Ryerson University
Department of Electrical, Computer & Biomedical Engineering
Faculty of Engineering & Architectural Science

Smart Wheelchair for Physically Challenged Individuals

by

Abdalrahman Alfakir, Hamza Mahdi, Nadim Hafez

Supervisor: Bassma Ghali

Biomedical Engineering Capstone Design Project

Ryerson University, 2019

Introduction

The first power wheelchair (PW) was invented after World War 2 to accommodate people that have spinal cord injury which causes paralysis in all four limbs [1]. Since then, many technologies have been used to control PW such as joystick operated with fingers, sip and puff systems, tongue drive systems, head and chin control and voice-activated commands [2]. Recent innovations include head tracking systems similar to head tracking mouse systems [3], and smart devices with cameras such as the Gryphon Shield and the iChair [3]. EEG based systems have been implemented and are commercially available but are not very reliable in applications because it is too sensitive for errors and requires a lot of signals filtering [4].

Therefore, this project used head tilt input, which is cheap, safe and intuitive to use. An accelerometer/IMU was used to track head tilting movements which can be mapped to motor commands. These sensors have a fast response time, are sensitive to small changes and have low power consumption and those changes are used to track sudden movement such as falling which can be useful in calling emergency services automatically. The user-friendly nature of such input method allows for modularity and ease of transfer between difference wheelchairs with small setup time and automatic calibration without expert help.

The user can activate/deactivate the motion of the wheelchair through head movement that is discussed later in the report. The wheelchair implemented innovative features such as detection of obstacles using light detection and ranging (LiDAR) which is suitable for indoor and outdoor use. A GPS is mounted on the wheelchair to track the coordinates of the wheelchair which can be used in cases of emergency to locate the wheelchair. The project is set up so those future revisions can include the implementation of an automation feature where the wheelchair can navigate indoors using simultaneous localization and mapping (SLAM). The GPS, coupled with magnetometer data from the IMU, may be useful for autonomous navigation of wheelchair without a prebuilt point cloud map. However, this functionality is outside of the scope of this project.

Acknowledgments

Special thanks to John and his wife for donating the wheelchair. This project would not have existed without their kindness.

The team would like to thank Ryerson Rams Robotics (R3) for lending the team their equipment and allowing the team to use their space. This project would not be of the same quality without their generosity.

Special thanks to Luka Subotincic for his help in taking the stock control system off the wheelchair.

Thanks to Feroz Balsara, Karan Guglani, Omar Sharif, Michel Kiflen, Anas Mahdi, Matthew Mirvish, Trevor Slawson, Kareem Al-Halib and Reem Alsati for testing the wheelchair and providing feedback.

Special thanks to Dr. Bassma Ghali for her guidance and supervision throughout the project.

Certification of Authorship

We certify that we are the authors of this paper and that any assistance we received in its preparation is fully acknowledged and disclosed in the paper. We have also cited any sources from which we used data, ideas or words; either quoted directly or paraphrased.

Table of Contents

Abstract

Individuals with quadriplegia cannot use joystick controlled power wheelchairs (PW) due to lack of force or psychomotor problems in the superior members [1]. For that reason, the need for an alternative method to control the wheelchair is a must. A smart wheelchair is a power wheelchair in which a multitude of sensors may be used to control it. There exist zero force control methods in which the user does not have to move their limbs [1]. In the "Smart Wheelchair" project, the traditional joystick is replaced with an Inertial Measurement Unit (IMU) which is placed on a wearable hat. The idea is to place the hat on the user's head, which will measure the roll, pitch, yaw angles of the head. The hat on the user's head is used as input to determine the desired velocity and steering direction of the wheelchair. Also, the "Smart Wheelchair" is equipped with a light detection and ranging (LIDAR) sensor that enables obstacle detection and avoidance which increases the safety of both user and surroundings. The "Smart Wheelchair" aims to increase the user's independence in terms of mobility. With the project being wholly modular and open source, it is easily transferable to any power wheelchair and can be replicated by anyone who has access to the documentation.

Keywords: Power wheelchair, Quadriplegia, IMU, LIDAR, Object detection, Open source

Objectives

There are two main objectives behind this capstone project:(1) to create an alternative control method for power wheelchairs with quadriplegic users, and (2) implement obstacle detection as a safety feature to prevent the wheelchair from running into objects, walls and such.

An additional goal of our team is to present a cheap and open source solution to the wheelchair problem while assisting individuals with spinal cord injury and other disorders of the central nervous system. To date, there is no open source hardware and software platform for full-sized electric wheelchairs. Our project is the first of its kind providing people with a reconfigurable wheelchair software platform that is entirely free. Our code is on a public GitHub repository which enables code sharing and reusability especially as every data point goes through the ROS (Robot Operating System) ecosystem, which mean that the code used in this project can be used regardless of the sensor brand used with minimal modifications required.

A long term goal is to achieve autonomous navigation through the use of GPS, IMU, and LIDAR and make it intuitive to use. This was not accomplished in the span of BME800 due to time constraints. The project is set up to enable autonomous navigation in future iterations of the project.

Background

Many technologies have been used to control PW such as joystick operated with fingers, sip and puff systems, tongue drive systems, head and chin control and voice-activated commands [2]. Recent innovations include head tracking systems similar to head tracking mouse systems [3], and smart devices with cameras such as the Gryphon Shield and the iChair [3]. EEG based systems have been implemented and are commercially available but are not very reliable in applications too sensitive to error [4].

Sip and Puff systems are simple to implement and relatively cheap. However, they provide limited input, may be a source of strain on the user, and has high chances of microorganisms inhabiting the air tubes. Besides, research has shown that other systems such as tongue drive systems are more effective than sip-n-puff systems in complex motions such as turning and maze navigation [5]. Head and chin controls require individuals with head movement and may use limit switches, touch sensors or accelerometers. These systems can be cost-effective and easy to use with minimal training [1]. Head and eye tracking devices are easy to implement nowadays as personal smart devices can be mounted on a mount-n-mover system which is modular, relatively cheap and can be transferred from wheelchair to another which is desired especially for children who change their wheelchairs regularly [2]. EEG signal collection, while alluring, present a challenge in signal processing, the strength of signal collected and require a lot of research and development work. Some attempts at collecting stronger motor cortex signals involve invasive methods such as surgical implantation of electrodes [7] which is not desired by users nor is it in the scope of our project.

Theory

The wheelchair uses a few basic principles for collecting sensory data, processing them, and actuating its motors upon such data. In this section, the theory behind sensors (IMU, LIDAR), motion control (brushed dc motors, PWM, H-bridge motor controllers, closed-loop feedback) and data exchange (Bluetooth, various serial protocols) are discussed.

Inertial Measurement Unit (IMU):

The IMU is an electronic element that is composed of measure three sensors: accelerometer, gyroscope, and a magnetometer.

The accelerometer allows the determination of pitch, yaw and roll angles for which the math will be shown later on in this section. The accelerometer sensor works by measuring the difference between any linear acceleration in the accelerometer's reference frame and the earth's gravitational field vector [8]. In the absence of linear acceleration, the accelerometer output is a measurement of the rotated gravitational field vector and can be used to determine the accelerometer pitch and roll orientation angles [8]. In short, a proof mass exists on the surface of the Micro-Electrical-Mechanical-System, and the deflection of this mass will change the capacitance between fingers of the proof mass and the sensing plates. The capacitance is converted to voltage and amplified, then digitized and sent over serial to the microcontroller. Figure 1 shows a simple transducer model and its equivalent circuit :

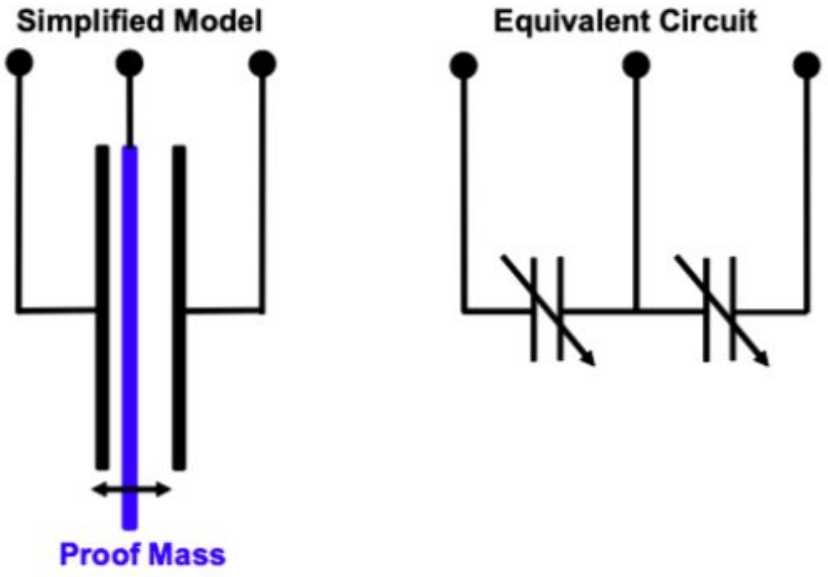

Figure 1: Simplified Transducer Model and Equivalent Electrical Circuit [8]

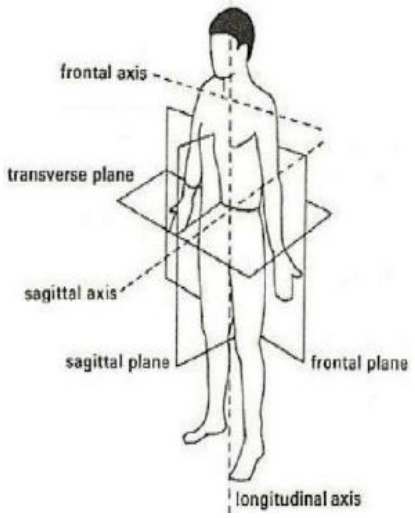

Figure 2: Body Planes [9]

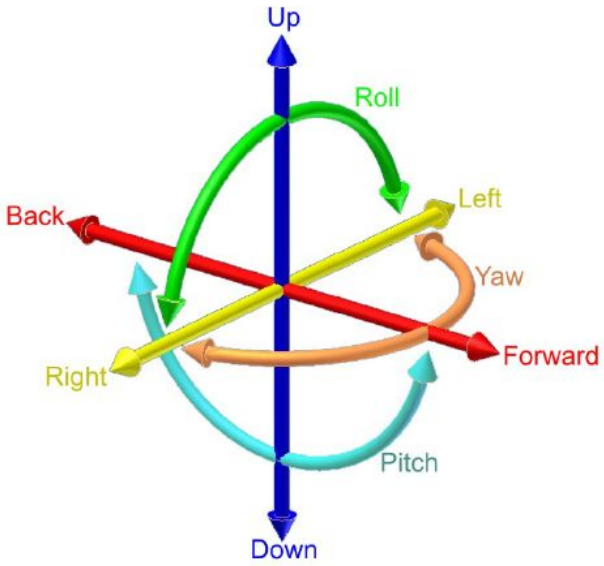

Figure 3: 3D axises [10]

The accelerometer acceleration is measured in the earth's reference frame, r in the earth's gravitational field g will output Gp [8]:

$$G_p = (G_{px}\, G_{py}\, G_{pz}) = R(g - a_r) \qquad (1)$$

where R is the rotation matrix describing the orientation of the sensor relative to the earth's coordinate frame.

There are different ordering methods that can be used when calculating the rotation matrix: R_{xyz} , R_{yxz} , R_{xzy} , R_{yzx} , R_{zxy} , R_{zyx}

We use Rxyz which is termed the "aerospace rotation sequence"

$$\boldsymbol{R}_{xyz}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)\begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$= \begin{pmatrix}-\sin\theta\\ \cos\theta\sin\phi\\ \cos\theta\cos\phi\end{pmatrix}$$

Figure 4: Aerospace Rotation Sequence [8]

The roll and pitch angles can be calculated as shown below:

$$\tan(\varphi_{xyz}) = \frac{G_{py}}{G_{pz}} \qquad (2)$$

$$tan(\theta_{xyz}) = \frac{-G_{px}}{\sqrt{G_{py}^2 + G_{pz}^2}} \quad (3)$$

Where : ϕ roll angle, θ = pitch angle, Ψ = yaw angle, G_p = matrix of constants, R = rotation matrix, g = gravitational constant, a_r = accelerometer acceleration.

The yaw ψ is not calculated around the IMU z-axis which is initially aligned with the gravitational field and pointing downwards. All accelerometers are entirely insensitive to rotations about the gravitational field vector and cannot be used to determine such a rotation. For that reason, the yaw rotation will be calculated using the magnetometer sensor inside the IMU.

