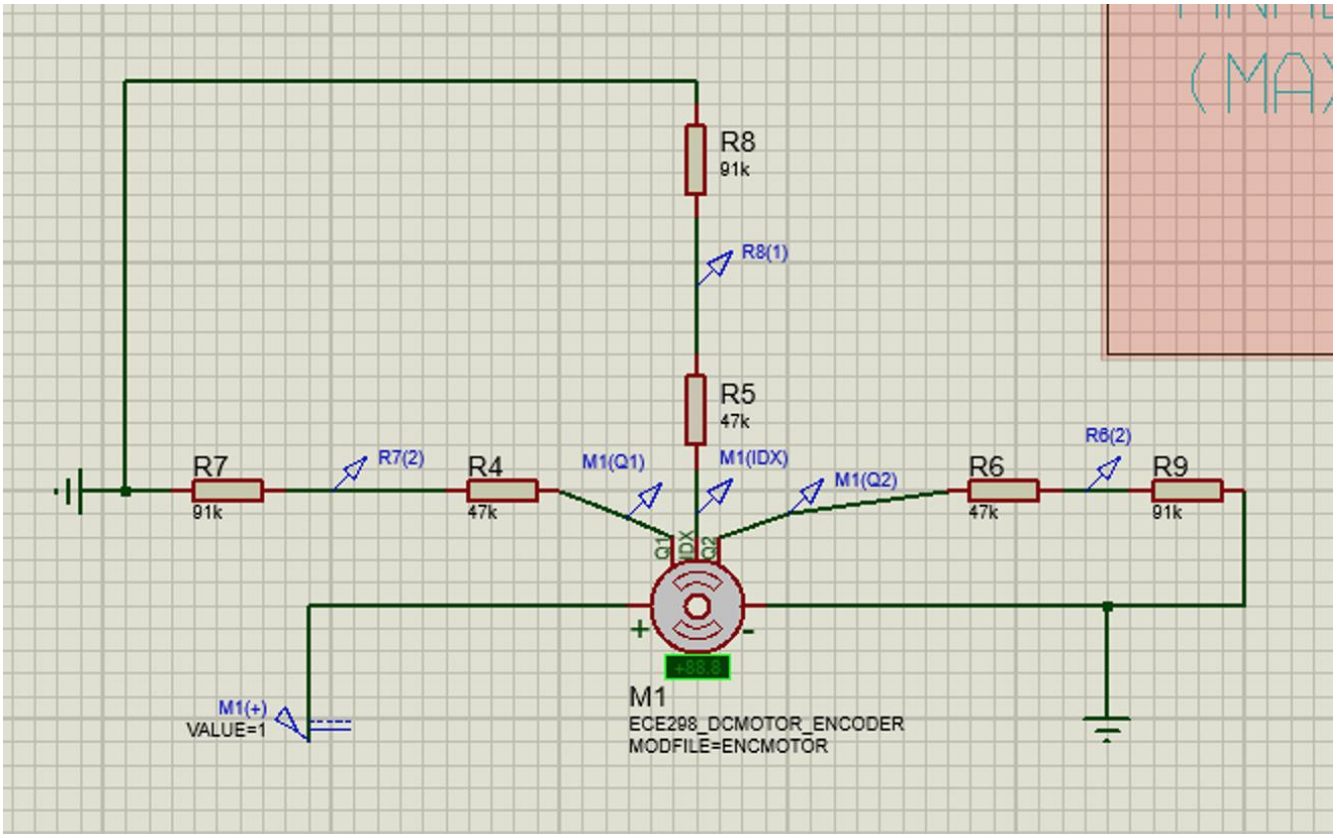

Fig. 18: DC Motor Encoder Interface Circuit

Figure 19 shows the three outputs of the encoder. All three signals have a high of 5V which is too high for the MCU to read. The pulses of the signal are dependent on the rotational speed and direction of the motor and are not relevant currently. Using digital analysis they can be more accurately read and deciphered but that is not necessary for only interfacing testing with the MCU.

