Lab 1 – Feasibility Model Phase 1

ECE 298 – S2021

Team Members		Team Number	
	Sophia Liu Kevin Sisjayawan		156

Part 1 – Project Design Requirements

1. **Functional Requirements**
    1. The system must be able to sense movement of objects once they are in a specified range of distance to the door.
    2. The system must be able to open the door once the detected object has entered the specified proximity range and close the door once there are no objects in the given range.
    3. The system must be able to stop the door from travelling once a collision is detected, and once it is no longer detected, must allow for the door to continue at a limited linear rate
    4. The door must open/close with a speed ramp-up and ramp-down rate when starting and stopping movement.
    5. The system must allow the user to specify which mode of operation the door opener will operate in (i.e., locked, setup, and run).
    6. The system must allow customization of outside/inside sense distances, time to remain open, speed ramp-up and ramp-down constants when operating under 'setup' mode.
    7. The system must have a Liquid Crystal Display (LCD) to show the current mode of operation.
    8. The system must have two green Light Emitting Diodes (LED) to show when an object has been detected within the range of the inside and outside proximity sensors.
    9. The system must have a flashing red LED when the door is in motion.
    10. The system must have a flashing yellow LED when a collision with the door is detected.
2. **Non-functional Requirements**
    1. The door closing/opening operations should be time efficient.
    2. Energy losses due to heating and cooling (primarily from hot/cool air coming through the door) should be minimized.
    3. The door system must operate in a safe manner that will not cause harm to the user.
    4. The door motor drive must be power efficient.
3. **Constraint Requirements**
    1. Only parts from the Proteus libraries can be used.
    2. The Printed Circuit Board (PCB) shape must be as per the PCB layout template specified in ECE 298 Lab 4.
    3. The final schematic and PCB design must include suitable decoupling capacitors for power supply voltage filtering.
    4. Voltage level translations (3.3V to 5V, and 5V to 3.3V) must be used for signals between the STM32 Microcontroller Unit (MCU) and 5V devices where needed.

Part 2 – Project Considerations for I/O

Project Sensors and User Inputs

- HCSR04 (Ultrasonic Ranger Module)
  - Description and use: For the outside and inside proximity sensors.
  - Parameters:
    - Max Range: 4m
    - Min Range: 2cm
    - Measuring Angle: 15 degrees
    - Trigger Input Signal: 10uS TTL pulse
    - Working Voltage: 5V
    - Echo Back Pulse Width: every uS / 148 = every inch
  - Connection to MCU: MCU's general purpose input/output ports (GPIO) to output trigger pulse and accept returning echo signal.
- ECE298_DCMOTOR_ENCODER
  - Description and use: To sense and monitor the direction and number of revolutions of the door motor. Note that this is the same device used as the motor to open and close the door.
  - Parameters:
    - Operating Voltage: 4.5V to 5.5V
    - Output: Two digital outputs (quadrature waveform)
  - Connection to MCU: MCU's general purpose input/output ports (GPIO) to accept the two output signals, with a pull-up resistor.
- SPST Push Button
  - Description and use: For the door position limit switches, collision switch, and the user inputs as a back, enter, up, and down button.
  - Parameters:
    - Current Rating: 3A @ 125V AC
    - Initial Contact Resistance: 20 MΩ (max)
  - Connection to MCU: MCU's general purpose input/output ports (GPIO) to accept the output signals, with a voltage source.

Project Actuators and User Outputs

- ECE298_DCMOTOR_ENCODER
  - Description and use: Motor to open and close the door as specified by the system
  - Parameters:
    - Zero load rotations per minute (RPM): 360 RPM
    - Load/max torque ratio: 50%
    - Pulses per revolution: 24 pulses
    - Nominal voltage: 12 V
    - Initial Contact Resistance: 20 MΩ (max)
  - Connection to MCU: MCU’s serial interface using the serial peripheral interface to output to a motor controller integrated circuit that will control the motor
- LM016L
  - Description and use: LCD to display the current mode the door opener is operating under
  - Parameters:
    - Number of columns: 16
    - Number of rows: 2

        - Clock frequency: 250 kHz
    - Connection to MCU: Use the MCU's serial interface to output digital signals to the LCD with the appropriate bit sequence to match the alphanumeric description of the modes
- LED-RED, LED-YELLOW, LED-GREEN
    - Description and use: LEDs to signify when an object is within the inside and outside proximity range (LED-GREEN), when the door is in motion (LED-RED), and when the door has collided with an object (LED-YELLOW)
    - Parameters[1]:
        - Forward Voltage @ 20mA: 2.2V
        - Full Drive Current: 10 mA
        - Breakdown Voltage: 4V
        - Series Resistance: 3 Ohms
        - Minimum on-time to Light: 10 ms
    - Connection to MCU: Use the MCU's GPIO to output a digital signal to an N-channel MOSFET transistor that will 'switch' the LED on or off with a pull up resistor (circuit shown in Part 3)

[1] The essential parameters of the different LEDs are taken from the parts properties, which show that all have the same parameters despite emitting different colors (this may be different in the prototyping phase)

Project MCU Internal Resources

Sensors and User Inputs:

For the proximity distance sensor, a timer/clock within the MCU is needed to calculate the length of the signal received. This directly correlates to how far an object is away from the sensor and so the time is used to verify the distance and determine the corresponding functionality. Also software will be needed to convert from the time the signal is active to a usable distance or have an error handler for distances outside a suitable range. The timer and the ratio of motor revolutions to wheel rotations will both be needed to calculate the RPM of the motor. Software to compare the pulses of the two channels will output the direction of the motor. For the position limit switch and collision switches, they can be linked to an interrupt so that the priority will be to halt door movement for safety. After the collision switch is open, the door speed is limited to 0.5ft/sec. To process user inputs, software is needed to handle the different operating modes and the selections that the user will make using the push buttons. All the parameters specified in the set-up mode (inside and outside sense distance, time to remain open at full position, speed ramp-up and ramp-down constant) will be needed by the MCU to properly run the system.

Actuators and User Outputs:

The ECE298_DCMOTOR_ENCODER will require either the serial peripheral interface or inter-integrated circuit interface in order to properly output signals to the motor controller integrated circuit (connected to the motor itself) from the MCU. Parameters needed for the software to properly output to the motor include the desired speed of the motor (to control how fast the motor opens or closes the door), as well as the motor direction (to control whether the door is opening or closing). Controlling motor speed will most likely require a pulse width modulated input. The LM016L LCD display will also require the SPI to set the bit pattern needed by the LCD inputs. In terms of LCD functionality, the MCU software will require as parameters: the LCD initialization/configuration bits, the current operating mode of the door, and the bit encodings (potentially implemented as a look up table) to translate alphanumeric characters to bits. Lastly, the LEDs (i.e. LED-RED, LED-YELLOW, LED-GREEN) will require a clock to make them flash when turned on. MCU GPIO pins should be set to digital output to act as switches to allow for the LED to be turned on or off. The software should have as parameters: the current door operation mode to determine which LED (i.e. which color) should be flashed, and some sort of data structure to map what mode corresponds to which color.

