HEAD-MOTION CONTROLLED WHEELCHAIR

By

Dev Manaktala

Arnav Jain

Jiayuan Liu

Final Report for ECE 445, Senior Design, Spring 2021

TA: Yichi Zhang

May 2021

Project No. 55

Abstract

The purpose of this report is to showcase a head-motion controlled wheelchair for people who cannot operate conventional wheelchairs such as those with amputated or paralalyzed limbs. In addition to motion control, the wheelchair consists of an object detection system that provides vibration feedback to the user based on objects in the vicinity. The scope of our project included motorizing a non-motorized wheelchair to be able to be used by people with existing wheelchairs by attaching the system. The wheelchair consists of 3 main subsystems: Motion detection, object detection, and traction control. Through the course of this report, we discuss the various design considerations, implementations, testing procedures and results of each subsystem that led to the final design of the wheelchair. Finally we discuss the achievements, ethical considerations and future improvements required to make this a viable product.

Contents

1. Introduction

1.1 Objective

Classical wheelchairs are designed to be operated by people who have the ability to maneuver them with a functioning arm. There are people, such as amputees and those suffering from paralysis, that cannot operate such wheelchairs.

We propose a head-motion controlled wheelchair that would allow people with such disabilities to be able to travel. The wheelchair would have a mounted camera facing the user and use computer vision to detect head motion to move and turn the wheelchair. The wheelchair would also have a vibration module allowing users with compromised vision to maneuver. This would be done through IR/ultrasonic sensors on the wheelchair detecting obstacles in the path and giving vibrational feedback to the user.

1.2 Background

Existing prototypes of such wheelchairs either use eye movement as user input or require the user to wear some sort of device on their head. We believe that our design could provide a much simpler and more viable solution compared to these since head motion is a more natural way to provide input. Furthermore, one of the features that distinguishes us from current alternative options is the object detection system as this would add another layer of safety to the wheelchair.

1.3 High-Level Requirements

- Wheelchair should be able to move instantaneously with head motion with minimal lag (less than 1 sec)
- Wheelchair motors should have a good enough torque to move any person upto 250 lbs between speeds of 4.5 and 8 mph
- Wheel-chair should not have false-positive inputs - the wheelchair should not move if the user did not intend it to.
- The ultrasonic sensor should be able to detect object upto 1 meters away accurately

2 Design

The design consists of four major subsystems and a microcontroller. The power supply subsystem consisting of both high and low voltage batteries provides power to the microcontroller, the Raspberry Pi, and the motor controller circuits. The motion detection subsystem involves the camera and the Raspberry Pi. Frames are sent from the camera to the Raspberry Pi [1]. The Raspberry Pi processes the data and sends output to the microcontroller. The microcontroller gives instructions to the motor controller circuits and thus drives both traction motors. The object detection subsystem measures objects close to the surface by ultrasonic sensors and gives vibration feedback to the users through the vibration motors. The Raspberry Pi, the microcontroller, and two sets of motor control circuits will be on a single board to avoid external wiring and extra power distribution.

2.1 Motion Detection

The head-motion detection subsystem consists of the mounted Raspberry Pi and the Raspberry Pi camera. It is an essential component of the system since it is responsible for determining the intended user input, maintaining the state machine, and providing the correct outputs to the motor controller. The requirements of the subsystem include low latency detections and calculations of intended user input, and a low false positivity rate.

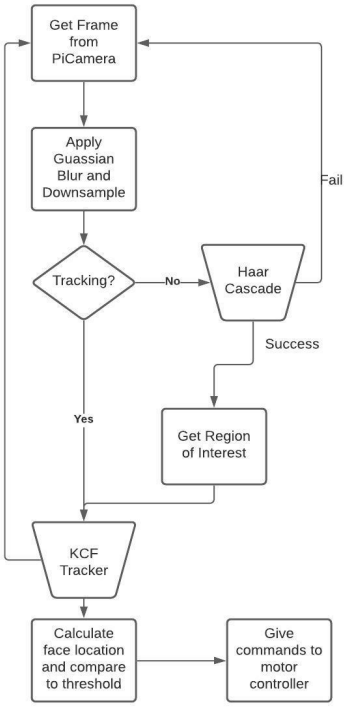

Figure 1. Flow Chart of Motion Detection Subsystem

2.1.1 Optimizations to Meet Requirements

Several Optimizations were made to the motion detection subsystem to increase accuracy and reduce latency. These optimizations are as follows:

- Input frames to face tracking and detection models were blurred, downsized, grayscale images. The accuracy of the system remained unaffected due to these changes since the model outputs are invariant of color and only rely on feature extraction and background differentiation. It improved the frame rate significantly since the input sizes were reduced.
- Face detection was only performed when face tracking failed over multiple frames. Performing face detection is more computationally expensive than performing face tracking and thus it is only performed when region of interest is required for the face tracker. Moreover, the KCF face tracker has the ability to retrieve the object once lost. Thus, face detection is only performed when this failure lasts over multiple frames.
- False positive face detections are minimized by setting a minimum size of detection. This not only increases accuracy of detection but also reduces chances of incorrect face detections in the background of the frame.

2.1.2 Intended Input Calculation

Designing a system that determines the intended input accurate was challenging. There are safety concerns of incorrectly recognizing a gesture as an input and thus this needed to be designed to minimize false positives.

There are three inputs that need to be determined namely: turn right, turn left and toggle between start and stop. These inputs are provided by tilting the head right, left and up respectively. The input choices were determined to eliminate movements of the head like turning or rotating the head, which could be made naturally. Calculations of the intended input are made based on the original position of detection of the face at rest. This is recognized as the stable position of the user and all input calculations are made with reference to this stable position. As the head is tilted the center position of the tracked face also changes. The difference between the stable position and the new position is calculated and compared against a threshold to determine the intended input. The signal to the motor is only given if this position is maintained for multiple frames to reduce false positives.

