Lab 2 – Feasibility Model Phase 2

ECE 298 – S2021

|  |  |  |  |
| --- | --- | --- | --- |
| Lab Section: |  | 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 must have a minimum rotation speed of -400 RPM and the maximum rotation speed must be 400 RPM. These numbers are estimates of just faster than the average human walking speed.  The coil inductance of the motor is 100 mH and the coil resistance of the motor chosen is 10 Ohms. The zero-load RPM of the motor is 800 RPM. |
| 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 | 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.  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 24th 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. |
| Device input / output range | The device’s surrounding transistor’s gate voltages will be on the range 0-3.3V to turn the device on and off. The power supply will be 12 V and the duty cycle of the PWM supplied will control how long the circuit is exposed to this power supply.  The devices output is a square wave whose range is 0-5V.  The format of the output are two signals that pules 24 times per revolution and indicate the direction of rotation of the motor, and another signal that pulses once per rotation to indicate the rotation speed. |
| MCU connectivity details | 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. |
| Device/MCU interfacing details | 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 interfaing. |

## Schematics and Simulations

Diagram

Description automatically generated with medium confidenceBelow 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:

Diagram

Description automatically generated with low confidenceThe motor rotates at an asymptotically approaching 200 RPM.

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

Graphical user interface, chart, treemap chart

Description automatically generated

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).

Below shows the screen capture of the V\_MOTOR\_ENCODER\_OUTPUT:

Graphical user interface

Description automatically generated

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 is negligible (on the order of 10 uA).

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):

Diagram, schematic

Description automatically generated

V\_MOTOR\_CONTROLLER\_IO:

Graphical user interface

Description automatically generated

# DC Motor Controller

## Summary

| Item | Description |
| --- | --- |
| Purpose | The purpose of the DC motor controller is to control the flow of current through a brushed DC motor. |
| Device physical domain and range | The DC motor controller must be able to supply the DC motor anywhere between -3.5 to +3.5 A. This corresponds to a minimum and maximum rotation speed of -400-400 RPM. |
| 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 | 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.  Since the device is driven with a PWM signal the device is a push-pull with strong highs and lows. |
| Device input / output range | The device must be powered by a maximum 50 V supply.  The logical inputs (driven by the PWM signal) has a V\_IL of maximum 0.5 V, a V\_IH of minimum 1.5 V, a max/min current of +- 1 A.  The maximum output current is 3.6 A. |
| MCU connectivity details | The two inputs IN1, IN2 will be connected to two PWM channels of the MCU. 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 schematics and simulations for the “DC motor encoder” can be found in the “DC motor” section.

Diagram

Description automatically generatedTo 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:

Chart

Description automatically generated

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.

# 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. |
| 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).  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.  The electrical characteristics of the LCD logic is:  Maximum power supply (V\_DD – V\_SS) = 7.0 V  V\_IH = 2.2 V min.  V\_IL = 0.6 V max.  V\_OH = 2.4 V min.  V\_OL = 0.4 V max.S  The electrical characteristics of the inverter when supplied a 5V VDD (4049) logic is:  V\_OL = 0.05 V  V\_OH = 4.95 V  V\_IL = 1.0 V  V\_IH = 4.0 V  I\_OL = 4.0 mA  I\_OH = 4.6 mA  I\_IN = -1.0 mA  The electrical characteristics of the NMOS transistor (2SK176) is:  Drain-source voltage rating = 200 V  Gate-source voltage rating = +- 20 V |
| 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 | 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.  The CMOS transistors that behave like a switch are open collectors that are pulled up to VCC when not active. This drives the inverter which is a push-pull (strong high and low) device. |
| 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:

Diagram, schematic

Description automatically generated

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.

Diagram, schematic

Description automatically generatedSince 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 5V 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.