Part 3 – Device Testing Methodology

Ultrasonic Ranger Module - HCSR04

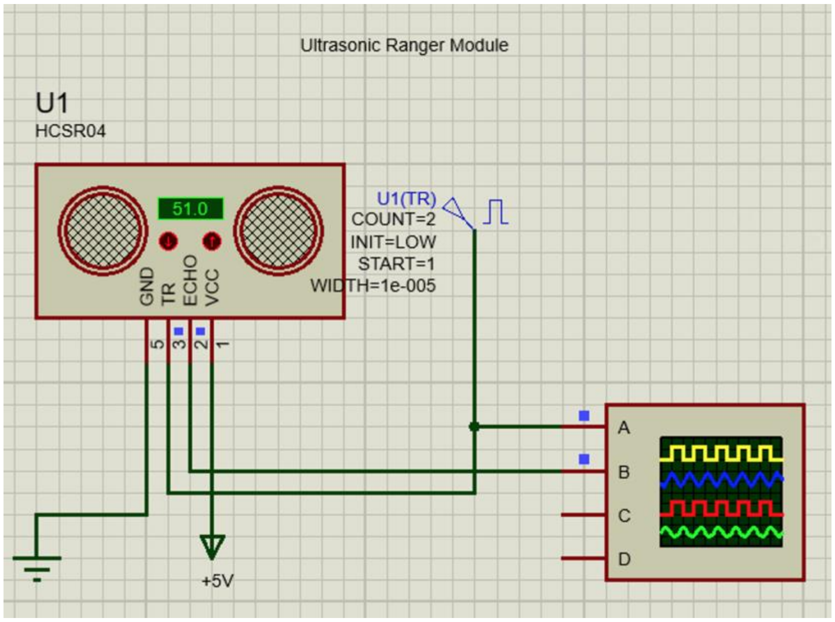

Fig. 1: Ultrasonic Ranger Circuit Schematic

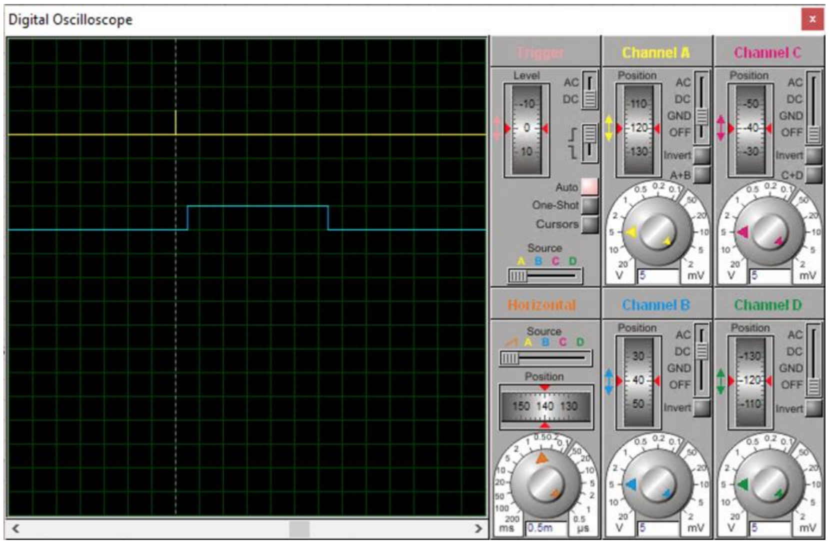

Fig. 2: Scope Output of Ultrasonic Ranger Circuit

To test the ultrasonic ranger module, the device is preset to detect an object at 51 cm (20 inches) and a trigger pulse of width 10 us is inputted to start the ranging. An output echo signal is detected, ensuring that the module is working correctly. To further verify the device's expected output, we can calculate that the width of the output signal should follow the equation: uS / 148 = inch. Since our input is 20 inches, the output width should be 2960 us. Looking at the digital analysis graph, the signal is high for (3230 - 272) us = 2958 us, which agrees with our theoretical calculations. Testing with parameters of 2, 4, 6, 8 feet all produce an expected outcome. This is satisfactory for our project requirements to be used as the inside and outside distance proximity sensors since it can successfully detect objects within the required sense distances and the output is readable and simple enough for the MCU to process. So, when an object is detected within the device's range but not within our system's sense distances, the MCU can decide to ignore the signal until a shorter, more suitable signal is detected.

Fig. 3: Digital Analysis Graph of Ultrasonic Ranger Circuit (1)

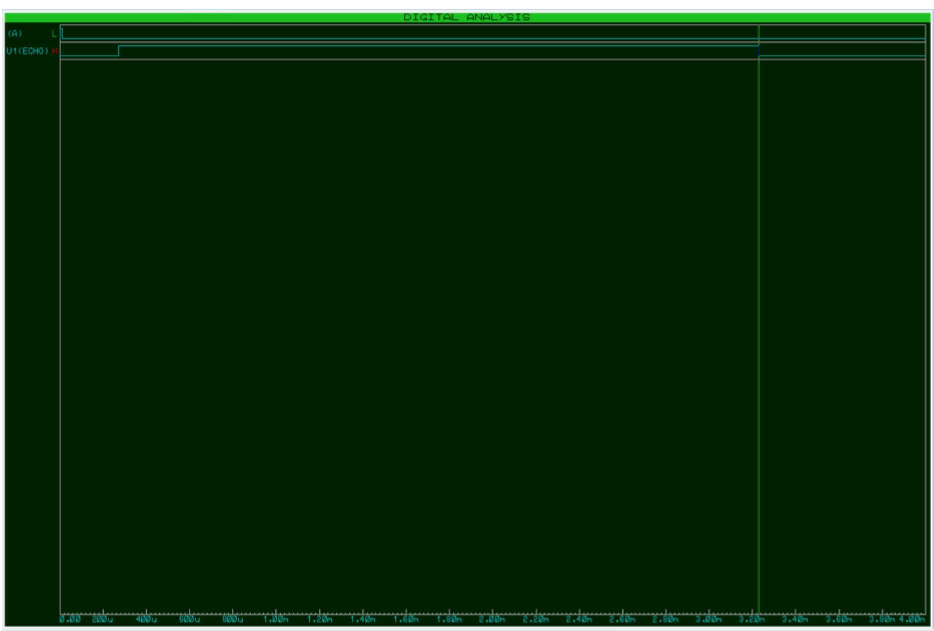

Fig. 4: Digital Analysis Graph of Ultrasonic Ranger Circuit (2)

Push Button - SPST Push Button

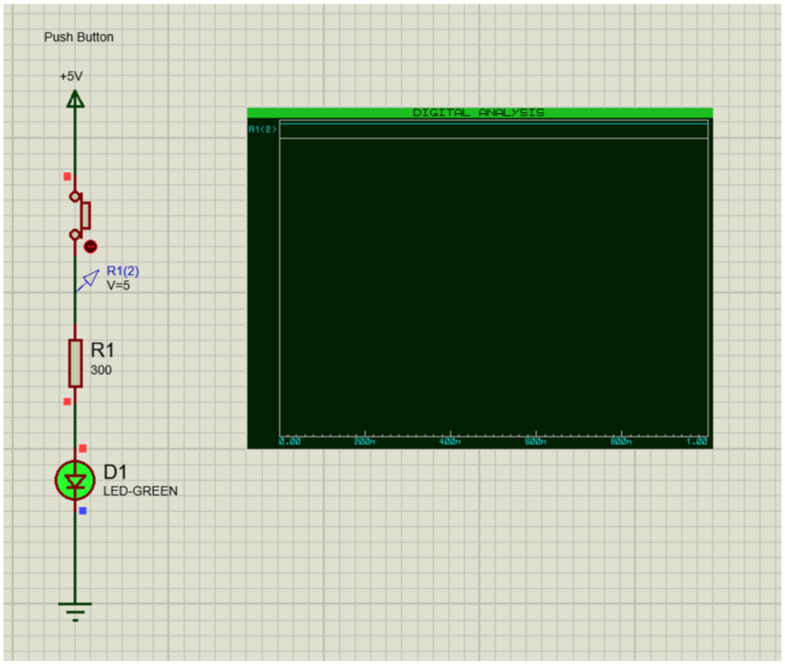

Fig. 5: Push Button Circuit Schematic (on)

