Interactive Mobility
End of Project Documention

Vadim Artishuk, Ryan Law, Riley Schofield, Alex Verstivskiy
Dept. Engineering and Computer Science: Students
CSU Sacramento

Abstract— ***This paper lists the different types of disabilities that people have and the statistics pertaining to those disabilities. It goes into detail about wheelchair disabilities including the technology that goes into the wheelchairs and the statistics of quadriplegic and cerebral palsy wheelchair users. This design will implement a head tracking mouse interface as its main interface for a user to communicate directly with the Raspberry Pi 2. As the user is assumed to have little to no use of the limbs, they are to use this interface to control the computer's mouse, allowing them to navigate around the system. A camera module will be used to give the Raspberry Pi the ability to take and record video. The secondary interface will be the onscreen keyboard. Using their head to control the mouse, a user will be able to type in various commands to further control the computer, which we can classify as the hardware layer user interface. The hardware layer of the interface will allow the user to safely avoid obstacles while traveling in their wheelchair. The relationship the hardware interface has is the communication between other microcontrollers and devices including sensors. Since their head movement is restricted to a certain amount, their field of vision is also very restricted and obstacle avoidance is a good safety precaution for the user. There is a software layer of the user interface as well. This software layer consists of a program that the user will be able to open up and either performs daily tasks, activities or even just monitoring a set of systems. Based on the preferences of the user and the resources that are provided to them, the program will have a series of shortcuts organized by those preferences which will be able to access those available resources. The software user interface has a relationship that can be described as the direct interaction between the Raspberry Pi and the user himself. The features within our design all have common goal: to improve the life of a quadriplegic disabled person by giving them independence through technology. Much time and research has been put into producing deployable prototype of our interactive mobility design. The idea began with inspiration to help a dear friend who suffers from the medical condition cerebral palsy, but over time***

has grown into something so much more. The design can be applied to all types of situations for people who are dependent upon others because of limitations to their movement abilities. It was interesting to see how our original goals changed and developed as the time went on. We were displeased with certain aspects and worked hard to bring them to a point that we were satisfied. We want our interactive mobility to be introduced to the world in a way that these people feel comfortable using our product, as they were always in our thoughts and served as inspiration during the process of design.

Keywords—GUI, IDE, GPIO, I/O, CMOS, ADA, SCI, IR, EEG, *CP, SCI*

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

Executive Summary

There are many different types of disabilities that limit people's abilities or hinder their capabilities to perform tasks that are normally done without thinking. To better the lives of a group of people with disabilities we focused on quadriplegics and people with cerebral palsy. People with these conditions typically have limited range of motion such as the inability to not control their arms and legs, leading to a dependence on a caretaker or another person to perform basic tasks. To extend these people's abilities and provide a greater sense of independence we created our system with the user in mind.

To allow the user of our system to achieve a sense of independence, we had to create our system in a way the user would be able to interact without the help of another person, and with a limited range of motion. Taking this into account we designed our system to use an open source head tracking software to control a cursor on a computer screen that would then allow them to navigate our user interface to perform certain tasks. Through the head tracking software, the user would be able to click on the screen to open and close a door, control a fan, and control a light in a room. In addition to these tasks, safety features such as object detection and an emergency contact system was included to make sure the user is safe while a caretaker doesn't see them. The object detection feature will scan the front of the wheelchair and show the user the direction of the object, while the emergency contact system will continuously check to see if the user hasn't tipped over, and if it does a caretaker will be notified of the user's distress.

To confirm our progress and ensure our end goal is met we created a timeline of all the tasks we would need to complete as well as milestones we would achieve along the way. Continuous changes were made to the timeline to accommodate changes and unforeseen setbacks. To fill in the timeline we created a work break down structure that detailed the work needed to be done for each feature as well as the team member that would carry out the task and the amount of time needed to complete each task. Additionally, a risk assessment was done to find problems in our system early in our design to mitigate the probability of a problem and the effect of the problem on the system.

A need for our product in the market is key for the success of design. For this reason, we conducted a market review looking at similar products to see how our product is different than what is already in the market, the size of the market, and the prospects of the market we are entering. Using this research, we could compare the budget we allocated for our design and the total cost of our design to similar products we saw in the market and see if we could achieve our goal of lowering the cost of products that performed similar functions to what we are doing.

Testing was conducted to ensure the problem statement was being accurately addressed as well as ensuring the correct metrics were being fulfilled in regard to the feature set and obtain the expected results from our design. After the tests were conducted changes, necessary changes were made to our design to better meet the feature set we outlined in the Fall semester. As a result of the testing and the changes we made, we were able to show that each of the feature sets were met and the correct metrics were satisfied. We outlined the feature sets and showed that they were met in status report of deployable prototype.

As our design is still a prototype showing a proof of concept, we will still need to make changes to the design to achieve a marketable device based on our research for the market review. One area we will need to improve on is the hardware we are using for some components of our system. This will enable us to have a more reliable system that will have a more realistic operating time. Further refinements such as reducing the number of microcontrollers used in the system. Even though we lowered the number of microcontrollers used from fall to spring semesters, a single microcontroller will be needed for a marketable system.

I. Introduction

Disabilities come in all shapes and sizes. Some can be relatively easy for a person to adapt to, while others come with a much more difficulties to overcome. They can happen to anyone; from the elderly getting older and developing conditions such as arthritis, to the children with cancer or serious diseases that prevent them from having control of their own body, causing them to require a wheelchair. It can even happen to someone in their twenties or thirties if they have an accident, not caused by some genetic illness. There are many different types of problems that people have to deal with every day. We have developed treatments and techniques to help many of them, but there are still many things that these people can't do. With the advent of new technologies there have been many solutions presented for these different types of illness' and new methods of helping these people are being presented in an ongoing basis. The purpose of this project is to engineer a way for these dependent people to become at least somewhat more independent, even though they will still probably need a caregiver to help them out. Helping the person inflicted with the disability can overall help both the caregiver and the person. There is a direct relationship between the caregiver and the patient because it can be demanding for the caregiver, and for the patient it is restricting and can be very discouraging. If it is too physically demanding for the caregiver their health is affected and providing one solution that can help even a little will help the relationship both between the caregiver and the patient.

Our design aims to lessen the extent in which a quadriplegic person is reliant on their caregiver by creating a device that is interfaced to allow them to travel in and around the home safely and independently. Our aim is not to create something that will help doing a task faster, because a caregiver provides the fastest form of help, but rather perform tasks themselves. Quadriplegia is having the inability to move their body except their heads, which creates a heavy dependence on caregivers. This solution lessens that demand and brings extra independence to the afflicted person or persons with quadriplegia. There are other ways to help quadriplegic people with getting their independence. We could do speech recognition to communicate with different microcontrollers to help with their independence. Speech recognition can be very difficult because sometimes the microphone may not be able to pick up the commands. The components needed for speech recognition can be very expensive as well. Purchasing software to be able to pick up commands spoken by the user is very expensive, and a microphone that is able to pick up sound with minimal noise is also very expensive. You can have the caregiver constantly by their side twenty-four hours a day, but our solution to this problem lifts some of the burden off of the caregiver while at the same time giving them some of their independence. Other technologies such as EEG sensors, or sensors that detect brain waves, have also been used to help the disabled. This approach however can be expensive like the speech recognition, as well as being difficult to work with as there can be a lot of noise that could get in the way of measuring the brain wave signals. The safety feature that we are planning to implement is also a plus since most wheelchairs only have basic safety device like seat straps and some sort of braking system. Although it adds more complexity for the design of the wheelchair, it also adds peace of mind, for both the caregiver and the individual using the wheelchair. Most wheelchairs also don't have a method of detecting if it will run into an object. Granted this isn't usually needed since the operator of the wheelchair can detect the object and maneuver the wheelchair to not hit the object, but for someone who doesn't use a conventional means of controlling their wheelchair it might be more difficult. Having a method for object detection is definitely an add-on compared to the capabilities of most wheelchairs.

The whole project is broken down into the smallest tasks so that smaller, simpler tasks may be designed and debugged independently. Trying to process and debug the entire project at once may be problematic, while trying to work on the project as whole without having any deadlines can also cause issues. Deadlines ensure that the project is moving along as well as making sure that there will be enough time to finish it. The features that we are implementing in our project are to be considered the top level of our design. These features will then be divided into sub tasks, which are then broken into smaller activities. Completing all activities within a sub task will determine that the sub task is complete. Once every sub task and activity are finished with a specific feature, we know that the feature is completed. Completing a feature doesn't necessarily mean that it will be able to merge with the other features and be functional right away. Every feature of this project like the obstacle detection and the emergency system will all be tied together through the graphical user interface. So there will need to be time dedicated to getting the main on board microcontroller to interface with all those other features. Ryan Law is in charge of the communication between the main control unit and wirelessly communicating to various appliances, Vadim Artishuk is in charge of the graphical user interface so they will both need to establish a communication that can later be programmed into the graphical user interface.

The graphical user interface is broken down into three parts that include installing and using the free camera mouse head tracking software, creating a GUI, and establishing a communication between the different microcontroller. The head tracking software uses a webcam to track the head movements of the user and translate them into mouse movement. Once this is installed and running, the user may operate a mouse to traverse the screen or display, so that they may select which feature they want to use. The GUI will be the application that the user uses to communicate with the other microcontrollers. That application needs to be organized based on safety and preference of tasks. Once everything is organized, a template will be made to specify how the GUI will appear to the user. Vadim Artishuk will then be able to program the application. The ability to navigate through the application and between different features of the wheelchair will determine that the sub task is complete. The other features, such as the emergency call system, will need to be able to communicate back and forth between each other to ensure the user is informed of any alerts through the user interface. The microcontrollers must constantly be able to communicate back and forth between the main control unit. Once this has been accomplished, the GUI can be programmed to execute its various commands through the established communication.

Our design's next feature, object detection, can be broken down into five different subtasks with each subtask having one or more activity related to the subtask. The first task will be to measure the accuracy and capability of the sensor we will be using so that we can maximize the use of each sensor and cover as much scanning area with a single sensor as possible. Then we will set up the sensors and program the microcontroller to receive data from the sensors and perform the correct actions based on the inputs it is given. The object detection system will communicate with the user interface when needed to notify the user if something is found.

The feature allowing a user to control household appliances wirelessly will be broken down into six different subtasks. First of all, we will need to communicate the signals between the various microcontrollers that control appliances found within the home. Some basic I/O programming will be needed to control the various hardware associated with opening and closing doors, turning on and off lights, and adjusting the speeds of fans. We are then to establish wireless communication between the microcontrollers and the main control unit. Simple signals will be sent and received, ideally through some form of existing wireless technology, between the microcontrollers and the main control unit. We are then to integrate this technology into the basic automation of household appliances. Finally, these commands are to be programmed so that they are easily accessible within the graphical user interface.

Our final feature is probably the most crucial feature to our design, as it will allow for the user to be safe while at the same time being able to navigate around the home independently without the constant watch of as a caregiver. This feature is an emergency call system that will allow a user to alert either emergency assistance or a caregiver that they are in need of help. This feature can be broken down into two main subtasks. The first task will be to implement a rotational system that monitors various statuses of the wheelchair through the use of a gyroscope. The second task will be to create a communication system that allows the user to communicate between the main control system, call system, and necessary microcontrollers. Like the other features within our design, this feature will eventually be integrated into the graphical user interface so that the user may easily access it.

The risk of our design is to be measured by the impact of a certain component of a project and the probability that it may happen. So if the component has a very high probability of happening and a very high impact to the project, the risk can be seen as very high as well. The earlier stages of the project are the riskiest because of lots of unknown variables. For example, not having all the research done that is needed for a project will increase the probability that something can go wrong will increase the total risk. Uncertainties can be related to risk because the more uncertain the group is with their project, the higher the chances that their tasks can go wrong which effectively increases risk. So researching more on each task can decrease risk, and researching alternate routes or more detailed descriptions of the task at hand can decrease that risk. Our project risks can be divided into three different categories, financial risk, hardware/software risk, and time. In each of these sections we discuss the different risks that could appear and our plans on the methods we could use to avoid the risk or handle the situation if something appears. After listing out the risks with our project we then estimated the probability that each specific risk and the impact that it could have on the project. Using this analysis of the different risks we can better see the problems that could occur with the project and be better prepared to fix a problem if something were to occur.

Starting with our financial risks, we broke that down into three of our features which include the user interface, object detection and our safety feature. The user interface is the most important feature because it allows the user of the wheelchair to interact with the rest of the system which means that the user interface has the highest impact on our project completion if something were to go wrong. Our goal is to make a low cost solution but depending on the way we implement the user interface there could be a risk of us spending too much money to implement this feature. Our use of head tracking software relies on a windows operating system, and buying a device that could support this software could make us go over our budget. The object detection and safety features are also included in the financial risks. These features have a risk because if the initial sensors do not work for the application then costlier sensors will need to be bought, which also takes away from our budget and our goal of creating a low cost system.

Our next section of risk includes the hardware and software risks. This section is broken down into the user interface, wireless device control, object detection, and safety features. This section includes topics such as the probability that a component would break or fail, or if a programming or software error were to occur. As like the financial risk section, the highest risk is if something fails in the user interface feature. The user interface has to take multiple inputs from the other components and act on those inputs and display the correct screen to the user. This means that we have to make sure that the device controlling the interface can reliably communicate with the rest of the devices in the system, as well as be able to run for a long period of time. The wireless device control also poses a programming risk because of the language that is needed to program the wireless device, which means that communication between the user interface and wireless may have to be programmed in the same language. The object detection and safety features have risks of getting false data from the sensors, which in turn could display incorrect messages to the user. These risks are lower than the user interface feature but can still affect the system.

The last section we talk about are the time risks that impact this project. This section is split up into the same sections as the hardware and software risks stated above. These risks include the inaccuracies of the estimations made for the time needed to implement a feature, the time it takes waiting for parts to ship, and the risk of time being wasted by implementing a part of a feature and then having to redo it because the implementation doesn't work.

II. Societal Problem

A. Different Types of Disabilities

According to studies done in the US by the University of New Hampshire, the percentage of those who had an ambulatory disability (difficulty walking) was fairly low for those aged 5-17, but it jumps when you go into the 18-64 age group, and it then soars in the 65 and over age group [1]. Figures 1 and 2 below show a few graphs on the percentages and areas where these disabilities tend to occur.

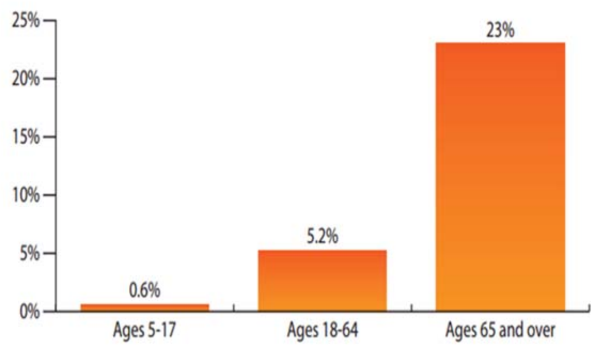

Figure 1 List of percentage of people with ambulatory disabilities by age group[1]

According to the same source, when asked if anyone in the household "had difficulty dressing or bathing" the numbers were much less, but they were still mostly prevalent in the southern United States [1]. This type of disability is called self-care disability and the data is shown in Figures 2,3 and 4. This type of disability is important because someone in a wheelchair could also have a self-care disability.

