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. |
| 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 in an H-circuit with 4 transistor’s whose gate voltages will be controlled by the MCU from 0-3.3V. Turning opposite pairs of transistors on will close the circuit to the DC power supply and cause the motor to spin forwards or backwards. A PWM input applied to these transistors 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.  utputs can also be open drain /  open collector  , but not for the  purpose of bus signalling. The output acts as a switch to ground,  powering high  -current devices  by en  abling or interrupting the  current like a switch |
| 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.  State the  working voltage of the digital I/O (e.g., 3.3  V or 5.0  V).  Each digital  output  will have a  minimum  output  high  voltage  ,  V  OH  ,min  , and  maximum  output low voltage  , V  OL,max  . Each digital  input will have an  maximum  input high  voltage  , V  IH,m  ax  , and  minimum input low voltage  , V  IL,min  , respectively.  Note them here  or  note if the values are unavailabl  T |
| MCU connectivity details | The device will be connected to the MCU via an H-bridge circuit with 4 NMOS transistors. The 4 NMOS transistors will be controlled via 2 GPIO output pins of the MCU and 2 timer pins. These two timer pins will control the exposure of the motor to current using a PWM signal and the 2 GPIO output pins will control which way the current is sunk to ground across the H circuit/ |
| Device/MCU interfacing details | The input to the motor is a digital signal from 0-3.3V. The output of the motor is a digital signal rom 0-5V, which will be stepped down to 0-3.3V. Thus, the connectivity to the MCU is Digital to digital. The connection will be unidirectional for inputs and outputs separately. |

## Schematics and Simulations

# LM016L - LCD

## 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 the LM016L. This is a parallel communicative device that is asynchronous in its communications. |
| 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). |
| 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 required to be stepped down through a voltage divider to 0-3.3V and the inputs from the MCU will be connected to a transistor that, when on, will pull the input to the LCD data lines low. The logic that the GPIO pins will drive the LCD on will therefore be inverted. |
| Device/MCU interfacing details | The |

## Schematics and Simulations

# 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 |  |
| 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 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

# 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 device 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 |  |

## Schematics and Simulations

# 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

# 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 |  |
| 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