# Motor-Aided Wheelchair

By

Guangxun Zhai

Jiacheng Huang

Zilin Zhao

Final Report for ECE 445, Senior Design, Fall 2021

TA: Yichi Zhang

07 December 2021

Project No. 26

# Abstract

This report mainly discusses our project, Motor-Aided Wheelchair, the goal of which is to help medical personnel and family members who take care of the people with disabilities that are not able to control current powered wheelchairs in the market themselves. A brief introduction and a detailed description of our design is given first, followed by the expenditure we spent and the reflection we conclude after completing the project.

# Contents

# 1 Introduction

## 1.1 Objectives

An accurate epidemiological data on the use of wheelchairs is hard to obtain, but an estimation states that around 1% of the world's population, or about 65 million people, are living with a disability needing a wheelchair. The United States has an estimated 3.3 million wheelchair users, and the number is increasing each year. In Canada, there are approximately 288,800 wheelchair and scooter users aged 15 and over, about 1% of the Canadian population. The actual number of wheelchair users might be even greater, given the fact that the above data includes only manual wheelchair users instead of those in residences or long-term care facilities. Data given by the EU statistical office Eurostatin claims that there are an estimated 5 million wheelchair users in Europe. However, in 2003 it was estimated that 20 million of those who needed a wheelchair to get around did not have one, and of those who did, very few had a wheelchair adequate to meet their needs. vspace.15cm

Our goal is to develop a motorized wheelchair to help pushers. Instead of being controlled only by the person sitting in through a panel on the wheelchair, our wheelchair detects the force exerted by the pusher and gives an assistive force to make it easy to push. The assistive force is calculated by a smooth function to give the pusher an intuitive and fluent experience.

## 1.2 Background

Nowadays, motorized wheelchairs are becoming more widespread, but there remain people with disabilities that need more care who are not able to control the motorized wheelchairs themselves. In such situations, an ordinary motorized wheelchair controlled by the person sitting in is unlikely to help, given the fact that the medical personnel or family members who push the wheelchair could not be benefited. Therefore, developing a motor-aided wheelchair that gives feedback assistive power to the wheels makes people easier to push.