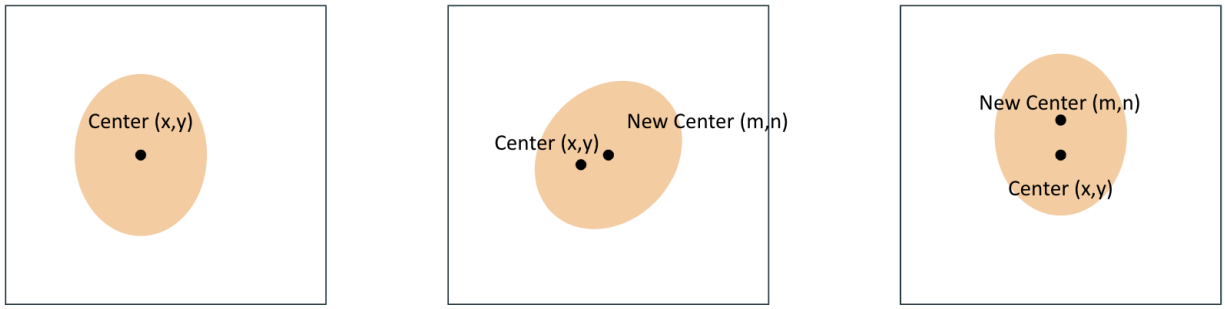

Figure 2. Intended Input Calculation

The left and right (turning) inputs are determined as follows:

$$|m - x| > Threshold \quad (eq.\ 1)$$

The up (start and stop) input are determined as follows:

$$|n - y| > Threshold \quad (eq.\ 2)$$

2.1.3 Face Detection Model Performance Analysis

We tried three different models for face detection and tested them for the resulting frame rate and accuracy. The results were as follows:

Model	Accuracy	Frame Rate
Haar Cascade	0.88	30 fps
Histogram of Graphs (HoG)	0.90	24 fps
MTCNN	0.86	27 fps

Table 1. Three Different Models for Face Detection

Haar Cascade provided a good balance between accuracy and frame rate. HoG had the highest accuracy but resulted in a lower frame rate. MTCNN was disqualified due to the lesser accuracy and frame rate compared to the Haar Cascade filter.

Based on these results, we chose to go with **Haar Cascade Filters** since the model seemed to provide similar to HoG but resulted in a much higher frame rate.

2.1.4 Face Tracking Model Performance Analysis

We tried three different models for face tracking and tested them for the resulting frame rate and accuracy. The results were as follows:

Model	False Positive Rate	Frame Rate
MOSSE	0.17	35 fps
KCF	0.07	9 fps
CSRT	0.05	5 fps

Table 2. Three Different Models for Face Tracking

MOSSE had the highest frame rate but also the highest false positive rate. KCF had a significantly lower frame rate but also a lower false positivity rate. CSRT had the lowest false positive rate but almost half the frame rate as KCF.

We decided to go with the **KCF model** because based on our field tests. Even though MOSSE provided the highest frame rate, its false positive rate proved to be a hazard as it would give incorrect signals to the motor. CSRT on the other hand was too slow and resulted in a very delayed response from the system.

2.2 Object Detection

The object detection subsystem measures objects close to the surface and gives vibration feedback to the users. When an object is detected by the ultrasonic sensors, a signal is transmitted to the microcontroller. The microcontroller toggles a signal to make the DC motor drive to operate the vibration motor. The vibration motor would be activated based on a gradient depending on the proximity of the object - closer objects would give higher vibrations.

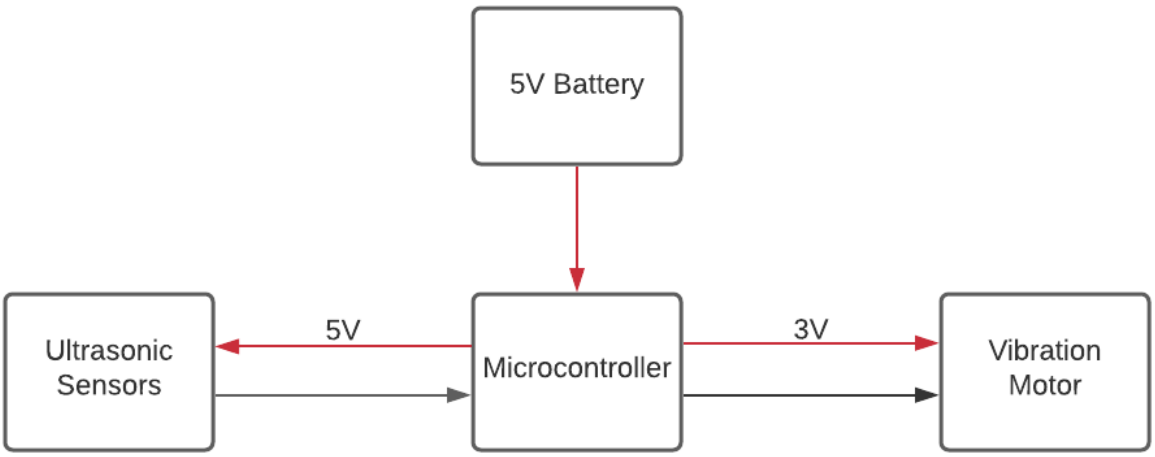

Figure 3. Object Detection Subsystem Block Diagram

The following equation was used for the calculation of the distance from the closest object:

$$distance = time \cdot 340/20000 \quad \text{(eq. 3)}$$

The limitation of the sensor only allowed us to calculate accurately upto 3 meters.
The following equation was used to calculate the vibration gradient based on which an input signal was sent to the vibration motor.

$$Vibration\ coefficient = distance/9000 \quad \text{(eq. 4)}$$

2.2.1 Software Overview

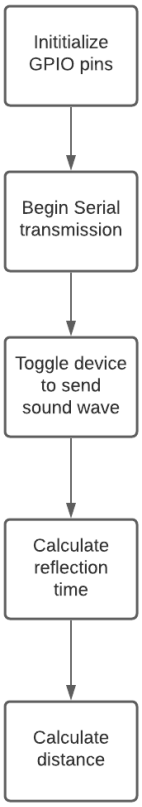

Figure 4. Object Detection software flowchart

2.2.2 Performance Analysis

The Object Detection system did not work as well as we had expected. One of the major issues was the ultrasonic sensor itself. Due to our budget constraints we had to use a cheaper sensor which gave a lot of inconsistent values. It was also extremely sensitive to bumps and vibrations from the ground. This can easily be solved by getting a more effective sensor like the UGT581 ultrasonic sensor. A lot of work was done to make software improvements to ignore inconsistencies in the values from the ultrasonic sensor. Another part of this subsystem that can be worked on is getting a vibration motor that has a larger operating voltage so that we can give a larger range of vibrations. This can be particularly useful in being able to differentiate the distances of objects.