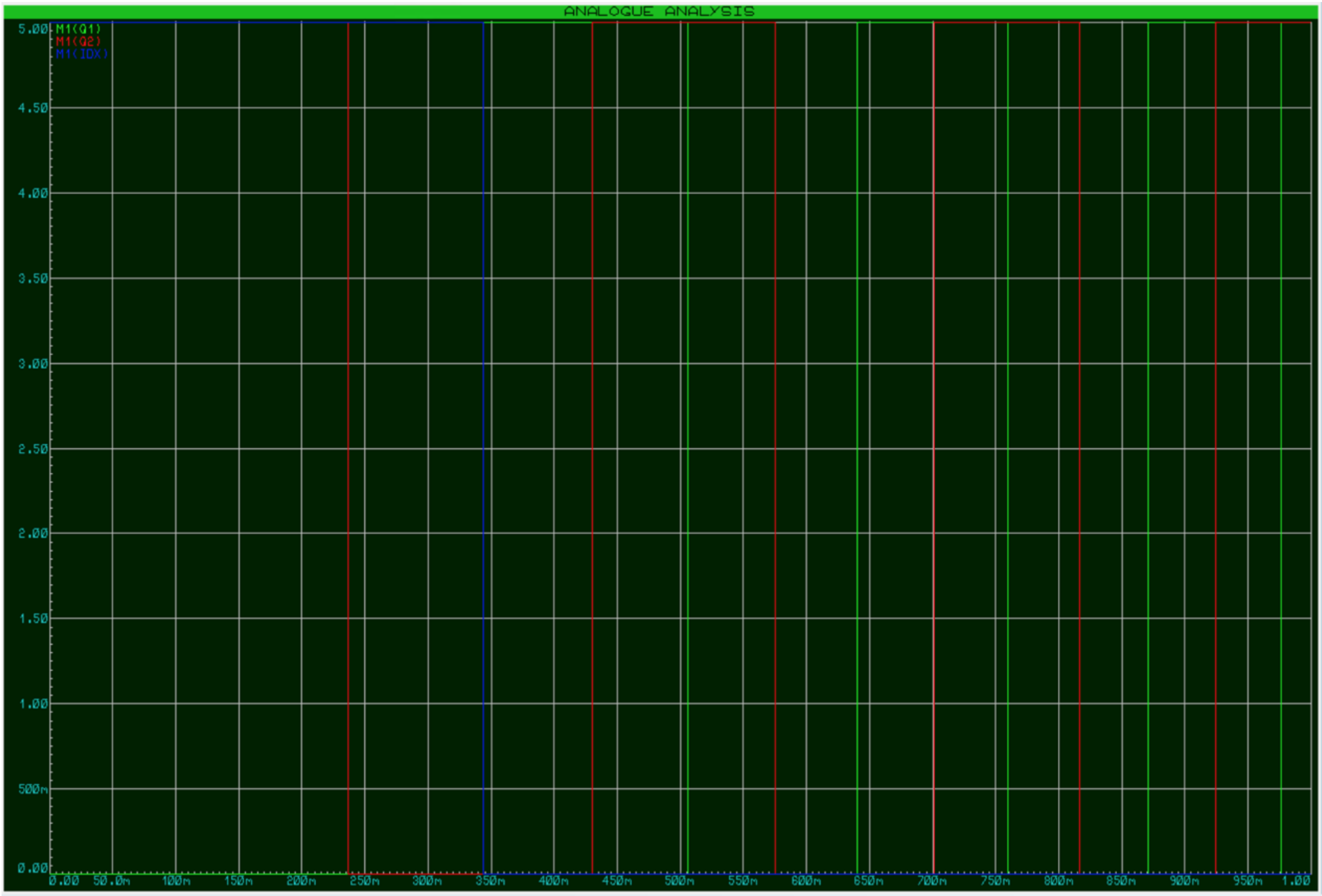

Fig. 19: Analog Analysis of the Outputs of the Encoder

In figure 20, the analog analysis shows the readings of the probes, simulating the signal into the MCU. The signal has been reduced to 3.3V which is what we need. Also, when comparing with Figure 19, there is no difference between the output of the encoder and the input into the MCU other than the magnitude of the signal which is important since we need to preserve the information transmitted by the encoder.