B. Disabilities Requiring Wheelchairs

Quadriplegics are individuals who are paralyzed in all four of their limbs. Quadriplegia can be caused either by an accident that in some way injures the spine, or there are also several documented medical conditions that cause this condition. Around 7800 spinal cord injuries occur per year in the US, and that is only those which are reported [2]. Most researchers believe that this number is under representing of the actual amount. They consider that, patients who die immediately or soon after the injury, and those who have neurological problems secondary to trauma, but are not included in a Spinal Cord Injury (SCI), may not be counted among the statistics. This would increase the number by 4860 more per year [2]. Table 1 below shows some of the causes leading to SCI.

Table I
Causes of SCI [2]

Cause	Percentage
Motor Vehicle Accidents	44%
Acts of Violence	24%
Falls	22%
Sports	8%
Other	2%

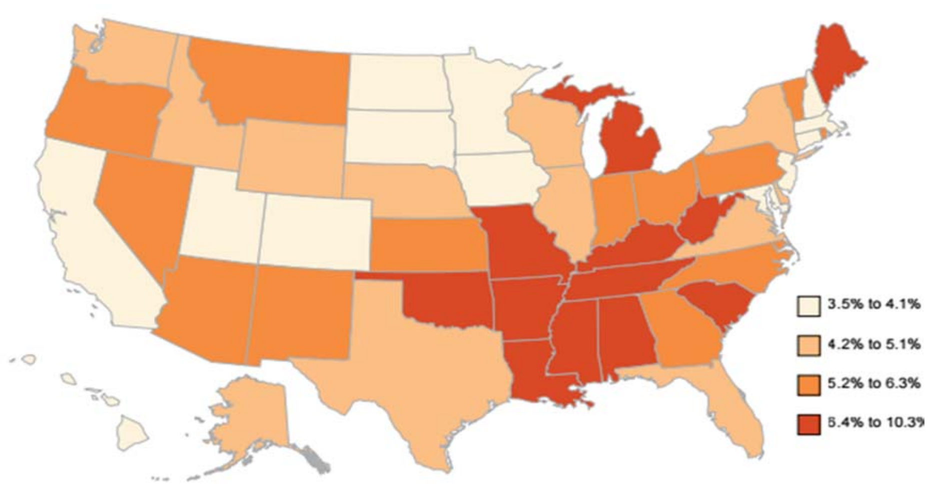

Figure 2 Areas in US where ambulatory disability is most prevalent among 18-64-year-olds[1]

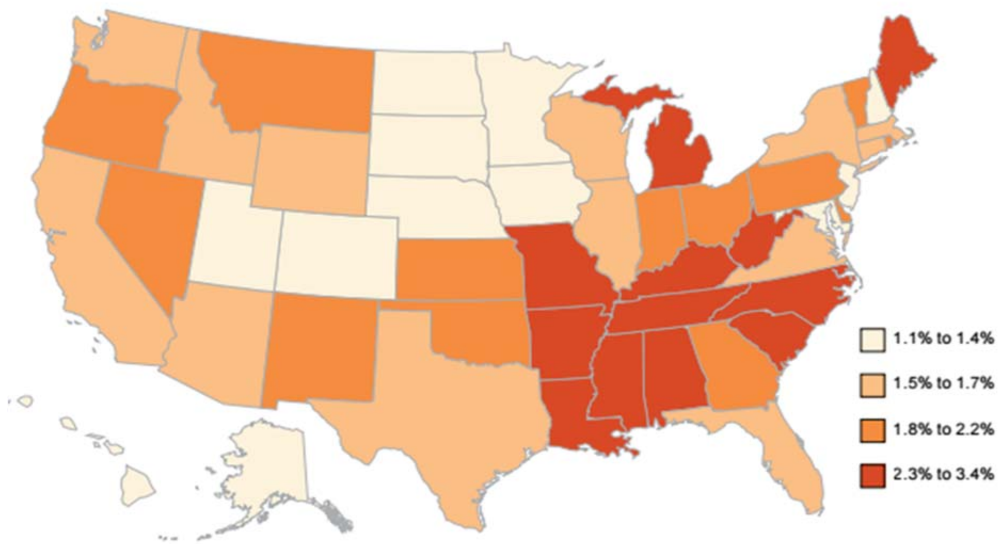

Figure 3 Areas in US where self-care disability is most prevalent in ages 65 and older[1]

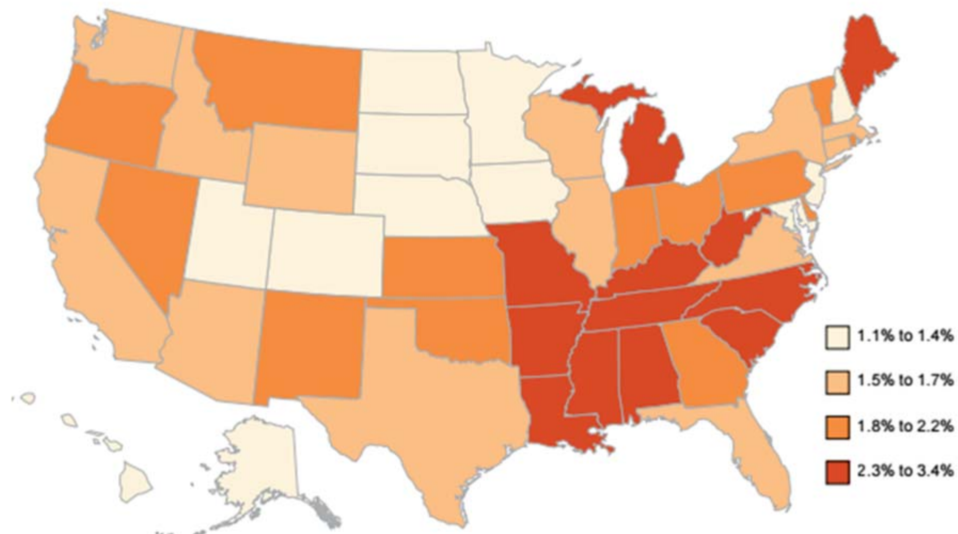

Figure 4 Areas in US where self-care disability is most prevalent among 18-64-year-olds [1]

Cerebral Palsy (CP) is a condition which can cause a person to become quadriplegic. CP is a neurological disorder that generally appears during infancy or childhood and affects body movement, muscle coordination, and balance of the individual [3]. The majority of people with CP are born with it although it may not be diagnosed until months or even years after birth [3]. CP doesn't always cause profound disabilities and it doesn't seem to affect life expectancy [3]. Most children who have CP also have either average or above average intelligence [3]. There are many treatments, and medications out there which can improve the motor skills of a person with CP [3]. While there are cases of people with CP who require little to no special assistance, a person with severe CP might need lifelong care [3]. Because of their needs people with cerebral palsy or who are quadriplegics can be part of the percentages that make up the ambulatory disabilities and self-care disabilities stated above.

While we have been able to support those requiring wheelchairs it has not been an easy task. We have come up with a set of guidelines that enable us to take better care of the physically disabled, while still allowing them to participate in society without being discriminated against. The ADA Compliance laws, or Americans with Disabilities Act, has allowed us to care for these people to a limited degree, but we still do not fully understand the needs associated with taking care of them. One of the key things spelled out in the compliance laws is the need for wheelchair access to all public areas. This supplies that any wheelchair ramps in public areas will not be above a specific angle of 4.8 degrees. These along with other requirements are named in the ADA compliance laws which allow exceptions for these disabled individuals.

C. *Wheelchair Technology*

Since the first wheelchair was invented, not many advancements have been made to benefit the disabled until 1933 where Harry Jennings and Herbert Everest invented the first lightweight and collapsible wheelchair called the "X-brace" [4]. Their design implemented a pivot bolt and cross bar system that allowed their device to collapse as shown in Figure 5. Ever since then many different types of wheelchairs have been invented to help the disabled with their mobility and independence. The wheelchair the Jennings and Everest invented was a manual wheelchair where the user had to propel themselves by pushing the rods next to their wheels, but now there are things such as electric wheelchairs, sports wheelchairs, standing wheelchairs, and even dog wheelchairs.

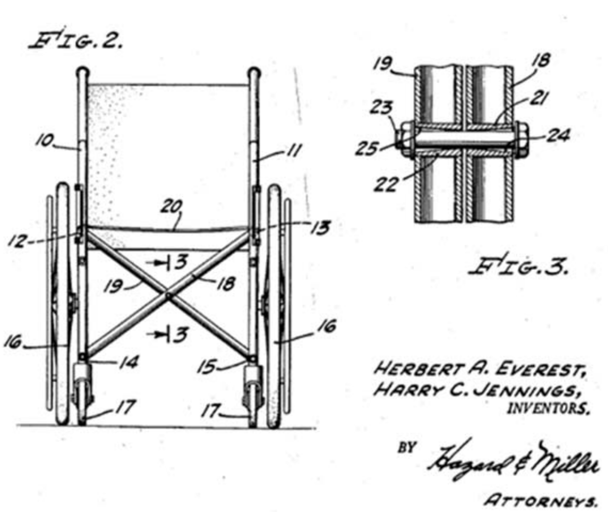

Figure 5 Herbert Everest's and Harry Jennings's design of wheelchair [4]

In its most basic form, a wheelchair consists of two main components: wheels and a seat. This device provides a means of transportation for people with disabilities. As the complications of disabilities vary for each individual condition, the technology regarding wheelchairs has grown to accommodate certain disabilities to help individuals live the lives more independently.

Currently, individuals lacking the motor skills necessary to propel a basic wheelchair often resort to using wheelchairs that are powered electrically. This gives them independence as they are not reliant on other people for transportation and improves their quality of life.

Unfortunately, the simple electric wheelchair is not a "fix-all" solution, since many individuals' motor skills are affected differently. Therefore, systems must be tailored to specific disabilities to ensure that the users able to fully benefit from the system.

As Permobil is one of the leading providers in rehab mobility solutions, a few of their products are examined to help understand what types of technologies exist to cope with the problems in dealing with a debilitating condition.

We looked at three of Permobil's seating system which they called Corpus 3G, MX, and F5 Corpus VS. When looking at the Corpus 3G we saw that it was really customizable. The seat's width and depth were both adjustable which allows for different sizes of users. Another feature of this chair is that it was compatible with many after-market backrests which allows the user to customize their chair to be more comfortable. The system has a powerful center mounted actuator that offers a full fifty degrees of tilt, and a full 175 degrees of recline. The armrests are complete with channels made for routing cables. Accessories may be easily mounted onto the system [5]. Table II below shows all of the features that the Corpus 3G offers.

Another chair offered by Permobil is called the MX which is designed to be used by children. It has a unique function that allows the seat to be lowered to the floor allowing for more activities such as playing with others or listening to stories. It is a modular system that can be adapted with many different accessories and customizations. It also has adjustable leg rests that enable the chair to grow along with the child [5]. Table III below shows all of the features that the MX offers.

The last chair that we looked at was called the F5 Corpus VS. This chair combines all the features of the Corpus seating system with a standing functionality. This functionality gives the user the ability to stand and drive simultaneously, complete with automatic anti-tipping wheels designed to provide greater stability while driving in the standing position. It features a fully independent suspension made to absorb everyday terrain and obstacles. This system also has positive health outcomes such as pressure relief and improved circulatory, GI, and respiratory systems [5]. Table IV below shows all of the features that available for the F5 Corpus VS.

Another feature of wheelchairs that we looked at was the drive controls. Because people with disabilities are limited in certain motions, the wheelchairs that they use have to allow for different ways to control the wheelchair.

The first and most used control for wheelchairs is the joystick. Permobil's joystick module has simple joystick is used to interact with a wheelchair's control system. This component features a 2.75-inch display along with a 1/8-inch switch jack designed for programming. The module is also available in both compact and heavy duty versions [5]. This type of control is easy to control and implement but cannot be used by quadriplegics and people with CP.

Another way that the user can control the wheelchair is through the "Total Control Head Array". The Total Control Head Array fits behind the user's head and allows them to access different pressure sensing buttons allowing for multiple thresholds of control. Telescoping arms and mountable brackets allow for switches to be mounted within the range of the user's head mobility [5]. This type of control is better for people who have no use of their arms but impractical for those that can use their arms.

A third method of motor control is call the "Sip and Puff System". This device is designed to be used by individuals with no extremity function. This device fits into the user's mouth similar to the way that a straw does. The device translates pressurized breaths into various commands through a fiber optic switch [5]. This type of control system is good for people that cannot move their arms or legs and possible even their neck, however it does not allow for the user to turn as the user can only sip or puff.

Table II
Specifications for the Corpus 3G [5]

CORPUS 3G SPECIFICATIONS

Legrest Elevation	(POWER)	85° — 170°	Tilt Options	(POWER)	0° — 50°
	(MANUAL)	85° — 170°		(MANUAL)	-5° — 45°
Armrest Pad Lengths	10", 13", 16"	260, 335, 410 mm	Recline Options	(POWER)	85° — 175°
Seat Widths	17", 19", 21", 23"	420, 470, 520, 570 mm		(MANUAL)	85° — 120°
Seat Depths	14" — 22"	370 — 570 mm			
Backrest Heights	20", 23" — 28"	500, 570 — 710 mm			

Table III
Specifications for the MX [5]

MX SPECIFICATIONS

Legrest Elevation	(POWER)	No Power Option	
	(MANUAL)	90° — 125°	
Armrest Pad Lengths	8", 12.5"	200, 315 mm	
Seat Widths	10", 12", 14", 16"	260, 310, 360, 410 mm	
Seat Depths	10" — 18"	260 — 460 mm	
Backrest Heights	14" — 22.5"	360 — 560 mm	

Tilt Options	(POWER)	0° — 45°
	(MANUAL)	No Manual Option
Recline Options	(POWER)	No Power Option
	(MANUAL)	85° — 105°

Table IV
Specifications for the Corpus VS [5]

Maximum User Weight	300lbs	136 kg
Maximum Speed	7.5 mph	12 km/h
Driving Range (1)	16 miles	26 km
Base Width	25.5"	648 mm
Base Length	43"	1093 mm
Minimum Turning Radius	30"	762 mm
Ground Clearance	3"	75 mm
Drive Electronics	R-net 120A	
Battery Type	Group 24 Gel	

Optional Seat Elevator	14"	350 mm
Seat to Floor Height	17.5", 18.5", 19.5"	445, 470, 495 mm
Power Tilt Options	0° - 50° Posterior	
Recline Options	85°-180° (power)	
Legrest Elevation	90° - 180°	
Armrest Pad Lengths	10", 13", 16" & 18"	
Seat Widths	17" - 23" (by 2" increments)	420 - 570 mm
Seat Depths	14" - 22" (by 1" increments)	370 - 570 mm
Backrest Heights	23" - 28" (by 1" incr.)	580 - 710 mm

The last method of motion control is through eye movement. While still in a prototype stage, a control system using a CMOS sensor camera detects a user's eye movements and interprets them into commands. The image produced by the camera is interpreted with the functions provided in Matlab's "Image Processing" toolbox. The image is processed into three different regions: left, right, and forward. Depending upon where the iris is positioned within the processed image determines which function the user is attempting to activate, as shown in the figure below. The system processes no iris (closed eye) as if the user is trying to deactivate the systems current mode. Laser and distance sensors are predicted to be implemented in the future to provide more safety with this system [6].