2.3 Power Supply

The power supply consists of two 36V high voltage battery packs and a power regulation circuit. The high voltage batteries power a pair of motor control circuits and thus power two traction motors. The 36V voltage is regulated to both 24V and 5V by a power regulation circuit. The 5V voltage powers both the microcontroller, the ultrasonic sensors, and the vibration motor. The 24V voltage is the pull-up voltage for the power mosfets.

2.3.1 Design Procedure

The initial plan of the power supply was to use one 36V battery to power the traction motors and use a power regulation circuit to generate the 24V and 5V required by all the other electrical components. However, during the design review, we realized that the Raspberry Pi can draw up to 2.5A of current. This means that the power consumption of the LM317 linear regulator that steps down from 24V to 5V is consuming

$$(24V - 5V) \cdot 2.5A = 27.5W \qquad (eq.\ 5)$$

of power. This is obviously more than the heat capacity of the metal pad on a reasonable size of PCB. Therefore, we chose to power up the Raspberry Pi using a separate 5V low voltage battery, while using the power regulation circuit to power all the circuits on the PCB. The components on the PCB that are drawing current from the linear regulators include: the microcontroller, the ultrasonic sensor and a number of BJTs. Since the current drawn by the base of BJTs is negligible, the majority of current is drawn by the microcontroller and the ultrasonic sensor. The current design has just one ultrasonic sensor that requires 15mA [2] and one ATMega328P microcontroller that requires 42mA [3]. So, the power dissipation on the linear regulator that steps down from 24V to 5V can be approximated as

$$(24V - 5V) \cdot (15mA + 42mA) = 1.083W \qquad (eq.\ 6)$$

This is a reasonable amount of power dissipation on a metal pad.

Another change is that we chose to use two 36V batteries to power two motor controller circuits. The first reason is that the connector that we choose to connect the battery to the motor controller is the TE Connectivity 770166-1 Connector. It has a current rating of 9.5A [4]. This is not enough to provide about 20A of current for both motors. The second reason is that the two-battery configuration doubles the operation time of the wheelchair, since our wheelchair needs to have a reasonable time of operation. With two batteries of 8000 mAh batteries, we can provide 8A of current to each of the motors for

$$8000\ mAh\ /\ 8A = 1\ hour \qquad (eq.\ 7)$$

Therefore, we decided to use two 36V batteries to power the two traction motors.

2.3.2 Design Details

To achieve both 24V and 5V from the 36V battery input, we need to step down twice. We chose the LM317 linear regulator because it has a 40V maximum input voltage rating, which is higher than the 36V of our supply voltage. It also has a 1.5A output current rating, which is enough for the microcontroller, Raspberry Pi, and also the vibration motor. The output of the linear regulator [5] can be calculated as

$$V_{out} = 1.25 \cdot (1 + R_2/R_1) \quad (eq.\ 8)$$

Therefore, we selected R2 = 3.65k and R1 = 200 for stepping down to 24V. Besides, we selected R2 = 3.6k and R1 = 1.2k to step down from 24V to 5V. Since the power of a linear regulator is approximately

$$(V_{out} - V_{in}) \cdot I \quad (eq.\ 9)$$

A lot of power is dissipated by it. Therefore, we chose a surface mount package and implemented a large metal polygon to the heat dissipation pad of the linear regulator.

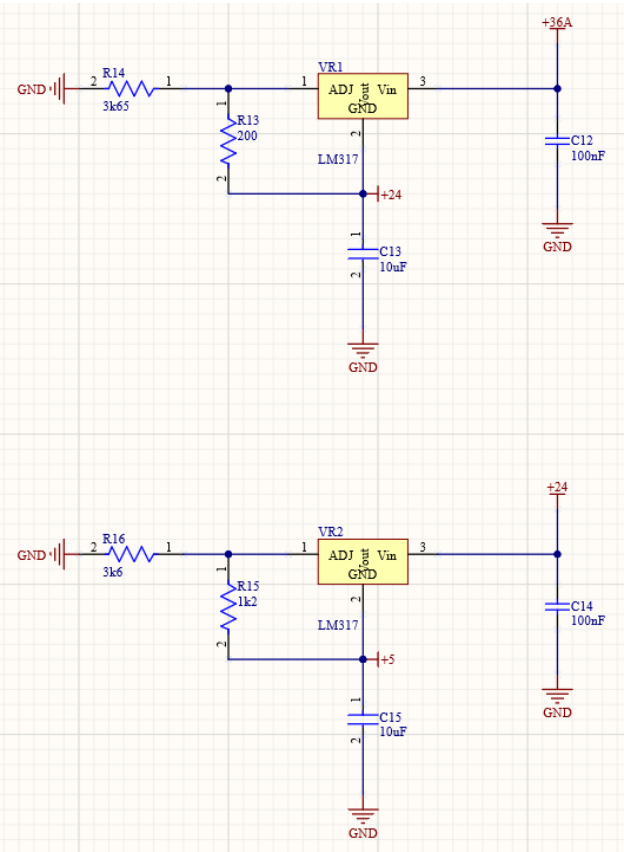

Figure 5. Power Regulation Circuit Schematic

2.3.3 Design Verification

We verified the power supply system by connecting the battery to the PCB and measuring the voltage at different circuit nets. The measured voltage at the +24 net was 23.5V and the measured voltage at the +5 net was 5.20V. These values are close enough to the designed values. Although the 5.20V at the +5 net seems to be a little higher than 5V, it is actually very close to the default Arduino Vcc value, which is measured to be 5.19V.

When testing the wheelchair as a complete system, the power supply system operated well for the first few minutes and went fault. It turned out that the +36 input net is shorted to the +5 net, so that all the low voltage components on the PCB were damaged and the motor always ran at its full speed since the microcontroller was also damaged by the high voltage. The reason is that in our design described in section 2.3.1, we ignored the current drawn by the vibration motor. Equation 6 was missing the term of vibration motor current. The total current drawn from the power regulation circuit when testing the whole system was measured to be 220mA, then equation 6 would become

$$(24V - 5V) \cdot 220mA = 4.18W \qquad (eq.\ 10)$$

This is more than what an LM317 linear regulator can afford. The best solution to this problem is to use a DC-DC converter to replace the system of two linear regulators. A well designed DC-DC converter will be able to continuously supply the required current, but the design will be much more complicated and will require more components.