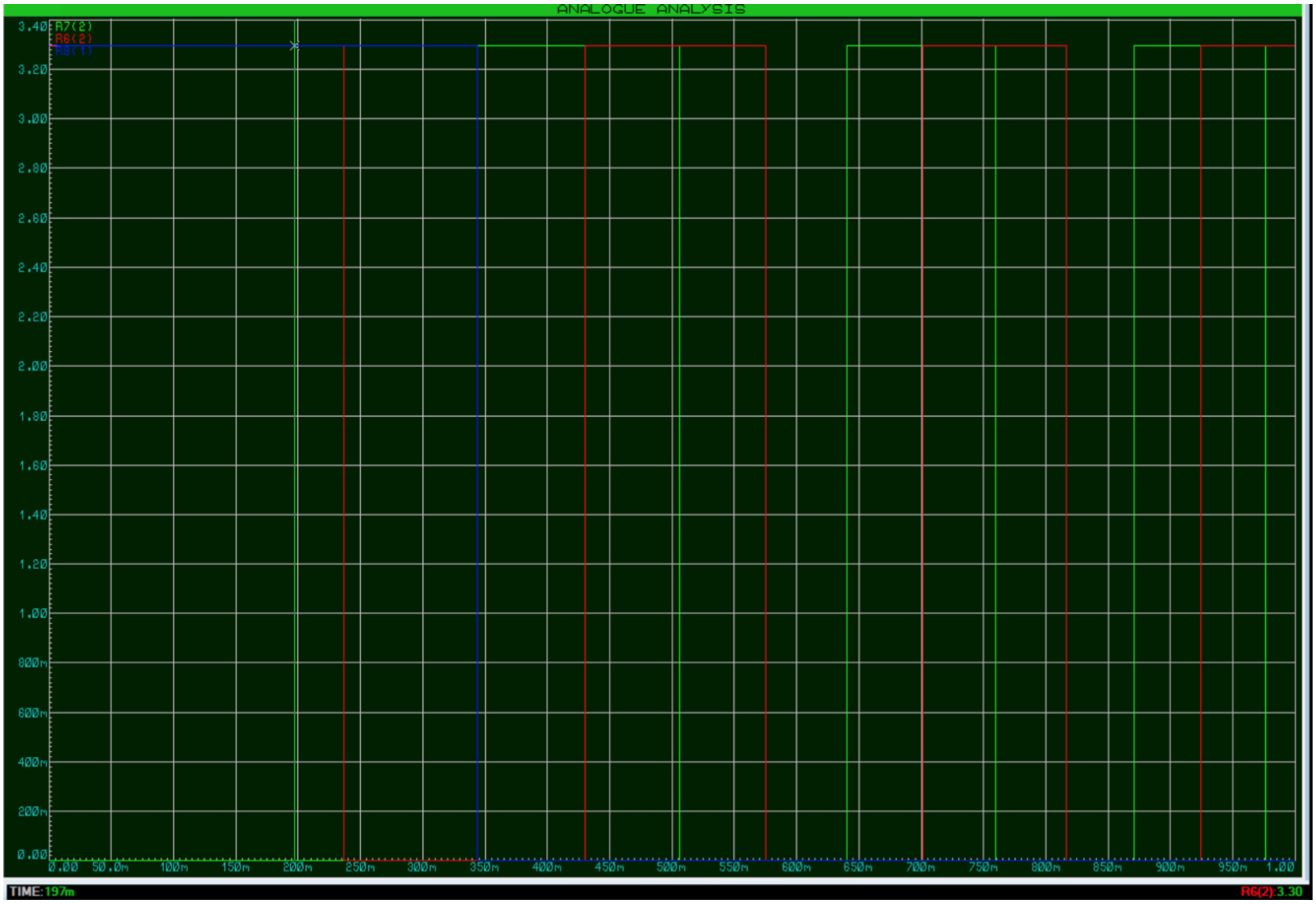

Fig. 20: Analog Analysis of the adjusted simulated input to the MCU

SPST Push Button

Summary

Item	Description
Purpose	A simple switch mechanism that will be used for user input as key buttons and as limit and collision switches.
Device physical domain and range	The device will detect when a physical object interacts with it.
Device type chosen	Push Button
Proteus Library component name	BUTTON
Device input / output properties	The device will input a constant voltage and output high if an object interacts with it. Although when not enabled, the output is an open circuit, it acts as ground.
Device input / output range	The device has a working voltage of 3.3V. There is no min/max input/output voltage since it is directly communicating a constant voltage.
MCU connectivity details	A digital input pin will be needed to read each push button.
Device/MCU interfacing details	Output: (D/D), no interfacing will be needed.

Schematics and Simulations

Figure 21 shows the schematic for testing the push button. It is very straightforward as the push button can just be directly connected into the MCU with no interfacing. In figure 22, the analog analysis correctly asserts the when the push button is pressed, the MCU will receive a signal of 3.3V.