Figure 6 below shows a model of how camera uses eye movement to figure out direction.

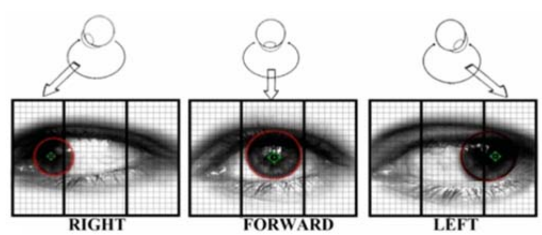

Figure 6 Detection of Iris Position [6]

Home automation is just a luxury for many people in the world, but for somebody with a disability, this may actually be their world. Home automation helps these people to live their lives independently, without the constant need of another person in which they are dependent upon. Fortunately, this kind of technology is becoming more readily available.

There are many forms of assistive technology associated with helping someone with a disability live independently. Automatics locks may be controlled remotely with the touch of a button. Security systems allow someone to monitor their property or even just to see who is knocking on the door. TV's and entertainment systems may be controlled via a central multimedia hub. Thermostats may be accessed and adjusted over the internet. All of these things can be done from almost anywhere in the world, and for our purpose, from the control of a disabled persons wheelchair.

D. *Wheelchair Market*

According to a study done by the University of New Hampshire, the amount of disable people in the US has been slowly climbing from 12.1% to 12.6% from 2008 to 2014 as shown in Figure 7 [1]. One reason for this increase is because of the "baby boomer" generation. As shown in Figure 1 the amount of people that need wheelchairs are mostly people over the age of 65. This generation of people is getting closer to the age of 65 and older where the most amount of people need wheelchairs. This means that there is a higher chance of increase in the wheelchair market.

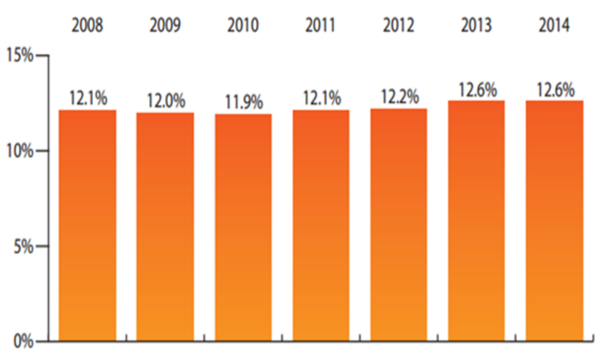

Figure 7 Percentage of people in the US with Disabilities [1]

As of 2015 the expected market for manual wheelchairs in 2018 is 2.9 billion dollars and 3.9 billion for electric or power wheelchairs. As of 2015 there were an estimated 3.3 million wheelchair users in just the US alone and the number continues to increase every year due to the "baby boomer" generation. From the same figures, there was an increase in wheelchair revenue by 2.5% from 2009 to 2014.

E. *Caregiver Health*

It is understood that when a child has a disability that family centered services are a lot better. This will cause an increased demand on the family members to provide for the child and can be very exhausting. The family member can feel extremely burdened to the point where the attention the child or person needs is not met [7]. It might seem that caregiver health might be irrelevant when the child is the one that has the problem, but you can't dismiss the fact that it is important.

It was shown that even though the mother caregivers were caring for the child more, the father was experiencing similar health effects as the mother, not as severe but the trend was there. In D.A. Hurley's article, he conducted a study by having the different families complete questionnaires. A total of 312 questionnaires were sent to homes of 156 children [8]. The study provided enough evidence to show that the health of the caregivers was significantly lower than expected. The questions included amount of time spent per day spent with the child as well as the SF-36 version of the health survey [8]. This survey is a good way to measure generic health. The survey gives the person a score, a 50 is considered to be average in the United States [8]. The SF-36 score is transformed into a 0-100 scale. They made a conclusion that caregivers with more dependent children scored significantly lower on the bodily pain measurements. According to the study they were unable to figure out the exact reason for that but it is safe to say that the increased care needed by the child involves greater amounts of physical activity [8]. They weren't able to make the connection where the severity of the child's disability can affect the care givers health because you can't study that directly.

Table V
Caregiver Health [8]

SF-36 domain	Male caregivers (n= 60) Mean (SD)/median	Female caregivers (n= 94–99) Mean (SD)/median
Physical functioning (PF)	52.79 (5.19)/54.9	48.91 (8.33)/50.7
Role physical (RP)	52.12 (6.74)/55.7	46.99 (9.89)/49.5
Bodily pain (BP)	52.47 (9.84)/55.4	47.72 (11.29)/46.1
General health (GH)	50.26 (9.69)/52.0	47.66 (11.62)/50.6
Vitality (VT)	51.01 (7.75)/49.0	43.33 (10.29)/42.7
Social functioning (SF)	51.27 (7.60)/56.8	44.87 (11.29)/45.9
Role emotional (RE)	50.45 (9.23)/55.9	43.17 (12.19)/44.2
Mental health (MH)	51.37 (8.18)/52.8	44.05 (11.49)/47.2

Tests showed the comparison of a female caregiver compared to the general population; take a look at Table V. The primary caregivers for the patient were mothers and you can see the drastic effect in mental health. The mother experiences the worse effect because she is in constant contact with the child and those effects can only accumulate from there. The father isn't as involved as the mother and therefore experiences less of an effect but still the graph in Table V shows that the score is still lower when compared to the general population.

There are lots of other factors that can affect a caregiver's health, for example the money that is available, the behavior of the child and even the family function. Here in Figure 8 you can see how the income, child behavior, caregiving demands and family function play a role in the health of the caregiver.

All the different connections have an effect and they are measured by β. You can make the connection that the bigger the β the more beneficial the effect. For example, the physical health associated with less caregiving demands gives a β of .23 when compared to the psychological health that has a β of .12 [7]. The information shows the benefits the caregiver can experience by a less demanding child or patient. Providing the child or patient with the ability to direct their attention elsewhere means less demands for the caregiver. If you take a look at figure 8 again you notice that caregiving demands are direct effects of the health of the caregiver. The conclusion they came up with from that graph was that child behavior and caregiving demands were the strongest factors in the health of caregivers [7].

The point that is being made here is that providing a way to allow the child or person with disability to be more independent is beneficial to both the person and caregiver. The psychological and physical health of caregivers was very influenced by caregiving demands [7]. The person with the disability wants to be able to do things without having to constantly ask for help, the sense of independence is very important to them. Providing them with maybe a little sense of independence is what the caregiver wants as well. You can make the connection that by helping the disabled person with their limitations that in a sense you are also helping the caregiver. For example, being able to access the web by themselves without the help of a caregiver will give them the freedom to do so even when a caregiver is not around to help. This can help increase their quality of life and sense of belonging. Having access to current events is almost an essential part of life for everybody and having this access will be satisfying for them. If the caregiver is a parent of the child, it will be in their best interest to have a way to help the child be more independent so that the parent can be more capable of taking care of the child [9].

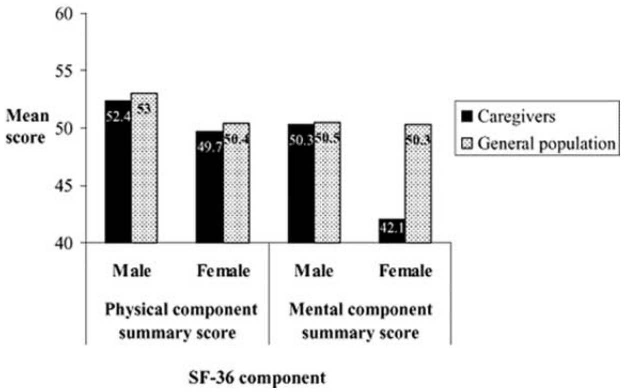

Figure 8 Caregiver Health Compared to General Population [9]

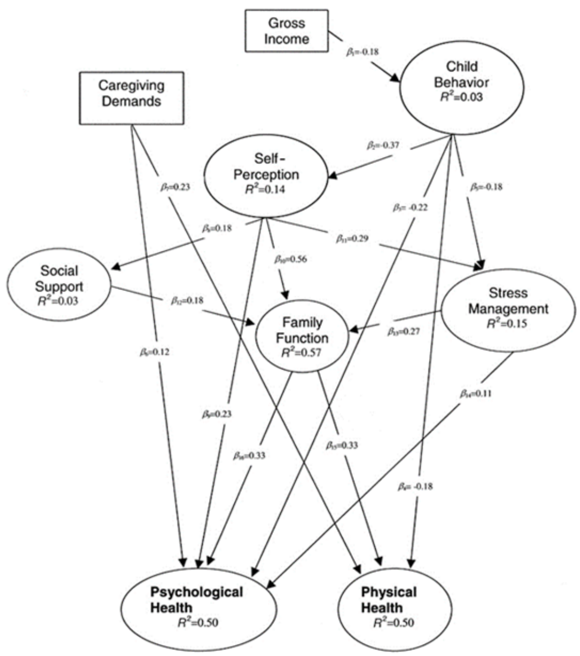

Figure 9 Aspects that can affect a Caregiver's Health [9]

After doing this research and looking at what has been done to help people with disabilities, we can think about other ways that we could help or other solutions that have not been tested or implemented. One idea that was discussed by the group was the use of voice control to allow the user to control other devices in the home that would all them something like a smart home. Another implantation that was suggested was the use of brain sensors to control the chair. When thinking of these designs we did find that with the voice control it would be hard to recognize a command and then execute it the first time that user would say something. The problem with the brain sensor is that they are relatively hard to work with and configure depending on the type of sensors that are bought. Typically, the better and easier to work with sensors cost a lot of money and it could be out of our budget.

Another design that we could implement would be a combination of the solutions already stated. As stated in section four of this paper the sip and puff method was one way that users of the wheelchair were able to control it, but it does not give the user a way to tell direction. If combined with the eye detection a user of the wheelchair could then be able tell the chair the direction that they want to travel as well as if they would like to move forward or backward.

The design that we do decide to implement should be cost efficient, as much of the user e assistive technology that exists happens to cost thousands of dollars. Something such as the Raspberry Pi would be ideal for prototyping since it is cheap and will give us easy access for manipulating the IO required within our design. A system that allows the user to interact with it physically, such as the open source software CameraMouse, which will track small tilts from the user. Systems will be implemented with the design to monitor the status of its user. We aim to give them freedom and independence, all while keeping safety at the forefront of our design.

III. Design Idea Contract

A. *Resources Needed*

Money will definitely play a big part in the resources we need. Buying parts based on our budget will help decide what is needed. Some consulting from a student or teacher to help debug the graphical user interface may be needed. We will need microcontrollers, this may include the Raspberry Pi and Arduino. Multiple Arduino’s and Raspberry Pi's may be needed. This project will be implemented first with a prototype so lots of simulation to show proof of concept so lots of space will not be required. There is a free head tracking software that will be used to translate head movement to mouse movement on the screen. Also a python IDE will be used to create the graphical user interface such as QT designer. We will also be needing a sensor for detecting the angle that the wheelchair is at. This will probably utilize a gyroscope like what is used in most planes to keep it on a nearly horizontal axis of symmetry. A sensor such as an IR sensor to scan in front of the wheelchair and tell the user if there is something in front that they might not have seen.

B. *User Interface*

Giving quadriplegics the ability to communicate with a computer allows them to focus their time and energy on something else. This attention to the computer makes the person or persons afflicted with quadriplegia require less intervention from their caregiver. Independence is a very important factor, if not the most important, when it comes to a satisfying lifestyle. Allowing the patients to communicate with the computer by themselves provides them with a sense of independence, enough to better their daily lifestyle. It benefits the caregiver as well because the patient's attention is on the computer which requires a lot less demands from the patient. The disabled person experiences the interaction with technology and gives them the ability to access the web and do lots of activities on the web that the vast majority of people do. They now have the ability to read the news, play some simple games, watch video clips or listen to music. Sometimes the caregiver does help search the web for them, but with the user interface they can do it alone and have a sense of satisfaction from being able to do it themselves.

Figure 10 Raspberry Pi 3 model B [10]

With the user interface in place for the disabled, it should make finding the more frequently visited applications or programs are easier to access making their lives easier. The user interface can be done on a microcontroller such as Raspberry Pi which has a Linux operating system seen in Figure 1. It can be programmed in Python language and has the ability to send signals out the GPIO pins to different features of the chair or wirelessly. The webcam will be installed on the front of the wheelchair facing the persons face. The webcam will be used to traverse the user interface.

Rather than navigating through Linux operating system, there will be a medium that makes accessing different features easier for them. This will provide the patient or person with the ability to access internet, use some simple home automation features such as temperature control or television control, and even call for help from the caregiver in emergency situations or regular daily tasks. Something as simple as monitoring pet food and water is enough to give the patient a feel of a regular daily lifestyle which can bring lots of enjoyment. The user interface will be the medium through which all the features will communicate with. This is a program that turns on with the microcontroller and using the small display, the patient can begin with whatever task they feel they want to tackle first. One of the team members worked for a man with cerebral palsy and said that more than half the time he is translating what the man is saying. The user interface is translating what the patient wants to do into tasks that can make his day better and easier.

The software layer of the user interface will require the use of the webcam, Raspberry Pi and the user. The user will face the webcam initially for enough time for the head tracking software to recognize which part of the body it will be tracking. It automatically chooses the center of the screen and that can be adjusted. Once recognized, the mouse that appears on the display will begin to mimic the movements of the user’s head. To click, the program requires the mouse to hover over an icon for a given amount of time, that amount of time can be adjusted based on user preference, and the user is able to fully control the mouse with their head movements. We can use a surrogate or something that can resemble a human head, by holding the surrogate in the same position for a short period of time will cause the mouse to click. Depending on the severity of the quadriplegia, ranging from C1-C8, the ability to move the head is affected. This software has the ability to change sensitivity of the head tracking software to increase the mouse's ability to traverse the screen. Movements can range up to 180 degrees and sensitivity can be adjusted based on the limitations of the person's head movement. For a person that is only able to move their head 90 degrees from side to side as well as up and down, the sensitivity may be adjusted to be able to traverse the whole screen of the main control unit display, or the user interface. Moving the surrogates head left, right, up and down 90 degrees, will cause the mouse to traverse the screen. Using the screen while driving the wheelchair will be a problem if the sensitivity is set to very high so driving the wheelchair while accessing the user interface is not recommended because it may prove to not be useable in that case.