2.4 Traction System

The traction system is powered by the high voltage batteries at 36V. It allows the wheelchair to move straight forward or to turn left and right by adjusting the speeds of two motors according to the signals received from the Raspberry Pi. When receiving a forward signal from the microcontroller, the motor controller controls two motors to operate at the same speed. On the other hand, when receiving a turning signal, the motor controller moves the two motors in opposite directions to turn the wheelchair.

2.4.1 Design Procedure

The initial choice for motors was to use the BestEquip three-phase brushless DC motor. However, we found that the control circuit of a brushed DC motor is much more simple. The control logic in the MCU only needs to handle one power line and the gate driver chips are also no longer required. Besides, the performance of a brushed DC motor is more than enough for our wheelchair which typically moves at less than 1m/s. We decided to pick the BY1020D 500W 36V brushed DC motor, because both its voltage and power rating fit our application. Moreover, this motor comes with a mounting bracket to be mounted on our wheelchair.

For the traction motor controller circuit, we implemented a combination of BJT and power MOSFETs to control the traction motor. Since the current rating of a BJT is limited, we used a BJT to control three

MOSFETs. Since we have three MOSFETs in parallel, the large current through the motor can be divided into three paths so that each MOSFET only needs to handle 1/3 of the current.

The 5V DC vibration motor controller circuit is designed to be relatively simple. This time we just need a BJT to drive the current because the current requirement for a vibration motor is small enough to be handled by a BC548 BJT.

2.4.2 Design Details

The schematics of our traction motor controller circuits are shown in figure 6 and 7. The microcontroller reads the throttle signal between 0 to 5 volts from an analog input pin for each motor. When the throttle signal is high, the microcontroller outputs a low voltage to the base of the BJT. Since we use an NPN BJT, the BJT is now turned off. Then the gates of the power MOSFETs are pulled up to 24V, so the MOSFETs are on. 36V of voltage is supplied to the motors and turns them on. When the throttle signal is low, the microcontroller outputs a high signal to the base of the BJT to turn it on. Then the gates of the power mosfets are grounded, so the mosfets are off. The V- terminals of the motors are not connected to ground and they are turned off in this way. The speed of the motors can be controlled by adjusting the signal frequency at the base of the BJTs, since the microcontroller has the ability to assign an arbitrary PWM frequency to its digital output pins.

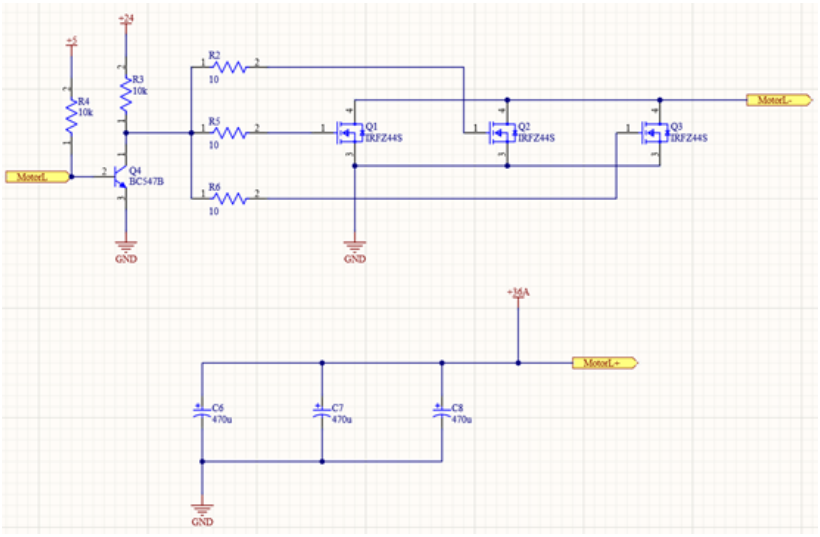

Figure 6. Left Traction Motor Controller Circuit Schematic

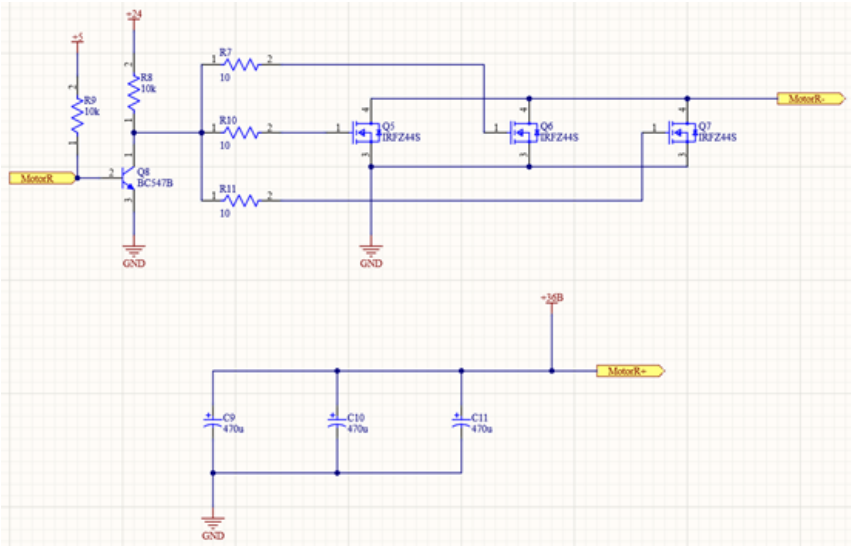

Figure 7. Right Traction Motor Controller Circuit Schematic

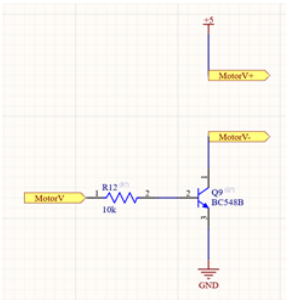

Figure 8. Vibration Motor Controller Circuit Schematic

The MOSFET that we picked for the traction motor controller circuit was the IRFZ44S Power MOSFET. This power MOSFET has a high V_{DS} rating of 60V and a high continuous I_D of 36A [6]. These parameters make it very suitable for high power applications. We chose the surface mount package so that it can dissipate up to 2W through the drain pad (heat dissipation pad) [6].

We chose BC547 as the BJT. BC547 is the high voltage version of the commonly used BC548 BJT and it has a 45V breakdown voltage [7], which is safer for our design of 24V. Although the 30V breakdown voltage rating of BC548 [7] is also above 24V, we always want to leave a larger safety margin.

