Lab 1 – Feasibility Model Phase 1

ECE 298 – S2021

Team Members	Kehan Xing (k5xing@uwaterloo.ca) Yiru Wu (y645wu@uwaterloo.ca)	Team Number	50

Part 1 – Project Design Requirements

1. **Functional Requirements**
   A. Automatically open the doorway to 8 feet wide when detecting a customer

   B. Stop the door movement within 0.2 seconds after the collision switch is closed

   C. Limit the door speed to 0.5 ft/sec when the collision switch opens again until the door is closed

2. **Non-functional Requirements**
   A. Safety: The automatic door needs to ensure the safety of users during use.

   B. Power efficiency: The automatic door needs to ensure the power effect and reduces losses.

   C. Durability: The automatic door needs to guarantee the quality of use within a certain period of time.

3. **Constraint Requirements**
   A. Maximum cost of $150 to develop the project

   B. Maximum weight of 2 kg for the hardware components

   C. Maximum PCB size of 9cm x 15cm

Part 2 – Project Considerations for I/O

Project Sensors and User Inputs

- **Outside/inside Distance Proximity Sensor**: The sensor will react to the trigger and output the reaction pulse signal, we could use an opamp to scale the sensor reaction pulse signal into the MCU digital input range.
- **Door Position Limit Switches(2)**: The output of the switch can be regarded as the presence or absence of current, so we could use a signal converter to transform the current into a voltage range for the MCU, and then use the internal ADC to get the digital input.

- **Motor Sensor**: The motor sensor will output continuous digital signals, so we could use an opamp to scale the pulse signal output into the MCU digital input range.
- **Collision Switch**: The output of the switch can be regarded as the presence or absence of current, so we could use a signal converter to transform the current into a voltage range for the MCU, and then use the internal ADC to get the digital input
- **Keypad**: The pins on the MCU will receive a voltage input when a key is pressed, so we use an opamp to scale the keypad output voltage to the MCU input pin voltage range, and then use the ADC to convert the input to digital input.

Project Actuators and User Outputs

- **LCD Display**: Use MCU's parallel interface to input signals to display characters.
- **LED (4)**: Use a transistor so the MCU can switch the high-current device on and off.
- **Door Motor**: Use the internal PWM to control the motor
- **Motor Controller IC**: Use the MCU's serial interface to control a motor controller IC.

Project MCU Internal Resources

a. Internal Resources

- **ADC:** The ADC will be used to convert Analog input to Digital input. For example, the input we receive from a switch will be a voltage of 0V or 3.3V, and the ADC will convert it to 0 or 1.
- **Interrupt:** The motor needs to be stopped once the collision switch or the door position limit switch is closed, and that requires an interrupt module to handle.
- **GPIO:** As we listed above, there are many user inputs and outputs as well as sensors and actuators in this project. They need GPIO functions to read or write.
- **Timer:** With the timer and the motor sensor, we can calculate the speed and the acceleration of the motor.
- **PWM:** Pulse-width modulation is needed to control the motor.

b. Software parameters

- **Duration:** We need a parameter to read the timer and use it to calculate speed and acceleration.
- **Conversion factor:** There could be many conversion factors, such as conversion between different distance units.

- **Sensor status:** We need parameters to read sensor status and use the signals we receive to control the motor.
- **Motorcycle:** There should be a counter to track how many cycles the motor has spinned in order to calculate the distance.
- **Keypad input:** Keypad inputs including selected mode and parameters in the setup mode need software parameters to read them.

Part 3 – Device Testing Methodology

Device 1 – ECE_GEN_PBUTTON (Switch)

We chose this pushbutton switch to represent the Door Position Limit switches and the Collision Switch. To test the switch, a 3.3V input voltage and a voltage scope are used. One end of the switch is connected to the input voltage, and the other end is connected to the scope. We expect to see the scope show 3.3V when the switch is closed and show 0V when it's open. The schematic in Figures 1&2 with a pull-down resistor R shows our idea.

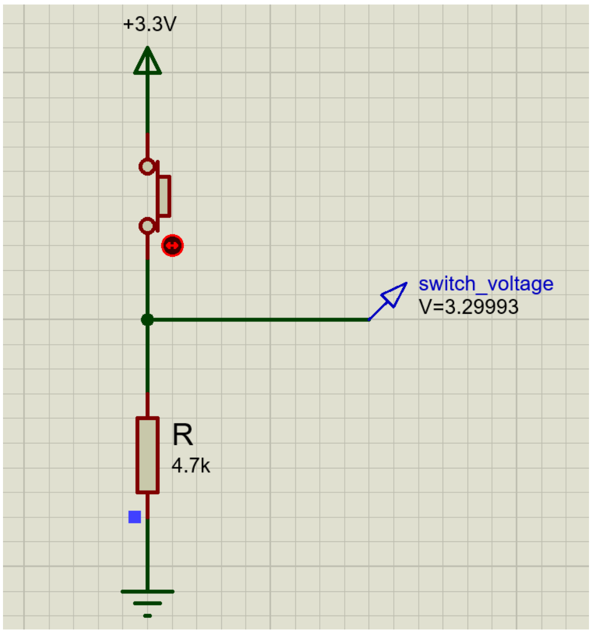

Figure 1: The schematic of ECE_GEN_PBUTTON -- Switch closed

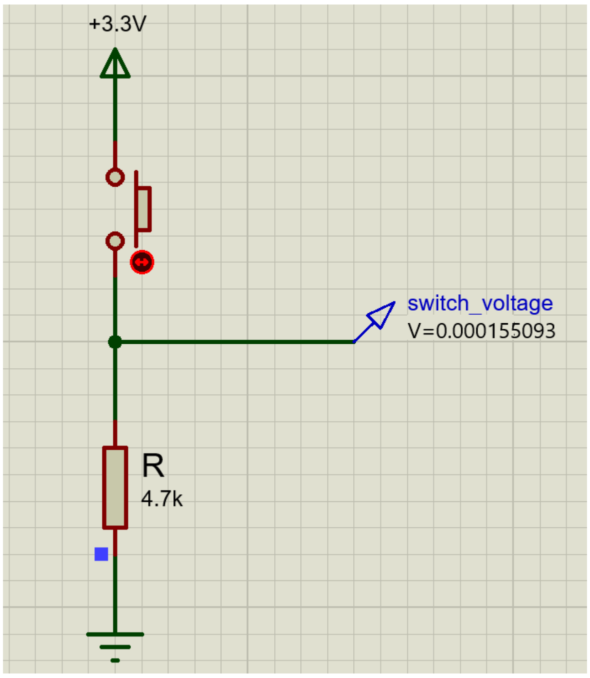

Figure 2: The schematic of ECE_GEN_PBUTTON -- Switch open

Verification: The switch works as expected. When the switch is closed, the voltage scope named switch_voltage shows a voltage value very close to 3.3V. Also, when the switch is open, the voltage scope shows a value close to 0V. This means the logic level is high when the switch is closed and is low when the switch is open.

Validation: As claimed in the verification section, a voltage of 5V indicates a high logic level, and a voltage of 0V indicates a low logic level. When using the switch in our project, we want the voltage scope to be replaced by an input pin on the MCU. In this case, the pin will read a high state when the switch is closed and a low state when the switch is open with the internal ADC, so this will work properly in our project.

Device 2 – HCSR04 (Distance Proximity Sensor)

This ultrasonic ranging module is chosen for the Outside/Inside Distance Proximity Sensors. To test this sensor, we used a pulse voltage for its trigger input and an oscilloscope to show the trigger input and receive the echo pulse output. We expect to see both the signals on the oscilloscope.

Figure 3 and Figure 4 show the schematic and the signal displayed by the oscilloscope with distance = 50cm, and Figure 5 and 6 show those with distance = 200cm.