The magnetometer outputs x and y readings which correspond to the Hx and Hy components of the earth's magnetic field (He) [11]. The compass heading can be determined using the following set of equations [11]:

Direction (y>0) = 90 - [actan(x/y)]*180/pi (4)
Direction (y<0) = 270 [actan(x/y)]*180/pi
Direction (y=0, x<0) = 180.0
Direction (y=0, x>0) = 0.0

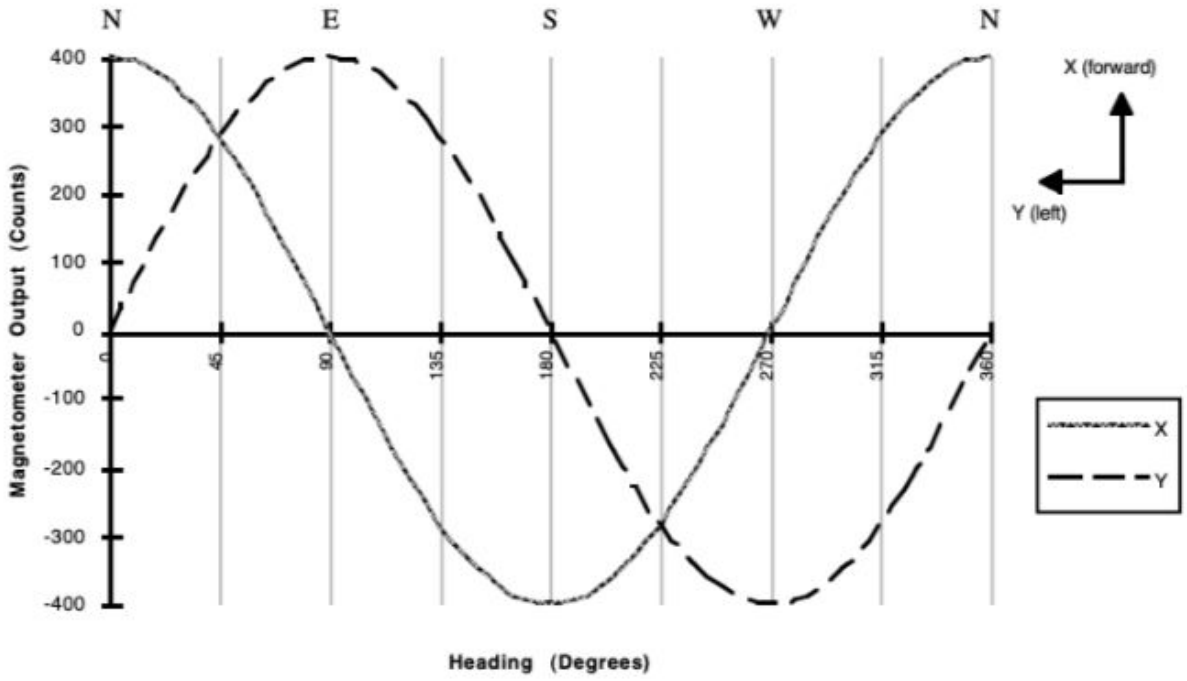

Figure 5: Hx and Hy Magnetometer Readings for Different Compass Headings [11]

The Gyro sensor inside the IMU can be used to sense rotations since it detects the relative rotations based on Coriolis force [12]. The sensor outputs data in deg/s which can be used to calculate the degrees relative to when the gyro was reset. This concept is of little importance to the project and is not used in the current design (see design section).

The components of the IMU must also be calibrated especially the Gyroscope and Magnetometer. Many of the cheap IMUs such as the one used in this project, will contain systematic errors which are a byproduct of incorrect scaling factors and axis misalignments [13].

What this does is lowers the accuracy of the output reading of the sensors. Another notable problem present with 'off the shelf' IMUs is the gyro drift that occurs over time [14]. The gyro drift gradually introduces a drift in the sensor of the gyroscope due to two main components: bias instability and angular random walk [14]. Many filters have been developed to remove these biases and calibrate the sensors to decent output results. One of these examples is the Kalman filter which uses mathematical equations to find recursive solutions for past, present and future states of the system, in this case, the IMU [15]. A more simplified version of the Kalman filter is the RTQF filter which is the default filter used in the RTIMULIB library [15]. The way that this filter works is by calculating different quaternions for the gyroscope and another for the accelerometer and magnetometer [15]. The gyroscope quaternion coordinates are initially stable but contain drift while the second quaternion is unstable and drift absent [15]. Therefore, these two coordinates are compared with one another through the Spherical Linear Interpolation technique (Slerp) to find an intermediate coordinate [15]. Slerp takes into consideration a power threshold that ranges from 0 to 1 which is set at 0.02 for RTIMULIB [15]. To elaborate even further about the power, a value of 0 would correspond to the sole use of the gyroscope quaternion while a 1 would only depend on the accelerometer and magnetometer quaternion [15]. Therefore, this technique depends more heavily on the gyroscope than on the other two components.

The actual calibration acquisition consist of recording the compass minimum and maximum values of the magnetometer in the x, y, and z directions. It then takes these values and scales the reading based on these magnitudes [15].

Light Detection and Ranging (LIDAR):

The LIDAR sensor works by creating a 2D plane of its surroundings. This is done by pulsing light waves within a specific frequency to its environment and then detecting the time it takes for the wave to come back. By keeping track of the time it made the light wave to go and come back, the device can calculate the distance of the object that the wave hit from the LIDAR source with the use of equation 5 [16].

$$Distance = \frac{time\ of\ travel \times speed\ of\ light}{2} \quad (5)$$

The sensor is spun on a rotating table allowing it to scan in all directions. For that reason, the sensor enables the creation of planar environment maps which will allow obstacle detection. The data can be stored for use in SLAM based navigation or processed in real time to avoid obstacles via windowing data.

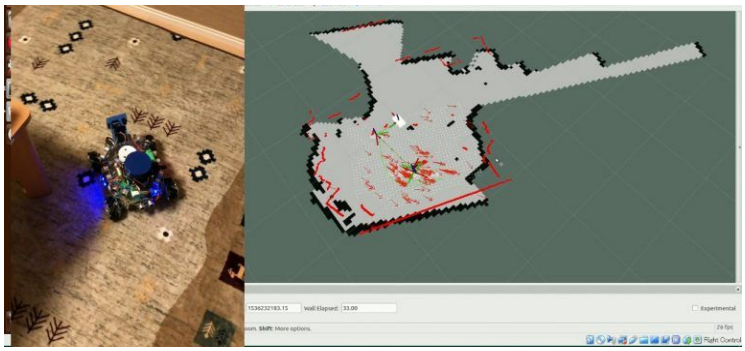

Figure 6: Example of LIDAR Use in Simultaneous Localization and Mapping (SLAM) [17]

Brushed DC Motor:

Brushed DC motors rely on the interaction between the electromagnetic fields of a fixed part (stator), and a rotating part (rotor). The magnetic attraction/repulsion induce an electromotive force which results in motion of the motor [18]. As the motor moves, magnetic poles switch orientation which ensures continuity of movement. The magnetic field switching is accomplished via the brushes as shown in the figure below; as the motor rotates, the current supplied by the brushes to the armature switch every half turn [18].

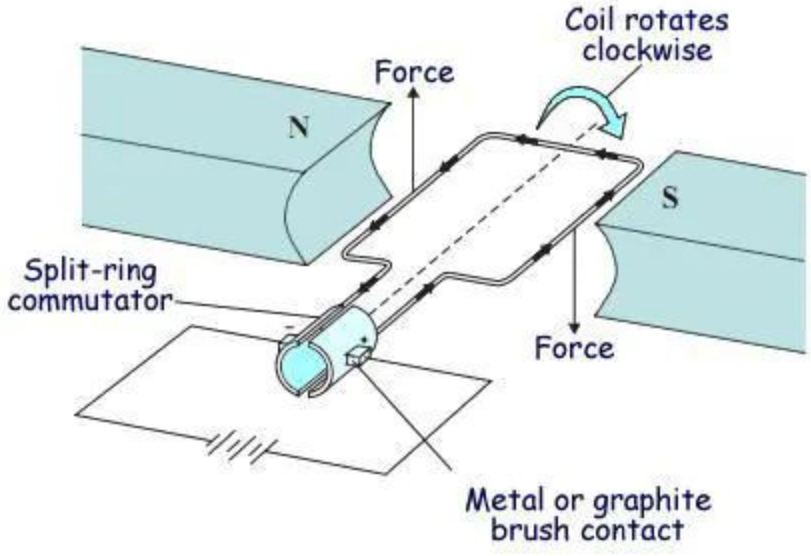

Figure 7: Generic DC Motor Setup [18]

The simplicity of brushed DC motors allow them to run without a controller given a power source, however, this does not provide torque/speed control. Motor controllers are used to vary the power delivered to DC motors resulting in variations in the torque/speed output curve.

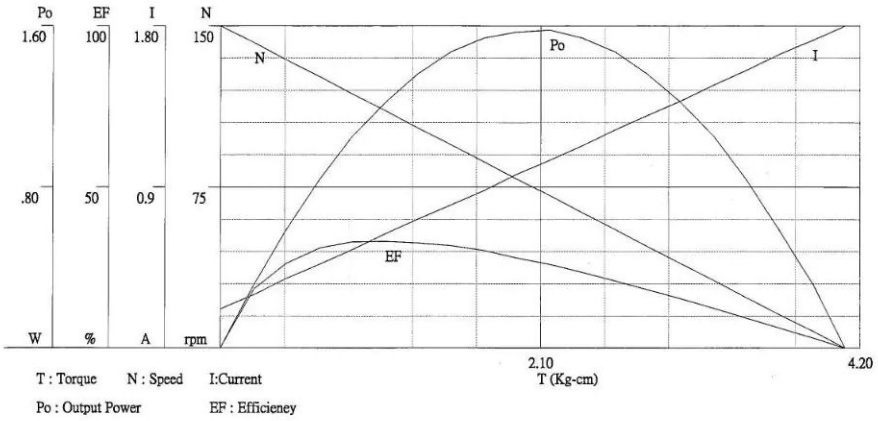

Figure 8: Operating Characteristics of a Generic Brushed DC Motors [18]

Motor Control Using Pulse Width Modulation (PWM):

The wheelchair electric DC motors' power needs to be regulated. A connection straight to the battery with a controlled switch in between would result in maximum motor output whenever the switch is closed. This a widely known issue and DC motors are usually controlled via Pulse-Width-Modulation (PWM). An electrical signal is discretized and sent over a specific frequency/period. The PWM value is said to have a duty cycle of 100% when the digital signal is high for the entirety of its period, and a duty cycle of 0% when the signal is low for the entire period. Figure 9 below illustrates the PWM duty cycle control:

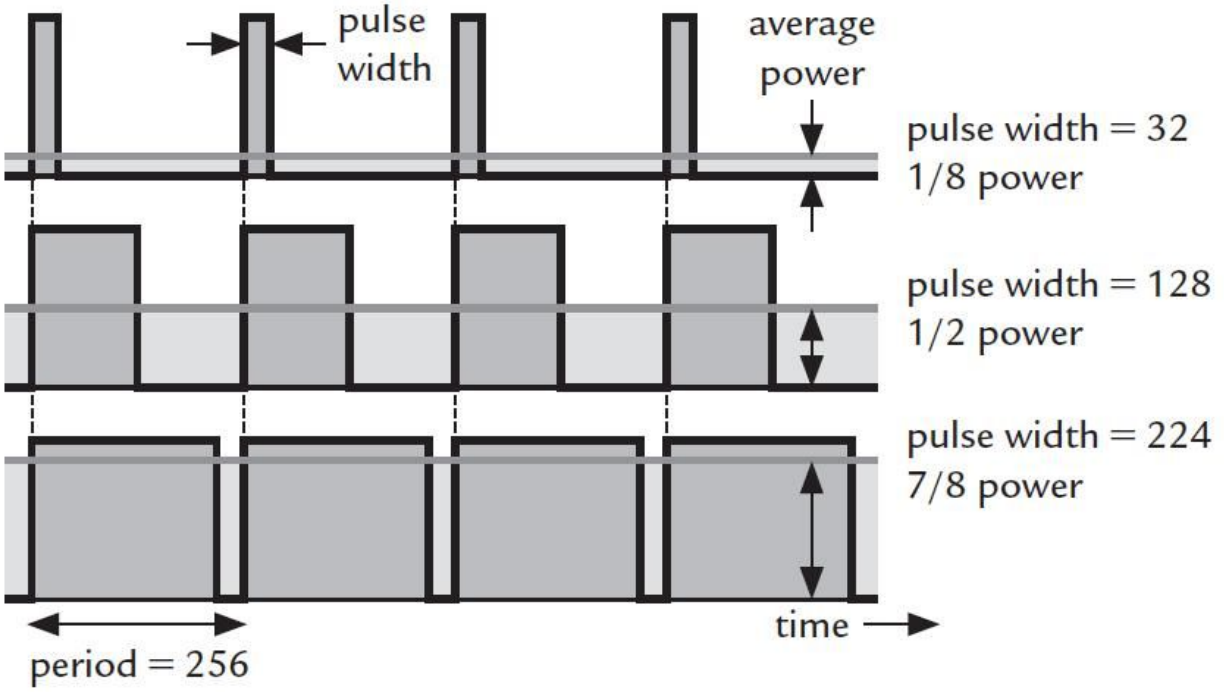

Figure 9: Duty Cycle in Pulse Width Modulation [19]

DC motors are driven by motor controllers that receive PWM signals and modulate the motor power. A motor controller is necessary to isolate the motor power from the logic controller as a microcontroller cannot provide enough current for the motor load. The diagram below shows a typical brushed DC motor control scheme.