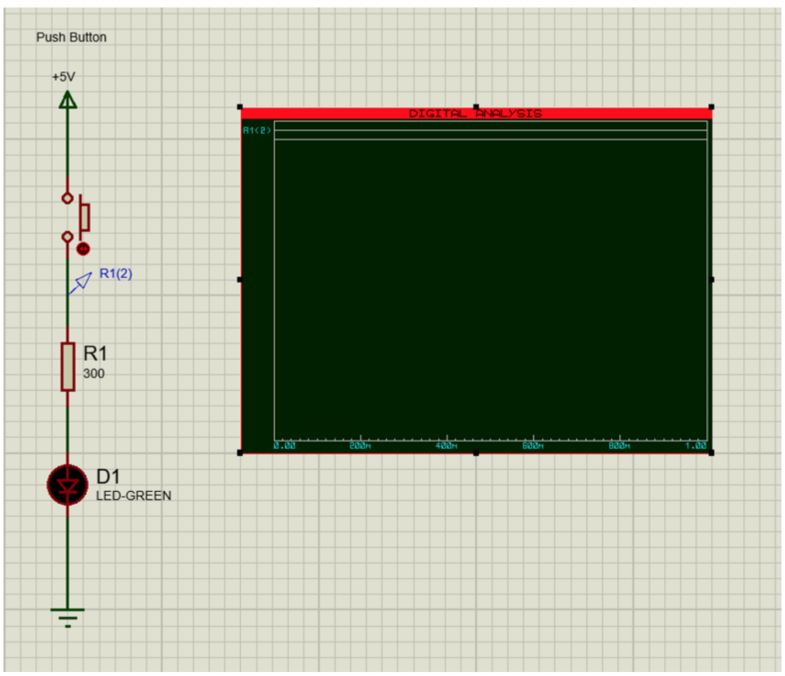

Fig. 6: Push Button Circuit Schematic (off)

Pushing the push button is tested by observing the voltage change when the button is in the down and up position. An LED is also added to show that when the button is not pushed, it is unlit, but when the button is pushed down, it is on. Based on the LED and the digital analysis graph, you can see that the voltage only passes through when the button is pushed down, to form a complete circuit. This push button's behaviour will be useful throughout many aspects of our project. Firstly, for the door limit switches, when the door hits its max and min travel limits, the door will push onto the button, telling the MCU the door needs to stop travelling. Similarly, for the collision switch, any time the button on the door is pushed, an interrupt signal will stop all door movement for the duration of the button press. The push button will also be used to detect user inputs. Four buttons are used to indicate back, enter, up and down so the user can easily navigate through the different types of operating modes and select the parameters they would like to customize for the system.

DC Motor and Encoder - ECE298_DCMOTOR_ENCODER

There are several aspects of the DC Motor and Encoder which need to be tested, and as a result there are multiple test schematics for each respective attribute to be investigated.

The first test is to show that the DC motor is able to be reversed in direction and that the attached encoder will appropriately show this reverse in polarity. Two circuits are made to model this behavior. The first circuit is a simple 1V DC power supply output into the motor, which is grounded. Voltage probes are placed at the Q1, IDX, and Q2 outputs from the encoder and are output into a digital analysis graph. The second circuit is the same as the first, with the DC power supply changed to -1V to reverse polarity. The digital analysis graph shows that the IDX output creates a pulse every 24 revolutions. This verifies the property as set in part 1 of this report that the motor will have 24 pulses per revolution. Furthermore, it means that it would be possible to calculate the number of revolutions the motor has spun for – a parameter that will be required. As a sanity check, one can also verify that M1(Q2) and M2(Q2) have the same output, as they should. The graph shows that when M1(Q1) has a rising edge, M2(Q1) has a falling edge and vice versa. This indicates the reverse in polarity, showing that whichever direction the motor measured by M1(Q1) is spinning, the motor measured by M2(Q1) is spinning the opposite direction. One can also verify this in the circuit schematic that shows that motor 1 is spinning at +15 RPM, whereas motor 2 is spinning at -15 RPM. This not only shows that the motor is reversible, but also that it is possible to detect which direction the motor is turning by comparing Q1 and Q2 rising/falling edges – another parameter that will be needed. Thus, this motor will be viable for the purposes of this project.