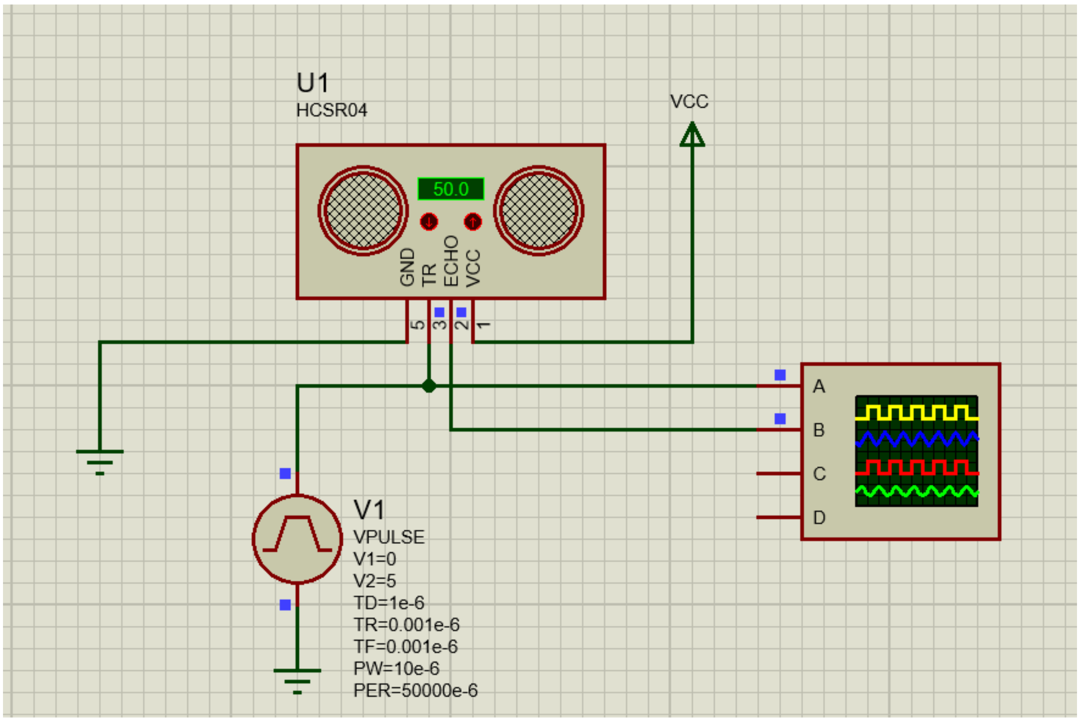

Figure 3: The schematic of HCSR04 ultrasonic ranging module with distance = 50cm

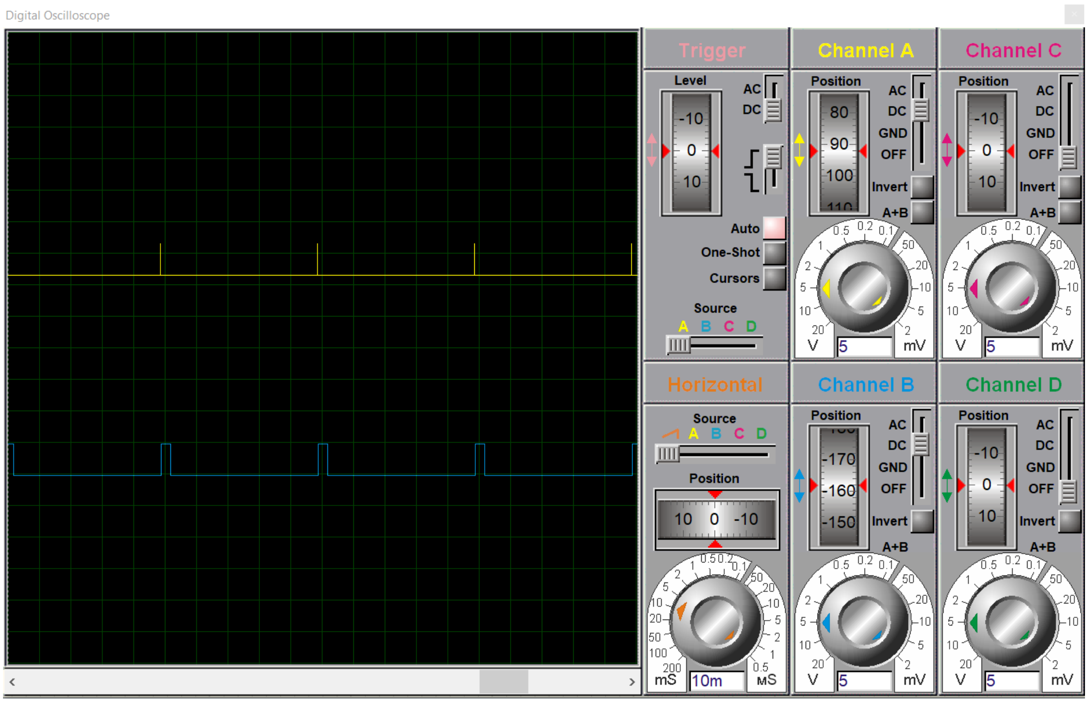

Figure4: The signals displayed by the oscilloscope with distance = 50cm (Yellow: trigger input, Blue: echo output)