The graphical user interface will include, once the program is opened, inputs that provide shortcuts such as access to obstacle avoidance alerts, environmental monitoring or control, emergency contacts, small set of daily tasks. The user will choose what is needed to be done and chooses from the list of inputs that are on the screen that are organized based on frequency of use and emergency situations. If the user chooses to control the climate of the house, there is a inputs that opens up another window that provides some house temperature and climate control options. If increasing the temperature of the house is the goal there is an input for that and by pushing the increase temperature input, the main control unit sends out the command. Pushing the decrease temperature will send out the command. Once the user is satisfied he can move onto a different task. The user goes back to the main window to choose from another set of tasks. For example, if they choose to turn on an appliance such as a TV, they can push the input which opens the window with some typical remote capabilities like channel up or down, volume up and down. Pushing the input that they so choose will cause the main control unit to send out that specific command. There will be inputs for emergency situations as well. There may be a situation where the wheelchair may malfunction and the graphical user interface will be able to send either a stress call or just a txt to the caregiver or emergency contacts. We will have to consider the different problems that the user interface may encounter such as, freezing or an inoperable display. In the event of any of similar inconveniences where the main control unit is unresponsive, a fail safe may be implemented such as a simple restart of the main control unit. The restart of the main control unit can be done either manually with the help of a caregiver or an automatic reboot sequence can be implemented. There needs to be a form of communication for help in case the main control unit is not functional. There will also need to be a way to turn off the user interface if the user is not going to be using it. This will ensure unwanted or not needed commands being sent unintentionally.

The hardware required for this part of the design idea requires the use of web cam and a compatible micro controller such as a Raspberry Pi. Using information seen on a 2D plane through the camera you can assign a frame that you would want to follow and if you recognize that frame moved slightly you will follow it. You can get good enough information from a webcam with a resolution of 320x240 pixels and to be able to process that information. A display to show the graphical user interface to the user. It will need to be able to remove disturbances that may appear in the frame such as objects. There are lots of different programs that can track a person's head movement and the one that this project will use is the free Camera Mouse program developed by Prof. James Gips of Boston College and Prof. Margrit Betke of Boston University. This project has been worked on since 2007 specifically for children with disabilities and later was made readily available, and about 2,500,000 copies were downloaded. Lots of improvements were made in order to be compatible with lots of different cameras and different hardware specifications. That application is very simple to use and has been tested to work well with a built in webcam on a laptop. To implement the user interface design, it will take about 40-60 hours to get a fully functioning interface that can do the bare minimum. Depending on the difficulty of coding in Python, I can decide to use Java which I am a lot more comfortable with. Vadim Artishuk will be in charge of implementing this feature.

C. *Wireless Communication to Household Components*

One of the key features of this design is to give people that are disabled with quadriplegia control over household tasks of their daily life without the assistance of other individuals. Examples of common issues that this design is to address include turning on and off lights, adjusting a thermostat, and opening and closing doors. As a person who is disabled with quadriplegia is likely to have issues completing these tasks, this design aims to help complete these tasks through the communication between a microcontroller and the main control unit. This allows them to complete tasks individually without the assistance of a caregiver or a family member. This would increase the individual's overall quality of life.

A person who is quadriplegic will have little to no movement below their neck. These people usually are only able to move their neck and heads, typically with limitations on their range of motion. This design plans to use some sort of computer device such as a Raspberry Pi as a main control unit to complete tasks around the home. The control unit is to be interfaced so that the user may interact with the device using their head as discussed in the "User Interface" section of this design [11]. A wireless control system will be used to control the various household components remotely via some sort of Bluetooth or Wi-Fi technology. The main control unit is to be attached to some sort of display. A graphical user interface is to be programmed and implemented so that the user may easily understand how to interact with the various features of the design. This GUI will be written as an application in the programming language Python, to be accessed by the user when they desire to control their home through the main control unit. It is estimated that about 120 hours will be spent on this feature of the design. Much is time will be spent on the functionality of wireless communication between the main control unit and the microcontrollers (approximately 75-90 hours). Ryan has experience programming components such as servos, input devices, and microcontroller communication, therefore, physically manipulating household components such as lights, fans, and doors should not take too much time (approximately 10-15 hours). Programming a graphical user interface to access the commands should take about 2-3 hours to complete as Vadim is handling a majority of this function.

Hardware and Software:

- Virtual Interface:

"Most users of computers use standard peripherals, keyboard and mouse. It is difficult to imagine the work without these devices. However, a number of users exist who are unable to operate these devices due to their physical impairment." [12]

There is various technology made to assist people with disabilities such as interaction techniques such as simple hand motion, eye-tracking, and the detection of myoelectric signals. Unfortunately, these hardware interfaces are limited in that they are only able to provide a small amount of control signals. A user trying to control an on-screen keyboard may find this either tedious or difficult to accomplish with their limited accessibility. Therefore, it is important to optimize the speed and efficiency at which the user is able to accomplish this task [12].

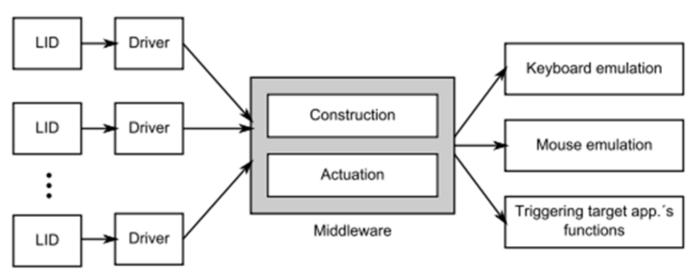

Figure 11 Optimization of Interface's Middleware [12]

- Open Source Software IDE:

This open source IDE will be used to control the software that is to be run on the micro controller. It will be used to program the various sensors used in obstacle avoidance and any other sensors that may be necessary for obtaining a functioning product.

- Wireless interface:

This device will be used to communicate a command from the main control unit to a microcontroller that controls basic commands for household components such as lights, fans, and bedroom doors. Ideally, some existing, wireless interfacing technology such as Zigbee will be incorporated into the design. This would be beneficial in that much work has already gone into building and improving these devices. Less work will be required to implement basic tasks as the device will serve as a foundation for implementing more complex features.

- Microcontroller:

Much of modern society's technology is aimed to assist the average individual with their wants and needs in daily life. As most of this technology is aimed at the average individual, people with disabilities are often hindered in the fact that they often are unable to interact with these devices. Fortunately, the Raspberry Pi is both affordable and powerful. It uses its GPIO (General Purpose Input Output), to easily interact with various hardware and accessories. The Raspberry Pi would be incredibly useful to somebody with a disability in that they would have the ability to adapt various components with modern technology in a way that they could easily interact with this technology [13].

The Raspberry Pi would be valuable to a disabled person in that they would access to communicate with the world. As the Raspberry Pi is actually a fully functioning computer, it could easily be interfaced to allow someone with a disability to use a computer. They would be able accomplish specific tasks such as browsing the web, instant messaging, typing documents, etc. This could be beneficial in that they would be able to apply these abilities towards developing a vocational or educational background.

Another way in which the Raspberry Pi would be useful within this design is that it can be programed to interact with various microcontrollers such as the Arduino. Its GPIO allows it to both send and receive signals to and from the microcontrollers in the form of voltages. Since there are many resources available for the Microcontroller such as tutorials and open-source software, and therefore will be used to interact with the Raspberry Pi within this design.

Arduino also has a variety of microcontrollers. The Arduino is one of the more fundamental models that the company produces. The Arduino Mega 2560 is quite similar to the Arduino Uno specification wise, but it has many more connection ports as shown in the table below. This may be necessary for implementing the amount of hardware associated with this design such as the sensors implemented in obstacle avoidance [14]. Dependent upon the whether the Arduino Uno has enough ports to control all of the components within this design, an Arduino Mega may be acquired to simply the logic implemented within the individual circuits.

D. *Emergency Contact System*

Since individuals who possess quadriplegia are primarily incapable of using any of their limbs, one of the features we are planning to develop is an

Table VI
Specifications of Arduino Uno vs. Mega 2560 [14]

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1

emergency call system. Most smart phones now have the feature of being able to rotate the screen when an individual is looking at it based on the orientation that the individual currently has the phone stationed at. If a quadriplegic individual ever had a problem where their wheelchair tipped over when they were by themselves there would be no way for them to contact their caregiver or an emergency service, short of shouting for help for someone in the next room. However, there are some quadriplegics who would have a difficult time with this because they have a difficult enough time just talking, namely individuals with Cerebral Palsy. The purpose of this feature would be to add a sensor to their wheelchair that could measure the orientation of their chair, and if the sensor detected that the chair had gone past 60° the program could send a signal to automatically contact their caregiver and/or emergency services if needed.

There are two main sensor types that can measure how an object is rotating: the accelerometer, and the gyroscope. While both can measure the orientation/rotation of an object they are attached to with varying degrees of accuracy, each sensor has a very distinct way of going about measuring that rotation.

Hardware:

- Accelerometer

The accelerometer is an electromechanical device used to measure acceleration forces [15]. Since it measures acceleration it can be used to sense the acceleration of a device in a particular direction to see if the orientation has been changed, as it does in most of our mobile devices. Accelerometers have a wide variety of uses, anywhere from laptops to cars for knowing when a crash happens to deploy the airbags. They can also be used to measure the pull of gravity on object, which allows them to be able to know when to change the orientation of a phone for the user.

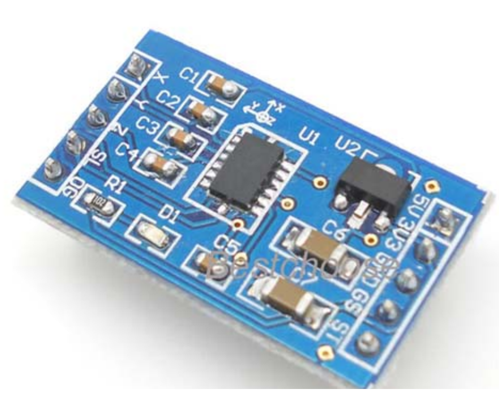

Figure 12 Accelerometer Sensor [16]

Although it may look like a rather simple device the accelerometer consists of many different parts and works in many ways, two of which are the piezoelectric effect and the capacitance sensor [15]. The piezoelectric effect uses microscopic crystal structures that become stressed due to accelerative forces [15]. These exerted forces are then converted into a voltage which is interpreted into a velocity and orientation [15]. When using a capacitance accelerometer changes in capacitance between microstructures located next to the device are detected and that change in capacitance is translated into a voltage [15]. With either accelerometer that is used it will generally be hooked up to some sort of microcontroller that can then be used to interpret the changes seen by the accelerometer into a command or series of commands depending on what the user desires.

- Gyroscope

A

Figure 13 Gyroscope Sensor [17]

gyroscope sensor is similar to an accelerometer in that it measures the velocity of an object; however, it is different in that it measures the angular velocity of the object, or the speed of its rotation [17]. Gyros are generally used to determine orientation and are often found in autonomous navigation systems [17]. So if you want to balance something like a robot, a gyroscope can be used to measure the rotation from a balanced position [17]. A gyroscope measures the angular velocity of an object in revolutions per second, or degrees per second [17]. Generally, gyros are used to measure the rotation of objects that are not supposed to rotate very much. Something like an aircraft isn't supposed to spin much more than a few degrees on each axis [17]. By detecting these small changes, gyros can help stabilize the flight of an aircraft [17]. When measuring the rotation of an aircraft it should be noted that the acceleration/linear velocity of the craft does not affect the measurement of the gyro, hence the different between gyros and accelerometers being that one measures linear velocity, while the other measures an angular velocity.

Hardware/Software:

- Microcontroller

Although we have decided to have the Raspberry Pi to run the main interface, we are unsure how difficult it would be to run the different sensors directly on the Pi and then interpret those signals into a series of commands necessary for our operations. Ideally we will use a microcontroller like the Arduino to read the signals coming in from the sensor and only if it meets certain conditions will it use the Pi to send a signal to contact their caregiver. Once we have successfully programmed the Arduino to react when the sensor goes past 60° and contact the caregiver then we have successfully completed this particular feature.

To set up and test the sensor it should take between 10-24 hours to have it functioning within the minimal set of necessary test parameters. It should take another 10-20 hours to set up an appropriate system that will then activate to contact a specified number, depending on the difficulty of finding a program to interact with outside of the running environment of the microcontroller. Depending on the decision of what microcontroller we use the budget for this part of the feature should be somewhere between $10-$40. Riley Schofield will be in charge of implementing this feature.

E. *Object Detection*

Another feature we intend to implement is obstacle detection. This feature will allow the user of the wheelchair to move around in their home without hitting things that are in the way. This feature could also be implemented together with an autonomous driving feature that would allow the user to select where they want to go and the chair would go there without hitting anything. In order to detect obstacles in the path of the wheelchair, it has to have some sort of hardware that will enable it do so. There are many different options that we can use as a means of object detection. To find the best solution for object detection we have to consider all of the possibilities and look at how they relate to our resources, time and money.

One possible way for object detection is through the use of an ultrasonic sensor. This sensor works by sending out a sound that is unable to be heard by humans and then measuring the amount of time it takes for an echo to return back to the sensor.

Looking at this sensor we saw that it would fit in well with the amount of time it would take to implement and would also be a low cost option. The problem with this sensor is that they become inaccurate as the distance increases between the sensor and the object meaning that it is better for short range object detection. This would be fine if the wheelchair was moving at a slow pace but could be unusable at faster speeds. Another problem that comes with ultrasonic sensors is that they have problems with interference. If used the ultrasonic sensor would pick up other loud sounds which means the sensor would pick up false objects in front of it. The last disadvantage of ultrasonic sensors is that they need high density surfaces for accurate results. This means that if something like foam or cloth is in the way the sound waves would get absorbed and not reflected back and the sensor would not detect the object.

Another type of object detection sensor is an infrared sensor. This sensor is also good because it is a low cost option and is also easy for the team to work with given our experience. It works the same way as the ultrasonic sensor but instead of using sound it uses an IR light and an IR LED to pick up the reflections. The only problems with this sensor would be if the wheelchair would need to detect objects outside. Because the sun gives of infrared light, it would interfere with the sensors ability to detect objects and could give false information.

The third type of sensor is a laser sensor. A laser sensor works by measuring the distance between the sensor and the object by calculating the speed of light and the time since emitted and until it is returned to the receiver. These sensors would be ideal because of their performance and long distance capabilities. The problem with these sensors is that they are very expensive and because we would need to cover a wide range for scanning we would need a couple of these sensors. This means that if used we might not have enough money in our budget to buy other things needed for the system.

The most popular means of object detection is through the use of RGB-D sensors or camera. These sensors are more expensive than the IR and ultrasonic sensors but less expensive than the laser sensors. The RGB-D sensor combines the IR sensor as well as a color mapping that tells the distance by using color. Figure 14 below shows how the camera is able to tell distance by the image that it captures. Using that image, we would be able to see if there is an object nearby based on how well the image appears.