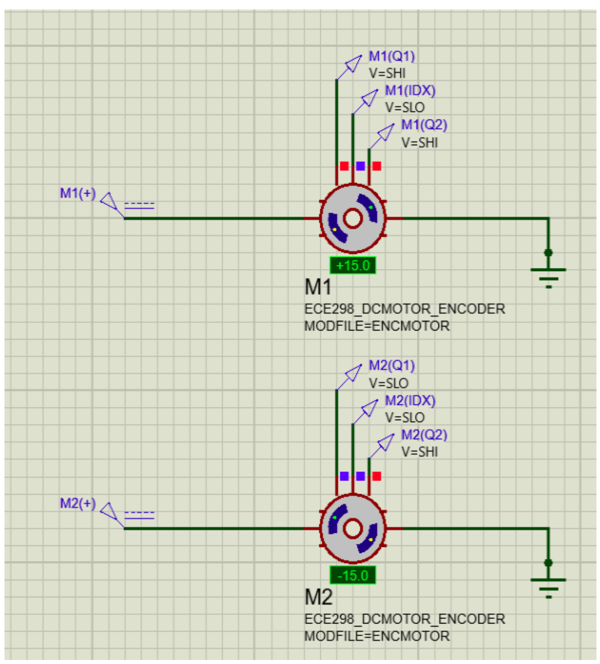

Fig. 7: Motor Direction Testing Schematic

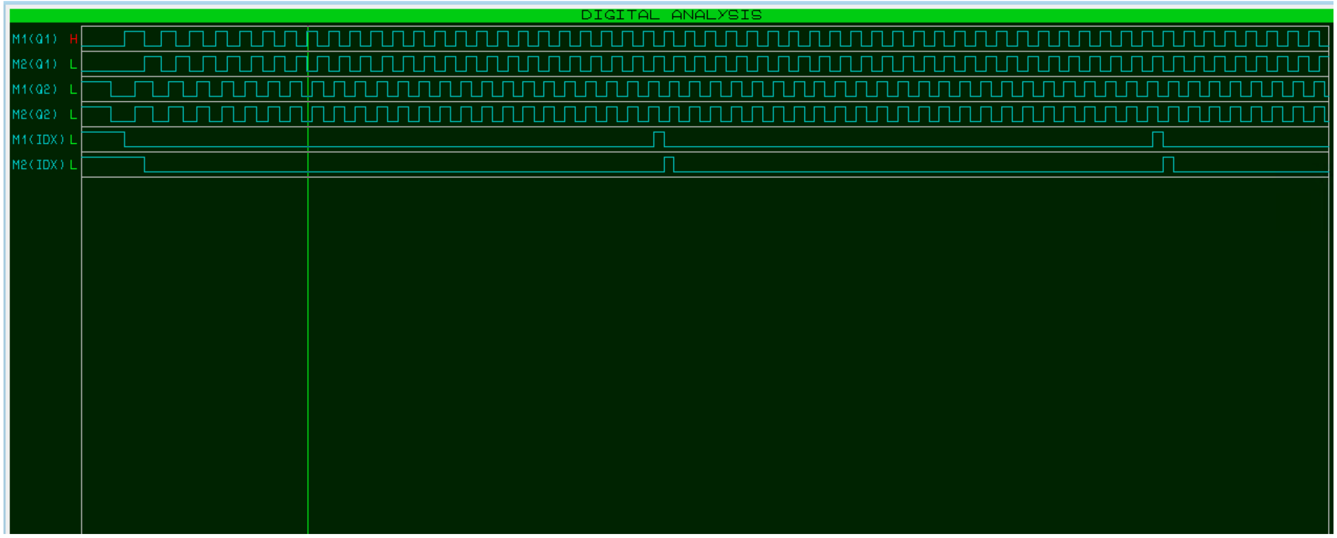

Fig. 8: Motor Direction Testing Output

The next test is to have a rough estimate of the full-power startup inrush current of the motor. The circuit required for this test is a 12V DC input into the motor with an ammeter in series, alongside one voltage probe METER1(-) and one current probe M3(-) that outputs to an analogue analysis graph. Upon running the simulation, one can see that the current jumps immediately to a value of around 1A. As such, when choosing an appropriate motor controller integrated circuit in the future, it is important to consider that it must be able to handle a maximum current of 1A. This is a relatively high current for integrated circuits, and as such will most probably limit the range of selection of motor controllers available for this project. At its maximum RPM, the motor is shown in the analogue analysis to approach a current minimum of around 500mA. As such, this would also be a good estimate of the operational low bound of the motor controller IC. However, given a range from 500mA–1A, this motor will be viable for this project.

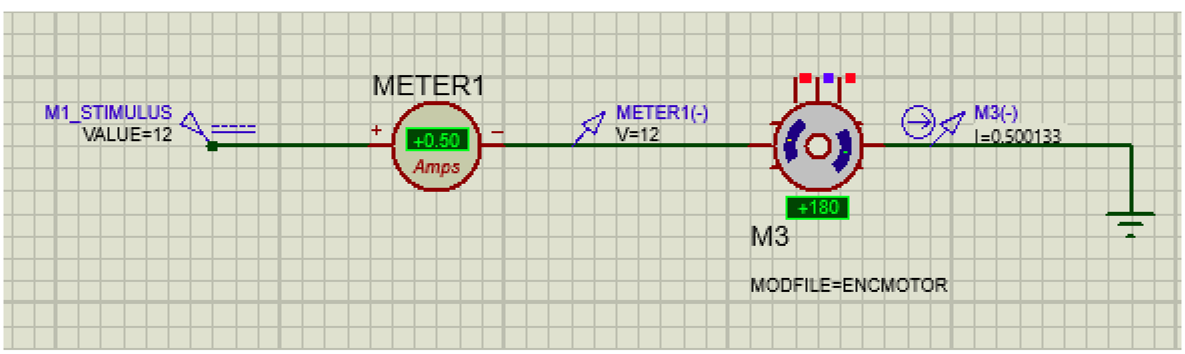

Fig. 9: Motor Inrush Current Test Schematic

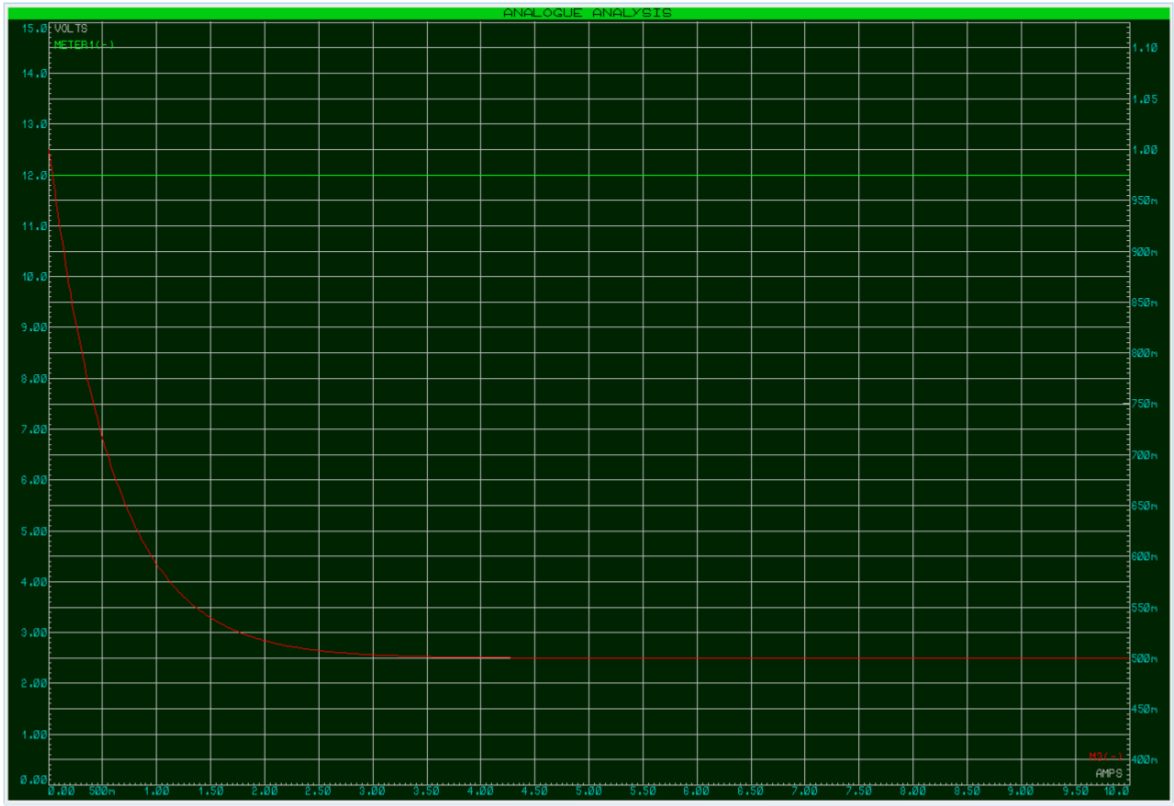

Fig. 10: Motor Inrush Current Test Output

The next test is to see the performance of the motor under battery degradation conditions (i.e. at different input voltages). The motor is connected similarly to the previous test, the only difference being that the input is now modelled as varying voltages at different times. For testing purposes, the voltages are chosen to be 0V, 6V, 10V, 12V, 10V, 6V, 0V in that order. METER2(-) and M4(-) show the motor voltage and current, respectively. A calculation of the total power consumption is beyond the scope of this test, but one thing to note is that at its peak RPM of 180 RPM, the motor has a voltage of 12V, a current of around 525mA, and thus consumes around 6.3W of power. This is a relatively high value, and will be important for considerations when selecting the motor power supply, as it should be able to compensate for that upper limit.