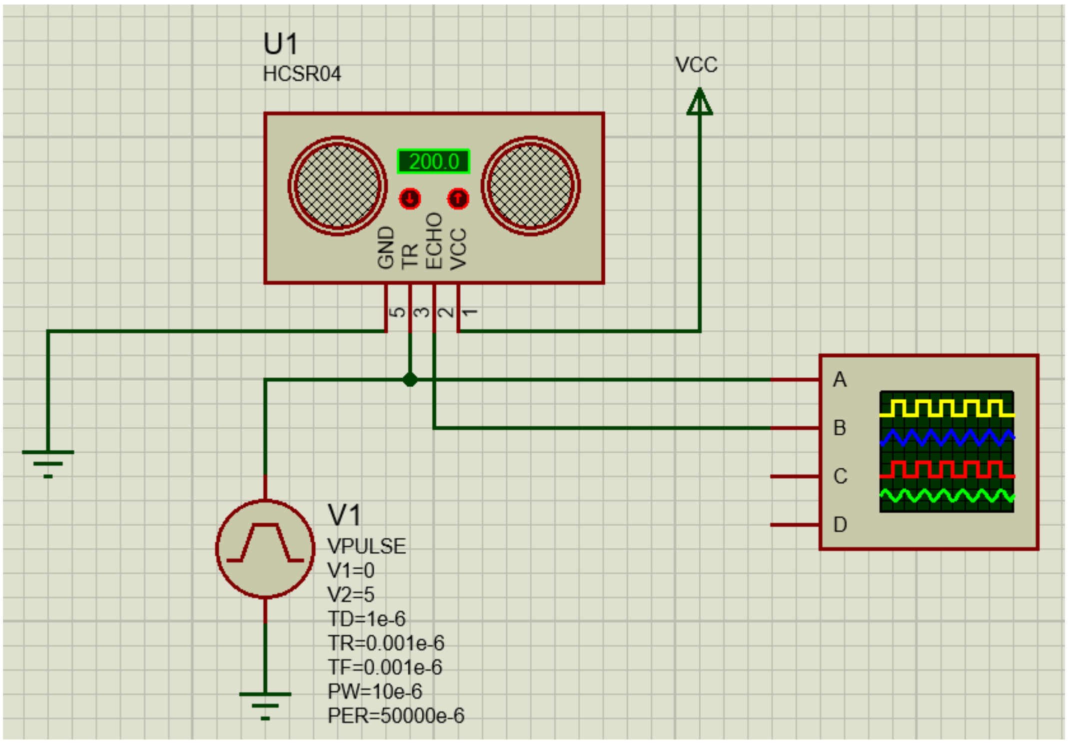

Figure 5: The schematic of HCSR04 ultrasonic ranging module with distance = 50cm

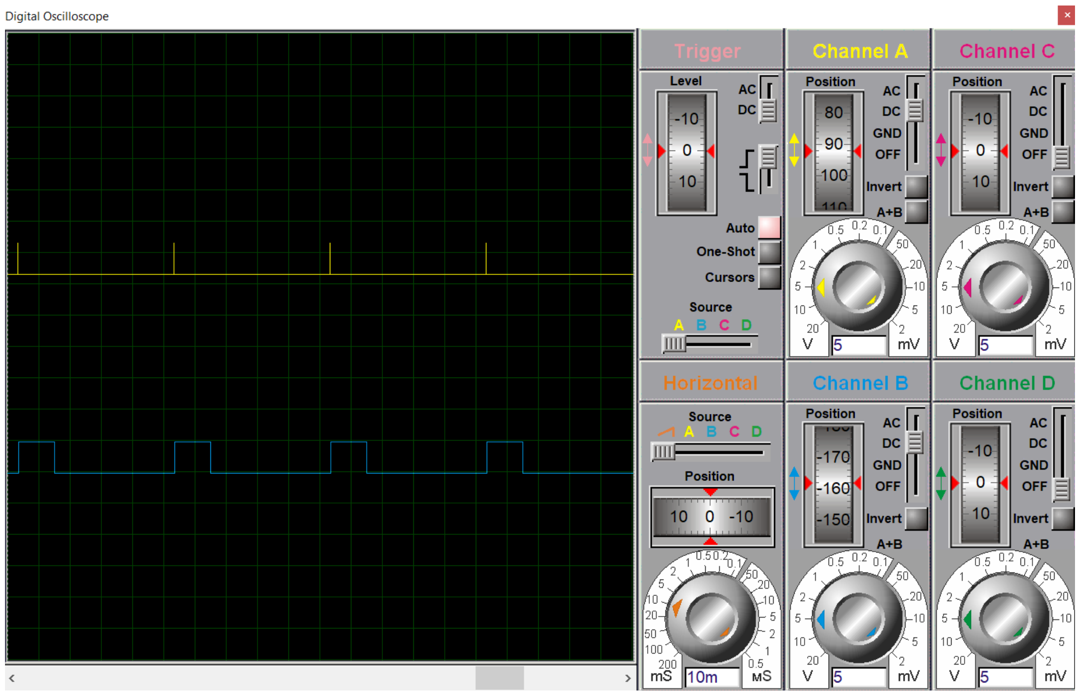

Figure 6: The signals displayed by the oscilloscope with distance = 200cm (Yellow: trigger input, Blue: echo output)

Verification: The ultrasound sensor works as expected. It can take the input signal and reflect the trigger input on the output signal. Also, the pulse width of the echo back signal is wider with distance = 200cm, and that matches the description on its datasheet.

Validation: The ultrasonic sensor is used to measure the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. This is exactly the functionality we want for the distance proximity sensor. Also, the detecting range of this ultrasonic ranging module is 2cm - 500cm, and the required range of our project is 60cm - 240cm. Therefore, this ultrasonic sensor can be used in our project.

Device 3 – KEYPAD - PHONE (Keypad)

The keypad - phone is chosen to be used in Setup Mode. To test this keypad, we selected four DC power supplies each providing different voltage. Then using three DC voltmeters to test the voltage changes when the keys were pushed. Figure 7, figure 8, figure 9 and figure 10 show the schematic and the voltage change.

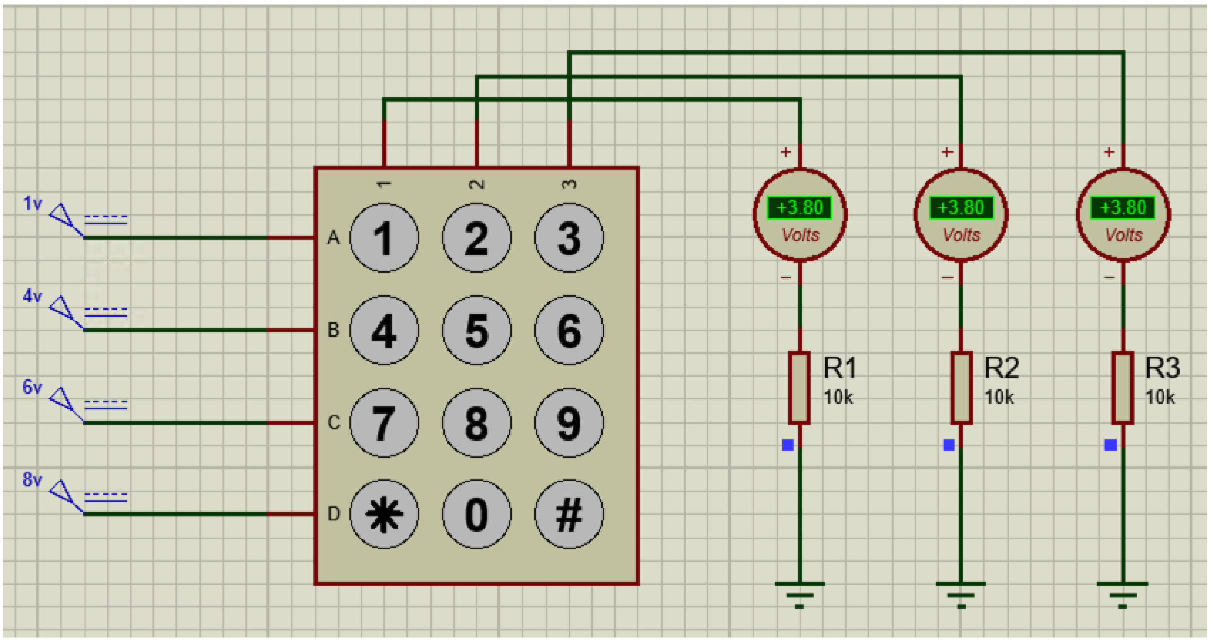

Figure 7: The schematic of KEYPAD module (initial state)

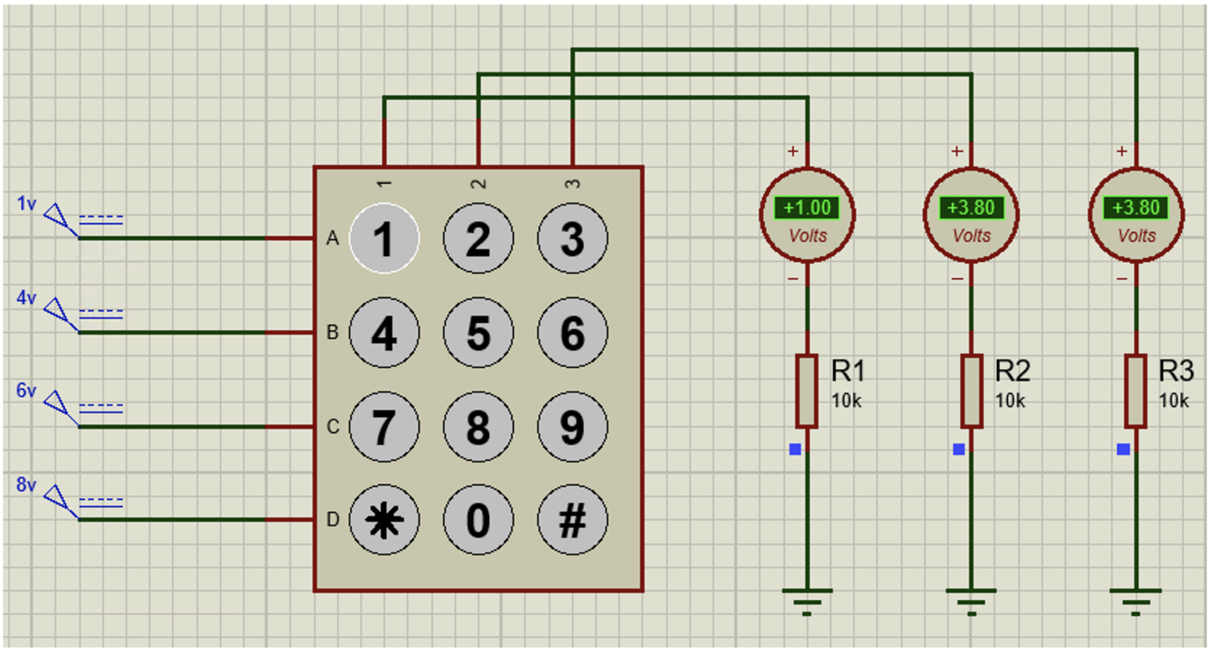

Figure 8: The schematic of KEYPAD module. When the key "1" (position A1) was pushed, the volts on the first voltmeter changed from 3.8v to 1v, which equal the voltage connected to pin A.