### 1.3 Block Diagram

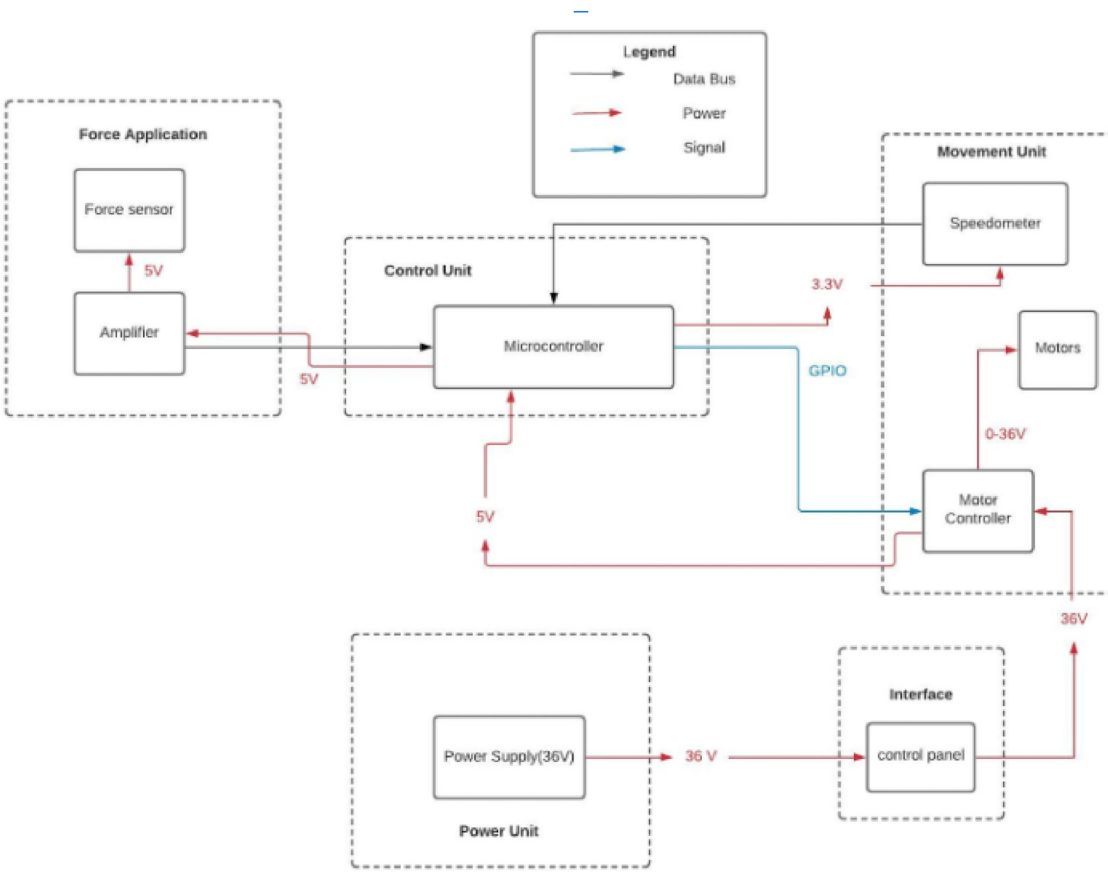



Figure 1: Block Diagram of Motor-Aided Wheelchair

Basically we have only one type of power supply which is two 36V batteries. The current from the batteries will go through our only interface (switch) to control on and off of the whole system. Then the motor controller could narrow the voltage to 5V and use this amount to power our PCB board. We will detect the magnitude of the force from a sensor and convey this data to our microcontroller to compute the corresponding duty cycle to the motor controller. Then the motor controller could make the motors and wheels spin at different speed depending on the duty cycle.

We made several changes compared to our initial design. We decreased the amount of power supplies since we found that the microcontroller could automatically power the PCB. We do not have to provide another 9V power. In addition, we add the motor controllers into our design to easily control the spinning speed of motors.

### 1.4 High Level Requirements

- We will connect the force sensors onto the handle and push the handles naturally. When the sensor detects the force, we divide the force value by 4500 to get a duty cycle number and use this to control the spinning speed of both two motors.
- The wheelchair will automatically keep its speed in a safety upper limit, which is 1.5 m/s.
- For safety concerns, the person sitting in the chair is able to use a switch to immediately halt the running of the board as well as the running of motors.

## 2 Design

The whole design includes five subsystems: Input Source, Control Unit, Movement Unit, Interface, Power Unit. Our Input Source has 2 load cells and 2 amplifier; The Control Unit is just our PCB board with a microcontroller ATmega328p; The Movement Unit includes one accelerometer, two motors, and two motor controllers; Our interface just has a switch button; And the Power is two 36V batteries, which could control the motors separately.

### 2.1 Input Source

#### 2.1.1 Design Procedure

The Input Source unit is where our main input, push force, generates. The load cell we are going to use is TAL221 [1], which is able to translate up to 100g of pressure into electrical signal. This is a Strain Gauge Load Cell. When we push, the force exerted on the handle will be detected by the implemented force sensors. Force sensors will be generating corresponding singles to the microcontroller.

However, due to the permanent damage on load cells, we replaced load cells with a ultrasonic sensor to sense the distance between the hand and the sensor. And the value of the distance will be transmitted to the microcontroller to do further computation.

#### 2.1.2 Design Details

For the load cell, there are three things to be considered when implementing load cells into the handles. First, the force capacity range of the load cell. We estimated the force that will be needed to push the wheelchair after the whole system is installed was around 1.5kg. Based on that, we chose TAL221 100g which has the force range of 0g to 1500g. Second, the voltage output from the load cell. Since the functional principle of a load cell is like a resistor reacting based on how much force you exert on it, the magnitude of the output voltage is limited. Thus, we will need to connect a voltage amplifier for our load cell in order for our PCB to read the signal. The voltage output for TAL221 100g is between 0.45 to 1.15 mv. Third, the placement of load cells. We decided to cut the handles in half and installed load cells in between handlebars.

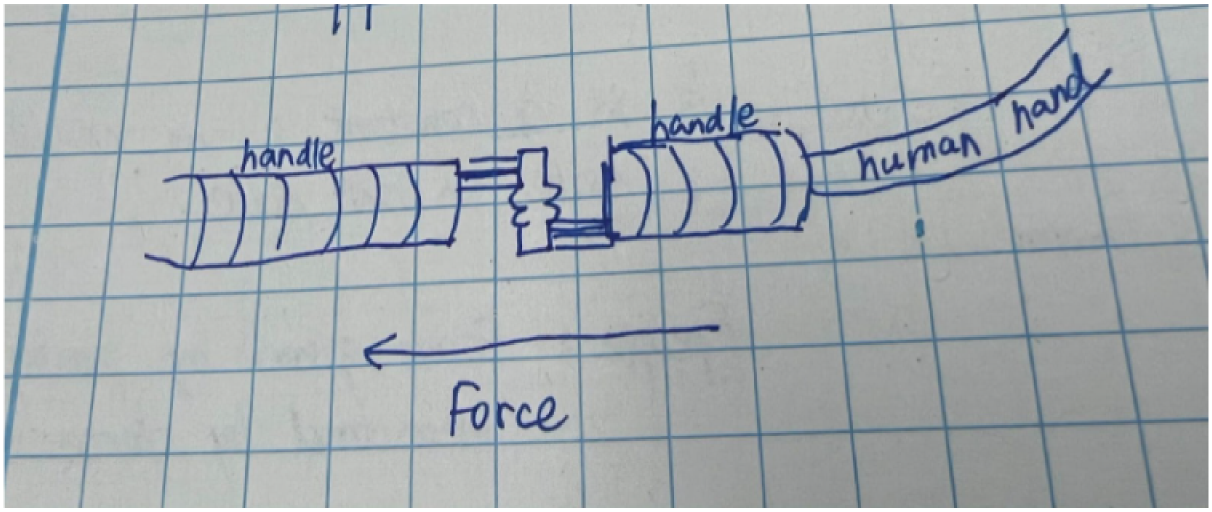



Figure 2: Load Cell Design Configuration

That way there is no interference of load cell implementation with the experience of pushing the wheelchair normally. However, when we got our product from the machine shop, load cells were installed in a different way and broke.