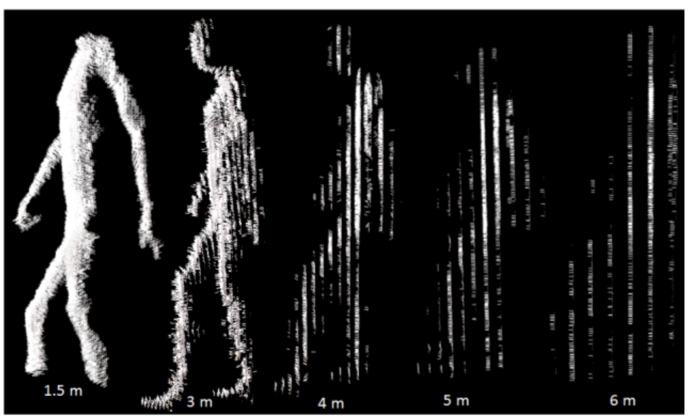

Figure 14 Depth data of a mannequin [18]

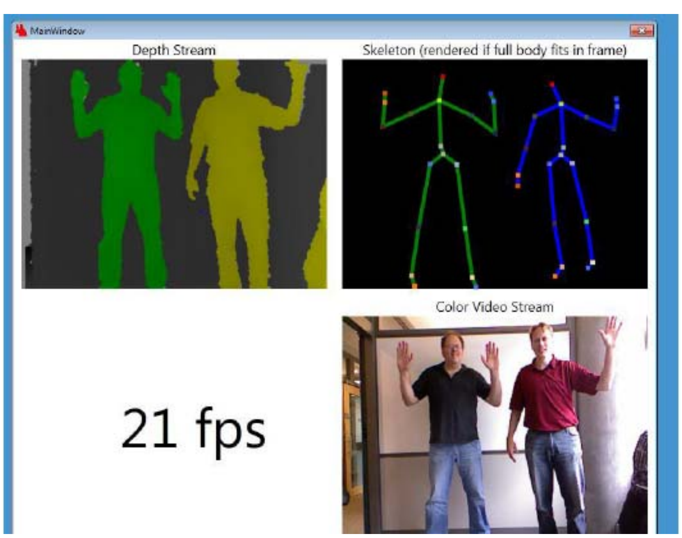

Figure 15 Microsoft's Kinect camera in use [18]

A benefit of this type of sensor is that there are commercial cameras that have software that can help with the programming. Microsoft's Kinect camera is one camera that is already used by many people to play video games and can be used in our implementation as well. Microsoft also has a Software Development Kit (SDK) that can be used along with their Kinect camera to allow for detection of objects as shown in Figure 15.

According to a study titled "Study on Stairs Detection using RGB-Depth Images", the depth sensor mounted on Xtion and Kinect irradiates infrared rays of special pattern to measure its warped pattern, but sunlight greatly interferes with the infrared rays of the depth sensor. Thus 3D photography is not possible without a projection pattern being reflected in the infrared camera well. Therefore, the system is limited to indoor use because it cannot acquire distance information under sunlight due to using infrared rays [19]. This means that these types of cameras or sensors might not be better than the IR sensor when used for outdoors, but they may be better for indoor uses because they could allow for greater distance tracking.

Depending on the type of sensor used to detect the objects the estimated amount of time could range from around 10 to 40 hours to implement. For example, taking into account the algorithms needed time to connect everything and programming, the ultrasonic sensors would be easier to implement when compared to the RGB-D camera. This is because the camera is a lot more complex and because no one in the group has worked with this type of technology before, meaning that there is a learning curve as well.

Just like the time ranges in hours needed to implement the design the amount of money needed also ranges. Assuming we already have a microcontroller, the amount of money needed to implement one of the sensors ranges from 15 to 150 dollars with the $150 option being a Microsoft Kinect camera or other commercially available product.

While testing this feature we will know that it is complete if we can detect an object that is three feet away from away from the wheelchair. In order to be detected the object would have to be 180 degrees from the front of the wheelchair. The object will be detected if it is no higher than 2 feet from the ground. If detected the user would be warned on the display to let them know that there is something in the way. This feature is intended to work in an indoor environment to detect common objects inside a house that would affect the mobility of the wheelchair, such as a table or chair.

Alex Verstivskiy will be implementing this feature, which includes programming the microcontroller to take inputs from sensors and testing to make sure that the correct data is taken. He will also make sure that the correct signals are provided to the user interface so that the user of the wheelchair will be notified of objects in the way of the wheelchair.

F. *Team Members*

Vadim Artishuk has great skill and abilities in identifying problems and debugging both hardware and software related problems. This will be very beneficial to the team because the project requires lots of adjustments and fixing as part of the process and that skill is very important in a team. His task is to create the graphical user interface and the programming skills he has fit well with it. The task is heavily based on programming.

Ryan Law excels in programming and debugging in various hardware and software programming languages such as Verilog, C, and Java. Also has experience programming various microcontrollers. These skills will help to interlink the various subsystems within this design to help them function smoothly with one another.

Alex Verstivskiy has the skills of programming in many languages as well as circuit building and testing. This will be helpful in our project when we are creating software, where coding and testing will be needed, as well as when we are connecting multiple device together and need to match the power requirements for each device.

Riley Schofield has experience in circuit building and troubleshooting as well as programming in several different languages. He has also had experience with many different electrical devices both in utilizing and understanding them. This will be helpful to know what the different capabilities are of the different electrical components and to know how it might be necessary to adjust a device so that it can work better for our purposes.

IV. FUNDING

Table VII
Cost of system [20]

Item	Cost
2 IR Sensors	$30
3 Servos	$33
Parallax Propeller Board	$50
Raspberry Pi	$35
Gyroscope	$25
4 Xbees	$130
Solenoid	$15
5 Relays	$10
2 Arduino Nano	$6
Vibrating Motor	$1.25
Wire	$10
Total	$345.25

One of our goals when designing our system was to keep the cost low. To ensure that we were not going over budget we tabulated the costs of all the components in our system as shown in the table above.

V. PROJECT MILESTONES

Our team devoted much time to this project, and there are many tasks that we are proud to have accomplished. Below, there is a list to display these accomplishments.

1. Problem statement written report finished
2. Design idea draft
3. Operate head tracking system
4. Work breakdown structure written report
5. Finished project timeline
6. Turn on and off lamp using microcontroller
7. Automate control of a lamp
8. Finish GUI and device communication

9. Use microcontroller to unlatch and open door to approximately 90 degrees
10. Measure when past 60 seconds and count to 15 seconds
11. Risk assessment written report
12. Vadim finishes with team leader duties
13. Ryan finishes with team leader duties
14. Alex finishes with team leader duties
15. Riley finishes with team leader duties
16. Navigate through GUI application
17. Risk assessment written report
18. GUI receives an input signal from the microcontroller
19. Establish communication with devices
20. Team Member Evaluations
21. Communicate simple wireless command using Xbee
22. Wirelessly execute wireless command using Xbee
23. Integrate commands to be used in GUI
24. Laboratory Prototype Demonstration
25. Revised problem statement written report
26. Device test plan written report
27. Market review and bulletin board review
28. Raspberry Pi receives and sends through serial ports
29. Midterm progress review
30. Deployable prototype technical review
31. Last weekly report
32. Final documentation report presentation
33. Deployable prototype presentation
34. Finish final presentation video
35. Practice market review presentation

VI. Project Work Breakdown Structure

A. *Graphical User Interface*

The graphical user interface needs to be simple and very easy to access. The goal of our design is making it easy to access the different features of the wheel chair without the assistance of another person. The different features of the wheelchair will be divided into 3 categories, accessing appliances, safety features, and an alert system in case of emergency. Accessing the appliances will have its own sub directory of different tasks that can be performed through this graphical user interface. This will include entertainment control, climate control and controlling doors. The emergency safety features will have options to send out help signals or make phone calls in case of a malfunction with the wheelchair. All the features of the project will be accessible through the user interface, so all the devices need to be able to communicate with the main control unit, which operates the user interface. The object detection system will send alerts to the main control unit, which will act accordingly. The emergency call system will send commands through the main control unit to be executed. The main control unit will operate with the user interface to enhance the user's experience. If the user is unable to access the interface, measures must be taken to ensure that the status of the user is transmitted to caregivers or nearby people. Navigating through the application and establishing communication between the main control unit and various microcontrollers make up a bulk of the tasks associated with the implementation of the graphical user interface feature. Navigation of the graphical user interface is the first subtask that is to be handled by Vadim Artishuk. After this, communication is also to be established between the main control unit and the microcontrollers by Vadim Artishuk.

1) *Operating the Head Tracking System*

The webcam needs to be installed for the head tracking system to be able to track head movement. The camera doesn't need to have a very high resolution because the software doesn't require high resolution to track head movement. The camera just needs to be compatible with the microcontroller.

The head tracking system needs to be able to traverse the operating system by clicking, moving and navigating the operating system. This means that it will be able to function alongside the user interface.

2) *Navigating Through the Application*

The different features of the wheelchair are to be organized, so that they are divided into separate categories and functions. Accessing the appliance section in the user interface gives the user the ability to choose from different tasks in that sub directory. The task here is to organize the categories and sub categories based on the likely hood of use. We also need to take into account that the safety features need to be very quickly accessed as well made very simple to navigate to within our design.

The main window of the application will be designed and programmed. The placement of the buttons relative to other buttons, specifically distance and size will be taken into account. The buttons need to be the typical size of a button, if the user will require a different size of the buttons so it is easier to see, that will be taken into account. An option to change the size of the buttons will be something implemented later.

The graphical user interface is to be programmed. This task requires the buttons only to be functional within the program. Accessing information outside the user application is not needed yet. The buttons in this task are only required to navigate through the program. When I am able to navigate through all the levels of the application, the task will be completed.

3) Communicating with the Various Microcontrollers

Communication is to be established with the other microcontrollers. We want the user to have the ability to open a door, this can be done with another microcontroller and communicating the response we want needs to be correctly interpreted by the receiving device. The safety features need to be able to send information and receive them from the main control unit.

Each separate device needs its own testing and debugging. So getting each device to communicate with the raspberry pi separately, is its own separate task. Each feature or task that that the user wants to use will be communicating with a different device. So opening the door will communicate to a different device to function so that is a separate task from turning on an appliance.

Buttons within our design will be programmed to interpret the communication between the GUI and the other devices. Once we can get the raspberry pi to communicate with the receiving device, we can then program the buttons in the application to control that device. The communication needs to be functional before we can get the inputs of the graphical user interface to establish that communication.

The application needs to be programmed to run on start up to give the user immediate access the physical and virtual user interfaces. The user needs to be able to access the microcontroller if it malfunctions and restarts, or when he runs it for the first time without the help of a caregiver. The user will not be able to use the device without the head tracking software running because the microcontroller will not be able to communicate with the person. This can be programmed into the microcontroller or through the program.

Tests will be performed to ensure the buttons can communicate with the other devices. This is the testing stage so everything should be functional and this will be the way we make sure that all of our features are working correctly.

The layout of the application will be altered to be more aesthetically pleasing. We will change the background of the application and the color and placement for the buttons and placement so that it will look better.

Multiple devices are to be controlled such that they are able to communicate to the main control unit in a way that allows specific functions of the design to operate in parallel. There will need to be a ticketing system or scheduling system in place for that. For example, the safety will be the first followed by what is necessary for the user.

The basic structure of how these components are to be broken down into basic tasks can be seen in the table below:

Table VIII
STRUCTURAL BREAKDOWN OF THE GRAPHICAL USER INTERFACE [21]

Feature	Subtask	Activity
Feature 1: Graphical User Interface		
	Subtask 1.1:	

	Operating the head tracking system (Estimated time to completion: 3 hours)	
		Activity 1.1.1: Install the Webcam and Camera Mouse head tracking software. (Estimated time to completion: 1 hour)
		Activity 1.1.2: Use the software and webcam to traverse the operating system. (Estimated time to completion: 2 hours)
	Subtask 1.2: Navigate through the application	
		Activity 1.2.1: Organize the different features of the wheelchair, dividing them up into separate categories and functions. (Estimated time to completion: 2 hours)
		Activity 1.2.2: Program and design the main window of the application. (Estimated time to completion: 2 hours)
		Activity 1.2.3: Program the graphical user interface. (Estimated time to completion: 30-50 hours)
	Subtask 1.3: Communicate between other microcontrollers	
		Activity 1.3.1: Establish a communication with the other devices. (Estimated time to completion: 15 hours)
		Activity 1.3.2: Each separate device needs its own testing and debugging. So getting each device to communicate with the raspberry pi separately, is its own separate task. (Estimated time to completion: 5-7 hours)
		Activity 1.3.3: Program buttons to interpret the communication between the GUI and the other devices. (Estimated time to completion: 10 hours)
		Activity 1.3.4: Program the application to run on start up. (Estimated time to completion: 20 hours)
		Activity 1.3.5: Test to make sure the buttons can communicate with the other devices. (Estimated time to completion: 50-70 hours, due to ongoing debugging)
		Activity 1.3.6: Change the layout of the application to be more aesthetically pleasing. (Estimated time to completion: 20 hours)
		Activity 1.3.7: Control multiple inputs to the main control unit, if two or more devices are trying to access the main control unit. (Estimated time to completion: 20-30 hours)

B. *Emergency Call System*

When it comes to an individual in a wheelchair who can't move any of their limbs, safety is of the utmost importance for their well-being. If we are to make this individual have more independence in their own home, then there also needs to be a failsafe mechanism in place that will automatically call for help if they end up needing it. For the emergency call system there will be two main categories. will be two main categories. Riley Schofield will be in charge of completing the Emergency Call System feature, and since it is connected to the Graphical User Interface he will have to work with Vadim Artishuk on what will be needed to communicate with the GUI.

1) *Rotational monitoring system for the wheelchair:*

The first will be the actual setup and programming of the device that is meant to monitor the user's wheelchair to see if they have had a catastrophic fall. The idea is that the device will measure the rotation of their chair from the horizontal plane and if the chair stays past 60° for 15 seconds then that will be interpreted by the microcontroller as having catastrophic fall. This will then proceed to the second step of making the emergency call.

The gyroscope sensor circuit schematics and programming will be researched so the sensor may be connected to and controlled by the microcontroller. Some devices require an electrical circuit as simple as ground, power, and a signal pin to read the data of the device. Others require can require more elaborate setups requiring resistors, transistors, diodes or other various circuit components in order to function properly. We must research the component number to find the data sheet to help to determine what setup is to be run between the gyroscope sensor and microcontroller. Once we have found the proper resources we can wire up the circuit and connect it to the microcontroller. As long as there is a significant amount of documentation for the gyroscope it should only take 3-4 hours at the most. The programming and wiring with troubleshooting for each could take anywhere from 5-12 hours.