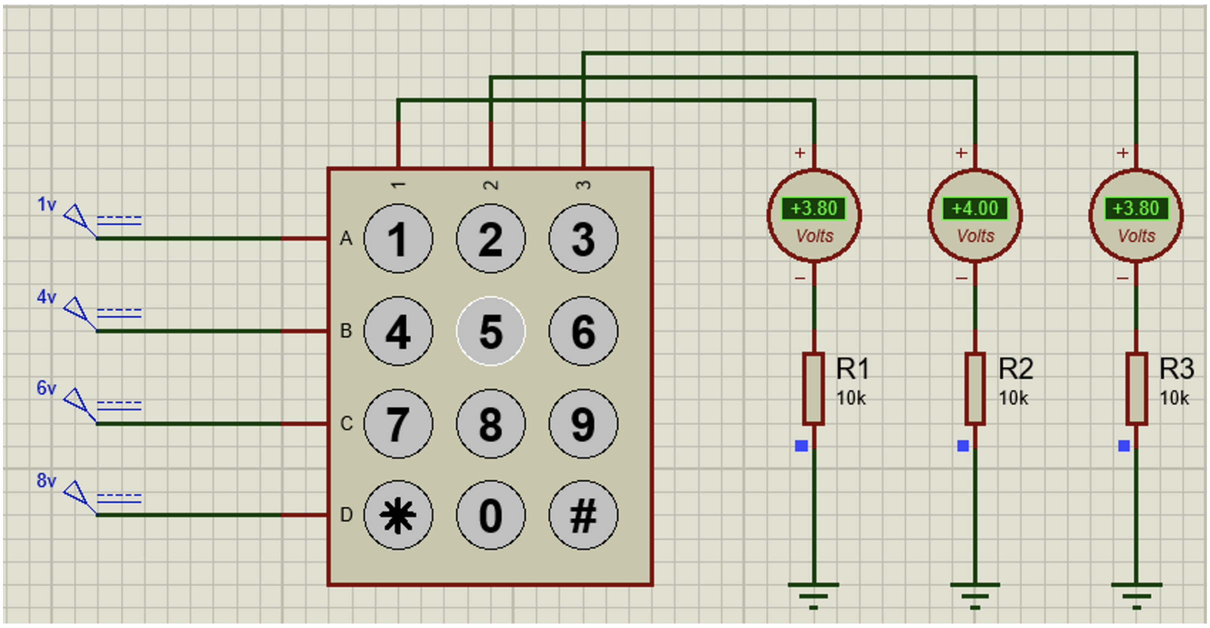

Figure 9: The schematic of KEYPAD module. When the key "5" (position B2) was pushed, the volts on the second voltmeter changed from 3.8v to 4v, which equal the voltage connected to pin B.

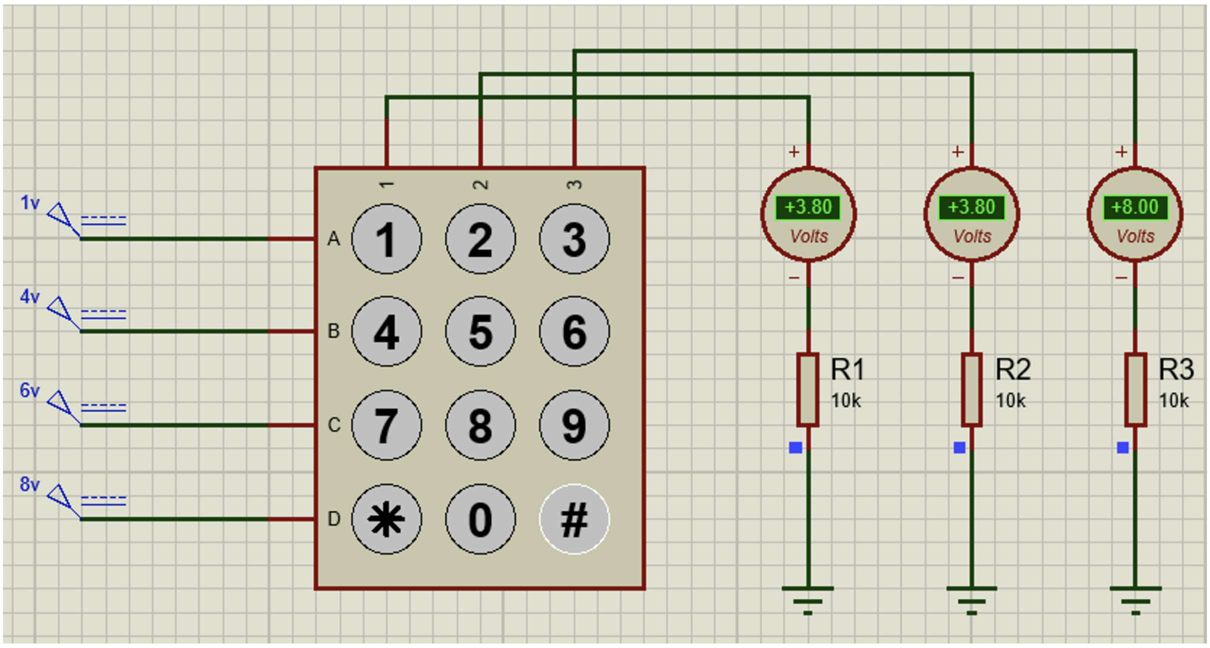

Figure 10: The schematic of KEYPAD module. When the key "#" (position D3) was pushed, the volts on the third voltmeter changed from 3.8v to 8v, which equal the voltage connected to pin D.

Verification: The keypad works as expected. When the key located at (x,y) be pressed, the pin x and the pin y are connected, so the voltage of pin y equals the voltage of pin x, as measured by voltmeter. Detailed information of all the cases is described under the figures.

Validation: When using the keypad in our project, the total 7 pins of the keypad will be connected to the MCU. Then we could detect which key is pressed by checking which pins are connected. Also, the user input required in our project are just numbers, so this keypad can satisfy the requirement.

Device 4 – ECE298_FAST_DCMOTOR_ENCODER (Motor/Motor Sensor)

This model is chosen to be used as motor and motor sensors. To test this motor, we need a DC voltage to supply the power and an oscilloscope to show the digital signal detected from the motor. Figure 11, figure 12 and figure 13 show the schematic and the signal displayed by the oscilloscope using a pulse voltage input. Figure 14, figure 15 and figure 16 shows that using a DC voltage.