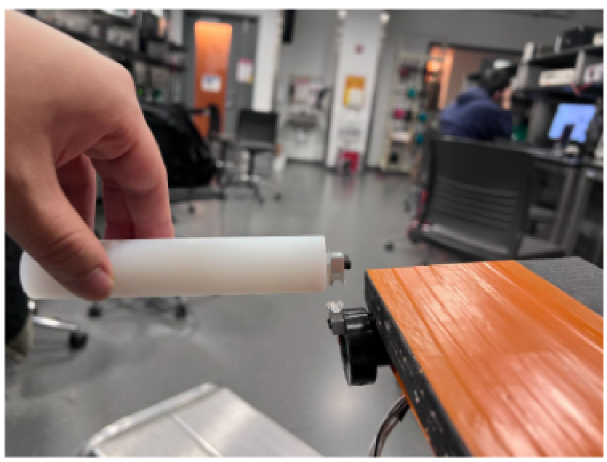


Figure 3: Broken Load Cell Configuration

We suspected that because of the way that load cells were installed differently, the torque from the handle exceeded the force from the handle to the load cell capacity limit and caused them to break.

Four colored wires on load cells, which are the same for all load cells, represent four different pin connections: red is voltage input, black is ground, white and green are voltage difference output. The voltage input for TAL221 100g is 5V.

The type of load cell amplifier we used was HX711.[2] The main function of this amplifier is to amplifier the voltage difference of the load cell's voltage output and transfer it into a readable signal by the microcontroller.

Due to permanent damage on load cells, we used a signal ultrasonic sensor as a replacement for our source of input, which takes the time signal of time the ultrasonic sensor returning to the sensor after it emitted. Since we only have one load cell, we placed a wood plank on two handlebars as a space to place the ultrasonic sensor. We placed the ultrasonic sensor in the middle of the wood plank so that the user can hover his or her hand on the sensor, which counts as intuitive feedback of the wheelchair movement based on the distance between the user's hand and the sensor.

The ultrasonic sensor we used is HC-SR04. Four pins connections are Vcc, Trig, Echo, and Gnd. Vcc is the voltage input, which is 5V. Gnd is the ground pin. Trig is a signal waveform from the microcontroller that triggers the output. Echo is a time signal waveform that tells how much time the ultrasonic wave takes to return back to the sensor from the blocking object.

### 2.1.3 Design Verification

The load cell itself is hard to test since the structure is a cuboid. While we got a voltage output when we twisted the load cell with our hand, we still needed a stable way to exert force on the load cell. We installed two layers of wood plank on each side of the load cell by screwing the plank into the holes on the load cell. We verified the load cell TAL221 by collecting the data of voltage output under different weights.

| Weight(g) | Voltage Output for 100g Load Cell TAL221(mv) | Suggested Output from Datasheet | maginal error(%) |
|---|---|---|---|
| 150 | 0.5978 | 0.6 | 0.366666667 |
| 174 | 0.6088 | | |
| 500 | 0.7011 | 0.7 | 0.157142857 |
| 1250 | 0.9996 | 1 | 0.04 |
| | | | 0.187936508 |

Table 1: Load Cell Test

As the data showed, the marginal error of the voltage output for our load cell is in our acceptable range. After testing the load cell, we connect the load cell with the voltage amplifier and our PCB. By using the same weights we used in voltage output testing, we were able to read the weight measurement on the Arduino application in our PCB by calling the HX711 library, which helped a lot in our later interaction with the motor controller.

## 2.2 Interface

### 2.2.1 Design Procedure

A switch button is provided for the person sitting in the wheelchair with a stop command. The stop command has the highest priority for safety concern, so that the person sitting in the wheelchair is able to stop it immediately. Because the top speed of running the wheelchair is controlled, when pressing the stop button, the momentum will not make the person sitting on the wheelchair fall off.

The type of the switch is "Tippette Full Sized Rocker Switches", which can be bought from the ECE Supply Center.

### 2.2.2 Design Verification

After the installation of the switch, we test its function under all circumstances including not moving, moving but not pushing, pushing, turning left/right. And in all situations, we could stop the running of the wheelchair to ensure safety.