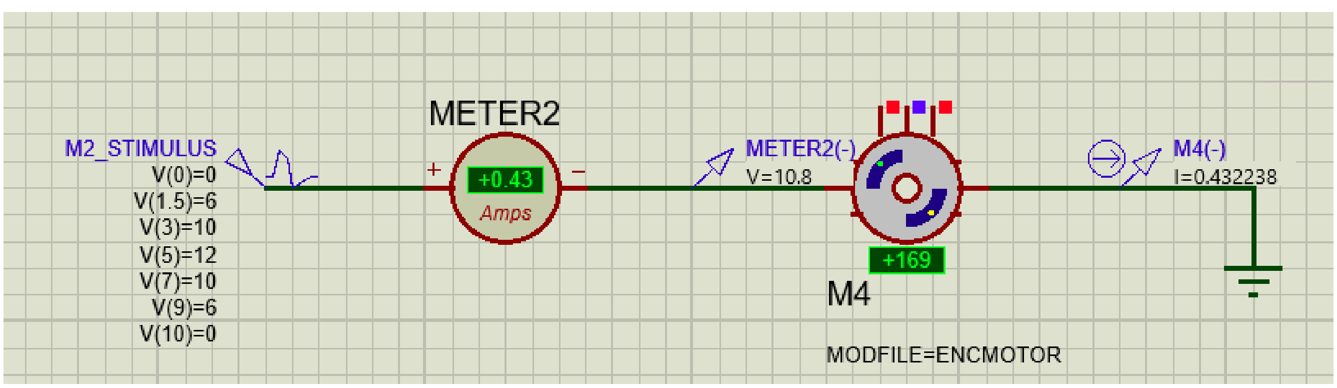

Fig. 11: Current Voltage Curve Motor Schematic

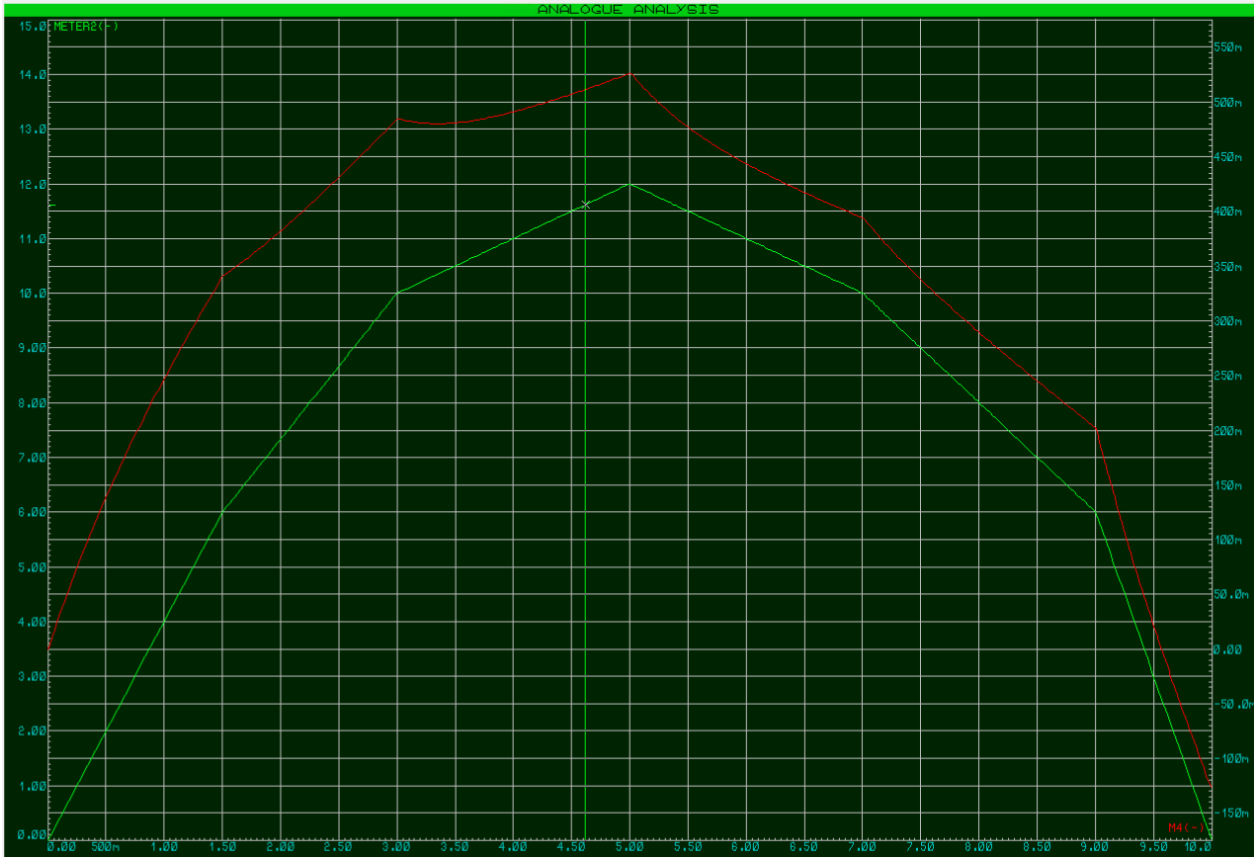

Fig. 12: Current Voltage Curve Motor Output

The last test is to simulate motor performance using a pulse width modulated input, as this best reflects what the actual implementation of the motor controller circuit will look like. The circuit schematic is similar to all tests before, with the only difference being that the input signal is now a user-defined PWM signal in a text file. METER3(-) and M5(-) measure the motor voltage and current respectively and are correspondingly plotted on the analogue analysis graph. The graph shows the current characteristics of the motor as varying widths of the PWM voltages (all of which are set at 12V) are input into the motor. Though it was rather difficult to measure the exact RPM and speed of the motor at the varying PWM inputs, in general, the graph shows sharp increases in current as the motor speed increases. It can also be seen that the decay of the current has a relatively long time constant, resulting in large current modulation as a result of the modulating input. The largest variance seems to come from the second 'step' of the current in the graph, ranging from 250mA to 600mA (i.e. a range of 350mA). This is a rather high range and will be a useful property for consideration in properly dealing with the current across the motor. In the case of this test, the motor is merely grounded and as such has no effect on anything. However, if the motor is connected to an IC or other device (as may be the case in the future), this modulating current may damage other components. But all in all, this test serves to show the performance of the motor in a more realistic setting for future considerations.