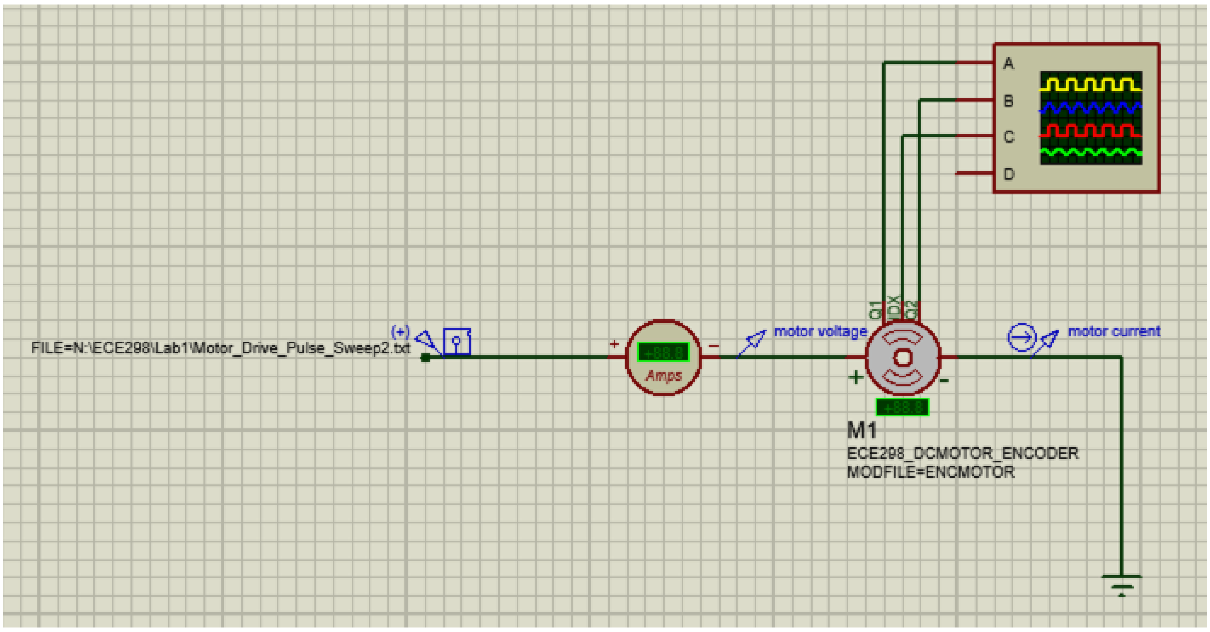

Figure 11: The schematic of ECE298_FAST_DCMOTOR_ENCODER using pulse voltage

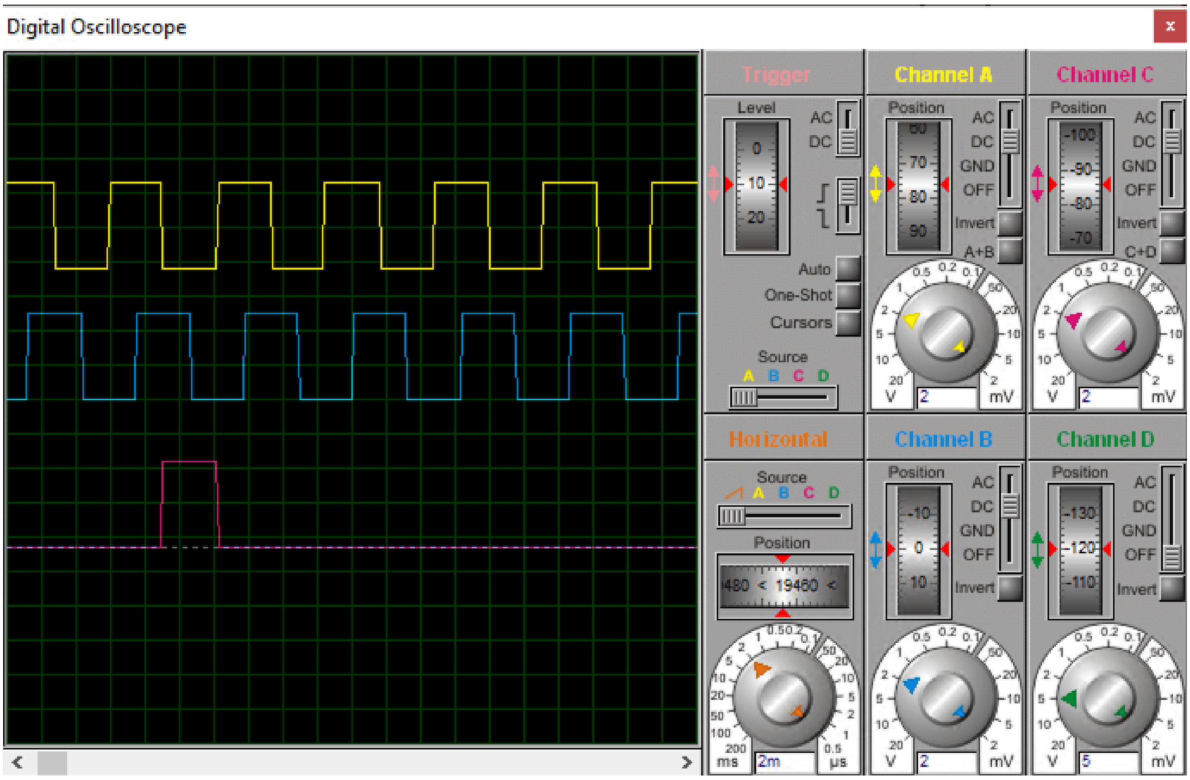

Figure 12: The signals output displayed by the oscilloscope using pulse voltage(Yellow: Q1, Blue: Q2, Red: IDX)