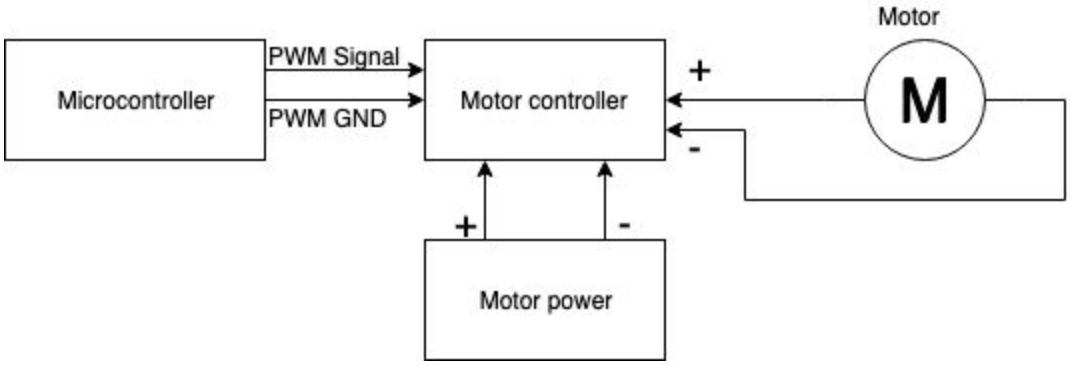

Figure 10: Typical DC Motor Control Scheme

Motor Controller:

As mentioned above, a motor controller is used to isolate the motor power source from the control signal and allows for torque/speed control of a motor. Brushed DC motors are often controlled using an H-bridge, which is a simple circuit containing four switching element, with the load at the center as shown in figure 11 below [20]:

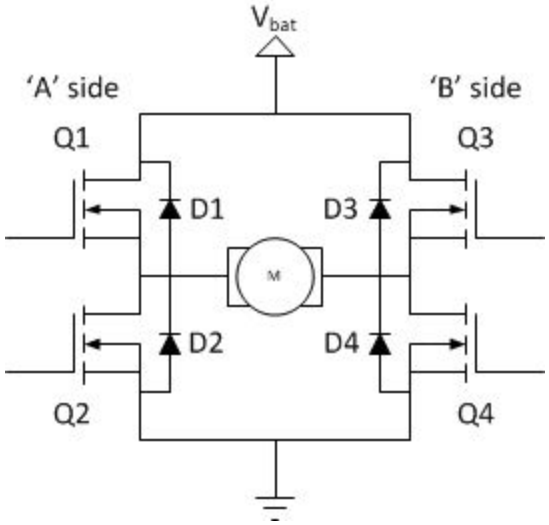

Figure 11: H-Bridge Configuration [20]

The switching elements shown in the figure above are typically FET transistors [20]. When Q3 and Q2 are closed, current flows through them allowing for the right side of the motor shown above to be the "A" side while the other side is grounded. This is shown in the figure below:

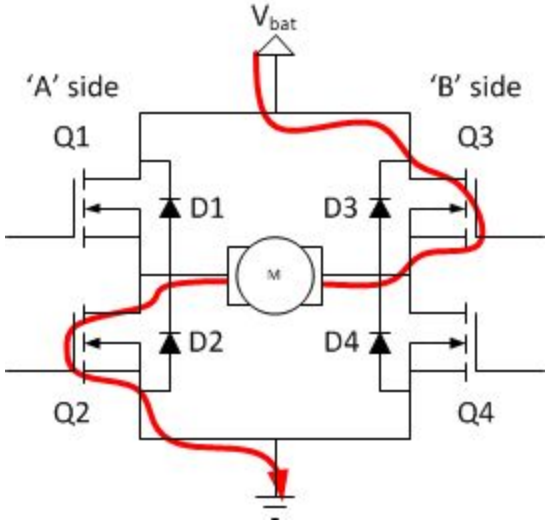

Figure 12: H-Bridge Operation [20]

If Q3 and Q2 are open and Q1 and Q4 are closed, current flows in the other direction resulting in opposite operating polarity as the motor's side "B" is being powered while side "A" is grounded. When the H-Bridge receives PWM signal, the switches open and close at the desired rate allowing for power control [20]. The diodes shown in the circuit are known as catch diodes and they provide a path for the motor current during the switching period [20]. This is very important to prevent damage to solid-state switches.

Another possible source of damage to H-Bridges is if two switches from the same side are on; this creates a short circuit situation since the resistant in the motor is too high compared to two switches (Q1 and Q2 or Q3 and Q4) [20]. This is shown in the figure below:

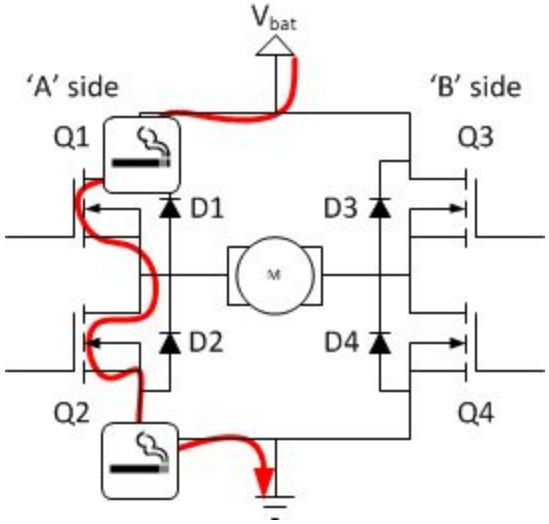

Figure 13: "Shoot-through" Situation Where Electrical Components are Damaged [20]

DC- DC buck converter:

A buck converter is used to step down the DC voltage from the input to the output. Its principle of operation is that the inductor inside the circuit of the buck converter works to prevent sudden changes in the input current [21]. The input of the buck converter is DC voltage, the load is a resistor, and the capacitor output is to help maintain the energy [21]. The buck converter has two switches, one of them is the MOSFET, the other one is a passive switch which includes the diode.

a. when the switch is on.

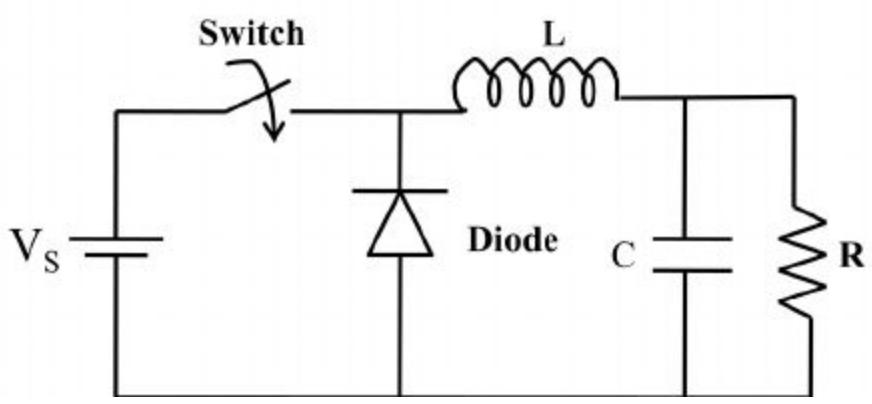

Figure 14: Buck Converter With Switch On [21]

When the switch is on, the Vx area is where the diode will be grounded, which means that the diode in the figure above is reverse biased diode, and accordingly the current will start to increase, the inductor will generate an opposing voltage throughout its terminal in reaction to the changing current and diode blocks.

b. when the switch is off.

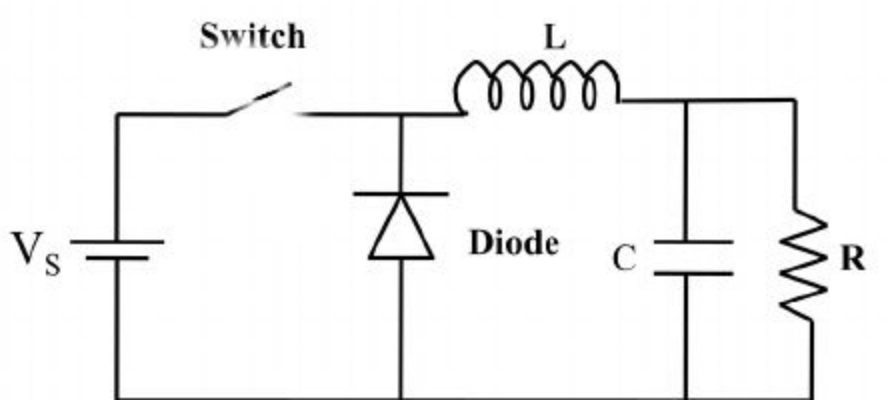

Figure 15: Buck Converter With Switch Off [21]

When the switch is off, which means the circuit is open, the voltage source will be removed from the circuit, and the current source will be the inductor. The current produced by the inductor will begin to decrease and therefore result in voltage drop within the inductor, and the current in inductor keeps decreasing until it reaches zero [21]

Closed Loop Control:

In order to accurately control motor shaft position/speed, a feedback loop is required. Motors are often paired with sensors to provide feedback on their rotational positions, with some motors even incorporating a sensor and a control circuit in their packaging (servos). An encoder can be used for relative, continuous rotational measurement, while a potentiometer can be used for absolute angle (range limited) measurement. Using a simple closed loop structure allows for a stable, predictable operation that can be fine-tuned by utilizing gains and lead/lag controllers.

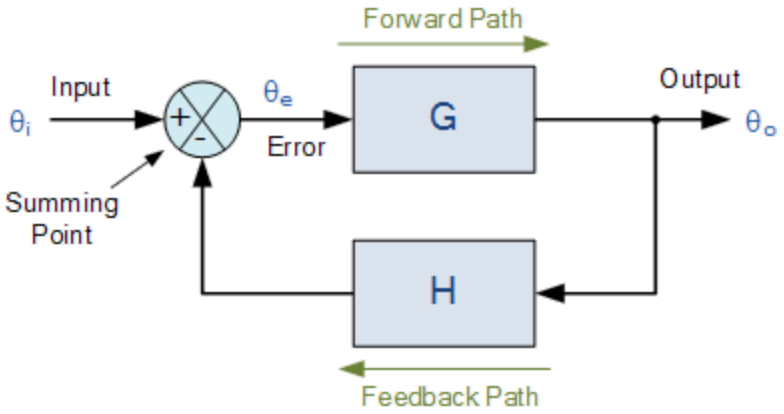

Figure 16: Typical Closed-loop System Representation [22]

The output is fed back to be added or subtracted from the input after scaling (H). If the feedback path is subtracted from the input, the loop is termed a "negative feedback loop". Similarly, if the feedback path is added to the input, it is a "positive feedback loop".

The use of closed loop feedback is quite simple to implement. A sample C code for the implementation of a feedback loop is shown below:

```
int target = TARGET;
int error, output;
float kp = 1.0;//can be tuned
while(true){
        error = target- sensor_value();
        output = error*kp;
        motor_power = output;
}
```

The addition of an integral term would decrease steady-state error but increase oscillations and transient response as a result. This can be combated by adding a derivative term which decreases oscillations and transient response as well as increasing the system stability. Sample code for a full proportional, integral, derivative loop (PID) is shown below:

```
int target = TARGET;
int error, error_integral, error_derivative,prev_error, output;
struct pid{
        float kp, ki, kd;
};//assign values to each constant after tuning the parameters
while(true){
        error = target- sensor_value();
        error_derivative = error - prev_error;
        error_integral += error;
```

```
        output = error*pid.kp + error_derivative*pid.kd + error_integral*pid.ki;
        motor_power = output;
        prev_error = error;
}
```

Bluetooth:

Bluetooth is a short wavelength wireless communication standard that operates in the 2.400 to 2.485 GHz range [23]. It is used in almost every modern laptop/phone due to its effectiveness at short range communication, low power, and low cost. Bluetooth uses point to point connection and is meant to replace cables/wiring between proximity devices [23]. Its core system consists of an RF transceiver, baseband and a protocol stack [23]. Transceiver complexity is minimized by employing a shaped, binary frequency modulation, and frequency hopping are used to reduce interference and fading [23]. Bluetooth operated in a master/slave manner, meaning that the master device is synchronizing all clock signals, while the slave device connects to the master. The Bluetooth logical link control and adaptation protocol (L2CAP) allows for packet sending between two devices which simplifies the process as a simple serial port emulation such as RFCOMM can be used to communicate with the onboard Bluetooth device while it handles all data sending and receiving automatically. Bluetooth connections are encrypted by a shared key when two devices are first paired, which provides a necessary layer of safety options [23].