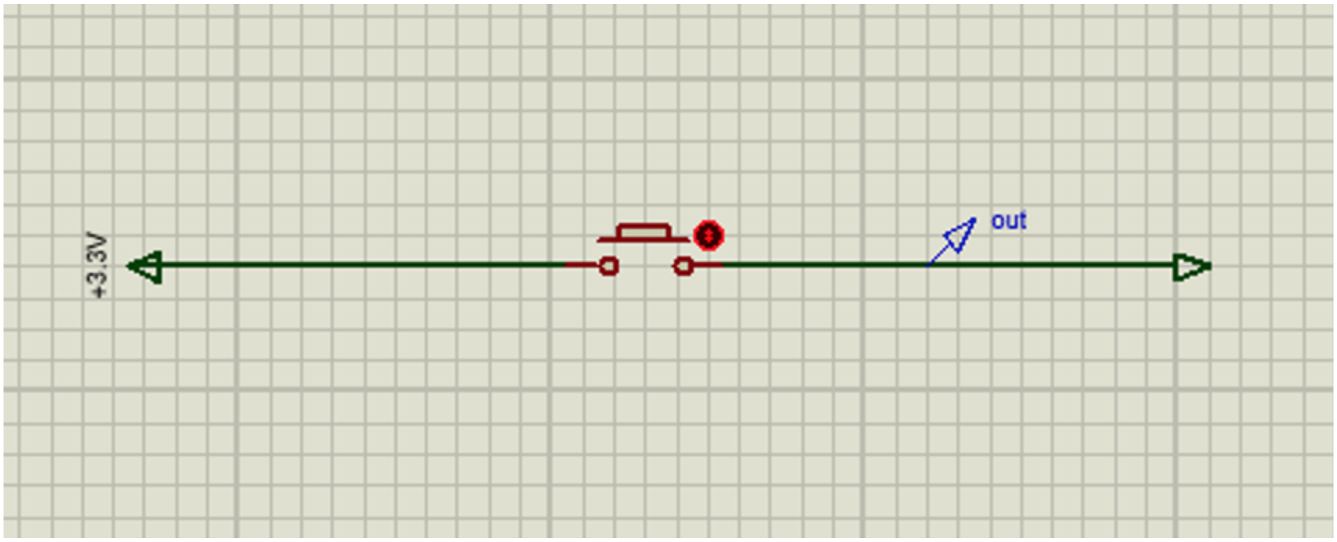

Fig. 21: Push Button Interface Circuit