A picture containing chart

Description automatically generated

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

A picture containing text, clock

Description automatically generated

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:

Diagram

Description automatically generated

Graphical user interface, schematic

Description automatically generated

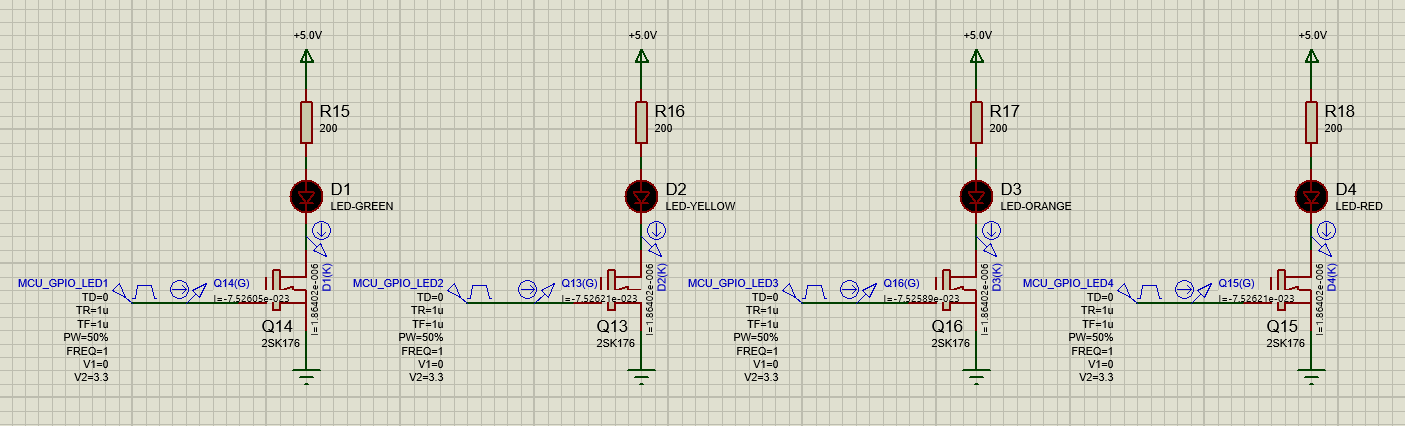
# Coloured LEDs

## Summary

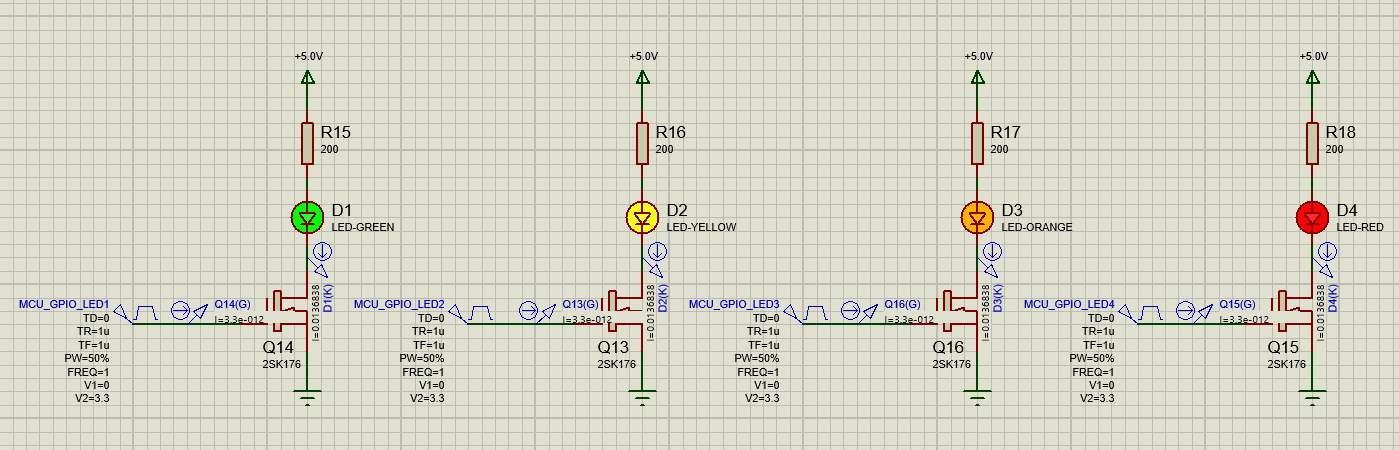
| Item | Description |
| --- | --- |
| Purpose | The purpose of the light emitting diodes (LEDs) is to convey information to the user regarding the controller mode and the battery levels. |
| Device physical domain and range | The current flowing through the LEDs during testing was 13.68 mA, which is 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 device chosen are simple light emitting diodes (LEDs) that are of 4 different colours: green, yellow, orange and red. |
| Proteus Library component name | The proteus library component names are:  LED-GREEN  LED-YELLOW  LED-ORANGE  LED-RED |
| Device input / output properties | All the LEDs that will be used in the project (every colour) will require an input to signal it to turn on/off. |
| Device input / output range | The LEDs will use a range of current that may vary anywhere between 0 mA to a maximum of 20 mA |
| MCU connectivity details | The devices will be controlled by an MCU General-purpose I/O (GPIO) pin. |
| 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. |