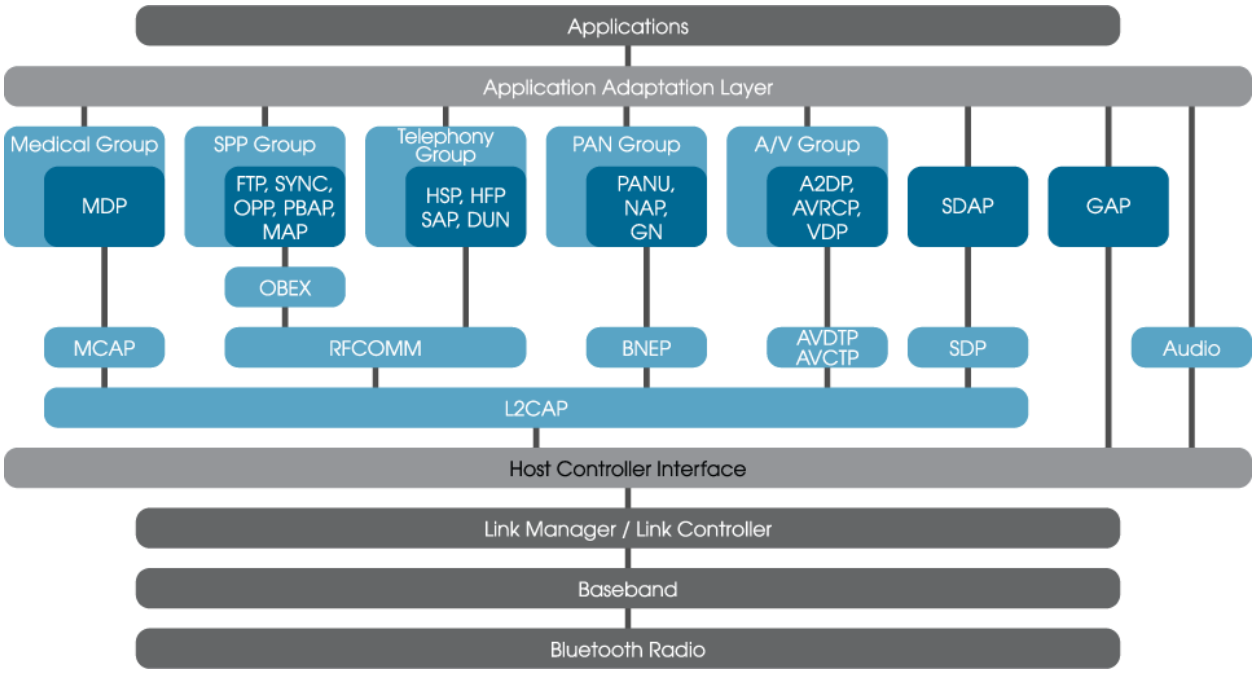

Figure 17: The Bluetooth Protocol Stack [24]

Serial Communication:

Serial communication is quite important in the field of microcontrollers as data needs to be relayed from one device to another. that method of transferring data from one device to

another could be a sensor sending data to a microcontroller or the other way around, or it could be one microcontroller to another. Many protocols for serial communication have been proposed such as CAN, SPI, I2C, and UART. Each has its advantages and disadvantages in terms of bandwidth, speed, reliability or even applicability. Because of the reliability of the previously mentioned protocols and their full use, modern microcontrollers contain built-in modules that allow the use of such protocols easily as a lot of background work such as data polling methods, data organization, and synchronous/asynchronous communications are already implemented and working almost autonomously.

	Synchronous		Asynchronous
Peripheral	**SPI**	**I²C**	**UART**
Max bit rate	20 Mbit/s	1 Mbit/s	500 kbit/s
Max bus size	Limited by number of pins	128 devices	Point to point (RS232), 256 devices (RS485)
Number of pins	3 + n × CS	2	2(+2)
Pros	Simple, low cost, high speed	Small pin count, allows multiple masters	Longer distance (use transceivers for improved noise immunity)
Cons	Single master, short distance	Slowest, short distance	Requires accurate clock frequency

Figure 18: Various Serial Protocols Comparison [25]

As shown in figure 18, I2C and SPI both use a clock signal to synchronize data exchange between devices. Both of I2C and SPI offer an advantage as they support multiple devices on the same connection with a master/slave architecture where a master device produces clock frequency while other devices connected are known as slave devices. I2C uses less wiring than SPI, resulting in simpler circuit design and modern software libraries allow for the use of any protocol easily given hardware support, which makes it suitable in applications where data transfer is not intensive, and the required rate is under 1 Mbit/s.

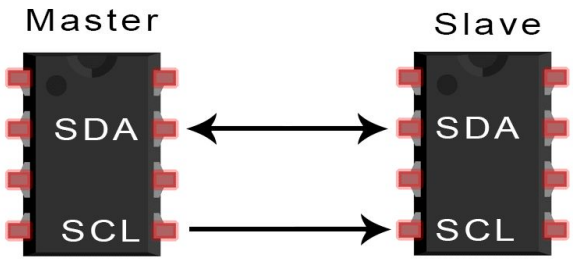

Figure 19: Data Flow in I2C Devices [26]

UART stands for Universal Asynchronous Receiver/Transmitter and is a widely used protocol in modern electronic devices. It only uses two wires and does not require a clock signal to operate. Alternatively, UART uses start and stop bits to signify when messages start and end [2]. Different data transfer rates can be used in UART (up to 115200 baud rate). Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps)[26].

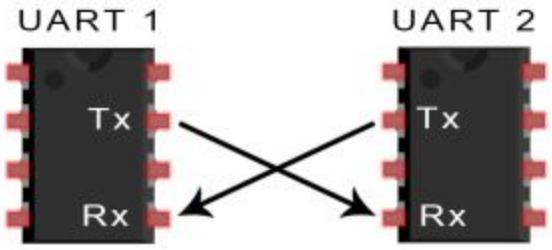

Figure 20: Data Flow in UART Devices [26]

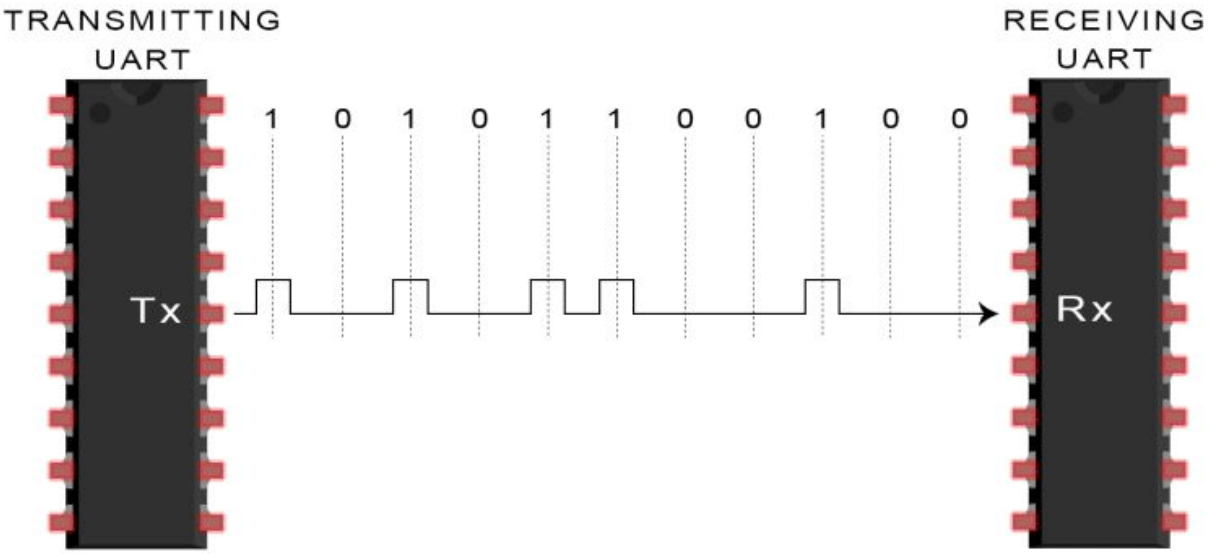

Figure 21: Sample Message Sent Between UART Devices [26]

In this section, the theory behind the basic building blocks of the smart wheelchair system was discussed.

Design

In this section, the mechanical, hardware and software components of the design are showcased and technical details behind design choices are discussed.

1. Mechanical

A used wheelchair was obtained to apply modifications to it which allowed the team to finish the project much faster as the mechanical design phase is skipped. The wheelchair's model is Invacare Storm TDX4 which has a center wheel-drive with two motors moving the drivetrain.

Figure 22: Invacare Storm TDX4 Power Wheelchair [27]

The wheelchair offers mechanical stability and comfort through the use of springs in its suspension which is configured for obstacles up to 3" [27]. The wheelchair has a 400-lb weight capacity and a top speed of 4.5 mph [27]. The motors are 4-pole traditional brushed DC motors and operate on a 24V battery [27].

The LIDAR was mounted on a tilting platform to enable scanning vertically, due to the fact that LIDAR is only able to scan in a horizontal plane and needs to be directed towards any desired vertical level. The platform consists of a motor mounted on an aluminum plate. A shaft extends from the motor and is used to move an independent aluminum plate where the LIDAR rests provide enough clearance for the platform to run, the LIDAR was raised using standoffs, and an acrylic board was used to keep the LIDAR on the delays. Below is a CAD model of the platform as well as the physical design:

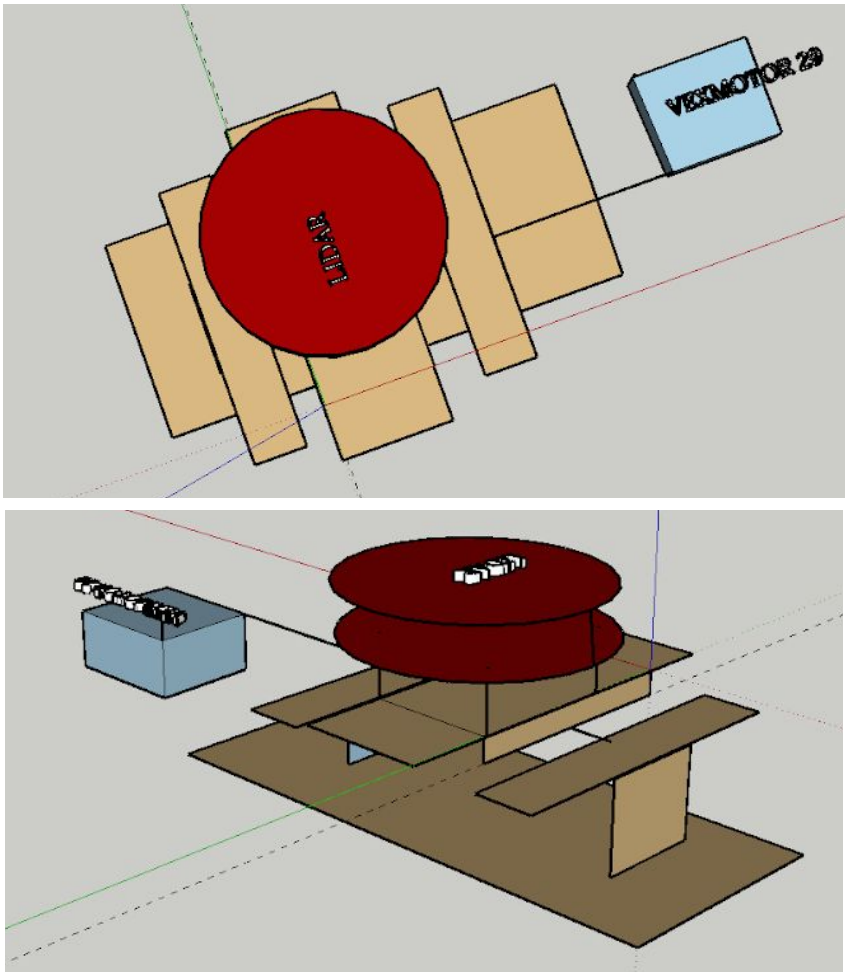

Figure 23: CAD Sketch of LIDAR Tilting Platform

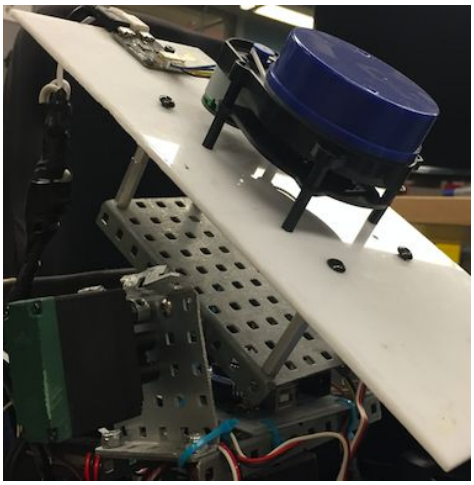

Figure 24: Side View of LIDAR Tilting Platform

The IMU (discussed in the next section) is mounted on a hat the user wears. A 3D printed enclosure was designed to secure the circuitry and appeal aesthetically. Below is a CAD model of the enclosure mounted on a hat:

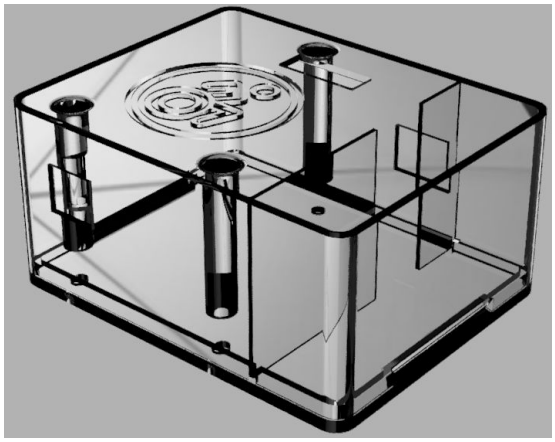

Figure 25: Transparent View of the 3D IMU Circuitry Case Design

2. Hardware

Since the wheelchair is electrical, there were two approaches to be used: (1) interface with the already existing control system, or (2) bypass the control system and interface the motors with a control system of our design. After a detailed analysis of the wheelchair documentation, the team decided that it is more useful to directly use a custom control system to avoid any complications later in the project, and to make prototyping faster by being able to add any features needed. A major advantage of creating a new control system is that it enables replicability of the project by anyone who has access to the documentation. This makes the "Smart Wheelchair" project open source on both software and hardware sides.