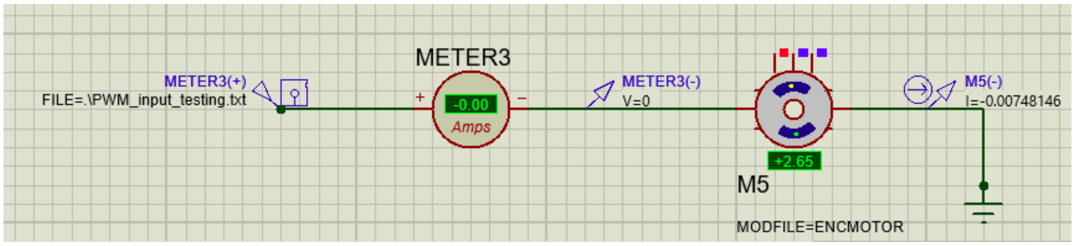

Fig. 13: PWM Motor Schematic

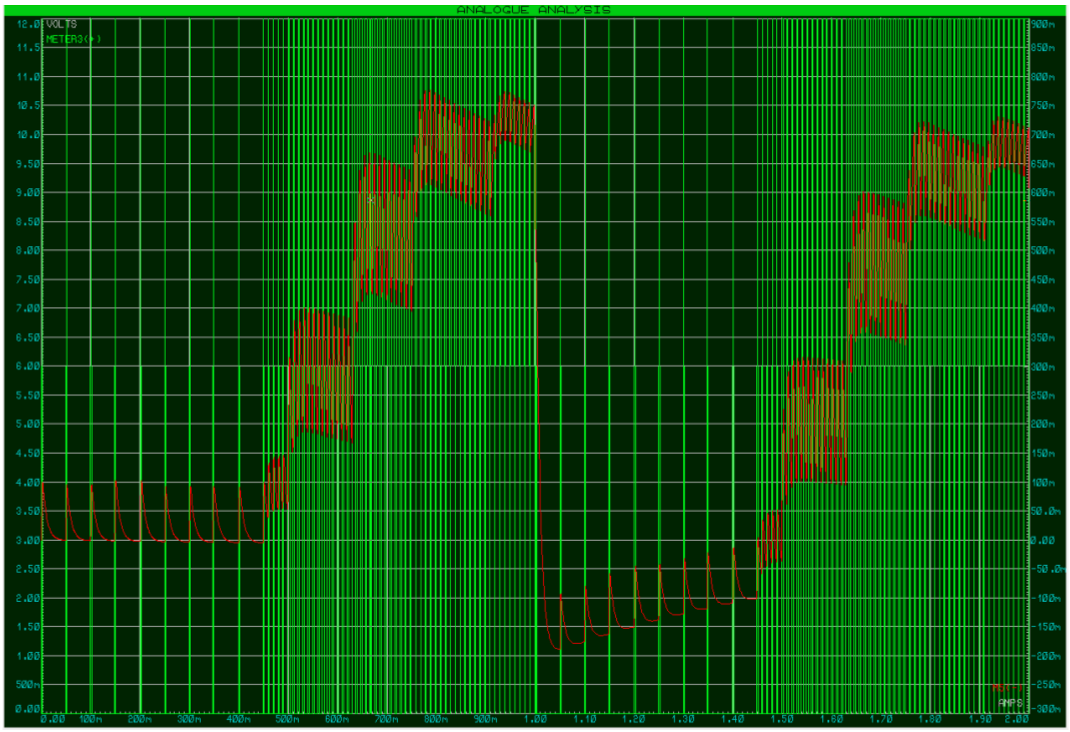

Fig. 14: PWM Motor Output

Liquid Crystal Display - LM016L

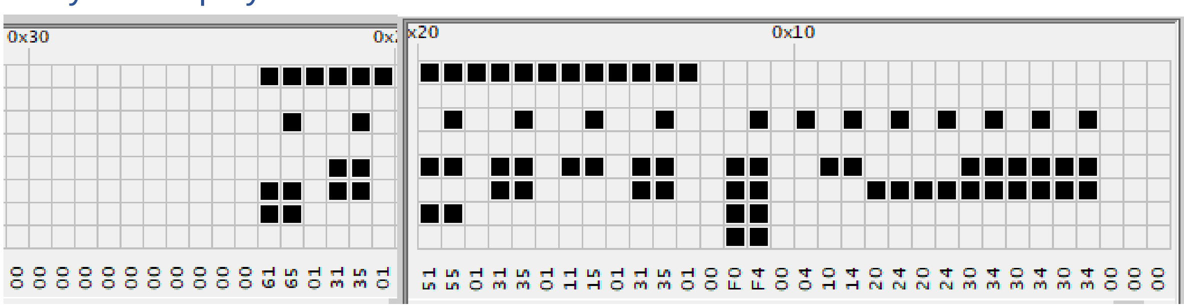

Fig. 15: LCD Test Bit Pattern

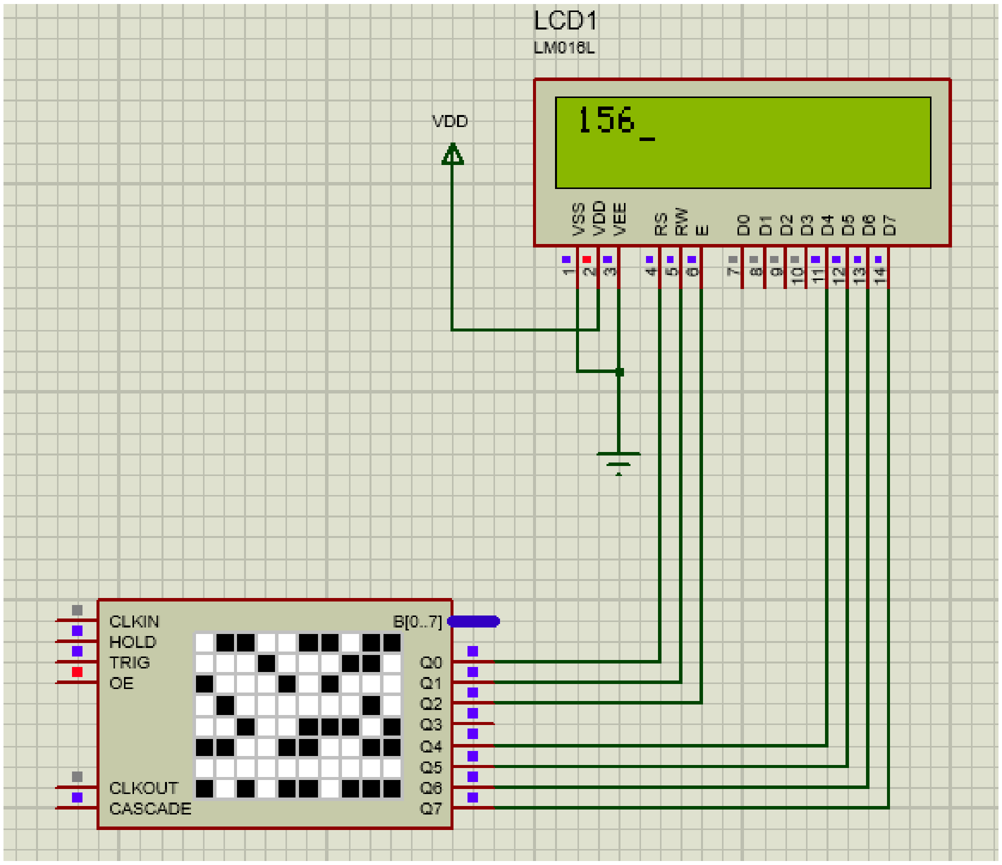

Fig. 16: LCD Test Schematic

To test the LCD, a pattern generator is used to output the bit pattern of the binary ASCII values corresponding to “1”, “5”, and “6” to spell out the group number 156. First, the pattern generator had to initialize the LM016L as described in the device datasheet (the bits can be seen in address 0x13 and below). Afterwards, the bit pattern (i.e. from address 0x15 to 0x26) is sent through the D4–D7 pins of the LCD. This sent 4 bits of the bit pattern at a time. The pattern generator uses an internal clock that runs at 4.403 Hz, and a reset rate at 500 mHz. For testing purposes, this is adequate to verify that the given bit pattern would produce the desired output. However, when prototyping it is important to note that the clock rate will be determined by the MCU and thus may be changed. To verify that the given bit pattern would output the correct result, the simulation once run will output “156” onto the LCD. No scope figures are required since this test produces a visual output that can be easily read. Given that outputting onto the LCD works, this device will be adequate enough for the purposes of our project. Furthermore, a 16x2 display would be large enough to output any required alphanumeric string symbolizing a given mode of operation.