## Schematics and Simulations

The circuit below shows a green, yellow, orange, and red LED connected via a 200 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).



The picture above has all the LEDs that are going to be used in the project in the off position, hence we don’t see any glow in them. The switches connected to the transistor are set to ground, causing the transistor to act like a switch in the open position, not allowing current to flow.



The picture above has all the LEDs that are going to be used in the project in the On position, hence we see a glow in them. If all of the switches are now set to the 3.3 V source, 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.

# 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 must be 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. |
| 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. 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 uA is drained from the battery to the sensor circuit, which itself only drains 33 uA. 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:

Diagram, schematic

Description automatically generated

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

Diagram, schematic

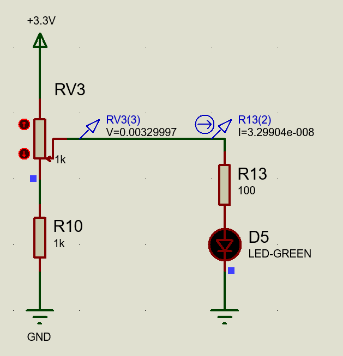
Description automatically generated

# ECE298\_GEN\_POTENTIOMETER

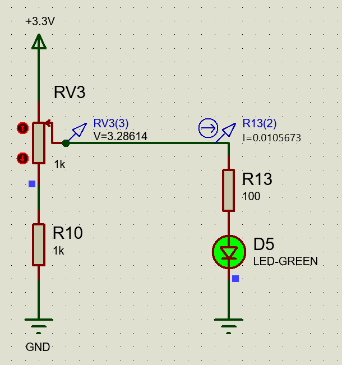
## Summary

| Item | Description |
| --- | --- |
| Purpose | The purpose of the potentiometer 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 about 1 k to 100 k. |
| Device type chosen | The device chosen to achieve the requirements in the general potentiometer, which simply put is a variable resistor. |
| Proteus Library component name | The Proteus library component name is ECE298\_GEN\_POTENTIOMETER |
| Device input / output properties | The potentiometers used will be connected as the first resistor in a voltage divider circuit, which is powered by the battery. |
| Device input / output range | This second resistor will be 300 k to ensure that the minimum sensed voltage of the ADC is 12 V = 0.03987 V and the maximum voltage is 12 V \* 3 V. This will correspond respectfully to maximum/minimum speed and maximum left/maximum right turning for speed and steering |
| MCU connectivity details | The potentiometer will be connected to an analog pin of the MCU in a voltage divider circuit. |
| Device/MCU interfacing details | To identify whether a change in the potentiometer has been made, the potentiometer’s input will trigger an interrupt that the MCU will deal with my changing the speed and steering direction of the wheelchair. |

## Schematics and Simulations



In the circuit above, the potentiometer is set to its highest resistance, and it is seen that all the current drawn from the source flows through R10 and there is no current flowing through the LED. The resistance shown is: (3.3V – 0.0032997V)/3.29904e-8 A = 99.93 M