When we have the circuit set up it is time to begin programming the microcontroller and testing the gyroscope. The main idea behind doing the initial programming is to become familiar with how the sensor works and to determine that the sensor actually works and isn't broken in some way. Programming and testing the gyroscope sensor for reading angles from the horizontal axis, we can start to understand what the different capabilities are of the sensor and if there are any changes that we might have to make to our code in order to make it work the way that we want it to. Once we have figured out exactly how the gyroscope works we can then program in our specific parameters we wish the sensor to react to, i.e. sending a signal for the caregiver to be contacted if the wheelchair stays past 60° for more than 15 seconds, that also reset when brought back into this threshold.

2) *Communication System Between the Microcontroller to the User Interface and Between the User Interface to the Call System:*

For making the call to their caregiver or an emergency service, there will need to be a communication system between the microcontroller reading the monitoring device and whatever device that will be making the call. This will most likely be the user interface that we set up for the individual to use to complete other tasks. When the command is received for the emergency call to be made a prompt will appear on the screen asking the user if they would still like to proceed with the call. The idea here is that if there is a false alarm then the user can cancel the call and the microcontroller will then go back to monitoring the rotation of the wheelchair. However, if a response is not received within 15 seconds of the command to the interface, then the system will assume that the user is somehow unable to access the interface and the call will automatically be sent to the caregiver.

The basic structure of how these components are to be broken down into basic tasks can be seen in the table below:

Table IX
STRUCTURAL BREAKDOWN OF THE EMERGENCY CALL SYSTEM [22]

Feature	Subtask	Activity
Feature 2: Emergency Call System		
	Subtask 2.1: Rotational monitoring system for the wheelchair (Estimated time to completion: 31-63 hours)	
		Activity 2.1.1: Research the gyroscope sensor circuit schematics and programming (Estimated time to completion: 3-4 hours)
		Activity 2.1.2: Wire the circuit to the microcontroller and attach gyroscope (Estimated time to completion: 2-6 hours)
		Activity 2.1.3: Program and test the gyroscope sensor for reading angles from the horizontal axis (Estimated time to completion: 2-5 hours)
		Activity 2.1.4: Establish a count system that will measure the amount of time the sensor exists 60° past the horizontal axis, (this count will reset if the wheelchair goes back within the 60° limit) (Estimated time to completion: 24-48 hours)
	Subtask 2.2: Communication system between the microcontroller to the user interface, and between the user interface to the call system (Estimated time to completion: 24-48 hours)	
		Activity 2.2.1: Have a system for the microcontroller to communicate to the user interface indicating that the wheelchair has fallen past 60° for more than 15 seconds, and then have a system to contact the caregiver once 15 seconds more have passed on the user interface without a response

		(Estimated time to completion: 24-48 hours)

C. Object Detection

Just like emergency call system feature the object detection feature will need to communicate with the user interface and display the needed messages. Alex Verstivskiy will be working on the object detection feature so he will need to work closely with Vadim Artishuk to allow the different microcontrollers to work together. In order to make the object detection feature work, we will need a microcontroller that will take the information from the sensors and process the information. The microcontroller will have to receive data from at least three different sensors for the field of view that we want to achieve.

1) Testing Sensor Capabilities:

The first task that will need to be done is to test the sensors to make sure how accurate they are when they read a certain distance. To test the sensor, we will use a tape measure to accurately measure the distance from the sensor to an object that we will place. This way we will know how accurate the readings from the sensor are and if we would need to change the programming accordingly. This test will need to be done with each sensor that would be used for this feature since there could be differences in all of the sensors that we use. This will take about an hour to two hours to collect all of this data for each individual sensor.

The next test we would need to do with each sensor is to test the scanning width of each sensor. To test this, we will need to two objects that would be the same distance from the sensor but they would be separated from each other. We will then move the objects closer together and scan with the sensor to see if it finds anything. If nothing is detected, we will move the objects closer together and rescan. We will continue to do this until we find that both of the objects are able to be detected separately. Then by measuring the distance between the two objects and the distance to the sensor from the two objects we will be able to make a triangle that would give us the angle in which a single sensor would operate and how many sensors we will need in order to fully cover our goal of detection in 180 degrees. This same test will also be done in a vertical direction to see if there is a range of operation in the y-axis. After this data is collected we will then know how far and how wide the sensors can scan so we can then calculate how many sensors that will be needed.

2) Connecting Sensors to a Microcontroller:

After testing the sensors and figuring out how many we will need we can then start mounting the sensors and writing the program that would take the data from the sensor and give it to the microcontroller. Initially we will work with one sensor and write the program to take the data. Once we can see that we can get the correct results with one sensor we will then connect them together so that the program will be taking in data from multiple sensors.

3) Connecting to User Interface Controller:

Once we see that the system works by getting the correct results from testing, we will then connect the microcontroller to the device that is controlling the user interface. This way the microcontroller that is used for the object detection feature would be able to communicate with the user interface and be able to notify the user of the wheelchair that something was detected. To allow the microcontroller that is receiving the data from the sensor to communicate with the user interface device we will use a serial bus so that we can directly connect the two devices.

4) Scaling Feature to an Actual Wheelchair:

Once the key tasks needed to complete a prototype for this feature are complete, we can then begin to refine the feature and scale up our design to work on an actual wheelchair. The first task that will be done is to calculate the number of sensors that would be needed to attain our goals set in the design document. We will do the same tests as stated before to calculate the lengths and widths the sensors can measure while mounted to the wheelchair and attach needed sensors accordingly.

5) Measuring the Usage of Power:

The next step would be to calculate and measure the power usage of the sensors and the microcontroller taking the data from the sensors. This is to see if we would be able to operate this feature for the amount of time that we specified in the design document. This means that if we are not able attain our goal for operation time we will need to change components that would require less power consumption or switch to a bigger battery that would allow us to get the amount of time we would like to operate for.

The basic structure of how these components are to be broken down into basic tasks can be seen in the table below:

Table X
STRUCTURAL BREAKDOWN OF OBJECT DETECTION [23]

Feature	Subtask	Activity
Feature 3: Object Detection		
	Subtask 3.1: Test sensor capability (Estimated time to completion: 13 hours)	
		Activity 3.1.1: Measure the usable distance in length the sensor can measure (Estimated time to completion: 6 hours)
		Activity 3.1.2: Measure degree that a single sensor can scan in x and y axis (Estimated time to completion: 5 hours)
		Activity 3.1.3: Calculate the number of sensors needed based upon measurements (Estimated time to completion: 2 hours)
	Subtask 3.2: Connect sensors to microcontroller (Estimated time to completion: 10 hours)	
		Activity 3.2.1: Position the sensors to be able to scan 180 degrees and measure 3 feet away from the sensor (Estimated time to completion: 3 hours)
		Activity 3.2.2: Program the microcontroller to read data from the sensors and make decision on when to communicate with user interface (Estimated time to completion: 7 hours)
	Subtask 3.3: Connect to user interface controller (Estimated time to completion: 10 hours)	
		Activity 3.3.1: Connect the serial bus of the object detection microcontroller to the device controlling the user interface (Estimated time to completion: 7 hours)
		Activity 3.3.2: Send signal to display message when object is

		detected (Estimated time to completion: 3 hours)
	Subtask 3.4: Scale feature to actual wheelchair (Estimated time to completion: 3-5 hours)	
		Activity 3.4.1: Calculate the number of sensors needed to scan 180 degrees and 3 feet ahead of wheelchair (Estimated time to completion: 3-5 hours)
	Subtask 3.5: Measure Usage (Estimated time to completion: 5-10 hours)	
		Activity 3.5.1: Calculate and measure the power usage of the object detection system. Make changes to system according to specifications (Estimated time to completion: 5-10 hours)

D. *Wireless Communication to Household Appliances*

The final feature within our design also aims to better enhance the freedom and independence of a quadriplegic person. As paraplegic individuals' motor skills are entirely different than the motor skills of people who are not disabled. Therefore, several factors must be considered when designing a system to automate the control of common household appliances such as fans and lights. These individuals are often limited to being stationed in their wheelchairs for most, if not the entirety, of the day. Their wheelchairs and accessories serve a purpose of transportation and also help them to accomplish various tasks throughout the day. Taking these factors into consideration, a person disabled with quadriplegia should also be able to control household appliances for the comfort of their wheelchair. As these wheelchairs are constantly moving around the home, it is important that this system is able to communicate wirelessly. Ryan Law will be working to implement this feature within our design.

1) Automate the control of common household appliances:

In order to create a system, we will need to first perform a series of subtasks to ensure that the final product is fully functional and performs all desired duties. First of all, two microcontrollers need to be able to communicate to one another and execute basic commands such as turning on and off lights, opening and closing doors, or adjusting the speed of a fan.

2) Wirelessly automate the control of common household appliances:

Perhaps the most tedious of all the activities regarding home automation will be transmitting and receiving the commands associated with wireless communication between the various microcontrollers. Ideally, we will be able to incorporate some existing wireless technology to simplify to implementation of this wireless within our design. This would also help with further integration between our design and other existing wireless technology. Although, as of now we will only focus on wirelessly communicating commands such as turning on and off lights, opening and closing doors, and increasing or decreasing the speed of a household fan.

3) Integrate Wireless Automation into Graphical User Interface:

Once a functional prototype is produced, Vadim Artishuk and Ryan Law are to work together towards making these automatic commands accessible through the graphical user interface, as the user will need to be able to experience all features through a single interface. As Vadim will have the basic architecture laid out for this graphical user interface, this task should be as simple as just mapping the commands to their appropriate location that has been programmed into our design. This will allow the user to allow the user to access these commands through the designs head tracking interface, all from the comfort of their wheelchair.

The basic structure of how these components are to be broken down into basic tasks can be seen in the table below:

Table XI
STRUCTURAL BREAKDOWN OF WIRELESS AUTOMATION OF HOUSEHOLD APPLIANCES [24]

Feature	Subtask	Activity
Feature 4: Automation of Household components via Wireless communication: (Estimated time to completion: 110-120 hours)		
	Subtask 4.1: Using a microcontroller, a common household lamp is to be turned on and off on command (Estimated time to completion: 10 hours)	
		Activity 4.1.1: Communication is to be programmed between the main control unit, microcontroller. A simple signal is to be sent from the main control unit to the microcontroller, which then will send a signal to the lamp (Estimated time to completion: 10 hours)
	Subtask 4.2: Control a fan that plugs into a household outlet with a microcontroller to simulate the control of a thermostat (Estimated time to completion: 15 hours)	
		Activity 4.2.1: Communication is to be programmed between the main control unit, microcontroller, and fan. Implement necessary circuitry for communication between input and output devices. (Estimated time to completion: 8 hours)
		Activity 4.2.2: Manipulate specific functions of the fun such as power and speed settings. (Estimated time to completion: 7 hours)
	Subtask 4.3: A closed, unlocked	

	bedroom door is to be unlocked using a microcontroller (Estimated time to completion: 15-20 hours)	
		Activity 4.3.1: A door is to be unlatched by a microcontroller so that it may be opened (Estimated time to completion: 10-15 hours)
		Activity 4.3.2: A door is to be rotated 90 degrees by a microcontroller by means of automation (Estimated time to completion: 5 hours)
	Subtask 4.4: A simple command is to be communicated between the main control unit and microcontroller (Estimated time to completion: 65 hours)	
		Activity 4.4.1: Preprogrammed colors will be displayed on RGB LED (Estimated time to completion: 65 hours)
	Subtask 4.5: Commands are to be communicated to control household components wirelessly (Estimated time to completion: 20-30 hours)	
		Activity 4.5.1: Wirelessly automate the control of household appliances: turn on and off a fan, open and close a door, adjust the various speeds of a household fan (Estimated time to completion: 20-30 hours)

VII. Risk Mitigation

A. *Background*

Risk can be measured by the impact of a certain component of a project and the probability that it may happen. So if the component has a very high probability of happening and a very high impact to the project, the risk can be seen as very high as well. The earlier stages of the project are the riskiest because of lots of unknown variables. For example, not having all the research done that is needed for a project will increase the probability that something can go wrong will increase the total risk. Uncertainties can be related to risk because the more uncertain the group is with their project, the higher the chances that their tasks can go wrong which effectively increases risk. So researching more on each task can decrease risk, and researching alternate routes or more detailed descriptions of the task at hand can decrease that risk. Our project risks can be divided into three different categories, financial risk, hardware/software risk, and time. In each of these sections we discuss the different risks that could appear and our plans on the methods we could use to avoid the risk or handle the situation if something appears. After listing out the risks with our project we then estimated the probability that each specific risk and the impact that it could have on the project. Using this analysis of the different risks we can better see the problems that could occur with the project and be better prepared to fix a problem if something were to occur.

Starting with our financial risks, we broke that down into three of our features which include the user interface, object detection and our safety feature. The user interface is the most important feature because it allows the user of the wheelchair to interact with the rest of the system which means that the user interface has the highest impact on our project completion if something were to go wrong. Our goal is to make a low cost solution but depending on the way we implement the user interface there could be a risk of us spending too much money to implement this feature. Our use of head tracking software relies on a windows operating system, and buying a device that could support this software could make us go over our budget. The object detection and safety features are also included in the financial risks. These features have a risk because if the initial sensors do not work for the application then costlier sensors will need to be bought, which also takes away from our budget and our goal of creating a low cost system.

Our next section of risk includes the hardware and software risks. This section is broken down into the user interface, wireless device control, object detection, and safety features. This section includes topics such as the probability that a component would break or fail, or if a programming or software error were to occur. As like the financial risk section, the highest risk is if something fails in the user interface feature. The user interface has to take multiple inputs from the other components and act on those inputs and display the correct screen to the user. This means that we have to make sure that the device controlling the interface can reliably communicate with the rest of the devices in the system, as well as be able to run for a long period of time. The wireless device control also poses a programming risk because of the language that is needed to program the wireless device, which means that communication between the user interface and wireless may have to be programmed in the same language. The object detection and safety features have risks of getting false data from the sensors, which in turn could display incorrect messages to the user. These risks are lower than the user interface feature but can still affect the system.

The last section we talk about are the time risks that impact this project. This section is split up into the same sections as the hardware and software risks stated above. These risks include the inaccuracies of the estimations made for the time needed to implement a feature, the time it takes waiting for parts to ship, and the risk of time being wasted by implementing a part of a feature and then having to redo it because the implementation doesn’t work.

B. *Financial Risk*

The goal of the project was to keep the project as inexpensive as possible while still maintaining the same quality product. Finance is a very important risk to consider because it has a profound effect on the quality of the project and the amount of time required as well. For example, there might be components that are very accurate and very easy to use but can be very expensive. Depending on how expensive they are effects how long it might take to finish the project which can, at the end, prove to be very time inefficient.