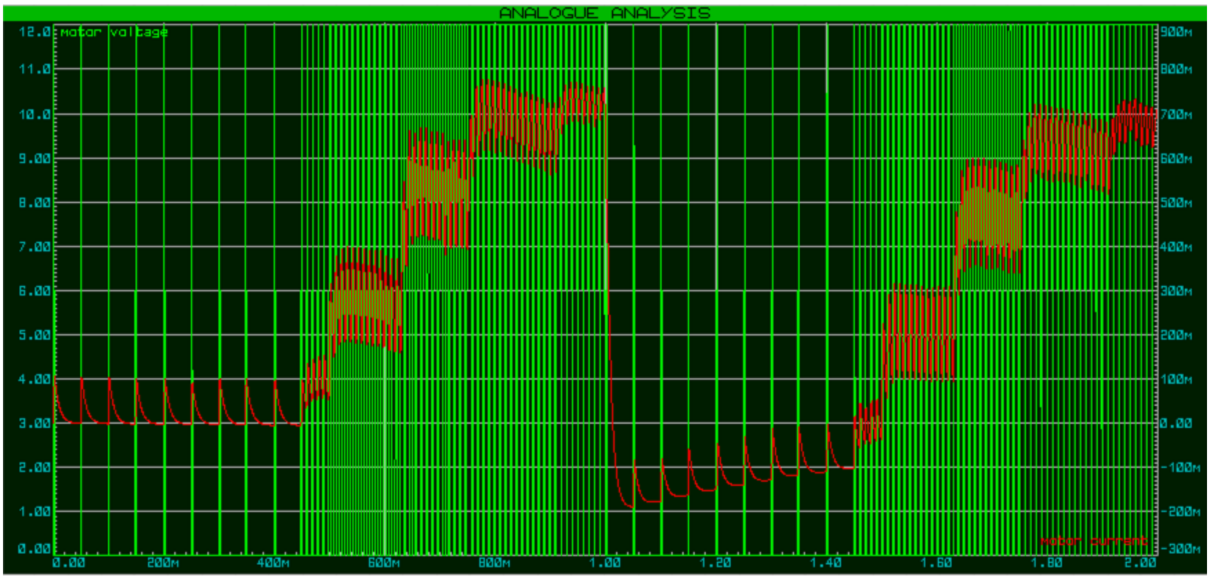

Figure 13: The current and voltage under PWM control

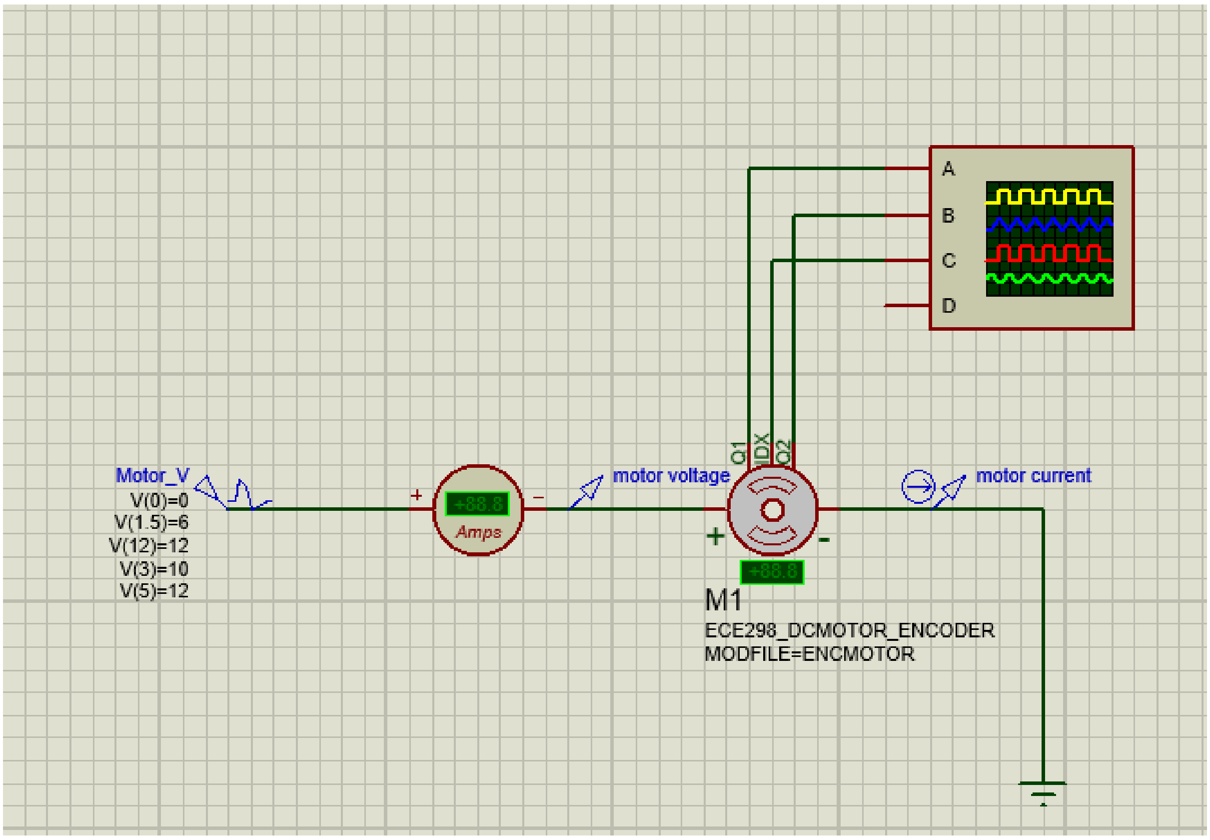

Figure 14: The schematic of ECE298_FAST_DCMOTOR_ENCODER using a DC voltage

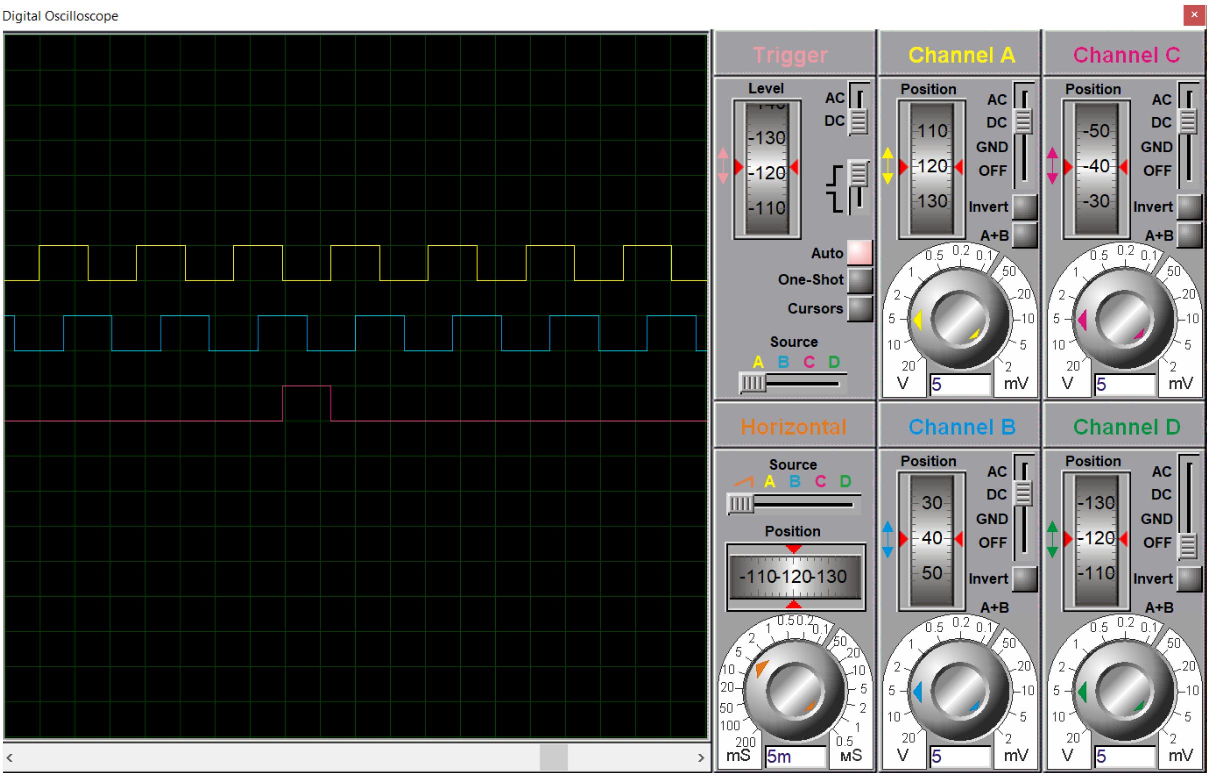

Figure 15: The signals output displayed by the oscilloscope using DC voltage(Yellow: Q1, Blue: Q2, Red: IDX)

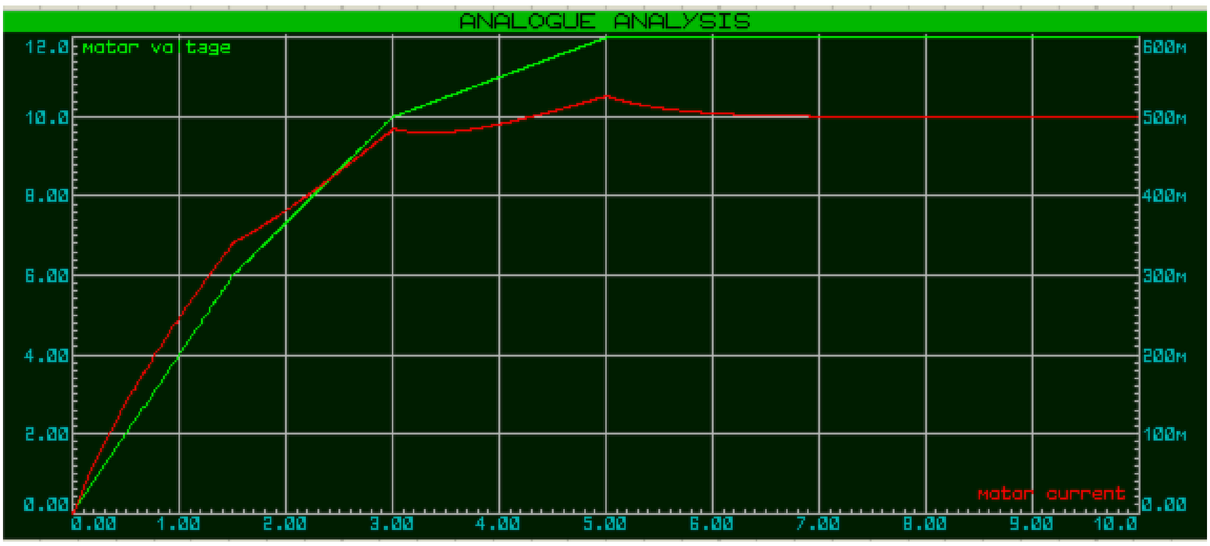

Figure 16: The current and voltage under DC voltage

Motor:

Verification: The motor can spin successfully when the power is on. From Figure 16, we could see the current changes with the voltage, which means the current powered DC motor could change with controlling voltage. We could also verify the current of the motor will vary by the control of PWM from Figure 13.