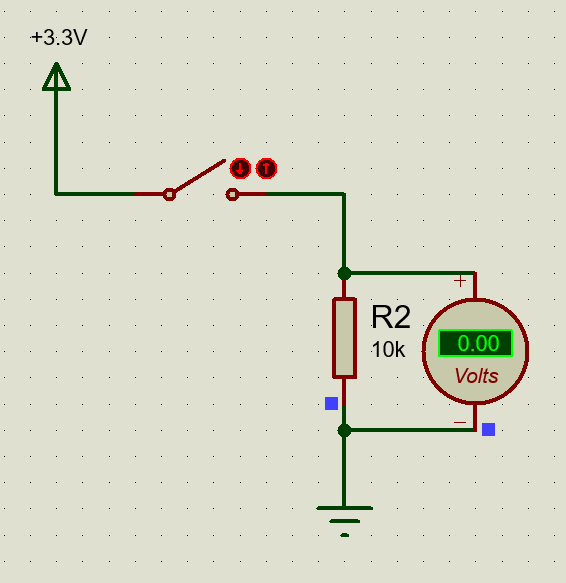
In the circuit above, where the potentiometer is set to its lowest setting, much larger current is allowed to flow through the LED. It is seen that the LED is now on since the current flowing through it is around 10 mA. Additionally, the voltage across the led and R13 is almost the entire voltage supplied by the 3.3V source. (3.3V – 3.28614V)/0.0105673 A = 1.311

# ECE298\_GEN\_SWITCH

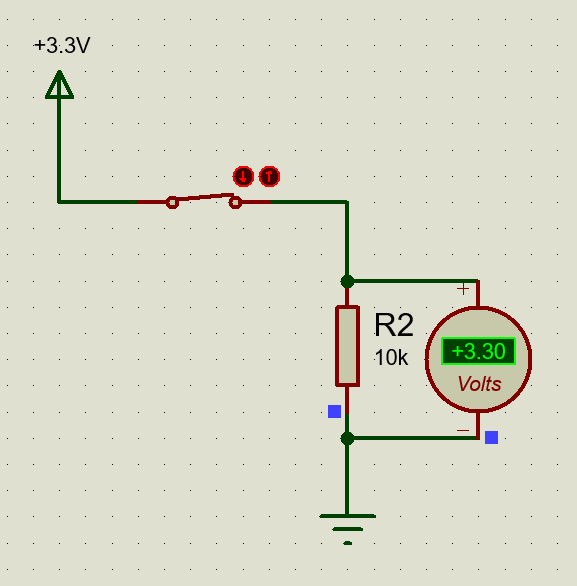
## Summary

| Item | Description |
| --- | --- |
| Purpose | The purpose of the switches is to be used as user inputs. The first switch will be used as an On/Off switch for the whole 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 of 0-3.3V where 3.3V will be a ‘1’ signal while on the other hand if the switch is closed, 0V will be passing through causing a ‘0’ signal. |
| Device type chosen | The device chosen is a simple switch that toggles between a ‘0’ and ‘1’ signal. |
| Proteus Library component name | The Proteus library component name is ECE298\_GEN\_SWITCH. |
| Device input / output properties | The switches that will be used for the project will have an analog voltage across it. |
| Device input / output range |  |
| MCU connectivity details | The devices will be controlled by an MCU General-purpose I/O (GPIO) pin. |
| Device/MCU interfacing details | The switch will be connected in series with a resistor that is sourced by a GPIO pin set to output 3.3 V. The output of the circuit will be sent to another GPIO input pin that will sense whether the voltage level is high or low. If the switch is closed, the GPIO input pin will sense just under 3.3V or ‘1’ (due to the voltage divider circuit). If the switch is closed, the GPIO pin will be connected to a pull-down resistor to ground, where it will sense a ‘0’. This signal will be processed to determine which mode the controller is in. This will be achieved via an interrupt. |

## Schematics and Simulations



The circuit to above shows us the switch in the open position, hence, the resistor is grounded with no current flowing across it, and the voltage across it is 0 V.



The circuit above shows the switch in the closed position. This allows current to flow through the resistor, whose voltage is shown to be 3.3 V.