1) *User Interface: Head tracking*

A very high financial risk that may affect project completion is using the head tracking software. Although there are lots of free and open sourced head tracking software, most of the functional and simple to use ones are made for windows operating systems. The Raspberry Pi is a Linux based operating system and the head tracking software for Linux may prove to be difficult to implement. This poses a pretty big problem because gathering input from the user will require purchasing a Windows operated computer or tablet which adds to the already high expenses. There is a possibility to purchase an Arduino micro along with gyroscope sensors to translate head movement to the USB. This will be cheaper alternative to the windows computer but it also adds additional time needed to complete the project. To be able to keep the quality of the project the same it may seem that purchasing the windows computer will be the better route due to time constraints. Purchasing a windows computer can cost anywhere beginning from 150 or 200 dollars. That is very expensive and adding that to the project can be very problematic with the budget that we have. The computer will be the most expensive part as well as the most fragile. There will be additional costs that will be included to secure the computer to the wheelchair so it will be safe and stable. This is poses a very high risk because if our group can't afford the computer then the project will have to either be canceled or we will have to take a route in which time will become the highest expense. The risk can be estimated to be of a value of 400 where 10000 is the highest. Refer to figure 16 and table 12 for the risk chart that is specifically associated for the GUI features of the project.

Table XII
GUI associated Risks Table [25]

Project	Probability	Impact	Potential Risk
Main control unit communication with other devices	30	95	2850
Corrupt data for GUI app	10	95	950
Not meeting deadlines	10	75	750
User Interface head tracking	80	5	400
Multiple inputs	20	10	200
Main unit malfunction	5	15	75

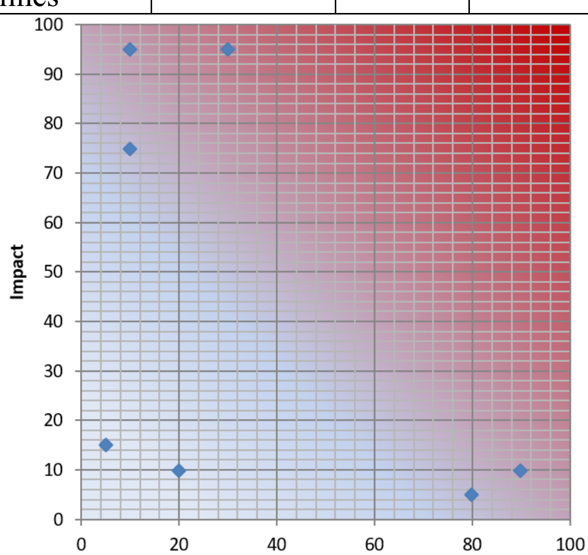

Figure 16 Risk chart for the GUI features [25]

There is a possibility that the Raspberry Pi may not be functional after the many trips it will take from being worked on at home to being worked on in college. There will be precautions that will be set in place to prevent that, such as placing the component in a safe box to protect from damage but it is important to understand that if the raspberry pi does break it will add to the already high costs. The Raspberry Pi breaking won't prevent the project from finishing because our group members have extra Raspberry Pi's that can be donated for the project. This part is considered to be very low risk and impact because it can be easily replaced and is relatively inexpensive. The risk can be estimated to be of a value of 75 where 10000 is the highest. Refer to figure 16 and table 12 for the risk chart.

2) Object Detection: Changing sensor

The object detection feature is another financial risk found in this project. Our initial approach to implementing this feature is through the use of IR sensors to detect if something is in the way of the wheelchair. Because of the way that these sensors work there is a possibility that we may run into problems such as noise or narrow field of view. The problem of noise can come into play when multiple sensors are sending out and IR signal and a sensor that isn't the one sending out the signal receives that signal. This could lead to false positive and false warnings to the user of the wheelchair. The other problem of field of view is that the sensors do not scan very wide meaning that we may need to use a lot of sensors to cover the full 180-degree field of view that we were planning on scanning. This means that we may need to move to another type of sensor for the object detection system. The other choice of sensor that we looked at was the RGB-D camera, or a Microsoft Kinect camera. This type of camera could fix some of the problem that we had with the IR sensors but it is very costly meaning it could not fit in our budget and our intended goal to make our system low-cost. To mitigate this risk, we will have to reduce the chances of noise coming from other sensors. This might be done by having a scanning order that would allow different sensors to scan at different times meaning that a sensor will not be picking up a signal that it did not send out. Another mitigation plan is to efficiently use the scanning range of the sensors and calculating the correct amount of sensors that we will need. This way we can still cover the full 180-degree field of view with the IR sensors and would not have to spend as much money when comparing this to the cost of the Microsoft Kinect Camera. The estimated probability of having to exchange the IR sensors for a different type of sensor is 20 percent with an impact of 40 meaning that it is a relatively low risk. After getting the estimates we tabulated the data and created a graph of it, as shown in the table and figure below.

Table XIII
Object detection sensor risk [26]

Project	Probability	Impact	Potential Risk
IR sensors don't work for design	20	40	800
Getting false values for IR sensor	20	40	800
Difficulty obtaining parts	10	10	100

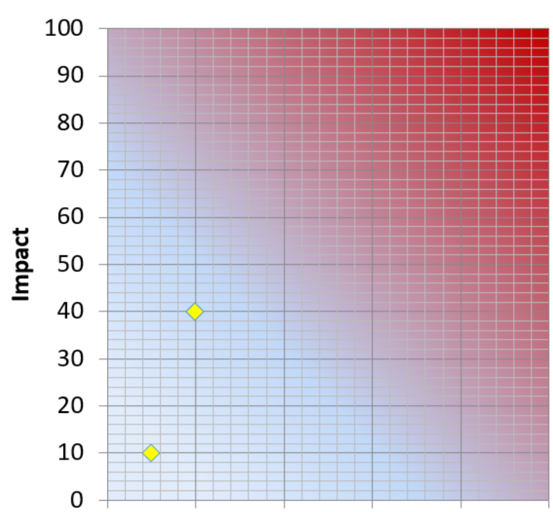

Figure 17 Probability vs impact for object detection [26]

3) Safety: Changing sensors

When it comes to the financial risk for the safety feature of this project there is only a minor amount of risk. With the most obvious way to monitor the angle of the wheelchair being the use of a gyroscope there are not very many parts necessary in this part. In research, there were several sensors that could cost up to $50, the one being used only cost about $8. The most expensive part of the feature was the actual microcontroller which cost around $20. The biggest risk would be if the sensor is not able to properly measure the desired values and a new sensor must be used, whether it is the same price or more expensive. The risk of this happening can be specified as a value of 50 out of 10000 [27].

Table XIV
Emergency Contact feature risks [27]

Project	Probability	Impact	Potential Risk
Gyroscope not measuring properly	30	10	300
Gyroscope component doesn't work for desired outcome	5	40	200
Cost of new gyroscope if first one doesn't work	5	10	50

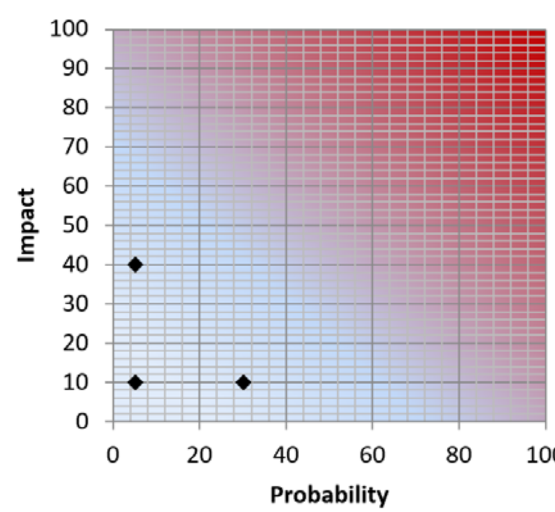

Figure 18 Emergency contact risk graph [27]

C. *Hardware/Software Risks*

1) *User Interface: Communicating with devices*

The highest risk of the whole project that has the ability to stop the project is the interfacing between the different microcontrollers and the main Raspberry Pi. Every feature of the project meets up to set up communication between the Raspberry Pi running the GUI. If some of the features like obstacle detection or safety or home automation are unable to communicate with the GUI the project will not be functional. The safety features of the project send alerts to the GUI if the user is indeed in a very unsafe situation, if the user does not reply within the time allotted it will automatically send a distress signal either through sound or a phone call. The automation of household appliances needs to communicate information from the raspberry pi to those components. The whole project is tied together by the GUI and the Raspberry Pi that is running that GUI The obstacle detection sends alerts to the user if there is an obstruction, so signals need to be sent to the Raspberry Pi. The project is supposed to allow the user control over the different features of the project, if there is no communication between the different sensors and microcontrollers the GUI will be pointless and not functional. There will be different routes that can be taken to establish the communication, originally communication between the different household appliances was going to be using Wi-Fi but if that proves to be too difficult, when considering time constraints, Bluetooth and hard wiring components will be considered. We are beginning the project and are somewhat not fully informed as to exactly what might be needed for the communication. That uncertainty increases the probability that the communication will not happen and the impact is already very high which makes this part a very high risk to project completion. To decrease the risk, we can find more information on communication to decrease the risk. So having alternative routes to communicating effectively decreases risk because that decreases uncertainty and uncertainty tends to increase risk. The risk can be estimated to be of a value of 2850 where 1000 is the highest. Refer to figure 16 and table 12 for the risk chart.

Multiple inputs from the safety features or obstacle detection into the Raspberry Pi may prove to be complicated. The reason that it will be important to consider controlling multiple inputs by prioritizing is because there is a possibility that some inputs may need to be dealt with earlier than others. Some inputs may need full control over others for example like the safety features because that is the highest priority in the project. Some sort of scheduling will need to be implemented. The user will not be aware of the different inputs because the processing happens so quickly so scheduling may not even need to be implemented. As far as the user is concerned the inputs may be dealt quick enough that it is unnoticeable. Priority will need to be implemented because safety will need to have full control over the GUI if the situation calls for it. There is a feature in our project that sends signals to the main controlling Raspberry Pi that there is an emergency situation that needs to be dealt with as soon as possible, if another task is currently occupying the Raspberry Pi, it will need to be interrupted because safety is always first. To help reduce the risk of this happening and not providing a safe system for users, some sort of parallelism may be implemented. This will help reduce the total impact it may have on the project. The risk can be estimated to be of a value of 200 where 1000 is the highest. Refer to figure 16 and table 12 for the risk chart.

If the Raspberry Pi begins to malfunction and the GUI and all the programs that will be used for the project become unusable the process to try and reclaim all the lost information is wasted time. To prevent this, lots of timely updates to the backup drive of the project needs to be done very frequently. There is always a possibility that something can become corrupted. If we consistently backup the programs and any changes to the GUI we effectively minimize the probability that a corrupt GUI will hinder our project progress. Since the probability decreases the risk that it poses for the project effectively decreases as well. The risk can be estimated to be of a value of 75 where 1000 is the highest. Refer to figure 16 and table 12 for the risk chart.

2) Controlling Wireless Devices: Unlatching the Door with Servo

Currently in our design, we plan to implement a servo in the mechanism that is to unlatch a door. The servo will be placed inside of the doorknob and will mimic the motion that occurs when the handle is turned, causing the latch to open. This design poses a couple of risks. First, there isn't much space available inside of the actual doorknob. The size of this servo must be small enough to fit within this area. If we are unable to fit the servo inside of the doorknob, we will need to unlatch the door externally. A mechanism could be implemented to unlatch the door externally, such as a pin that pushes the latch from inside of the strike plate. In addition to being small, this servo must have enough power to move the mechanism that unlatches the door. Tests should be done on this mechanism to determine the amount of force required to move the latch. The results of these tests will be used to select the servo that we are to implement within our design. Our team estimates that this risk will have a potential risk value of 500 out of a maximum value of 10,000, as shown in Table XV and Figure 19 below.

Table XV
Wireless Device control risks [28]

Project	Probability	Impact	Potential Risk
Wireless communication of commands	20	90	1800
Manipulating internal circuitry of fan	20	30	600
Unlatching door using a servo	50	10	500
Compatibility of programming languages	10	50	500

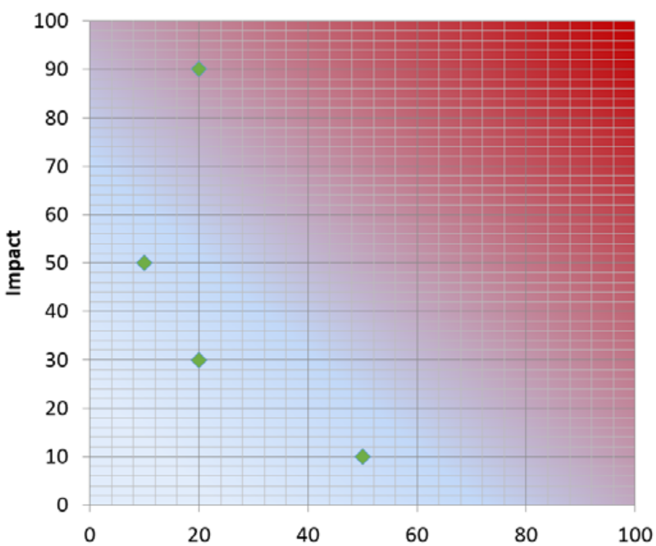

Figure 19 Wireless Device Control risks graph [28]

3) Controlling Wireless Devices: Merging programming languages

Another component that could potentially prevent the functionality of our design will be when we integrate the various programs into a single functioning program. It's unlikely that every little bit of code will be able to be programmed the same exact programming language. Therefore, our team members must effectively communicate how they are able to implement software so we are able to more easily coordinate how the programs are to be merged. Since the graphical user interface will be where these programs are accessed, the language that it is written in will likely be the language in which all other programs must be compatible. Establishing this language early will allow more time for making any changes that are necessary to ensure compatibility between the various programs. Our team estimates that this risk will have a potential risk value of 500 out of a maximum value of 10,000, as shown Table XV and Figure 19 above.

4) Safety: Gyroscope angle measuring

The hardware risk of the Emergency Contact feature is mainly if the gyroscope breaks. With electrical components, there is always some risk that the hardware could break or be short circuited if proper care is not taken with the component. The other possibility is if the gyroscope does not meet the requirements for the feature. This could be because it isn't accurate enough in the measurements that it outputs. There is also a small possibility that an alternative method must be used in order to achieve the goals of this feature.

In researching the gyroscope being used for this feature it has also been discovered that there is a possibility that the gyroscope alone will not accurately measure the angle of the wheelchair after it has been rotated around several times. In a

YouTube video of a woman demonstrating the MPU 6050, she brought to light the fact that after rotating the gyroscope and then setting it back on the flat surface it began to show some signs of offset from where it should be when the gyroscope was sitting on the flat surface at an angle of 0°. After doing this several times it was shown that the gyroscope reading appeared to be up to almost 90° when it was sitting on the flat surface. To counter this the woman used the built-in accelerometer to supplement the gyroscope in giving the actual rotational angle which the device was seeing. The probability of this happening is 300 out of 10000 as shown in Table XIV.

5) *Object Detection: Inaccurate readings*

When it comes to programming the IR sensor there is a risk of having the sensors reading values that could be inaccurate. This could then cause the user interface to display that an object is in front of the wheelchair even though there is nothing in the way or the object is farther away than the sensor thinks. To mitigate this, we can program the sensors in a way which makes them double check to make sure that there is something there. Another approach is to use another sensor to make sure that there is something there. This way a warning would not appear until at least two sensor detect something. The estimated probability for this is 20 percent with a risk of 40 percent as seen in Table XIII and Figure 17.