There are a few design considerations in the PCB layout. The main consideration is the current capacity of the traces. The circuit loop with the highest current is the one from battery input to the traction motor to the three IRFZ44S Power MOSFETs then to ground. We set the trace width of this path to 100mils. Another high-current loop is the one from 36V input to the three 470 microfarad capacitors

then to ground. The trace width of this path is also set to 100mils. The second widest traces are the three side streams of the main current loop. These traces are set to 50mils, since they only share ⅓ of the current in the main stream on average. The circuit loop with the third highest current is the one from 36V input through two linear regulators to 5V. This current path powers the MCU, the ultrasonic sensor, and the vibration motor and its width is set to 25mils.

Another thing to concern is the heat dissipation. We implemented polygons to the heat dissipation pads of the Power MOSFETs and linear regulators, since these components dissipate the most heat of the circuit. As shown in figure 9 and 10, we implemented four polygons in total: one for each LM317 linear regulator and one for each set of three IRFZ44S Power MOSFETs.

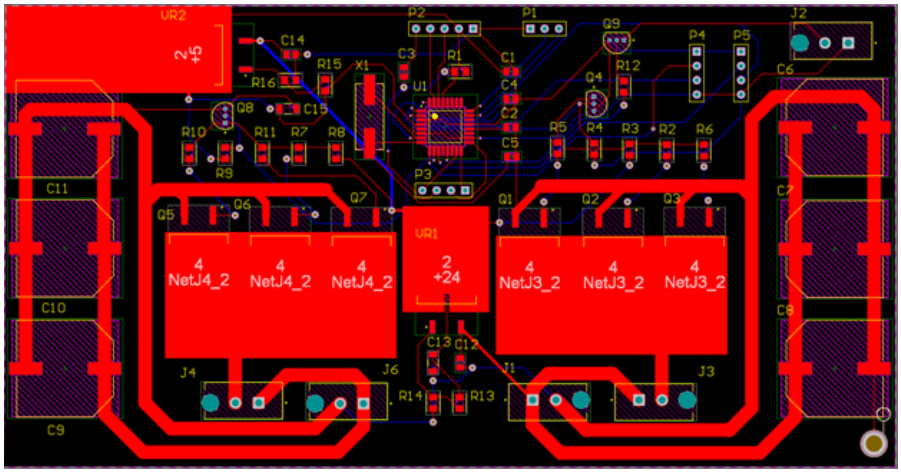

Figure 9. PCB Layout 2-D View

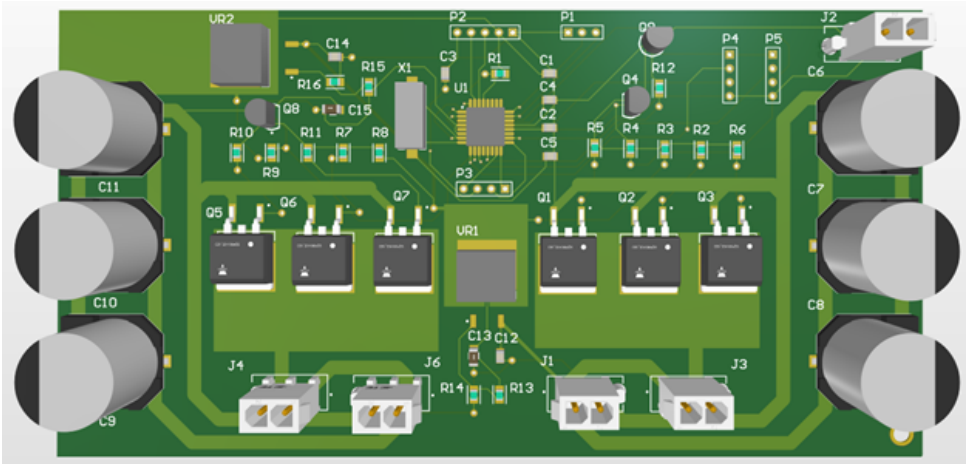

Figure 10. PCB Layout 3-D View

Brushed motors have an efficiency range between 75% to 80% [8], so the estimated heat dissipation of each of our motors is at most

$$0.25 \cdot 500W = 125W \quad (eq.\ 11)$$

Since we are running our motors at at most 9.5A, which is the current rating of the connectors, we are not running at full power. Our heat dissipation of each motor is at most

$$0.25 \cdot 9.5A \cdot 36V = 85.5W \quad (eq.\ 12)$$

The length of the motor is more than 15cm and the radius is over 6cm, so the total surface area is more than $565cm^2$. Therefore, the power density over the motor is less than $0.151J/cm^2$. This value is securely under the safety line.

2.4.3 Design Verification

We verified the traction system by uploading the motor controller Arduino code into the microcontroller. The code was successful and we could easily modify the PWM frequency on the digital output pins to control the speed of the motors.

Since the traction motor controller circuit is controlled by two signals from the Raspberry Pi, it needs to be tested together with the Raspberry Pi. We first tested it by giving 3.3V to each of the input pins of the MCU. However, we found that no matter which pin is given 3.3V, the motor would always run. This was incorrect because each of the two motors should be separately controlled by the two input signals. We modified the face detection code in the Raspberry Pi to let it generate a LOW signal when the signal on that pin needs to be grounded. In this way, we could use the software to suppress the voltage at the ATmega328P chip. We implemented the code and connected the Raspberry Pi to the traction motor controller circuit. The two traction motors were successfully controlled by head motion in this way.

When testing the wheelchair as a complete system, the power MOSFETs heated up almost immediately after the motors started running. To improve the heat dissipation, heat sinks need to be installed on the heat dissipation pads of the power MOSFETs. Actually, there is space reserved for these heat sinks as shown in figure 9 and 10.

Another problem was that one of the traction motors was running in the wrong direction. Since the machine shop installed the two motors symmetrically on each wheel of the wheelchair, giving the same polarity to both motors will make the wheelchair always turn in one direction. Moreover, the motor we used does not spin in the other direction when switching the input voltage polarity. There are two solutions to this problem. One is to mechanically change the direction of one of the motors, but this requires a big modification to the structure of the wheelchair. The second solution is to research for other brushed DC motors that can spin in both directions. Besides, the 36V 500W motors that we chose turned out to be too powerful for the wheelchair application. If we have a chance to do another design, we will use a 24V 250W motor. This will both reduce the motor power and cut down the overall cost.