The team opted for a cheap, reliable and simple way to implement the control system. Design analysis was first performed, and specifications were derived based on the analysis. Parts with open sources libraries were first considered and the hardware was picked based on the cheapest prices. The architecture of the control system design is shown below:

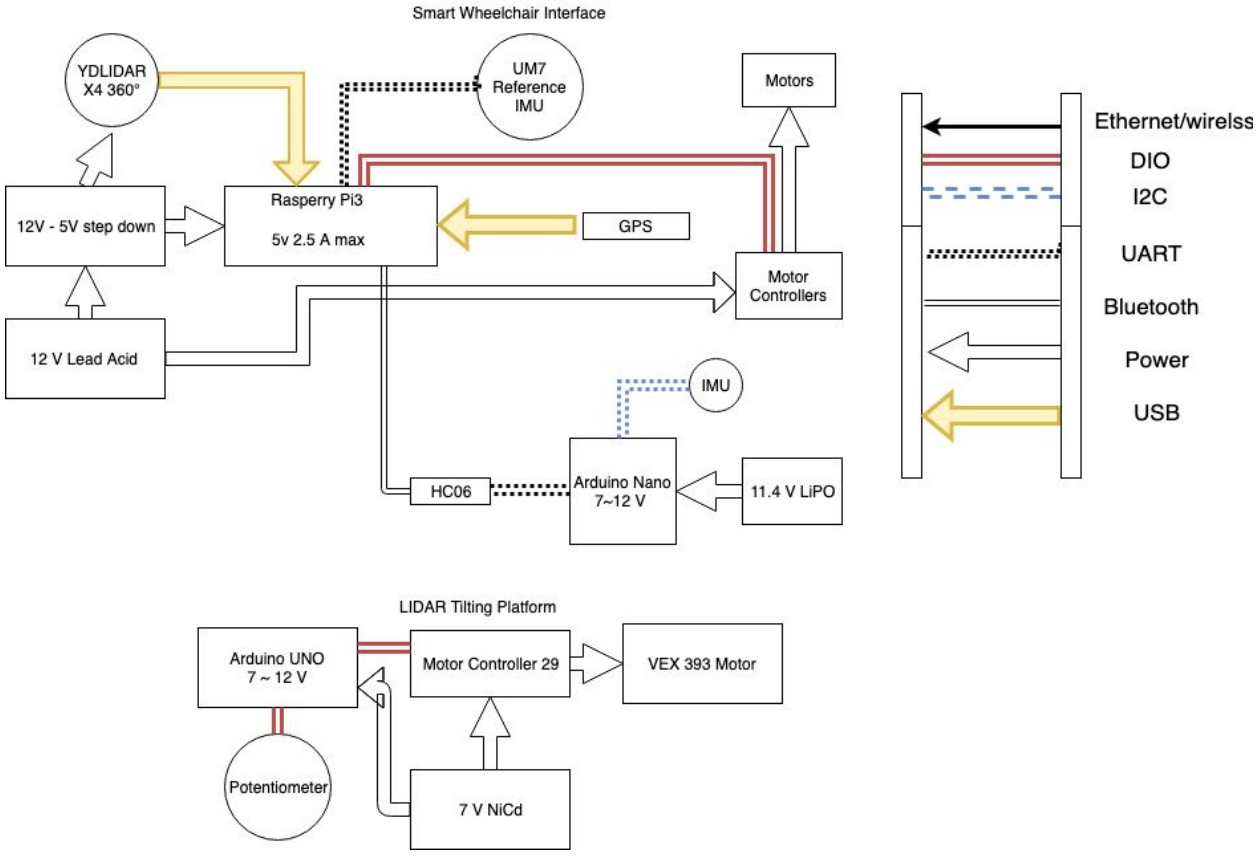

Figure 26: Overview of System Hardware Architecture

The primary wheelchair control hardware unit is an IMU mounted on the user's head. The reason behind choosing the IMU is due to its ease of use because it requires minimal force from the head and neck muscles. It can be calibrated for default resting positions and allows for a variety of sensor resolutions for motion and post-processing. Furthermore, an IMU sensor is affordable which is desired in such applications where accessibility is quite expensive due to user customization and lack of competition/alternatives. Modularity is quite important especially for people with spinal cord disorder and children who switch wheelchairs as they grow up, which is why the team wanted to avoid solutions that include mechanical compartments. An IMU offers an advantage over head-operated joysticks in the sense that they are mounted on the user rather than the wheelchair and are very easy to take off (ex: Velcro can be used for mounting the sensor). An LSM9DS1 IMU was chosen as it is one of the cheapest IMUs on the market. Some issues regarding calibration the team faced but overcome by writing a new calibration library. The IMU is connected to an Arduino nano due to the nano's small footprint, and the Arduino is, in turn, connected to the HC06 Bluetooth device. This combination is packaged in a small circuit board and powered by a LiPo battery. Figure 27 shows the circuit board design:

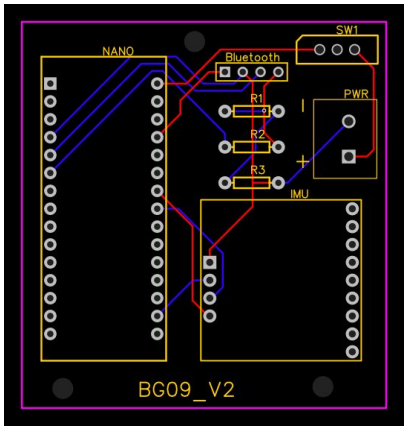

Figure 27: Head Mounted IMU PCB

The central computing unit of the wheelchair is the raspberry pi 3B+. It is a cheap, robust and widely used development board computer with many GPIO ports/communication protocols. It features I2C, UART, SPI and 40 digital pins including 2 PWM pins. It is also equipped with WIFI and Bluetooth support which means that no add-ons are required in terms of electronics. Additionally, it uses an ARM v8 processor with support for the Linux operating system which is essential as it simplifies the prototyping/coding process.

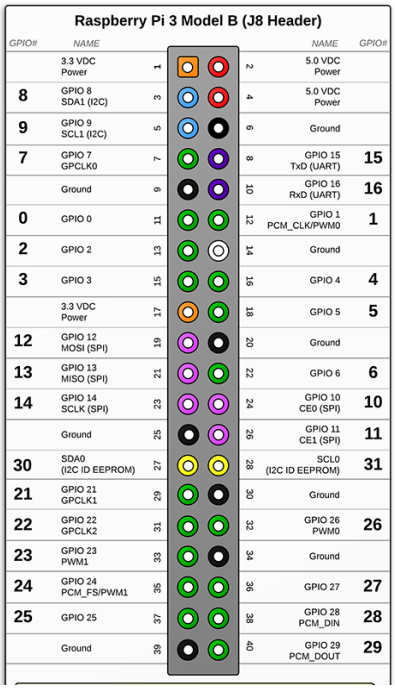

Figure 28: Raspberry Pi 3B+ GPIO Header Pins [28]

The motor controllers are used as an intermediate between the battery and the motor of the wheelchair and provide direct control via user signal while isolating the logic hardware from the power hardware. Originally, cheap motor controllers were used, but they were ineffective due to the current draw given the load of the wheelchair. The Victor SP controllers are reliable and can take up 40 Amps of continuous current draw. The Victor SP receive PWM signals from the pi's pin 32 (PWM0) and pin 33 (PWM1).

The battery used in this project is 12V 10Ah sealed lead acid. Sealed lead acid batteries are the most known batteries of lead-acid since they can work in different conditions. The electrolyte is hanging in a thin fiberglass mat that fits between the lead plates. As the battery is operating, the electrolyte is moved from glass mat to battery plates as needed. The mat has enough electrolyte. The movement electrolyte from glass mat to battery plates gives the battery the ability to handle vibrations, which make them a perfect fit for recreational vehicles and cars. They also can be placed in any position and most importantly can handle low-temperature degrees.

The DC-DC buck converter is used to step down the voltage of the sealed lead acid battery which is 12V to 5V, which is the maximum voltage that Raspberry Pi can handle. A second buck converter is used to power the LIDAR at 5V.

A LIDAR in this project is used for obstacle detection. The YDLIDAR can scan a total of 360 degrees. Additionally, it is mounted on a tilting platform as shown in the mechanical section to provide vertical scanning in addition to planar scanning. The LIDAR tilt platform features an independent control system from the wheelchair to decrease complexity. An Arduino Uno is used as the control unit, and a simple motor controller (VEX motor controller 29) is used to control the VEX 393 motor. The motor tilts the platform along the vertical axis, and the platform's angle is measured using a potentiometer. The angle is used as feedback in a closed-loop proportional control scheme which allows for consistent movement of the tilting platform.

Besides using the head mounted IMU to determine the user's heading, a reference IMU is mounted on the wheelchair to determine the relative angle between the wheelchair and the user as the yaw angle changes when the wheelchair turns even if the user did not move their head. The difference between the reference IMU and the head mounted IMU is used to calculate the yaw motion of the head for the wheelchair controls. The UM7 is connected to the pi using UART.

The GPS module requires no circuitry as it plugs straight into the USB port without any additional power requirements.

3. Software

Linux is one of the most powerful operating systems because of its being open source. Thousands of developers worldwide grow the platform by contributing to it yearly. A viral Linux distribution is Ubuntu, known for its stability, reliability and continuous support. In the vein of making our project completely open source, the team opted to use Linux Ubuntu 16.04, one of Ubuntu's best distributions. Ubuntu is free, and replication of the team's software system which is quite smooth and provides more options for physically challenged individuals. Also, it is quite easy to apply without the use of a video output device which is essential for a computer preserved inside the wheelchair control system with little access provided. Lastly, LUbuntu (Light Ubuntu) is efficient and runs on low-performance computers, for example, the raspberry pi.

Ubuntu's most significant advantage for the smart wheelchair project is its ability to run ROS (Robot Operating System). ROS is an open source software framework geared towards robotics. It provides standardized messaging between programs, hardware abstraction, and low-level hardware control [29]. ROS uses a peer-to-peer network in managing the processes under its roof and can have a distributed computing cluster working seamlessly provided a network system that connects devices together. Since its debut in 2007, ROS has been heavily used in advanced robotics research and been featured in countless papers. Also, it started making its way into the industrial world manifesting as ROS-Industrial [30].

ROS is useful for this project as it enables using multiple programming languages and programs seamlessly. This is accomplished mainly via the rospy and roscpp libraries which will allow C++ and Python programs to send and receive standardized messages to/from the ROS ecosystem. A program within the ROS ecosystem is called a node which is the term used in this report.

The LIDAR driver library uses C++ to obtain and process serialized information from the sensor. ROS has standardized LIDAR messages that make it easier to manipulate LIDAR data (sensor_msgs/LaserScan). A python node receives the LIDAR messages, processes data points, and determines if an object is nearby. It then sends a boolean via a ROS publisher to the motor control python node.

As for wireless communications, the Bluetooth, receiving is done via a python node which parses the messages received, processes data and sends it as PWM via a ROS publisher to the motor control node mentioned in the previous paragraph. The IMU data collected in this node consist of pitch, roll and magnetometer data which acts as the yaw data. Pitch and roll are scaled and transformed into PWM values while magnetometer data is used to determine the yaw angle which enables/disables wheelchair movement.