## 2.3 Control Unit

### 2.3.1 Design Procedure

All the functions on translating input into output is implemented on the microcontroller (ATmega328p). Based on force detection, the magnitude of the force will convey to the microcontroller. This number will divide by 4500 (maximum force) to get a duty cycle number. The data of duty cycle will be used as a control signal and further go into the motor controllers.

ATmega328p was planned to be placed on our PCB board, but we encounter some problems of that PCB board, so we changed it into a Red Board instead. And due to the final change into ultrasonic sensor, We also did some adjustment on the functions. Basically for the ultrasonic sensor, lower the distance between hand and the sensor, larger duty cycle number is generated.

### 2.3.2 Design Details

The control unit consists of an ATmega328P microcontroller, an ISP module, a power supply section, pin headers, and LEDs:

- **ATmega328P**[3]: The high-performance microcontroller combines a 32 KB ISP flash memory, 2 KB SRAM, 23 general purpose I/O pins, SPI serial port, and a 6-channel 10-bit A/D converter.
- **ISP**: The control unit includes an in-system programming (ISP) module that is connected to the Serial Peripheral Interface (SPI) port of the ATmega328P, which allows us to program the control unit through an ISP cable.
- **Power supply**: The power supply (for control unit, which is between 7 V and 15 V) goes through two regulators that convert the voltage to 5 V and then 3.3 V, providing stable voltage for the control circuit (the operating voltage of ATmega328P is in between 1.8 V and 5.5 V) and external sensors (the operating voltage of the accelerometer is 3.3 V).
- **Pin headers**: In case more pins are needed in the future, e.g. more sensors are added, the control unit is designed in such a way that most of the pins are connected to pin headers, so we can easily make use of any of the pins simply through jumper wires, which greatly helps us when developing and debugging as more information could be transferred through unused pins.
- **LEDs**: In addition, two LEDs are used to indicate the state of the control unit, one for the power and one for the ISP module. Whenever the control unit is powered or the ATmega328P is being programmed, the LEDs are turned on.

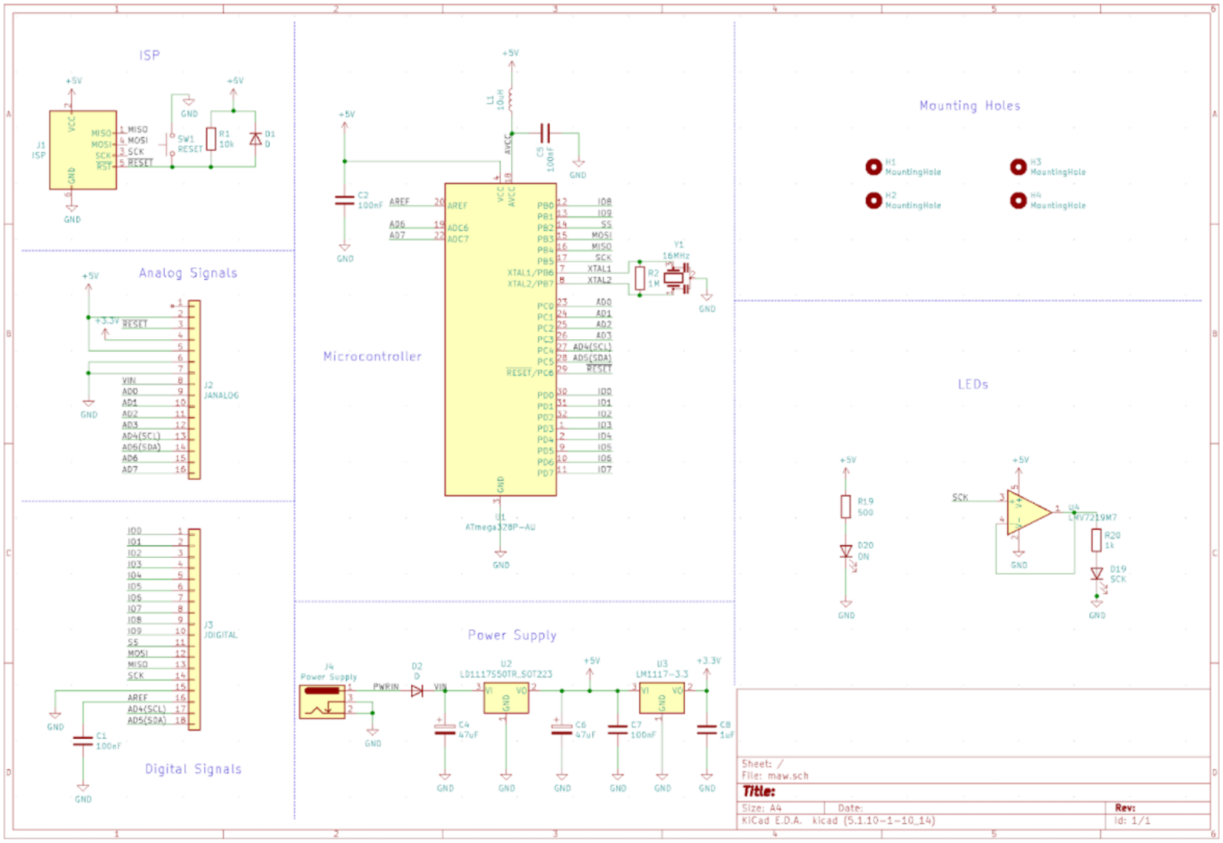
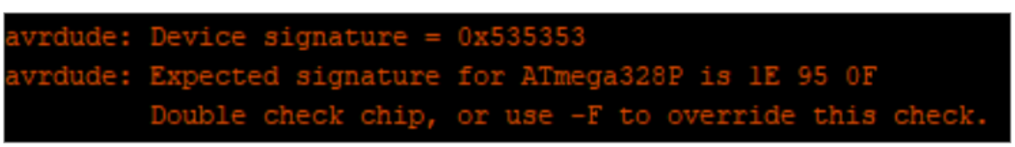