2.5 Microcontroller and Connectors

The microcontroller and connectors are common components that are required for almost all the electrical projects. The choice of microcontroller and connectors directly affects the quality of a product. Therefore, we are very careful on the choice and standard of these components.

2.5.1 Design Procedure

In the first place, we decided to use the most common Arduino ATMega328P chip because the circuit configuration, programming procedure, and code library are all mature and complete online. There are three different versions of ATMega328P chips: the Arduino development board, the ATMega328P-PU throughhole chip, and the ATMega328P-AU surface mount chip. We chose the ATMega328P-AU to make the design more compact and to keep consistent with all the other SMT components on the board, although it is more difficult to solder, program, and test.

We have two different connector systems: one for high voltage and high current components (high voltage battery, traction motors, and the vibration motor) and one for low voltage and low current components (Raspberry Pi, microcontroller, and ultrasonic sensor). We implemented a very complete connector system for high voltage components so that the current and voltage requirements at each connecting port are guaranteed to be reached. This connector system consists of the connector itself on the PCB, the receptacle housing to be plugged into the connector, and the female header sockets to be crimped at the terminals of 14 AWG wires.

2.5.2 Design Details

In the schematic of the ATMega328P-AU chip shown in figure 11, we need to connect all relevant passive components to the chip since we are using a custom chip instead of the whole Arduino board. Besides, we clarified all the digital/analog inputs and outputs that we were using in this project. For the 16MHz crystal labeled X1 in the figure, we chose to use the Fox Electronics FOXSDLF/160-20 crystal because it has a high frequency tolerance of +/- 30PPM [9] and has a surface mount package.

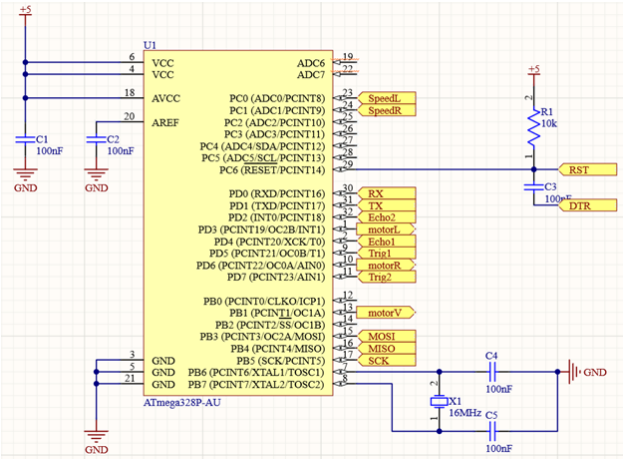

Figure 11. MCU Schematic

For the connectors design shown in figure 12, we used normal header pins for the MCU adapter ports and the ports that connect to the Raspberry Pi and ultrasonic sensors because these ports only need to handle data transfer at 5V and low current levels. However, we used the TE Connectivity AMP connectors for the battery connectors and the motor connectors to deal with high voltage and high current. For the battery input connectors, we chose to use the 1-770166-1 two-terminal connector. We also used this same 1-770166-1 connector for the two traction motor connectors and the vibration motor connector since they all have two pins of input. To match up the 1-770166-1 connector, we used the 172165-1 receptacle housing and the 794407-1 female socket. These three products are in the same series and all have the same current rating of 9.5 amps per line and voltage rating of 600 volts AC/DC [4]. These ratings well satisfy the requirement for our project.

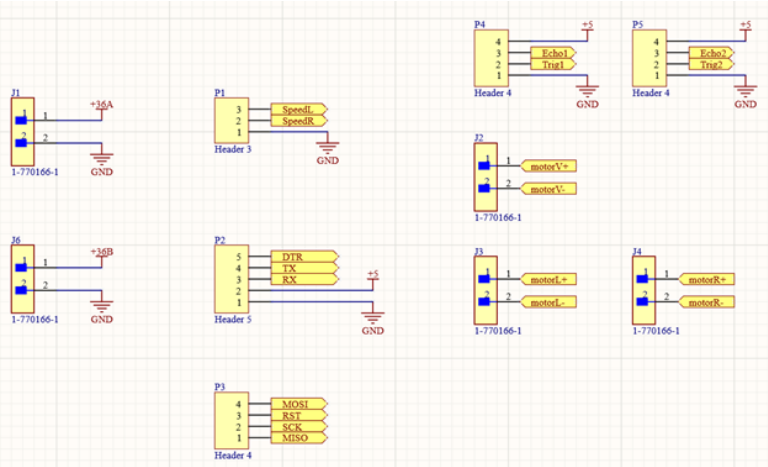

Figure 12. Connectors Schematic

2.5.3 Design Verification

We tried to verify the microcontroller design by using an FT232RL USB-to-TTL serial converter adapter to program the chip. However, we were not able to burn the bootloader. We realized that programming the SMT version of ATMega328P has a very specific procedure. We bought an XP-4627 ISP programmer that is specifically designed for microcontroller programming instead of the general USB-to-TTL serial converter. On the PCB side, we also made a few changes. First, we added the MOSI, MISO, SCK, and RST pins using a four-terminal header pin to help access the on-board SMT Arduino chip without using magnet wires. Besides, we changed the two 100 nanofarad capacitors siding the 16MHz crystal into two 22 picofarad capacitors to help the crystal oscillate in the right frequency range. Afterwards, we were able to burn the bootloader and upload sketches to the chip. The process of programming the SMT Arduino chip ATMega328P-AU is much more complicated than programming the throughhole version, but the SMT chip occupies a much smaller space and is more suitable for more advanced PCB designs.

Previously, since the lab did not provide the crimping tool, we just plugged the female header sockets directly into the connector on the PCB without the receptacle housing. There are current spikes whenever we connect the batteries to the PCB. To improve the safety index, we hand crimped all the 794407-1 female header sockets with a plier and plugged them tightly into the 172165-1 receptacle housing. However, the current spikes were still not eliminated. The current design is using three large 470 microfarad capacitors between the battery input and ground to prevent voltage spikes, but there are no passive components to reduce current spikes. We need to design a more complicated circuit instead of just parallel capacitors between battery input and ground to limit the inrush current.

2.6 System State Machine

The wheelchair was controlled by a state machine that determined what state the wheelchair would be at at any given time.