Table 1 illustrates the method of controlling the wheelchair with specific commands:

Table 1: Correlation between Wheelchair Control vs Head Movement

Axis	**Wheelchair Movement**
Pitch	Forward/Backward
Roll	Left/Right
Yaw	Start/Stop Movement

The motor control node mentioned earlier receives PWM information from a ROS topic and outputs to the raspberry pi PWM pins. The PWM runs at a rate of 500 Hz as the motor controller requires that specific frequency to operate. When the pulse width is 75%, the motors are at rest, and they run counter-clockwise when the pulse width is between 50 - 75 %. Besides, this node receives a boolean that stops the motors in case the LIDAR detects a close object.

Below is a node graph representing the current software architecture that runs on the raspberry pi. The rectangles represent topics, and the ovals represent nodes:

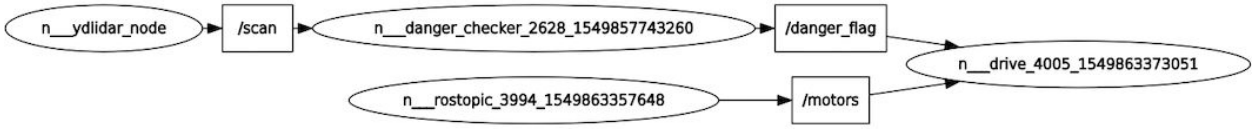

Figure 29: ROS Software Architecture of the Smart Wheelchair Project

Table 2: ROS topics and their respective ROS message standards

ROS Topic	Message Type	Format
/scan	sensor_msgs/LaserScan	std_msgs/Header header float32 angle_min float32 angle_max float32 angle_increment float32 time_increment float32 scan_time float32 range_min float32 range_max float32[] ranges float32[] intensities
/danger_flag	std_msgs/Bool	bool data
/motors	std_msgs/Float32MultiArray	std_msgs/MultiArrayLayout layout float32[] data

As for the Arduino microcontroller software, the RTIMULIB Arduino library was used. Unfortunately, the most recent version of this library doesn't have compatibility with the IMU used in this project which is the LSM9DS1 model. So, the library was manually altered to be compatible with this type of IMU model. The main C++ program that is used is called Arduino blue which is a modified version of the original RTIMULIB-Arduino Arduino-IMU code. The primary function of this program is to take the original raw data from the IMU and then applying the RTQF filter discussed in the theory section on that data. The three metrics that are calculated are the pitch, row, and yaw which are then sent to the HC06 Bluetooth module to be transmitted through Bluetooth to the raspberry pi. The calibration data should have already been stored by running the default ArduinoMagCal script which creates maximum and minimum data of the magnetometer which will be used in the RTQF filter in the Arduino blue program.

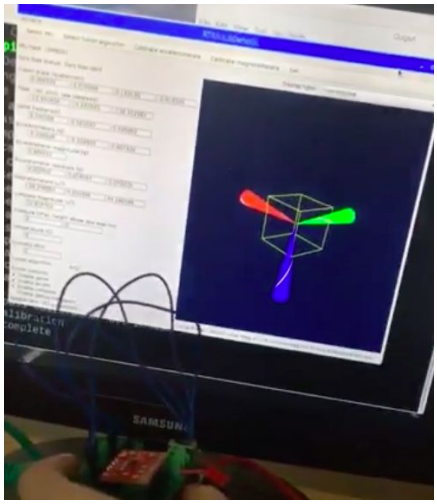

Figure 30: Original RTIMULIB Calibration Test of LSM9DS1 IMU

The LIDAR tilting platform is controlled by an Arduino Uno as mentioned in the previous section. The microcontroller polls data from the potentiometer and uses it as position feedback for the closed feedback loop that controls the motor output. The motor controller receives PWM signals generated by the *vex Motor* library. The programming of the platform is very simple as shown below:

```
#include <vexMotor.h> // includes the vexMotor library - uses
Servo.h
vexMotor myVexMotor1;  // creates an instance of the vexMotor
class
int error  = 0;
float kp = 0.8;
```

```
int target;
int power;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(A0,INPUT);
  myVexMotor1.attach(9);
}


void loop() {
  // put your main code here, to run repeatedly:
  while(true){
   target  = 150;
   error = target - analogRead(A0);
   while(abs(error)>30){
    error = target - analogRead(A0);
    Serial.print(analogRead(A0));
    Serial.print(',');
    Serial.print(error);
    power  = error*kp;
    if(abs(power)>255)
      power = power/abs(power)*255;
    Serial.print(',');
    Serial.println(power);
    myVexMotor1.write(power);
   }
   delay(100);
   target = 250;
   error = target - analogRead(A0);
   while(abs(error)>30){
    error = target - analogRead(A0);
    Serial.print(analogRead(A0));
    Serial.print(',');
    Serial.print(error);
    power  = error*kp;
    if(abs(power)>255)
      power = power/abs(power)*255;
```

```
        Serial.print(',');
        Serial.println(power);
        myVexMotor1.write(power);
       }
       //delay(1500);
     }
   }
```

An additional objective of the group was to achieve indoor autonomous navigation. So far, very little progress has been made on autonomous navigation using GPS as the team did not have enough personnel nor time to allocate resources for it. A GPS module was added, and data has been extracted into the ROS ecosystem through the nmea_navsat_driver library. The plan is to use GPS data to localize the robot and import that data into the navigation stack. The ROS navigation stack needs odometry information provided by the IMU transformed regarding the robot base frame [31]. In addition, LIDAR data can be used to create a static map of indoor spaces to facilitate autonomous navigation and enable obstacle detection while navigating [31]. Below is a figure showing the ROS navigation stack. Most blocks are provided within the ROS system. However, the team still need to give the sensor transforms,map_server, odometry source, and base controller blocks. The team has partially achieved the sensor sources block. More work is required to configure the wheelchair fully. The open source nature of our project allows for future collaborations and more finesse given enough time.

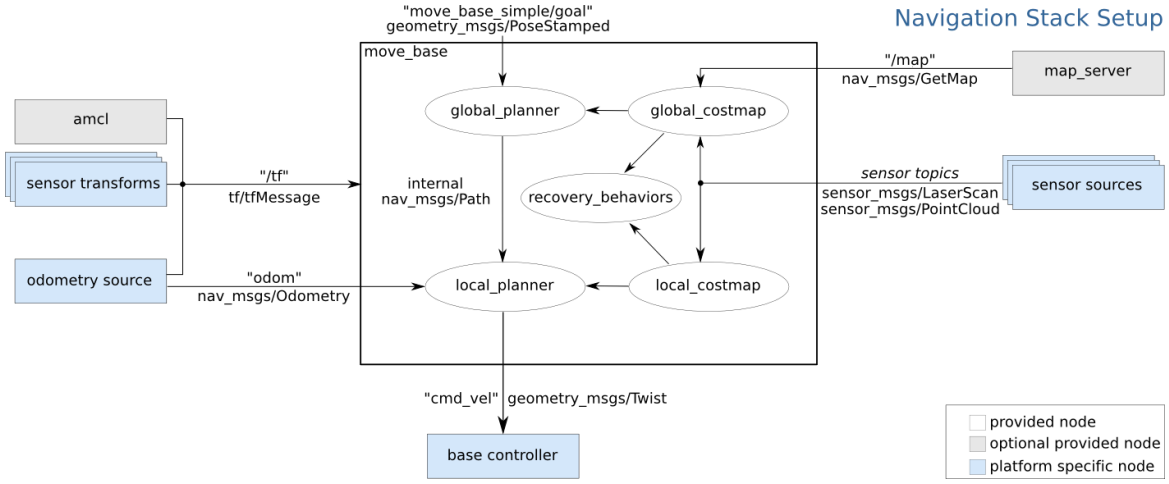

Figure 31: Generic Robot Navigation Stack Setup [31]

Detailed design elements of each subsystem in the smart wheelchair project were discussed in this section.

Alternative designs

The current design can be made more reliable and scalable for public use. Although it is open source, it is quite difficult for inexperienced programmers to use ROS. Microcontrollers are more reliable and less likely to crash than a computer with a full-fledged operating system (the raspberry pi). Microcontrollers are more accessible to program than creating ROS applications and require less power to operate for that reason the calls to shift for a microcontroller based control system instead of relying on the raspberry pi and Linux which will increase reliability, efficiency, and accessibility to the less technically inclined.

Additionally, this will reduce the cost as microcontrollers are inherently cheaper due to their lack of extra modules such as HDMI ports. The team recommends the use of the ESP32 as the main board controlling the wheelchair. It contains multiple serial ports including UART and I2C as well as built-in PWM, WIFI and Bluetooth capabilities which means no add-ons are required. The development board is shown below in figure 32:

Figure 32: ESP32 Development Board [32]

The ESP32 lacks a micro USB port which is easily remedied by a simple micro USB to UART circuit such as the FT232. The chip is powered by 3.3V meaning that the step-down buck converter would have to shift the values from 12 V to 3.3 V instead of 5V. This is not an issue because the current buck converter features variable voltage control.

As for sensors, our choice of sensors has proven to be robust for the application. However, the reference IMU (the UM7) can be swapped for an LSM9DS1 to provide consistency with the measurements provided from the head mounted IMU.

A 24V battery is recommended to ensure the wheelchair can overcome steep ramps at ease which results in changing the motor controllers as they are rated for a maximum of 16V. The TalonSRX motor controllers are an upgraded version of the motor controllers that are currently in use. The TalonSRX motor controllers feature smart current control and built-in PID as well as a maximum voltage of 28V.

Overall, some changes are recommended for hardware, but many successful choices remain unchanged in this proposed alternative design.

On the software side, since this design forgoes the use of the pi for a microcontroller, ROS cannot be used anymore. This is not an issue to the core functionality of the project as the software can be easily rewritten using C/C++. The possibility of implementing autonomous navigation, however, is less likely due to processing power limitations and lack of libraries that enable reliable SLAM navigation such as the ones existing in ROS.

Table 3 compares the current design with the proposed alternative design.

Table 3: Current Design and Alternative Design Comparison in Terms of Hardware and Software

Component	Current Design	Alternative Design
Central Computer	Raspberry PI 3B+	ESP32
Power	12V Lead Acid	24V Lead Acid
Motor Controllers	VictorSP	TalonSRX
Reference on-chair IMU	UM7	LSM9DS1
Software framework	ROS	Platform IO
Languages	C, C++, Python	C, C++
LIDAR	YDLIDAR X4	Unchanged
IMU Hat	Nano+HC06+LSM9DS1	Unchanged
LIDAR Tilting Platform	VEX 393 Motor + Motor controller 29+ Arduino Uno + Potentiometer	Unchanged

Material/Components

Name	Unit Price (CAD)	Quantity
IMU Breakout - LSM9DS1	20.99	4
Raspberry Pi 3 B	51.99	1
Raspberry Pi 3 B+	66.99	1
YDLIDAR X4 360° Laser Scanner	132.39	1
GPS	25.99	1
Cytron 10A 5-30V Dual Channel DC Motor Driver (OLD(	30.91	1
Victor SP DC Motor controller (NEW)	77.99	2
DROK Auto Boost Buck Converter (OLD)	25.99	1
WEIKE Buck Converter (In Use)	12.51	1
DROK Adjustable Buck Converter (In Use)	18	1
12V 12Ah Lead Acid Battery	61.59	1
Bluetooth HC06	15.99	3
Arduino Nano	6	2
Printed Circuit Boards (OLD)	25	1
Printed Circuit Boards	2	10
Lipo Battery 3s 650mh	36.99	1
TOTAL no tax	808.26	
Total with Tax	913.3338	
Individual Cost	304.4446	

Measurement and Testing Procedures

Initially, components were tested individually to test their functionality/reliability and the code's effectiveness in each subsystem. For example, the LIDAR was checked off the wheelchair initially to ensure the code receives accurate distance measurements and classifies a dangerous situation correctly. Another example, is testing Bluetooth data transmitting by having the Arduino send dummy data before integrating it into the IMU code. Once all wheelchair subsystems were tested and proved to work, the control system was assembled, and real test cases were tried.

Since the wheelchair project is providing both indoor and outdoor mobility for users, it is paramount to produce a fail-proof system. The team heavily focused on testing and optimization of the design primarily as a working prototype was built at the end of phase 2. The testing procedure consisted of driving the wheelchair in different environments by numerous users who are unfamiliar with the project. This provided a fresh look at the performance of the wheelchair and the reliability/intuitiveness of the control methods.

The mobility testing was mainly performed in Kerr Hall in Ryerson University as it offers an excellent indoor simulation of wide/narrow passages as well as a ramp. The performance was tested qualitatively by asking testers for feedback on the controllability of the wheelchair. The wheelchair was used for up to an hour on some testing days by multiple users to test the longevity of the battery as well as discovering any problems with long term operation such as component overheating.

Multiple hats and head-mounting positions were tested to determine the best controllability with the least compromise in terms of comfort. This was tested qualitatively as well through user feedback. Similarly, since the obtained wheelchair does not have any padding for seating, the group tested various cheap padding options by sitting on the wheelchair and using it for extended periods.