Validation: This motor is a good choice for us because it has a sensor attached. It can also spin properly. In the project, we are able to change the speed of the motor by controlling the PWM and make the motor rotate in the opposite direction by reversing power.

Motor Sensor:
Verification: From the graph generated by the oscilloscope, we could see the digital signals from the pins of the motor, so the motor sensor is working properly to record the move of the sensor.

Validation: In the project, we could use the digital signals from Q1 and Q2 to determine the direction of the motor. Also, we could use an internal counter to record the number of edges of the digital signal as revolutions of the motor.

Device 5 – LED (LED)

Three kinds of LEDs are used in this project, and we decided to use animated LED models with different colors to represent them. These LEDs have a forward voltage of 1.95V(red), 2V(yellow) and 2.1V(green) according to the information on Piazza. And they all have a full drive current of 10mA, so we calculated the resistances of their current limiting resistors based on those values and MCU voltage of 3.3V and created a schematic.

$$R = \frac{V - V_{led}}{I_{led}}$$

The schematic also used a potentiometer to change the current flows through the LEDs. After changing the LED model type to analog, the brightness of the LEDs should change with the current. The figures below show the brightness with current of approximately 100%, 50% and 20% of the full drive current.

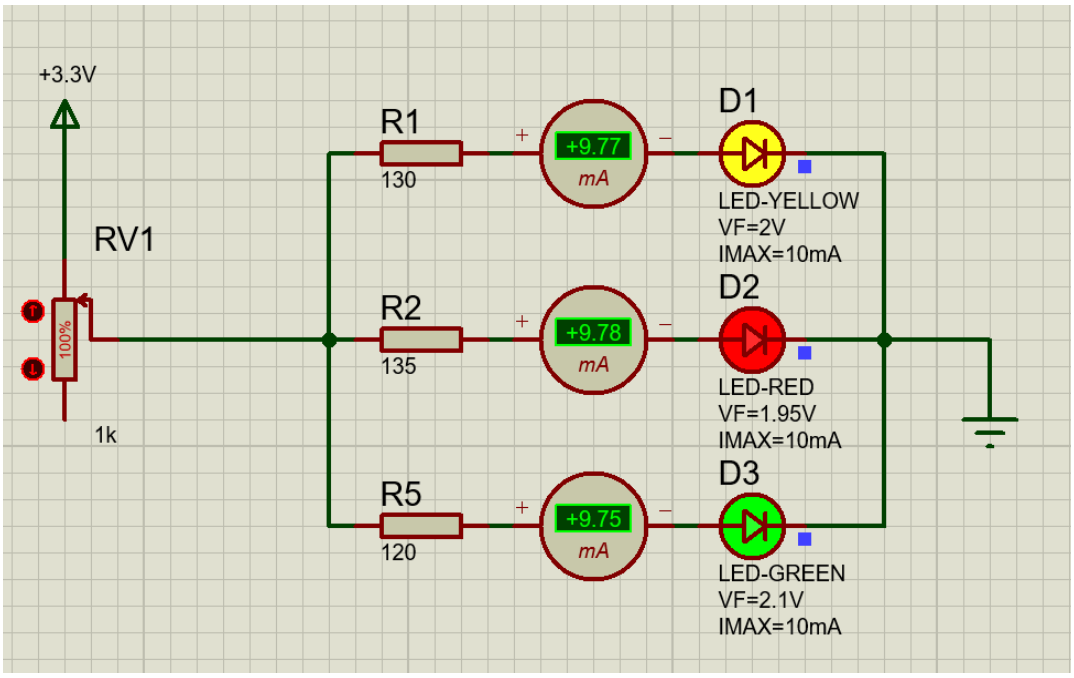

Figure 17: The schematic of the LEDs (100% of full drive current)

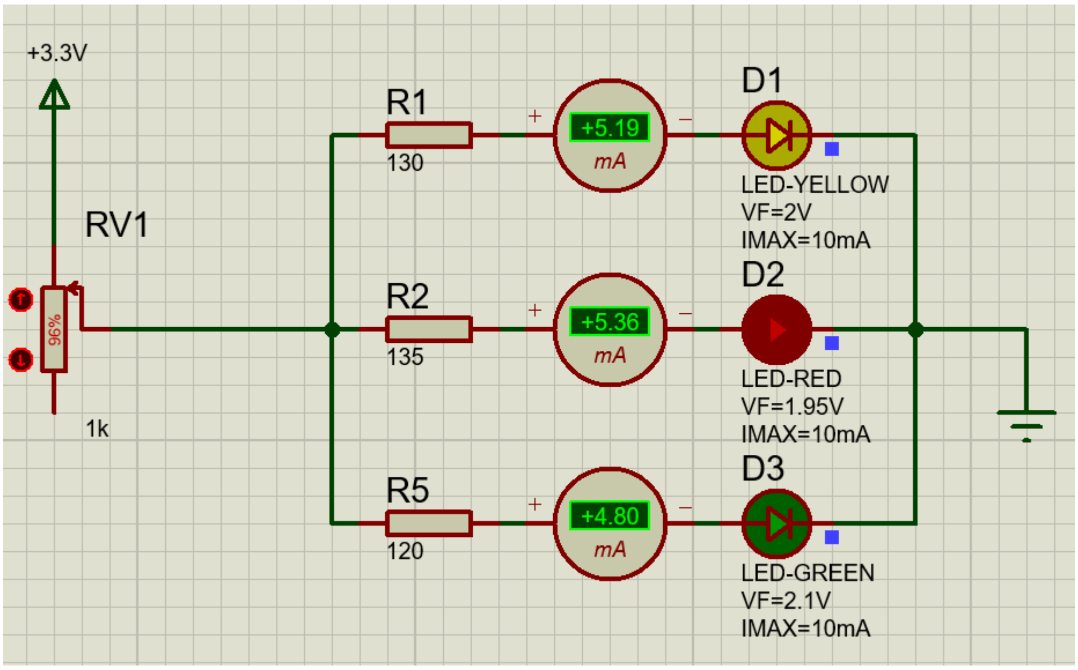

Figure 18: The schematic of LEDs (50% of full drive current)