Light Emitting Diodes - LED-RED, LED-GREEN, LED-YELLOW

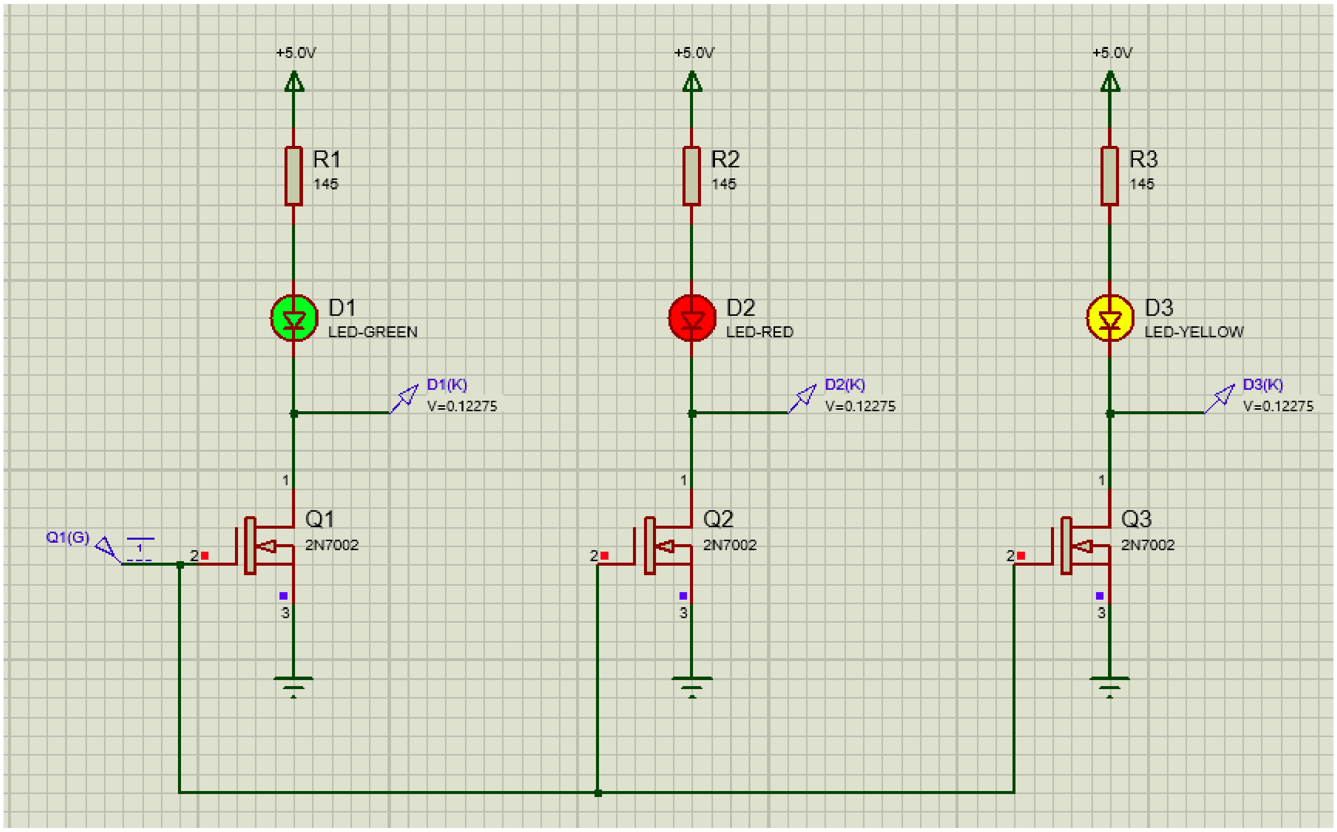

Fig. 17: LED Test Schematic (on)

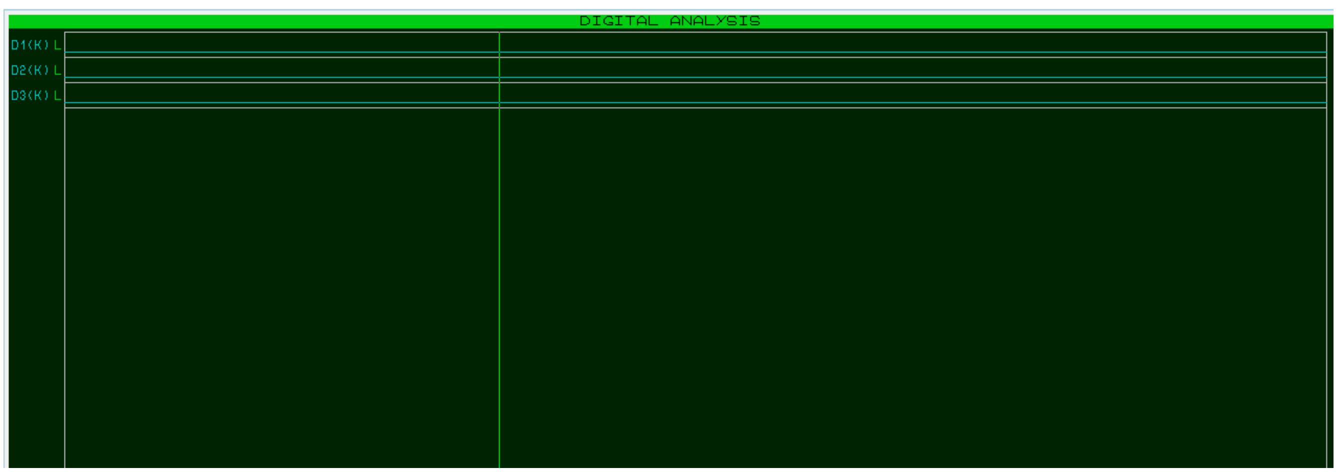

Fig. 18: LED Test Results (on)

To test the LEDs, the circuit of how an LED will connect to an MCU GPIO is modelled in the above circuit schematic by replacing the MCU with a digital signal Q1. This signal is connected to the gate channel of an n-channel Metal Oxide Semiconductor Field Effect Transistor (MOSFET) whose source is grounded and whose drain is connected to an LED powered with a 145 Ohm pull-up resistor and 5V DC power supply. Effectively, the n-channel MOSFET acts as a switch, where a high signal input into the gate will allow current to flow through the transistor, and thus also through the LED. This is reflected by the voltage drop measured at D1(K), D2(K), and D3(K) as shown both in the circuit schematic as well as the digital analysis graph. The voltage at those points is near to 0V, indicating a voltage drop across the LED and resistor of around 5V in total (which corresponds to the power supply). This shows that with a high digital input signal, the current flows properly through the LED and thus turns on (shown by the colored background of the LED symbols). Note that for testing purposes, all MOSFETS share the same input gate signal Q1. But in reality, each MOSFET will be connected to their individual MCU GPIO pins. Given that the LED functions as expected, it will be viable for the project. Furthermore, the use of a transistor as a 'switch' will reduce complications of having to convert MCU GPIO signals to valid input for the LED, given that the LED is powered by the DC supply of 5V.

Below is an example of a low signal being output into the transistors, symbolizing the 'off' state of the LEDs. With a low signal, no current is allowed to flow through the transistor to ground, and thus there is no observed voltage drop across the LED. This is reflected by the measurements at D1(K), D2(K), and D3(K) in both the circuit schematic as well as the digital analysis graph which all show 5V. The lack of voltage drop across the resistor and LED indicate that there is no current flowing through the circuit and thus the LED is turned off (as shown by the black colored background of the LED symbol). This verifies that the LED is also capable of being turned off using the same setup as above, and thus will be usable for the purposes of this project.