Figure 4: Schematic for the control unit, including ISP, analog signal header, digital signal header, microcontroller, power supply, LEDs, and mounting holes

### 2.3.3 Design Verification

The LED turns on when the battery is plugged into the barrel jack, and the pins expect GND, 3.3 V, and 5 V all provide the proper voltage. However, because of the incorrect crystal resonators purchased, the internal frequency of them are different from the microcontroller ones, thus causing an unexpected signature error, which prevented us from programming ATmega328P properly.



```
avrdude: Device signature = 0x535353
avrdude: Expected signature for ATmega328P is 1E 95 0F
         Double check chip, or use -F to override this check.
```

Figure 5: The error message shown during programming on the Arduino IDE

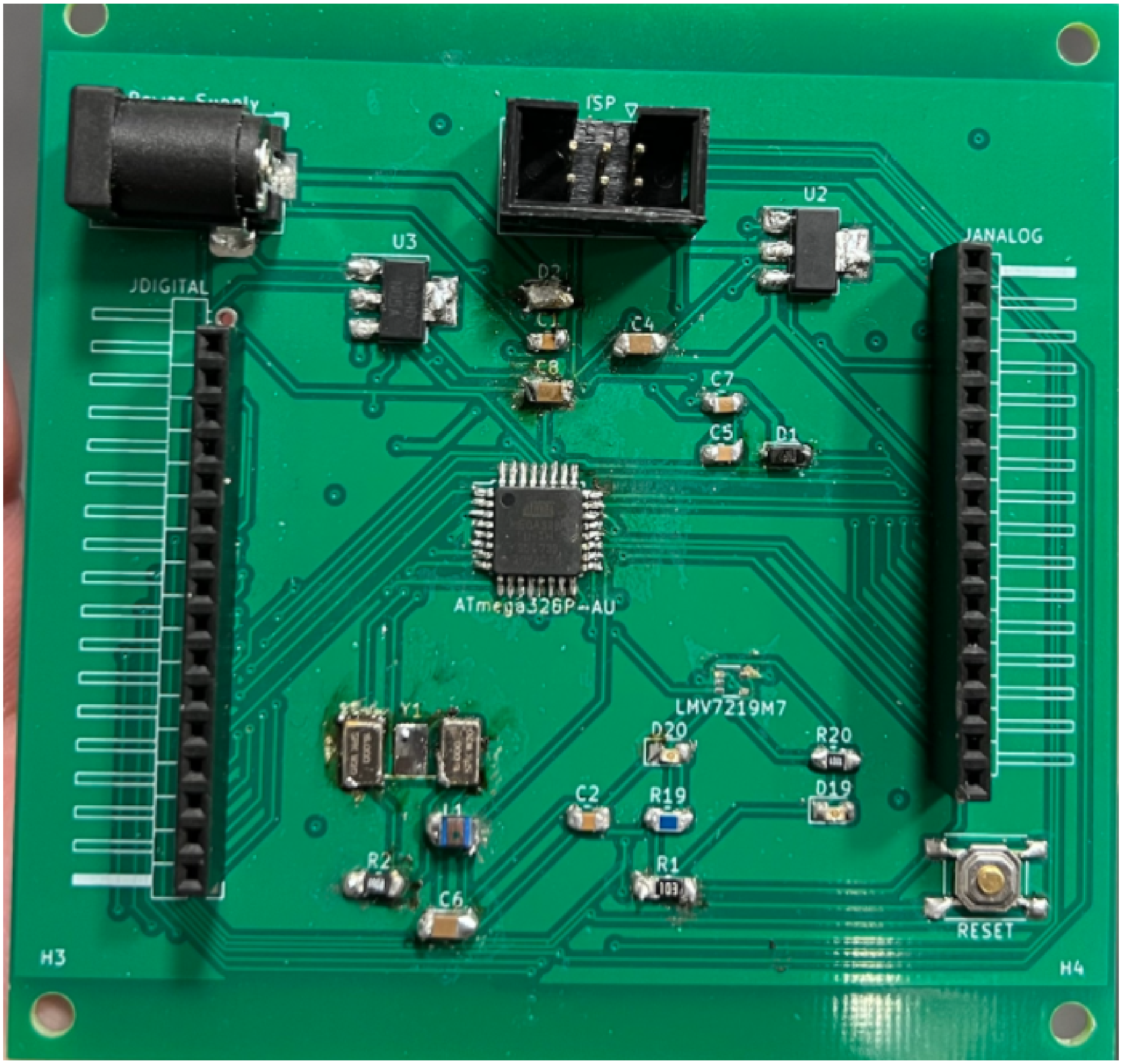



Figure 6: The printed circuit board that was built according to the schematic

## 2.4 Power Supply

### 2.4.1 Design Procedure

Our power supply is two Li batteries and both of them have an input voltage of 36 volts. This 36 volts will be given to the motor controllers and the motor controller will narrow down the voltage into 5V for powering into the PCB. PCB have regulators that could also adjust the voltage into 3.3V for the speed sensor. We used the motor controller to save one 9V voltage source.

### 2.4.2 Design Details

We choose to use two 36 Volts batteries to first power two motor controllers separately. We checked the parameter of the motor controller, which has a rated power at 500W.

$$500W/36V = 13.89A$$