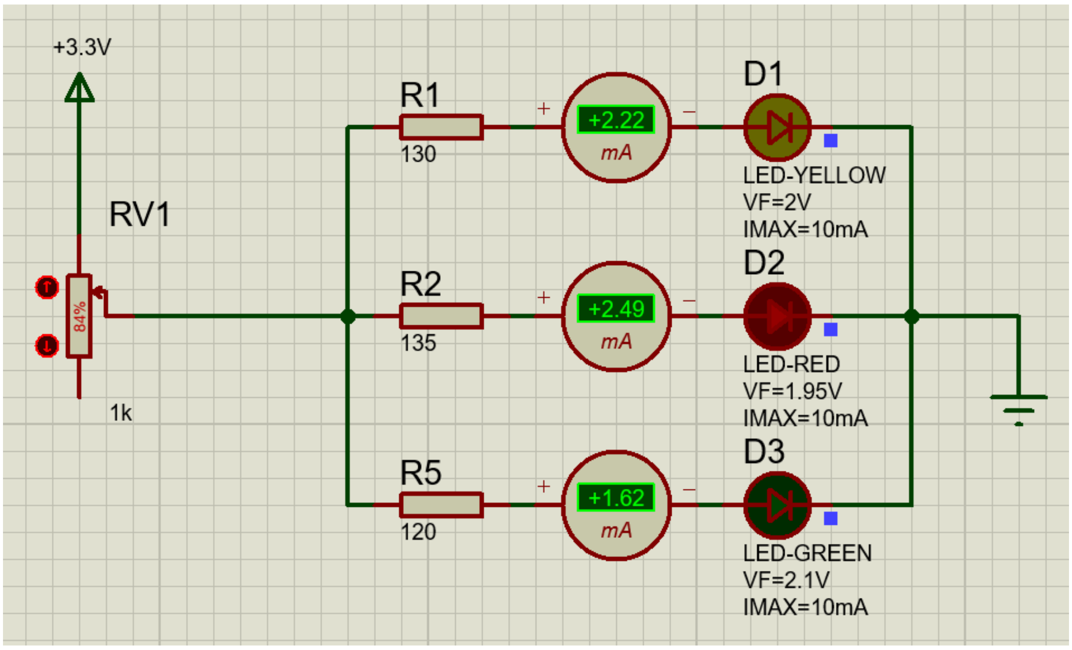

Figure 19: The schematic of LEDs (20% of full drive current)

Verification: The LEDs worked as expected. They can be turned on after powering the circuit, and the brightness of the LEDs changes with the current flow through them. We also confirmed the current-limiting resistors for these LEDs are calculated correctly because the ammeter displayed a value very close to 10mA when the potentiometer is not used in Figure 16.

Validation: These LEDs have a forward voltage of 1.95V, 2V and 2.1V, and our voltage source in this project will be 3.3V, so the LEDs can perform normally as shown above. For the project requirements, we only require a LED to turn on when a specific event happens. This can be achieved by addressing a high state to the pin to power the LED.

Device 6 – LM016L– (LCD display)

We chose this LCD for our project because it is simple to use and can display letters as well as numbers. To test this device, we connect the pattern generator to the LCD like Figure 20, and change the pattern to display our group number.

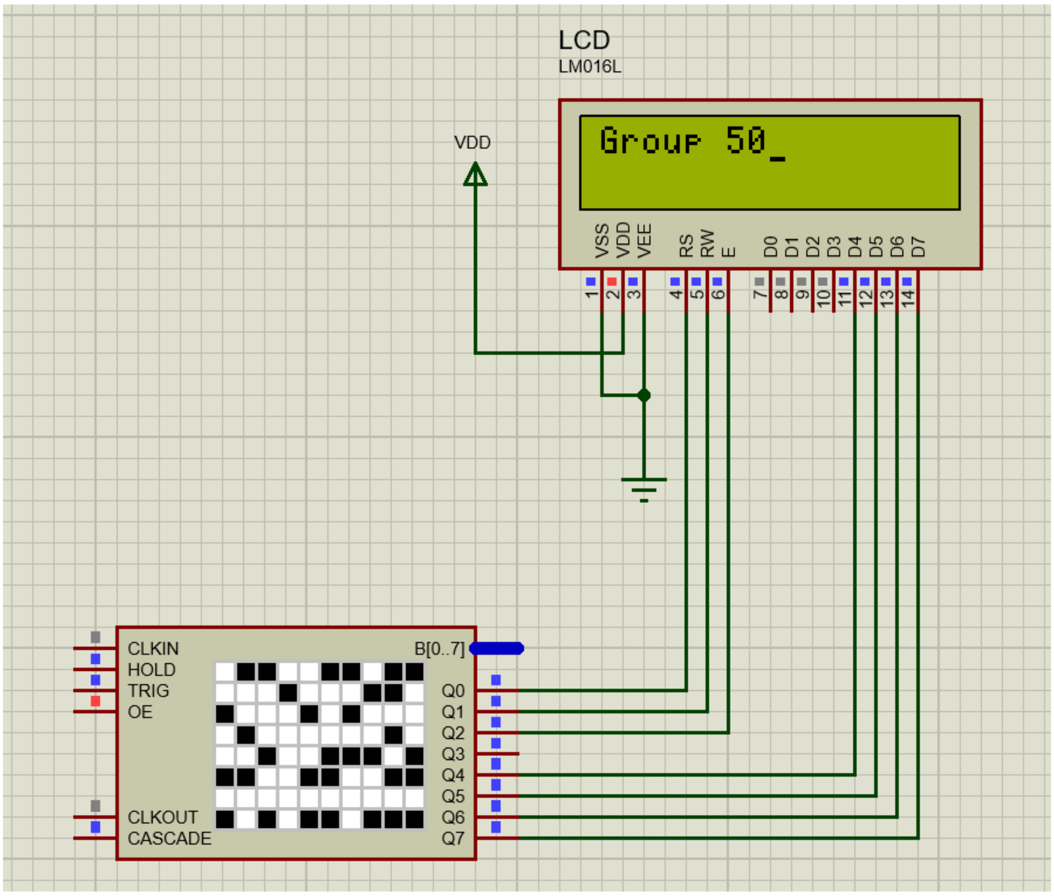

Figure 20: The schematic of LM016L

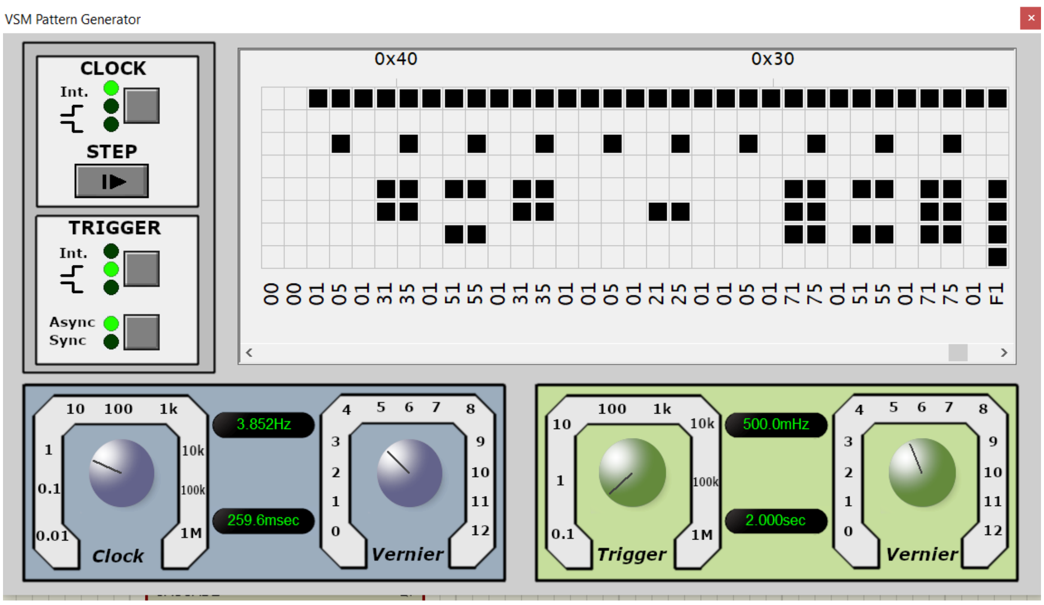

Figure 21: Part of the pattern of the pattern generator

Verification: The LCD successfully displays our group number as expected which is shown on Figure 20.

Validation: In our project, we want to display mode, parameter name and parameter value. They will be either letters or numbers. We tested that both of them can be displayed successfully on this LCD, so we can use the LCD in our project.

Part 4 – System-Level Design

System-Level Design