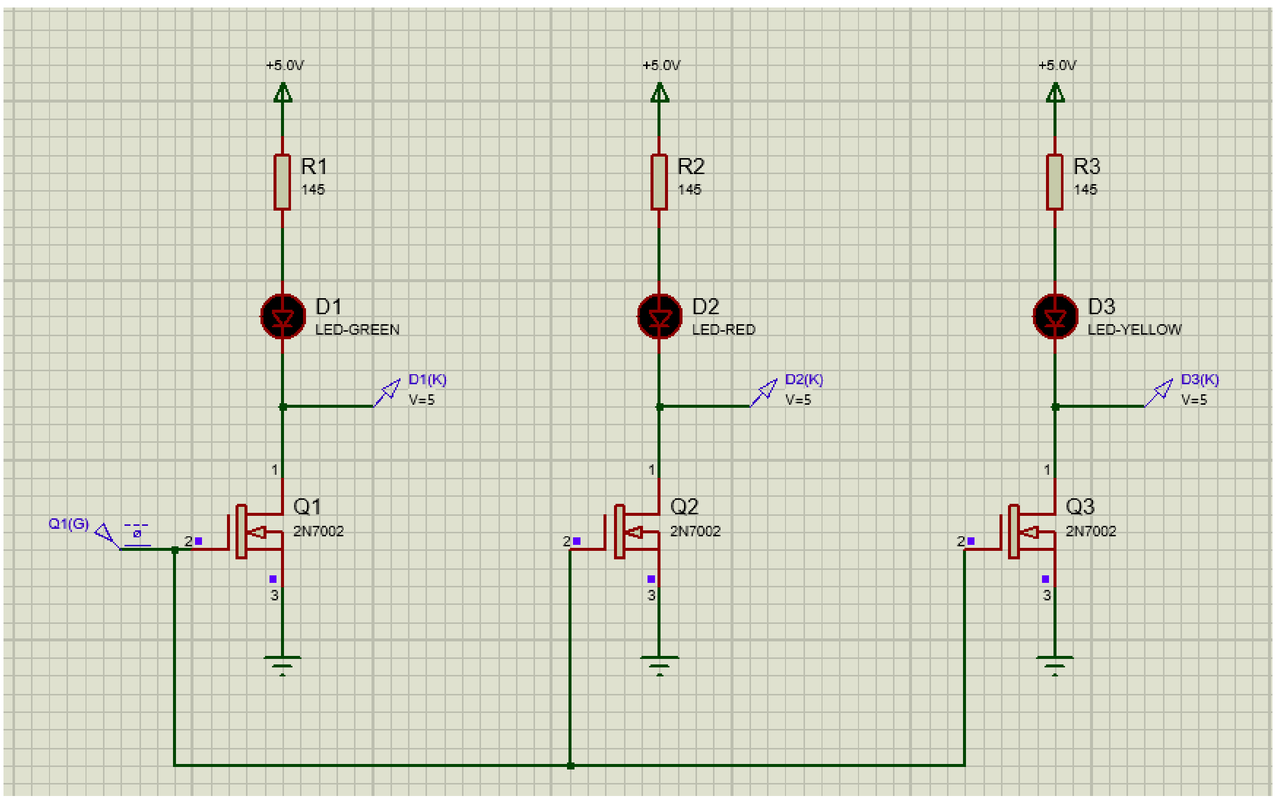

Fig. 19: LED Test Schematic (off)

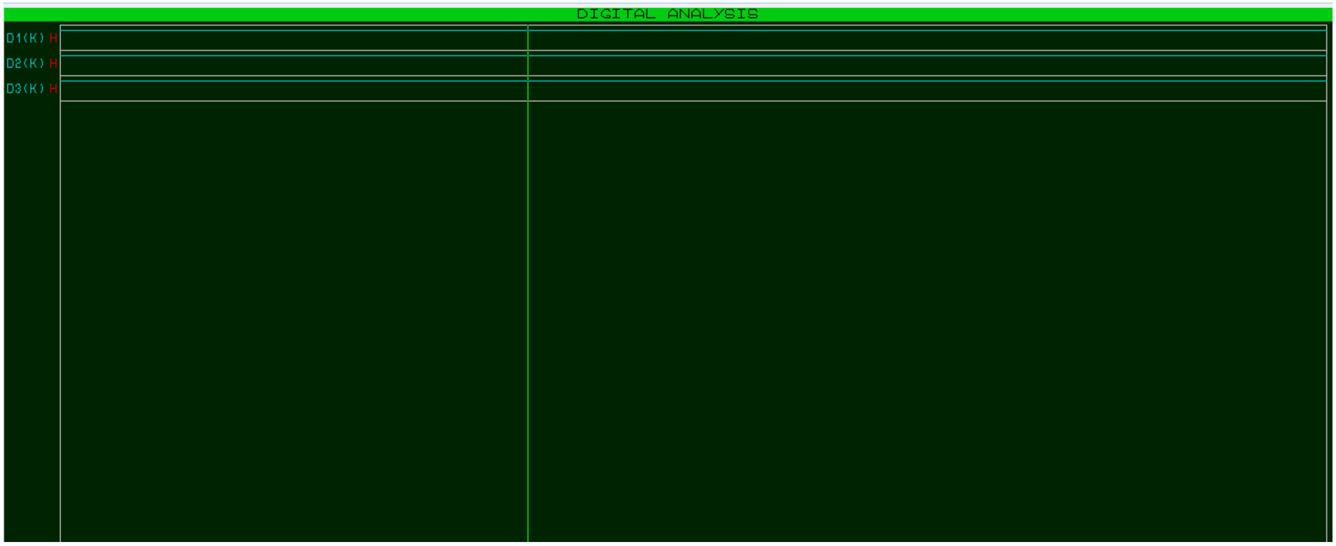

Fig. 20: LED Test Results (off)

Part 4 – System-Level Design

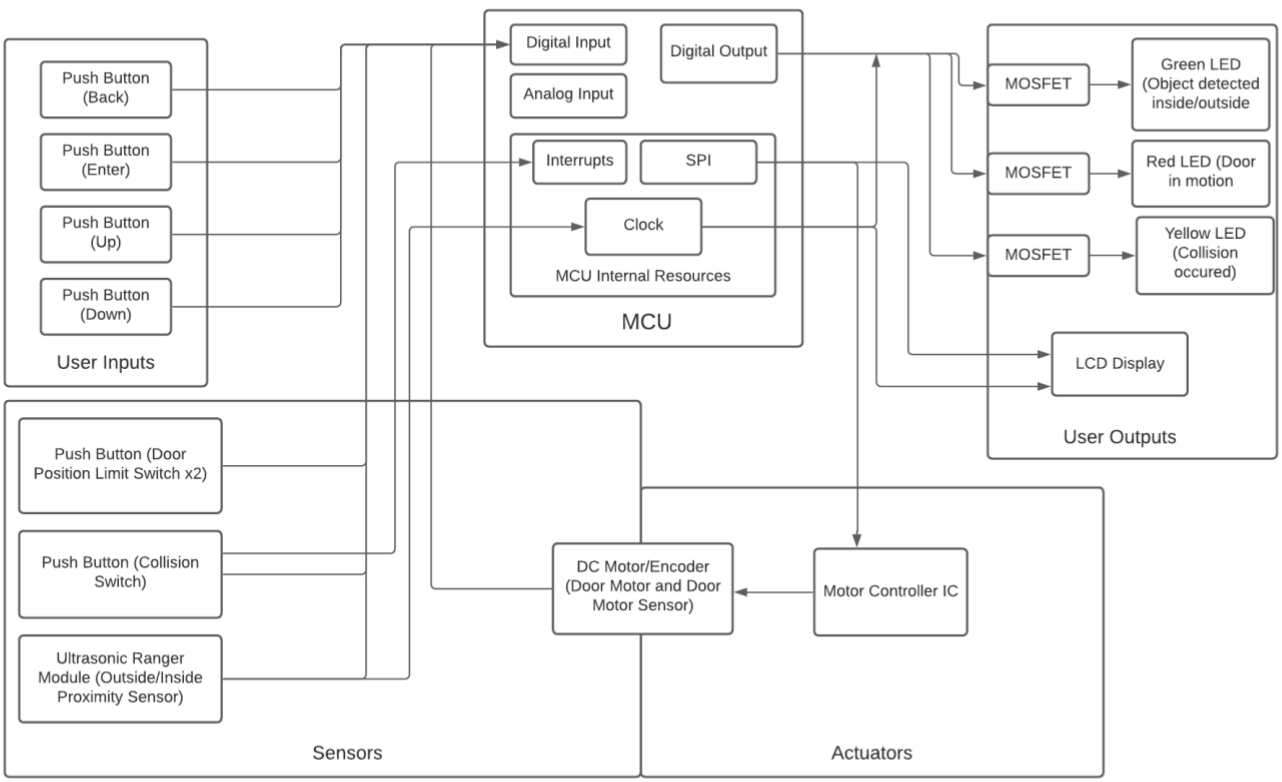

Fig. 21: High Level Design