For obstacle avoidance, users were asked to run into objects, walls, and people (human obstacles consisted of the group members only). The threshold distance and preferred angles to read from the LIDAR were tested and optimized on the spot. A successful obstacle avoidance setting is measured qualitatively by having the wheelchair not run into objects. On the other side, it is essential that the wheelchair is not too sensitive to obstacles and detects far away objects as a danger. Additionally, the LIDAR tilting platform was tested and optimized to provide a high enough sampling rate for all possible scanned planes so that obstacles are detected even in a dynamic environment.

As for outdoor testing, the primary concern was the LIDAR working in bright day-light. The wheelchair was brought outdoors around 2 - 3 pm on multiple occasions in both cloudy and sunny settings. The group attempted to run the wheelchair into an obstacle by commanding it to move forward while facing a wall. The wheelchair was not tested outdoors as heavily as indoor testing due to the harsh conditions of Canadian winter and the wheelchair not being weatherproof in this testing stage.

Measurement Results

In phase 1, many problems were encountered in terms of power sources/motor controllers that can withstand loading. Many parts failed, and the requirements had to be reanalyzed for new and improved designs.

Once all subsystems worked, the wheelchair was tested in actual use cases. Initially, users found that the wheelchair movement is too "jumpy" and found it uncomfortable to use. Additionally, most users found it difficult to control the wheelchair with our initial settings since it could only be moved forward or turned at a time. Moreover, users found that the head tilting was too drastic and difficult to maintain for extended periods. Some users found that turning was too fast and recommended decreasing the speed as it felt dangerous.

After remedying the issues pointed out in initial testing, the wheelchair was tested again. Users found that controllability is much smoother and the added vectoring (being able to turn while moving forward) provided more depth in terms of driving the wheelchair. In addition, the wheelchair was much safer as it moved slower and was less "jumpy." The addition of padding to the wheelchair seating made the experience a lot more positive for users.

As for the LIDAR, initial results showed some ineffectiveness due to the way signals were handled; all data points were averaged together which created some sensitivity issues when the object was too small and did not occupy a large portion of the LIDAR's field of view. This issue was remedied, and currently, the LIDAR can detect obstacles while being able to move in narrow passages. The LIDAR was tested outdoors, and it proved useful as it was able to identify walls effectively.

Finally, the wheelchair's stopping algorithm using differences in magnetometer angle between the reference magnetometer and the head mounted one was tested. The algorithm works effectively in almost all situations. One problem is the magnetometer does not function well in basements that are heavily shielded with pipes and express magnetic field distortions.

Post Analysis

There are some improvements to be introduced into the systems. First of all, reliability can be increased by avoiding the use of a computer with a fully fledged operating system as mentioned in the alternative design section. While this did not pose any problem in the testing phase, everyday use would require a more stable system such as a microcontroller.

Secondly, the team found that there would be more benefit in allowing users to recalibrate the default position of the IMU which adds quality of life improvements. Additionally, there should be different speed settings for users who are comfortable with the IMU control method as the current settings of the wheelchair dictate slow travel speed.

On extended periods of operation, the team noticed overheating in the head mounted IMU. This could be a result of the unregulated power supply from the battery to the Arduino Nano. If that is the case, the issue can be solved by just adding a voltage regulator between the battery and the Arduino microcontroller.

The LIDAR system proved to be very effective at obstacle detection. One issue is that currently, only the front area of the wheelchair is being scanned. Another LIDAR can be added on the back for improved safety at an additional cost.

Lastly, the magnetometer works effectively outdoors and indoors for the most part. It fails to detect the correct orientation in heavily shielded areas such as basements with many pipes. A possible solution is to use gyroscope data as it is relative and does not use the earth magnetic field as a reference which means it is not subject to interference.

Conclusions

A power wheelchair was modified to be operated by head tilt controls. The wheelchair features obstacle detection and safety features such as preventing the wheelchair from running into detected obstacles. The project features open source software platforms and code available to the public for future use/improvement. In addition, the stock hardware of the wheelchair was completely replaced except for motors. This allows for reproducibility of the project for any powered wheelchair regardless of the brand/manufacturer as the factory hardware is completely bypassed.

The project has successfully met the requirement of using an alternative control method for quadriplegic users effectively. The second requirement, which is stopping the wheelchair if it is running into an obstacle, has been met as well. The head tilt is measured using an IMU and then sent via Bluetooth to the central computer in the wheelchair. The wheelchair moves according to various scaling factors that transform the head roll, pitch, and yaw angles into PWM commands that run the motors. A LIDAR mounted on the front of the wheelchair which allows for full frontal scanning of the environment, and the use of scanned data in obstacle detection. The wheelchair can stop whenever it gets too close to people, various sized objects, and walls which demonstrates the effectiveness of the LIDAR.

Currently, there is no way to check for obstacles when the wheelchair is going back as a second LIDAR is needed for the back of the wheelchair. It was not added in this project due to financial limitations. However, it is quite easy to add to the system and the same code but with different thresholding values can be used.

The software architecture is set up so that indoor autonomous navigation is possible in future iterations of the project. This is an important goal for the team as it could be easily implemented on the ROS navigation stack given enough time and personnel are available. Future versions of the project could improve reliability and efficiency while reducing cost as discussed in the alternative design section. We hope this project can be used as a stepping stone towards a more accessible and independent future for quadriplegic users.

References

[1] J. Leaman and H. M. La, "A Comprehensive Review of Smart Wheelchairs: Past, Present, and Future," IEEE Transactions on Human-Machine Systems, vol. 47, (4), pp. 486-499, 2017.

[2] Christopher and Dana Reeve Foundation. One Degree of Separation:: Paralysis and Spinal Cord Injury in the United States. Short Hills, NJ: 2009.http://www.christopherreeve.org/site/c.ddJFKRNoFiG/b.5091685/k.58BD/One_Degree_of_ Separation.htm

[3] J. Leaman and H. M. La. iChair: Intell. powerchair for severely disabled people. In ISSAT Int. Conf. Modeling of Complex Syst. and Environments, Da Nang, Vietnam, Jun. 2015.

[4] F. Carrino *et al*, "A self-paced BCI system to control an electric wheelchair: Evaluation of a commercial, low-cost EEG device," in 2012, . DOI: 10.1109/BRC.2012.6222185.

[5] "Power Wheelchair Driving Controls," *Spinal Cord Injury Research Evidence*. [Online]. Available: https://scireproject.com/evidence/rehabilitation-evidence/wheeled-mobility-and-seating-equipment/power-wheelchairs/power-wheelchair-driving/. [Accessed: 26-Oct-2018].

[6] BlueSkyDesigns. Mount-n-mover. http://www.mountnmover.com/ product-options/dual-arm/, Feb. 2015.

[7] J. Kim *et al.*, "The tongue enables computer and wheelchair control for people with spinal cord injury," *Sci. Transl. Med.*, vol. 5, no. 213, p. 213ra166-213ra166, Nov. 2013.

[8] M. Pedley, "Tilt Sensing Using a Three-Axis Accelerometer." Freescale Semiconductor, Mar-2013. http://cache.freescale.com/files/sensors/doc/app_note/AN3461.pdf?fpsp=1

[9] T. Poluyi, Control System Development for Six Degree of Feedom Spine Simulator. 2014.

[10] "What is Pitch, Roll and Yaw ?," Emissary Drones, 19-Mar-2018. [Online]. Available: https://emissarydrones.com/what-is-roll-pitch-and-yaw. [Accessed: 15-Nov-2018].

[11] "Compass Heading Using Magnetometer." Honeywell, Jul-1995. https://aerocontent.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/Magnetic__Literature_Application_notes-documents/AN203_Compass_Heading_Using_Magnetometers.pdf

[12] "Gyro sensors - How they work and what's ahead | about Gyro sensor | Technical Information | other Information,"Seiko Epson Corporation. [Online]. Available: https://www5.epsondevice.com/en/information/technical_info/gyro/. [Accessed: 15-Nov-2018].

[13] J. Ferguson, "Calibration of Deterministic IMU Errors", Undergraduate, Embry-Riddle Aeronautical University, 2015.

[14] I. Beavers, "The Case of the Misguided Gyro", Analog Dialogue, no. 139, 2017.

[15] D. Sportillo, "Addressing the problem of Interactionin fully Immersive Virtual Environments:from raw sensor data to effective devices", Masters, Universit`a di Pisa, 2019.

[16] An Introduction to Lidar Technology, Data, and Applications. [Online] https://coast.noaa.gov/digitalcoast/_/pdf/lidar101.pdf. 4.

[17] Coding Videos. (2018). *Autonomous navigation robot with ROS (Raspberry pi 3 B+ + YDLIDAR X4) - Coding Videos*. [online] Available at: https://codingvideos.net/autonomous-navigation-robot-with-ros-raspberry-pi-3-b-ydlidar-x4/ [Accessed 8 Apr. 2019].

[18] Schweber, B. (n.d.). *Don't Ignore the Humble Brushed DC Motor| Mouser*. [online] Mouser.ca. Available at: https://www.mouser.ca/applications/dont-ignore-the-brushed-dc-motor/ [Accessed 8 Apr. 2019].

[19] M. Clement, "What is Pulse Width Modulation?", Brigham Young University, 2015.

[20] A. Tantos, "H-Bridges – the Basics | Modular Circuits", Modularcircuits.com. [Online]. Available: http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/. [Accessed: 08- Apr- 2019].

[21] Buck Converter. (2016). Retrieved April 10, 2019, from https://www.plexim.com/academy/power-electronics/buck-conv

[22] "Closed-loop Systems", Electronics Tutorials. [Online]. Available: https://www.electronics-tutorials.ws/systems/closed-loop-system.html. [Accessed: 08- Apr- 2019].

[23] "Bluetooth.com | Architecture - Core System", *Web.archive.org*, 2008. [Online]. Available: https://web.archive.org/web/20080123034431/http://bluetooth.com/Bluetooth/Technology/Works/Core_System_Architecture.htm. [Accessed: 08- Apr- 2019].

[24] C. Hills and C. Hills, "Micrium Bluetooth Stack at Phaedrus Systems", *Phaedsys.com*, 2017. [Online]. Available: https://www.phaedsys.com/principals/micrium/mbluetooth.html. [Accessed: 08- Apr- 2019].

[25] Hussain, "BME 538: Serial Communication", Ryerson University, 2017.

[26] C. Basics, "Basics of UART Communication", Circuit Basics. [Online]. Available: http://www.circuitbasics.com/basics-uart-communication/. [Accessed: 08- Apr- 2019].

[27] Invacare ® Storm Series ® TDX. Elyria: InvaCare Corporation, 2018.

[28] "The Pi4J Project – Pin Numbering - Raspberry Pi 3B+", Pi4j.com. [Online]. Available: https://pi4j.com/1.2/pins/model-3b-plus-rev1.html. [Accessed: 09- Apr- 2019].

[29] "Wiki," ros.org, 10-Jul-2018. [Online]. Available: http://wiki.ros.org/. [Accessed: 14-Feb-2019].

[30] "ROS-Industrial", rosindustrial.org. [Online]. Available: https://rosindustrial.org/. [Accessed:14-Feb-2019].

[31] “Setup and Configuration of the Navigation Stack on a Robot”, http://wiki.ros.org. [Online]. Available: http://wiki.ros.org/navigation/Tutorials/RobotSetup. [Accessed: 14-Feb-2019]

[32] A. Industries, "Espressif ESP32 Development Board - Developer Edition", Adafruit.com, 2019. [Online]. Available: https://www.adafruit.com/product/3269. [Accessed: 09- Apr- 2019]

Appendices

The entirety of the code base can be found at: https://github.com/hamzaMahdi/captstone

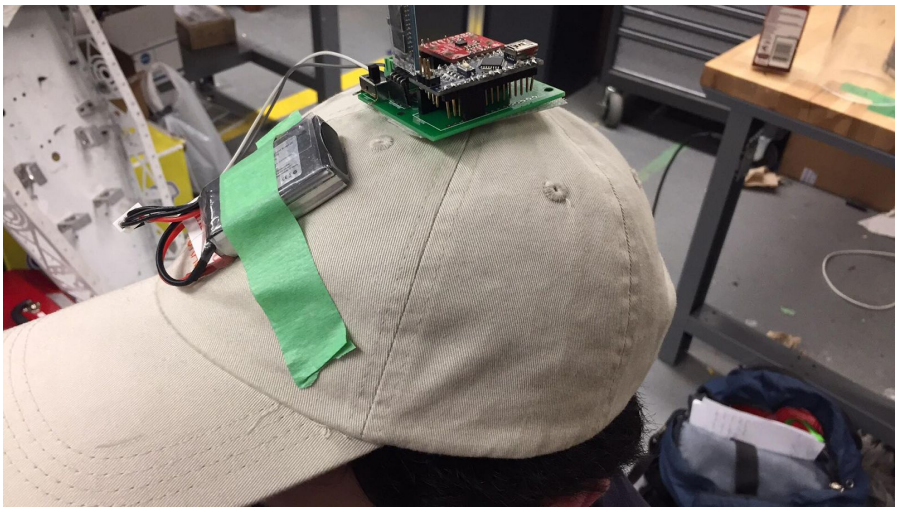

Figure 33: Early Prototype of Head Mounted IMU Circuit.