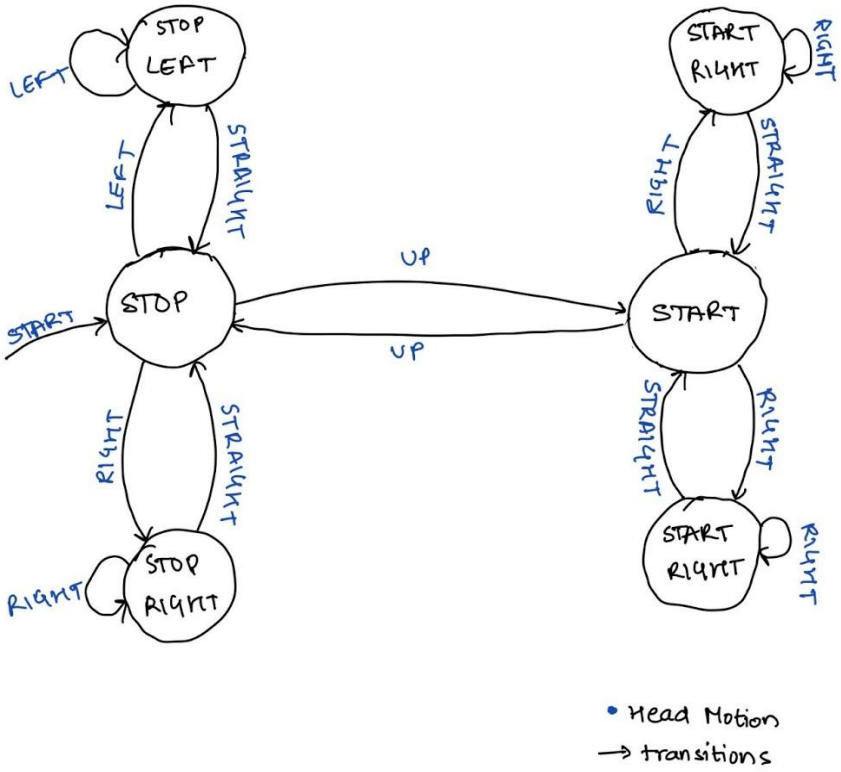

Figure 13. State Machine Diagram

In the future we plan on making a few changes to this state machine. We plan on adding another state called “Complete STOP” which would allow the user to be completely stopped such that no head motion would move the wheelchair. This would allow the user to be stationary while sitting on the wheelchair. This would change the state machine in the following way.

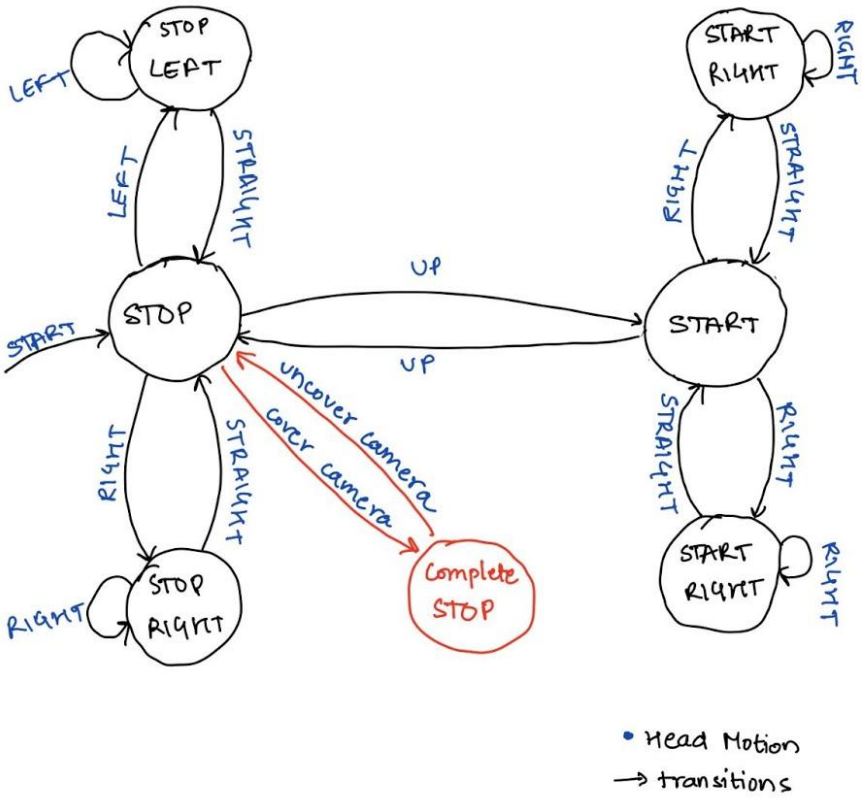

Figure 14. Revised State Machine Diagram

3. Costs

3.1 Parts

Description	Manufacturer	Part Number	Quantity	Cost/Unit	Total
Wheelchair	Walmart	/	1	$150	$150
Brushed DC Motor	Vevor	500W 36V	2	$60	$120

40kHz Ultrasonic Sensor	Adafruit Industries LLC	3942	2	$3	$6
USB to TTL Serial Converter	SongHe	FT232RL	1	$10	$10
USB ISP Programmer	ATMEL	XP-4627	2	$10	$20
8-bit Microcontroller	Microchip	ATMEGA328P-AU	6	$2	$12
Camera Module	Sony	DEV-14028	1	$25	$25
Li-ion Battery Pack	Okoman	36V 30Ah	2	$130	$260
Assorted resistors capacitors, MOSFETs, BJTs, crystals, linear regulators, connectors	Mouser	/	/	/	$200
PCBs	JLCPCB	/	10	$15	$150
Machine Shop Service	/	/	2 weeks	/	/
				Total	$953

Table 3. Cost of Parts

3.2 Labor

Assume that the labor cost for each member in our group is $30/hour and assume that each member works 15 hours per week. We have 3 members in our group and 16 weeks this semester to complete this project.The total labor cost is calculated as:

$$(\$30/hour) \cdot (15\ hours/week/member) \cdot (3\ members) \cdot (16\ weeks) \cdot 2.5 = \$54000 \quad (eq.\ 13)$$

4. Conclusion

4.1 Accomplishments

The user inputs are calculated inferred correctly when tracking and detection does not fail. Thus, we can control the motors correctly based on user input.