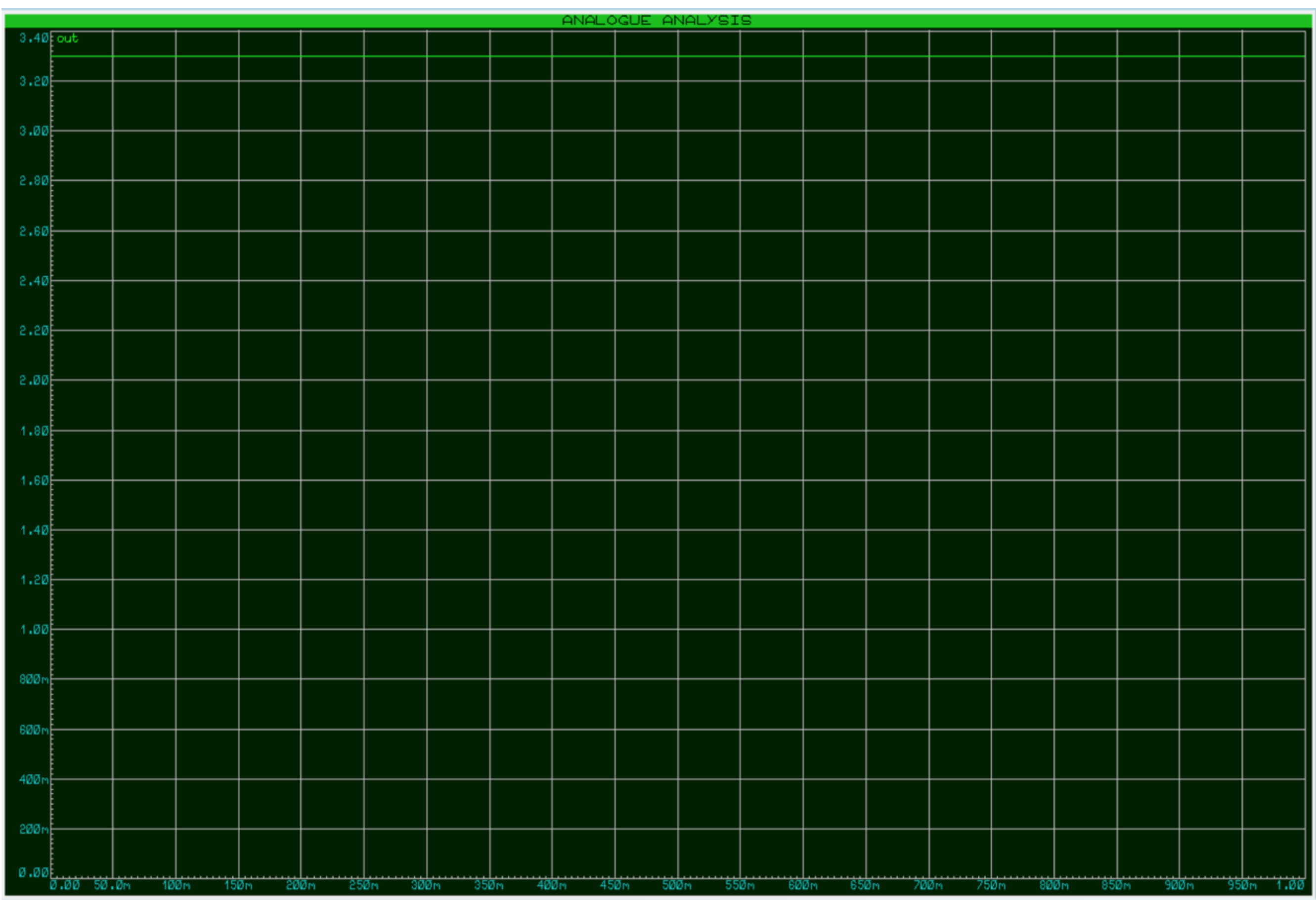

Fig. 22: Analog Analysis showing the output of the Push Button when pressed