We divide the rated power by the voltage to get the rated current. Then we used the 8000mAh to calculate the battery storage below.

$$8000mAh/13.89A = 0.576hour$$

Although the battery storage time is just around half an hour, the actual using time will be definitely longer. Because when we run the wheelchair, we will not run the motor at highest speed all the time. The actual current going through will be smaller.

### 2.4.3 Design Verification

We used the millimeter in the lab and connect the positive and negative bar on the battery to test its actual voltage.

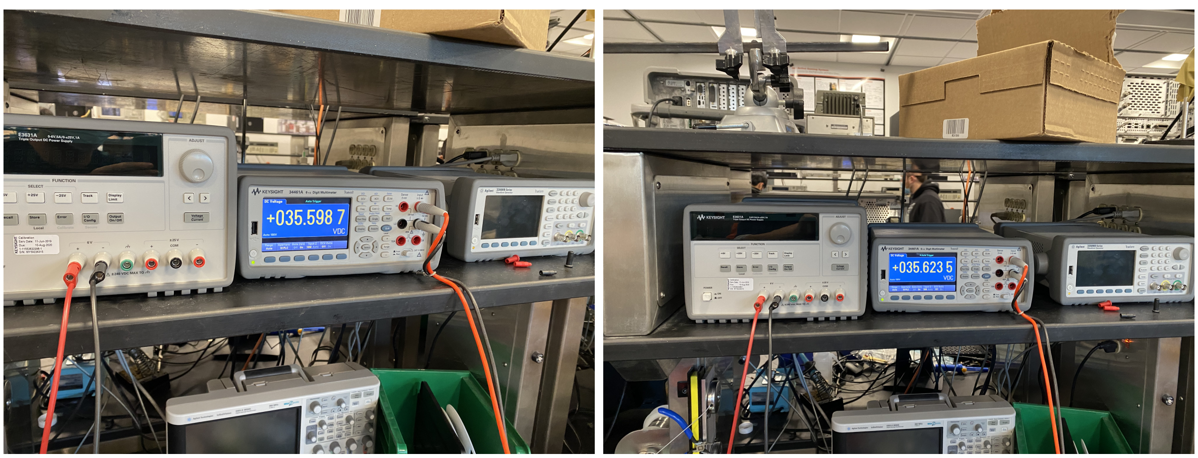



Figure 7: Voltage Tests on Batteries

## 2.5 Movement Unit

### 2.5.1 Design Procedure

The motor controller receive the duty cycle number from the microcontroller and use this to control the spinning speed of wheels. Larger the duty cycle, faster speed of wheels. Compared to the initial design, it cannot control the spinning direction of wheels, but the control of motors becomes easier.

We set a safety top limit speed at 1.5m per sec. However, we found that when we set the highest duty cycle to be 100, even sometimes when the motors spin very fast, it will slip on the wheels, so the forwarding speed will always below a safety speed.

### 2.5.2 Design Details

The movement unit consists of an accelerometer, two motor controllers, and two motors:

- **ADXL335**[4]: The digital output accelerometer that supports the 2 g and 4 g, and 8 g ranges.
- **YK-31C**[5]: The motor speed controller with rated voltage 36 V for motors with power up to 500 W.
- **BY1020D**: The motor with rated voltage 36 V, output 500 W, rated speed 2500 RPM, and rated current 17.8 A.