Figure 34: Current Design of IMU Hat.

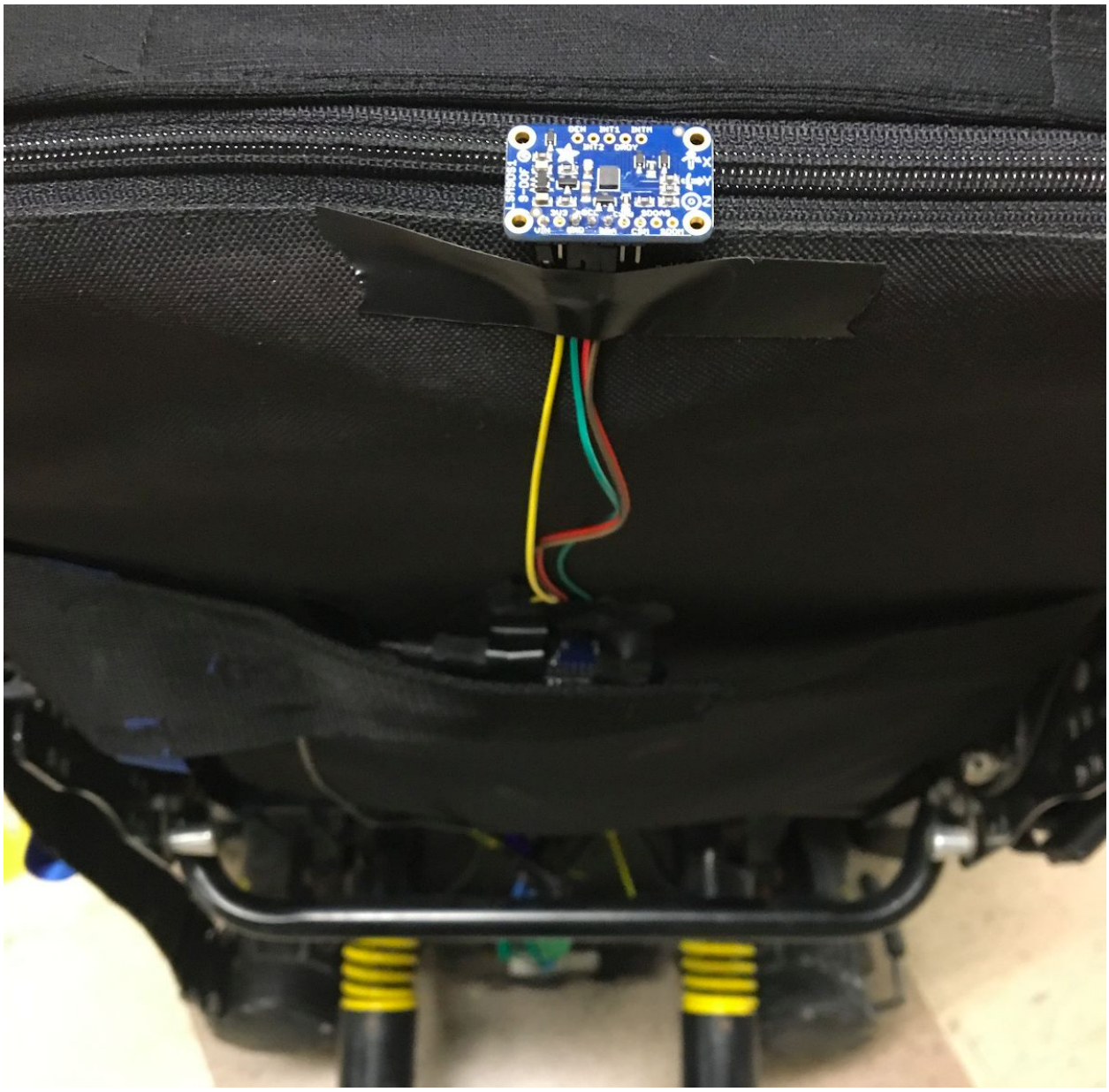

Figure 35: Reference IMU Mounted on the Back of the Wheelchair.

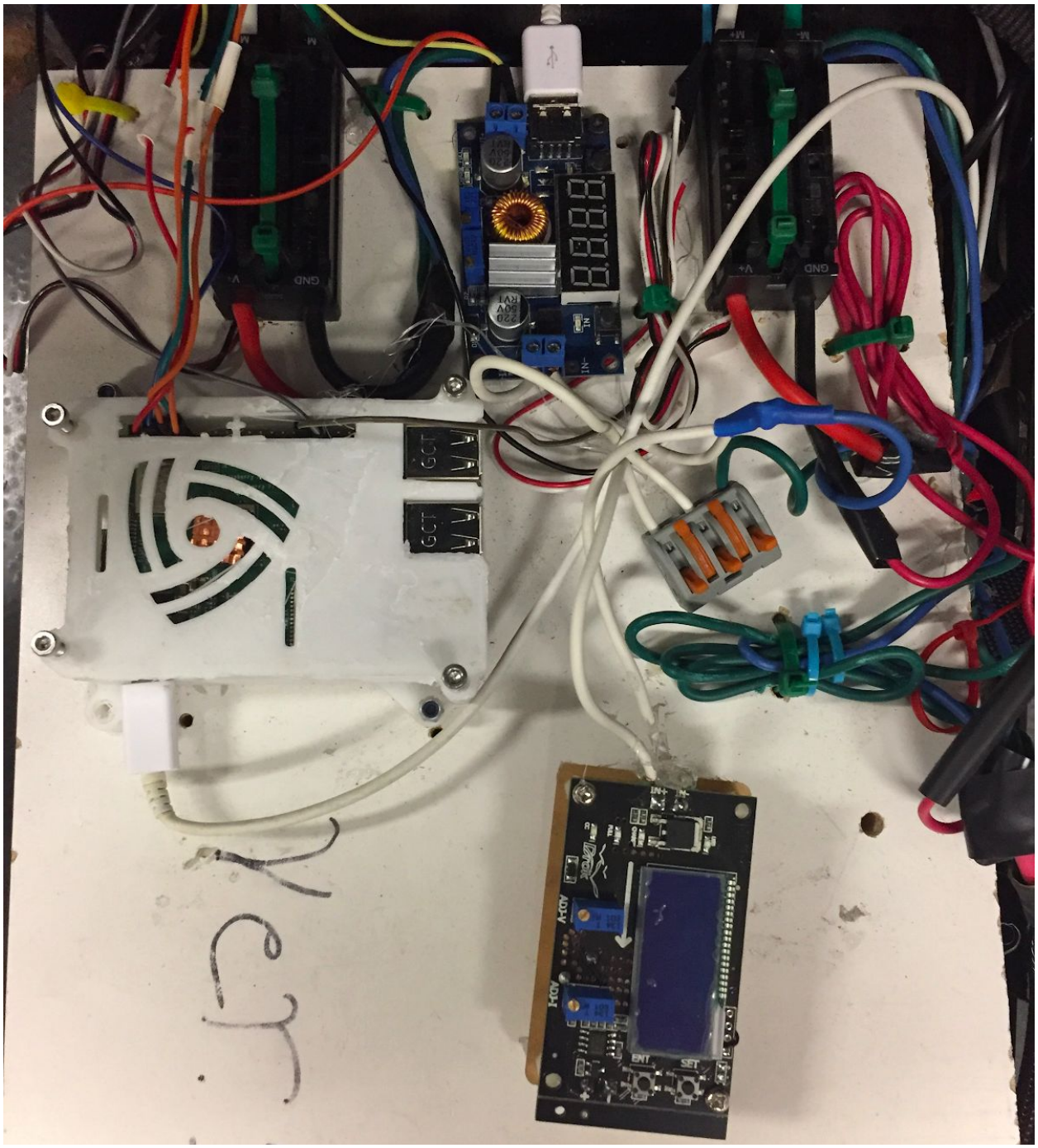

Figure 36: Wheelchair Circuitry Including Power Regulation, Raspberry Pi and Motor Controllers.

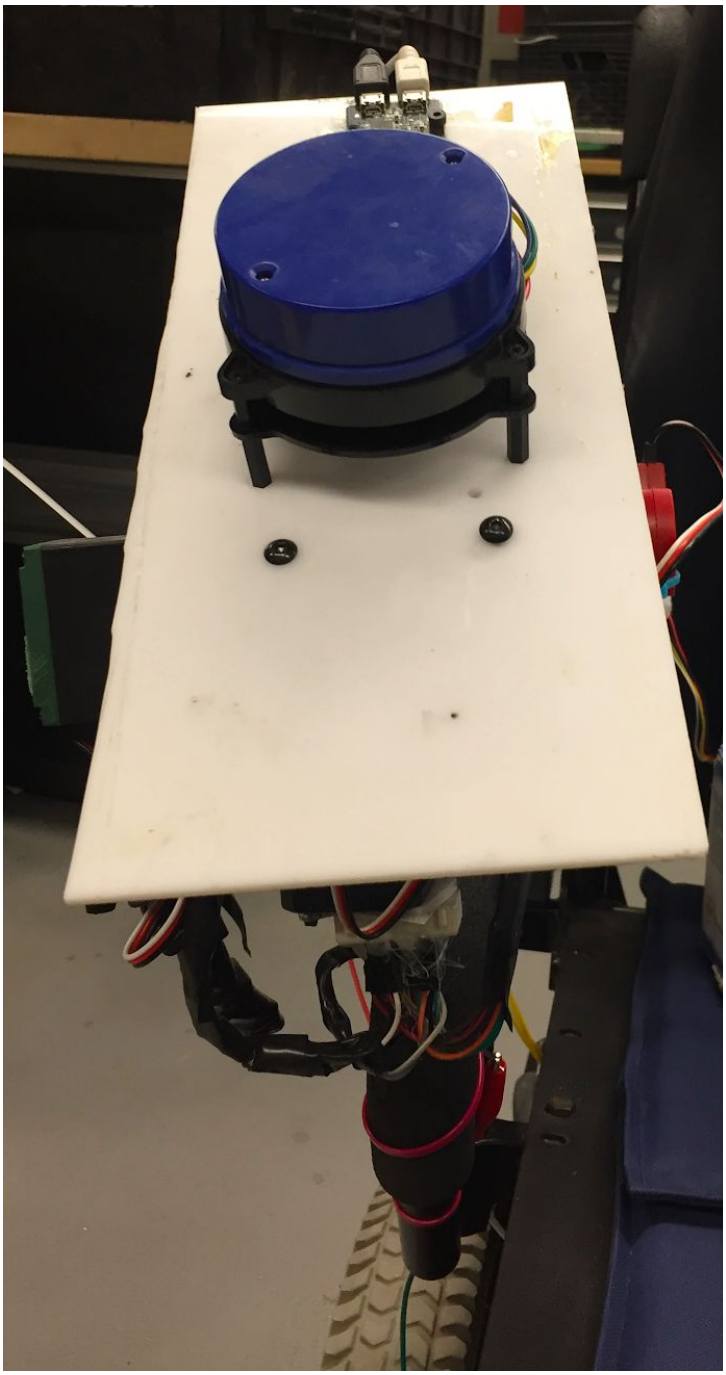

Figure 37: Frontal View of LIDAR Tilting Platform.

Figure 38: Testing of Early Smart Wheelchair Prototype.

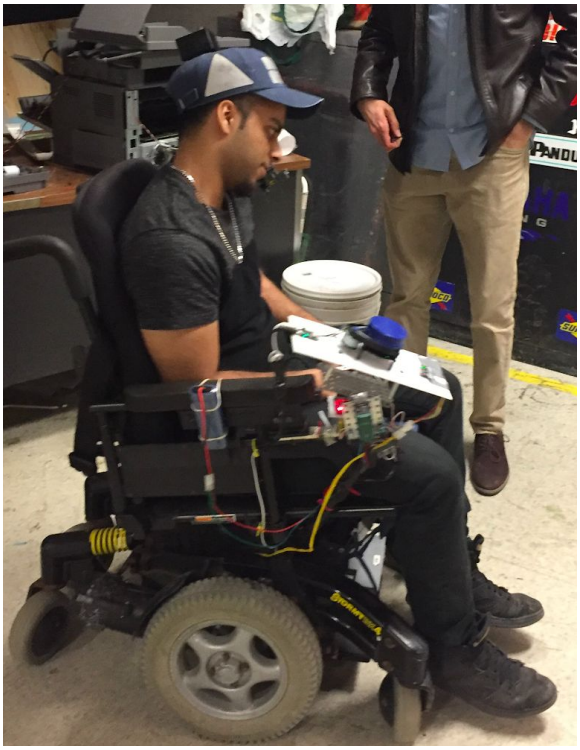

Figure 39: Testing of Smart Wheelchair Prototype.