4.2 Uncertainties

The accuracy of the head-motion detection subsystem is not high enough to ensure the safe movement of the user. The detection is prone to failure when the user moves too fast. Moreover, the user does not know what inputs he's providing since there is no screen. A screen would enable users to see themselves and the inputs they are providing to the system.

4.3 Ethical considerations

After testing the wheelchair, we realized many safety concerns of false positives, overheating of motors and PCB, and unintended movement. In order to make this wheelchair a viable product it is important to address all these issues and rigorously test the wheelchair. The wheelchair must be tested for safety and reliability in different environments.

4.4 Future work

We would like to work on improving the accuracy of the face detection even if it comes at the cost of a lower frame rate. Adding a recalibration period when face tracking fails would help the user adjust to a stable position and restart if needed. A screen would be required to convey this information to the user.

References

[1] ArduCam, ‘Raspberry Pi MIPI CSI Camera Pinout’, 2014. [Online]. Available: https://www.arducam.com/raspberry-pi-camera-pinout/. [Accessed: 18-Feb-2021].

[2] Murata, ‘Ultrasonic Sensors FAQ’, 2021. [Online]. Available: https://www.murata.com/en-us/support/faqs/products/sensor/ultrasonic/char/0002. [Accessed: 18-Feb-2021].

[3] Alan Mitchell, ‘Operating an Arduino for a Year from Batteries’, 2011. [Online]. Available: https://alanbmitchell.wordpress.com/2011/10/02/operate-arduino-for-year-from-batteries/. [Accessed: 3-April-2021]

[4] TE Connectivity, ‘770166-1 Product Details’, 2014. [Online]. Available: https://datasheet.octopart.com/770166-1-TE-Connectivity-datasheet-22148016.pdf. [Accessed: 5-April-2021].

[5] Texas Instruments, ‘LM317 3-Terminal Adjustable Regulator’, 1997. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm317.pdf. [Accessed: 4-March-2021]

[6] Vishay, ‘Power Mosfet’, 2011. [Online]. Available: https://www.vishay.com/docs/91293/91293.pdf. [Accessed: 4-March-2021]

[7] Fairchild Semiconductor, ‘BC546/547/548/549/550’, 2002. [Online]. Available: https://www.sparkfun.com/datasheets/Components/BC546.pdf. [Accessed: 4-March-2021]

[8] Quantum Devices, ‘Brushless Motors vs Brushed Motors, What's the Difference?’, 2014. [Online]. Available: https://www.quantumdev.com/brushless-motors-vs-brush-motors-whats-the-difference/. [Accessed: 4-March-2021]

[9] Fox Electronics, ‘Resistance Weld SMD Crystal’, 2021. [Online]. Available: https://datasheet.octopart.com/FOXSDLF/160-20-Fox-Electronics-datasheet-9568005.pdf. [Accessed: 5-April-2021].

Appendix A Requirement and Verification Table

Table 4. System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Face motion recognition must operate with minimal lag (<1 sec)	1. a. Implement Haar Cascade face detection model. b. Create test samples of different faces/backgrounds and calculate accuracy as detected faces/total samples. c. Calculate false positives as incorrect detections/total samples d. Expected values for accuracy is 90% and false positive rate is close to 0. e. Run a field test and calculate frames/second.	Y
2. The sensor would need to accurately detect objects upto a distance of 3 meters.	2. a. Assemble circuit on breadboard with ultrasonic sensors and microcontroller - get distance measurements on a monitor. b. Test sensors with the setup and place objects at 1, 2 and 3 meters in succession. c. Test multiple ultrasonic sensors to check which one gives the most accurate values.	Y
3. The output of the first linear regulator should be 24V and the output of the second linear regulator should be 5V.	3. a. Solder every component relevant to the power regulator. b. Apply 36V to the input of the first linear regulator. c. Measure the outputs of the two linear regulators. They should be 24V and 5V, respectively.	Y

4. The voltage variance at the 5V output cannot exceed 5% which is 250mV.	4. a. Use an oscilloscope to probe the output at the 5V terminal. b. Measure for at least a minute and note the highest and lowest values. Both values cannot be more than 250mV from 5V.	Y
5. The power regulator should be able to continuously provide at least 100mA at 5V to power the MCU, the ultrasonic sensor, and the vibration motor.	5. c. Connect a 50ohm resistor between the 5V output and ground. d. Use a multimeter to measure current through the resistor. The temperature of both LM 317 linear regulators should not increase significantly.	Y
6. When the SpeedL signal on PC0 is high at 5V, the MotorL signal on PD3 should be pulled to ground.	6. a. Apply 5V to PC0 on the MCU. b. Use a multimeter to measure the resistance between PD3 and ground. The resistance should be less than 2ohm, which means that PD3 is grounded.	Y
7. When the SpeedR signal on PC1 is high at 5V, the MotorR signal on PD6 should be pulled to ground.	7. a. Apply 5V to PC01 on the MCU. b. Use a multimeter to measure the resistance between PD6 and ground. The resistance should be less than 2ohm.	Y
8. When the object detection algorithm calculates a distance below 5m, the MotorV signal on PB1 should be 5V.	8. a. Operate the object detection algorithm. b. Use the Arduino console to display the detected distance. c. When the displayed distance is less than 5m, use a multimeter to measure the voltage at pin PB1. It	Y

	should be 5V.	
9. When the MotorL signal at the base of the left BC547B BJT is pulled to ground, all three left IRFZ44S MOSFETS should be turned on.	9. a. Apply ground to the base of the left BC547B BJT. b. Measure the resistance between the drain net (MotorL-) of the three left MOSFETS and ground. The resistance should be less than 2ohm, which means that this net is connected to ground.	Y
10. When the MotorR signal at the base of the right BC547B BJT is pulled to ground, all three right IRFZ44S MOSFETS should be turned on.	10. a. Apply ground to the base of the right BC547B BJT. b. Measure the resistance between the drain net (MotorR-) of the three right MOSFETS and ground. The resistance should be less than 2ohm, which means that this net is connected to ground.	Y
11. When the MotorV signal at the base of the BC548 BJT is powered at 5V, the BC 548 BJT should turn on.	11. a. Apply 5V to the base of the BC548 BJT. b. Measure the resistance between the drain net (MotorV-) of the BC548 BJT and ground. The resistance should be less than 2ohm.	Y