### 2.5.3 Design Verification

Three theoretical duty cycles, 83%, 67%, and 50% are picked to test the actual duty cycles of the PWM[6] signals from the microcontroller.

| Distance(cm) | Duty Cycle Measured(%) | Duty Cycle Theoretical(%) | Margin Error(%) | |
|---|---|---|---|---|
| 5 | 80.46 | 83 | 0.03060241 | |
| 10 | 65.03 | 67 | 0.029402985 | |
| 15 | 47.91 | 50 | 0.0418 | |
| | | | 0.033935132 | Total Margin Error |

Table 2: The actual duty cycle and the theoretical duty cycle

Three duty cycles 25%, 50%, and 75%, are picked to test the relationship between the duty cycle and the actual wheel speed. A piece of orange electric tape is attached to the wheel, and the movement was recorded by a camera. The wheel speed is given by $2\pi n/\Delta t$, where n is the number of circles the wheel rotates, and t is the time elapsed. The relationship is not strictly linear, but a clear increasing order can be seen from the data, which is enough for us to control the motors.

| Duty Cycle(%) | Wheel Speed(rad/s) |
|---|---|
| 25 | 3.15 |
| 50 | 8.05 |
| 75 | 9.51 |

Table 3: The relationship between the duty cycle from the microcontroller and the actual wheel speed

# 3 Costs

Our hourly salary is 30$/hour. We work 15 hours per week. We have three team members. Our labor fee is:

$$30\$/hour \cdot 2.5 \cdot 15hours/week \cdot 16weeks \cdot 3members = 54000\$$$

| Part | Manufacturer | Part Number | Quantity | Cost/Unit | Total |
|---|---|---|---|---|---|
| Wheelchair | Walmart | | 1 | $150 | $150 |
| Motor | Vevor | 500W 36V | 2 | $60 | $120 |
| Load Cell | Sparkfun | SEN-14727 | 2 | $8.95 | $17.9 |
| 16-pin Header | Adafruit | 2886 | 2 | $0.95 | $1.9 |
| AVR Programming Cable | Sparkfun | CAB-09215 | 1 | $1.95 | $1.95 |
| 2x3 Pin Header | Sparkfun | PRT-10877 | 2 | $1.5 | $3 |
| Crystal Resonator | Sparkfun | COM-00094 | 8 | $1.5 | $12 |
| DC Power Jack | Sparkfun | PRT-12748 | 1 | $1.5 | $1.5 |
| Push Button | Sparkfun | COM-08720 | 2 | $0.95 | $1.9 |
| Voltage Regulator-Adjustable | Sparkfun | COM-00595 | 4 | $1.77 | $7.08 |
| Voltage Regulator-5V | Sparkfun | COM-00107 | 1 | $0.95 | $0.95 |
| Motor Controller | Yiyun Tech | SPD-YK31C | 1 | $14.09 | $14.09 |
| Atmega328p | Sparkfun | COM-13448 | 5 | $4.25 | $21.25 |
| Accelerometer adxl335 | Sparkfun | SEN-09269 | 1 | $14.95 | $14.95 |
| Power Supply Jack | Adafruit | 373 | 2 | $0.95 | $1.9 |
| Voltage Regulator-5V | Digikey | 488-LM1117 | 5 | $0.6 | $3 |
| ISP Header | Mouser | 474-PRT-10877 | 3 | $1.5 | $4.5 |
| Comparator | RS-Component | 534-4771 | 3 | $3.42 | $10.26 |
| Load Cell Amplifier | Sparkfun | SEN-13879 | 2 | $9.95 | $9.95 |
| PCB Board | PCBway | | 20 | $2.05 | $41 |
| 100 nF Capacitor | Mouser | 80-C0805C104K5R | 20 | $0.08 | $1.6 |
| 10k Resistor | Digikey | 311-10KERCT-ND | 10 | $0.06 | $0.6 |

| | | | | | |
|---|---|---|---|---|---|
| 1M Resistor | Digikey | CSNL1206F T1L00CT | 10 | $0.49 | $4.9 |
| 500 Resistor | Digikey | TNPU08055 00RAWEN0 0CT | 3 | $4.9 | $14.7 |
| 1k Resistor | Digikey | RNCP0805 FTD1K00C T | 10 | $0.07 | $0.7 |
| LED | Mouser | APTD1608 QBC | 6 | $0.5 | $3 |
| Motor Controller | Amazon | YK31C | 2 | $15.99 | $31.98 |
| Red Board | Sparkfun | DEV-15444 | 1 | $19.95 | $19.95 |
| Electrical Tape | ECE Supply Center | | 2 | $3.5 | $7 |
| Ultrasonic Sensor | Digikey | 1528-2711 | 1 | $3.95 | $3.95 |
| | | | | **TOTAL** | $527.46 |

Table 4: Cost of the Project

# 4 Conclusion

## 4.1 Accomplishments

In this project, a sensor system that detects the push force or the distance from the hands to the handles and a control system that processes the data from the sensor system and then controls the speed of the motors is implemented. To sum up, a motor-aided wheelchair that helps people push is completed.

## 4.2 Uncertainties

The wiring was done by both soldering and electric type, but there might still be disconnections. In addition, the speed of the two wheels are not the same even given the same power, which increases the difficulty to keep the wheelchair move straight.

## 4.3 Ethical considerations[7]

Since our project is a motor-aided wheelchair, we took patients' safety into our consideration. The relationship between patients' safety and motors' power is a tricky problem to deal with. However, after experiments, we have balanced the maximum acceleration and maximum speed of the wheelchair and the efficiency so that it is able to both save significant effort from the pusher and give the person sitting in a comfortable experience.

## 4.4 Future work

Firstly, we want to resolve the issue of PCB and load cell to make our project be more close to our original design. Then we planed to add more weight in the front part of the wheelchair to change its center of mass since we have the whole system installed on the back side of the wheelchair. Third, we will need to move the position of the motor to the axis of the wheels so that we can get a more accurate feedback on how fast the wheel spins instead of doing measurement on our own. Last but not least, we can put more functions on the interface unit. For example, adding buttons for the person sitting on the wheelchair to control the turning direction of the wheelchair.

# References

[1] TAL221 MINIATURE LOAD CELL, datasheet, web page, Available at:
https://cdn.sparkfun.com/assets/9/9/a/f/3/TAL221.pdf

[2] Load Cell Amplifier - HX711, datasheet, web page, Available at:https://cdn.sparkfun.com/assets/b/f/5/a/e/hx711F_EN.pdf

[3] ATmega328P, Microchip, web page. Available at:
https://www.microchip.com/en-us/product/ATmega328p

[4] Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometers, Analog Devices, web page. Available at:
https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_adxl355.pdf

[5] Speed Controller SPD-YK31C Installation and Wiring, Electric Scooter Parts, web page. Available at:
http://www.electricscooterparts.com/hookup/SPD-YK31C.html

[6] IEEE Code of Ethics, web page. Available at:
https://www.ieee.org/about/corporate/governance/p7-8.html

[7] Pulse-width modulation, Wikipedia, web page. Available at:
https://en.wikipedia.org/wiki/Pulsewidth_modulation

[8J ames F. Cox and Leo Chartrand, "Nonsinusoidal oscillators" in Fundamentals of Linear Electronics: Integrated and Discrete (2 ed.), Cengage Learning, June 26, 2001.

# Appendix A Requirement and Verification Table

**Table 1: System Requirements and Verifications**

| Requirement | Verification | Verification status (Y or N) |
|---|---|---|
| 1. Requirement: Force cells are able to detect the magnitude of the force exerted with a margin of error of 3%: [0,1500g45g], and give voltage difference, which is proportional to the magnitude of force and has an error margin of 0.15 mV from the microcontroller. | 1. Verification: Connect load cells to a multimeter on the testbench. Put weights of 100g, 500g, and 1500g respectively on the top of the load cells. Read and record values shown on the multimeter. Check if the readings corresponding to the above weights are in the ranges [0.6 mV - 0.15 mV, 0.6 mV + 0.15 mV], [0.7 mV - 0.15 mV, 0.7 mV + 0.15 mV], and [1.0 mV - 0.15 mV, 1.0 mV + 0.15 mV] respectively. If the readings fall into each range properly, then requirement is fulfilled. | Y |
| 2. Requirement: A button that sends the stop signal to the microcontroller and has the highest priority. | 2. Verification: Connect the button with PCB. Press the button under each of the following situations: Not moving, Moving but not pushing (Decelerating), Pushing (Accelerating), Turning left, Turning right, If the LED on PCB turns off under all of the situations mentioned above, requirement is fulfilled. | Y |
| 3. Requirement: The microcontroller is able to track the signal from sensors. | 3. Verification: Connect the microcontroller to a computer with Arduino IDE. Print out the data from the sensors through the serial port by Serial Monitor in Arduino IDE. | Y |
| 4. Requirement: The microcontroller is able to output PWM signals, and the average voltage of the PWM signals should be from 0 to 36 V. | 4. Verification: With the last requirement verified, connect the PWM output pin to an Analog input pin. Print out the PWM signal received through Serial Monitor. The maximum of the PWM signal should be near 1024 and the minimum should be near 0. | Y |

Continued on next page

**Table 1 – continued from previous page**

| Requirement | Verification | Verification status (Y or N) |
|---|---|---|
| 4. Requirement: The power supply is able to output a voltage in the range [35 V, 36V]. | 4. Verification: Connect the output of the power supply to a multimeter, if the reading from the multimeter is in the range [35 V, 36V], then the requirement is fulfilled. | Y |
| 4. Requirement: The motors are able to drive the wheelchair with a maximum speed of 1.5 m/s. | 4. Verification: Keep increasing the push force until the motors output the highest power. When the wheelchair reaches the maximum speed, time 5 seconds. If the distance that the wheelchair moves is more than 7.5 m, then the requirement is fulfilled. | Y |