6) *Power Consumption*

After testing our lab prototype to make sure our system can work as expected we will then be setting up our system on actual wheelchair. This process will take an extra step in order to make sure that we do not alter the original functionality of the wheelchair, or if we do alter it, we do so in a manner that won't affect it too much. This means that there is a risk of using too much power for our system and resulting in a shorter range or operating time of the wheelchair. To handle this, we will have to calculate and measure the power that the individual components of the system and the total power that the system will use, and make sure that the consumption is in a reasonable range. Another mitigation method is to attach another battery to the wheelchair that would power our system and not be connected to the wheelchair power supply at all. This way we could calculate the average power consumption of our system and the amount of time that our system will be able to run and match it to the operating time of the wheelchair.

D. *Time Risks*

Deadlines pay a huge role in completing our project. Each group member has a good idea of what needs to be completed by what time because we created a timeline. Risks with deadlines can be very hard to gauge because trying to guess how much time a certain task may take can never be certain. To help mitigate those risks understanding exactly how long each task may take by getting outside help will help with prioritizing tasks that can reduce that risk. Knowing exactly how long it will take to complete a task reduces uncertainty which can reduce risk. To further mitigate these risks we can continually update our timeline to see how much time we have left versus how much we still need to do. This will ensure that we do not push back tasks too far and so that we could see the deadlines coming earlier.

1) *User Interface: Communicating with devices*

There has been time allotted more for some tasks then others. The problem with that is it may be a very inefficient use of time because a task that may have been perceived to take a shorter amount of time may end up taking the longest. That is part of the uncertainty of the GUI application. Creating the GUI shouldn't take a long time but communicating with the other devices may be the most difficult part. This can increase the risk of not finishing the project or task on time. To help mitigate that risk we can consult with an outside source that has had experience with a similar situation and that can reduce the risk. Seeking the outside help to help gauge the amount of time needed for tasks can help with a better more efficient use of time which effectively reduces risk.

2) *Object Detection: Parts*

On top of the time it takes to complete the features we have to include the time it takes for us to order the parts we need and how much time it will take for us to receive those parts. This could also include the time we would need to manufacture parts that we would need such as 3D printing custom parts for our design. To mitigate this risk, we ordered many of the parts that we would need early on in the semester so that the shipping time would not affect us too much. As for the custom parts we will need to consult with someone who knows more about the 3D building process so that we do not take too much time to learn how to create our parts. The probability and impact of this occurring is 10 percent as shown in Table XIII and Figure 17 above.

3) Controlling Wireless Devices: Working on Fans Circuitry

One of our tasks that could potentially waste time is when Ryan takes apart the fan to manipulate its inner circuitry. It is quite possible that the fan may be broken during this activity, requiring us to purchase a new fan. This new fan should be the same model as the fan that was broken. If for any reason we are unable to acquire the same model, we will be forced to use a model that has most likely been designed entirely different, requiring us to redo much of the work that was previously done on the initial fan. As time is critical to this project, we must ensure the model that we decide to manipulate has a reliable supply. If we are indeed forced to purchase a different model, it is ideal that we select a similar model with similar specifications, and also the same manufacturer to maximize the amount of overlapping work between to different models. Our team estimates that this risk will have a potential risk value of 600 out of a maximum value of 10,000, as shown Table XV and Figure 19.

4) Controlling Wireless Devices: Wireless Communication

Since our design aims to enhance the user's quality of life, we really do not want to just tie them down with wires. This is why we plan to implement a wireless system for communicating commands to the devices within their home. Unfortunately, there is a relatively steep learning curve for understanding wireless communication systems. Our team estimates that it will take about eighty-five to ninety-five hours to implement the required wireless commands specified within our design idea. As time passes, the overall impact of this specific risk increases. If our team decides to change any of this technology, we need to do it earlier rather than later, since we don't want these issues to hinder our progress. To reduce this risk, we will implement existing wireless technology with plentiful resources, such as the Xbee. We will also work to understand this wireless component as early as possible within the timeline of our design. Our team estimates that this risk will have a potential risk value of 1800 out of a maximum value of 10,000, as shown Table XV and Figure 19.

5) Safety: Wasted time

The biggest time risk for the Emergency Contact feature is in the possibility of wasted time on whatever device is being used to measure the angle of the wheelchair. If it turns out that using this gyroscope to measure the wheelchair will not work, then whatever time was spent in working to try to make it do what was necessary will have been wasted. A new part will then need to be researched and ordered, which will also take time and then the new part will need to be tested to see if it can perform the necessary function. The potential risk of this happening is 200 out of 10000 based on the risk assessment chart that was created shown in Table XIV and Figure 18.

VIII. DESIGN DOCUMENTATION

As specified in the problem statement, the goal of our design is to help quadriplegics gain a sense of independence using our system. To provide the functionality that gives a user this independence, we included four different features that work together to achieve this goal. These features include a user interface, home automation, object detection, and an emergency contact system.

The most important feature in our design is the user interface and the way the user can interact with our system. To allow the interaction, we used an open source software that was specifically built for the audience we are trying to target. The camera mouse software uses a webcam and to translate movement it sees into mouse cursor movement. We utilized this to allow the user of our system to control a computer mouse and select the various functions they would like to do on our user interface. The user interface displays different menus, information, and different functionalities our system can carry out. Selecting and clicking a button on the user interface the user will automatically be able to perform an action. This feature is exclusively run on a raspberry pi with the actual displaying of the interface being done on a computer screen in front of the user.

To be able to perform actions for the user we created a home automation feature that communicates with the user interface. With the home automation feature the user is able to open and close a door, turn a light on and off, and control a fan. When a user clicks a button on the user interface the corresponding signal will be wirelessly sent so that the action will be performed. This way the user is able to move around in their chair as they normally would, and still be able to perform the various tasks. This feature was implemented using an xbee mounted on the parallax propeller board that is located on the wheelchair, and another xbee that acts as the wireless receiver to then send signals to a relay and control the various device mentioned above.

With this independence, the user may be exposed to risks and dangers that might not occur under the watch of a caretaker or another person. For this reason, we decided that some safety features would need to be included to maintain the safety of the individual while they are not under supervision. A key component of the safety features is the emergency contact system. This feature allows the user to contact the caregiver at any time they feel necessary. Using the user interface, they will be able to navigate through the menu and select the button to contact their caregiver. Another aspect to this feature is the continuous gathering of wheelchair angle measurements. This means the system continuously checks to see if the chair has tipped over, and in the event the chair is tipped over, an automatic distress call will be made to notify the caretaker that the chair is tipped over. This feature utilizes the xbee and the gyroscope that is on the parallax board in order to measure the angle and wirelessly send the signal in the case of an emergency. This allows the user to move around without the need of constant attention from the caretaker. This gives the user the independence they desire and peace of mind for the caretaker to know that they can be contacted if something were to occur.

The other safety feature we included is the object detection. Since the user could be moving around without the supervision of a caretaker they may need extra help when they are navigating their surrounds. They may not be able to see things that are in front of them because of their limited range of motion, so the object detection feature would help with that aspect. Continuous scanning in front of the chair is done to check the surroundings and in the event detection the direction of the object is displayed on the user interface. This way the user is able to see where something could get in the way even though they did not actually see the object with their own eyes.

IX. Deployable Prototype Status

At the end of the spring semester we were able to achieve our goal as a deployable prototype. After implementing our initial design, we then tested the design to verify the design accurately addresses our problem statement. We tested each of the features in our system as well as the entire system after all the features were integrated together.

A. *User Interface*

Testing of the user interface was done to assure the interaction between our system and the user would go smoothly. Because the user would be using a webcam to control our system a test of the ideal distance between the users head and the location of the webcam was done. This test allowed us to see where we would need to place the webcam to get the best translation of head movement to cursor movement. Results from this test showed us that the best performance was seen when the distance between the webcam and the user’s head was 2 feet. We also tested the resolution of our display to see how the quality of the graphics we were using affected the system. For this test, we saw that when we would try to use a resolution that was too high the speed of the user interface was slowed down because of the extra rendering that was needed to be done. The final test for this feature was the aesthetics of the user interface. Initially the interface was plain and did not look appealing to the user. After this test we changed the color scheme to be more attractive to the user. After these tests and the refinements, we made to this feature we could fully meet the feature set for this feature of our system.

B. *Home Automation*

In addition to the user interface, tests were conducted on the home automation feature of our system. These tests were necessary to make sure that the actions the user would like to perform would be completed with the correct results, and could be done reliably. Since the home automation feature used the xbee to send the wireless signal to perform specific functions, the tests focused mainly on xbee. A range test and a reliability test were performed to find the reliability of the feature. After completing these tests, we found that the xbees we were using were able to send 97 out of 100 commands we were sending, which ultimately allowed the completion of the feature set for this feature and allowed the user to open and close a door, turn on and off a light, and control a fan.

C. *Object detection*

The third feature we tested in our system was the object detection. This feature is used to tell the user when an object is in the way and the direction of object detected. Tests for this feature include an accuracy test of the sensors used. Completion of the tests for this feature resulted in better sensor readings. Another test for this feature included a power test. This test was done because this feature is going to be on for the entire time the user is in the wheelchair and could possibly use too much power. This test showed us that we would be able to use our system for a reasonable operating time for a typical use. As the previous features, the object detection feature was able to fully meet its feature set and scan in 180 degree and three feet away from the chair.

D. *Emergency contact system*

The emergency contact system is included in the system to make sure the user is safe in the time that they are away from a caretaker or cannot be directly seen by the caretaker. This feature included a gyroscope to measure if the chair has tipped over, and in the case of this event, the caretaker would be notified. Testing of the gyroscope accuracy was done to validate that when the angle was measured, the wheelchair was indeed in the same angle and not in an upright position. This final feature of our system also met the goals we outlined in the design documentation to be able to measure the correct angle and notify someone in the case of the wheelchair tipping over.

X. Deployable Prototype Marketability Forcast

Based on our research there are many different customizing places out there which design custom add-ons for wheelchairs according to the preferences of the user. Since one of our goals is make a more economical product which still uses reliable parts we need to make sure that the things which are not directly mounted on the chair can be reasonably installed without having to take too much effort. The biggest part of that would be the different home automation parts. Since a light is a simple on off switch it should be something as easy as plugging in the XBEE relay system to a lamp the user has within their home and just making sure that it is connected to the XBEE system on the Pi. If the user wanted to have a light switch for a room controlled then that would require a customized system to be set up by us coming out and installing it. The same would have to be said for setting up the door. According to Ryan, it only took an hour to physically install the motors for the latch and movement of the door, which would again be installed by our company. With the fan we would actually have to sell them the fan with the relay system, otherwise if we tried to install it custom on a fan they already had then there would be a possibility that we could break it. The other more likely possibility would be if they wanted to control the thermostat for their home. Since most thermostat systems are already controlled be a type of relay all we would have to do is install our relay system so that the user could interact with it from the GUI. The biggest change from our current setup would be making it so that the user has two buttons to increase/decrease the temperature instead of the 1, 2, 3 and on/off for the fan speeds.

The other portion of the project which would need significant changes to make it marketable is the object detection system. Currently two sensors with servo motors are mounted on the chair in such a way that it can scan 180° in front of the chair. The mount system is specific to the chair and probably wouldn't be able to be used on many if any other wheelchairs. This would obviously need to be changed to something that was slightly more universal to make it possible to mount on other wheelchairs without having to do much if any significant customization. The other thing which is currently being done is to design a custom containment system for the servos and sensors just so that they are somewhat protected from being kicked and easily knocked off.

There are a few small things that would also need to be done to make sure this is a marketable product. With all the different components we use, we do have quite a few wires connected to multiple different places. To make this safe we are making sure to cover up the wires and bundle them together so that there will only be a few sets of wires going to the different components to keep it from looking like a mess of wires. Another thing which we are doing is creating a custom containment system for the microcontrollers so that they can be easily placed and protected on the chair. This containment system will need to be something that can be flexible in where it can be mounted on the chair since not all chairs will have the same setup in where all the parts are.

XI. CONSLUSION

It's important that the user of this project feels that this product is easy to use. With their disability, there isn't a lot of choices out there to be able to do some regular daily tasks, unless some people have the luxury of choosing a miracle. This project needs to be as simple as possible for them to use, the GUI will play a very large role in that simplicity. The GUI will be organized based on safety features first because the safety of the user is always the most important. After safety is addressed, the GUI will be prioritized so that the user can choose a feature more quickly based on how often the user may decide to use that feature. An example of this would be giving the user easy access to controlling the lights within their home. Many people turn on or off the lights whenever they enter or exit a room, especially when it is dark. Therefore, it would make sense to prioritize the access that the user has to this feature. We also want to make the emergency call system a priority as safety is crucial to the success of this design. The user should never have to navigate complicated menus if they are in a dangerous situation.

Once this graphical user interface can communicate with the other features of this project, the response time of the commands that were sent will be taken into account. This amount of time most likely will never be as fast we would like it to be, but with the resources we have, we expect it to be enough that the user does not experience a noticeable amount of delay when accessing the features implemented within this design. There needs to be a quick response in case a safety concern will present itself so a swift communication between the user interface and the safety feature is very important.

When it comes to safety one can never be too careful. For a quadriplegic, this is of the utmost importance. Since they are unable to move themselves around there usually are safety straps on their wheelchairs. Their caregivers secure them into the wheelchairs, so they don't accidently slide out onto the floor or get flung forward when their chair comes to a sudden stop. Although these straps help keep the user within their chair, they don't really help if the chair were to tip over. Granted, a caregiver usually accompanies them wherever they go so that this can be prevented, and there are also wide anti-tip wheels that many wheelchairs come with. If we are to design a system allowing quadriplegic patients the ability to be more independent, at least in-home, a safety system is a necessity. While there may not be many places in the home where a chair can be tipped from being on an uneven surface, it is still a possibility.

People often take for granted the freedom and independence associated with going about their everyday lives and routines. Individuals with quadriplegia often rely on the assistance of other people such as caregivers to perform even the most basic tasks within their day to day lives. The wireless automation component of this design aims to give these people that freedom and independence. Giving these people the ability to perform these tasks will be beneficial in that they will be less reliant on other individuals. This will inevitably benefit the quality of life for these disabled individuals.

On top of the wireless automation and sensing if a wheelchair tips, adding a system that could tell the user that there is something in the way of the wheelchair will allow the user to move around their surroundings safer. The user will be able to have their independence and move around with peace of mind that they will not hit anything and potentially cause injury to themselves.

It is important that the user experiences these features as a single unit, this is why we made everything accessible from within a single graphical user interface. These features were implemented within an actual wheelchair system so that various features could be tested to ensure that the user experiences an optimal version of our design.

As we worked on the project we ran into a few unexpected problems that caused delays in the completion of the project, but ultimately there was nothing that we couldn't find a work around for. To minimize the chances of these problems appearing, we created a list of the different types of problems, or risks, that could occur with the different components of our system and the system as a whole. On top of this we also came up with ideas of solutions to these problems so that if the event where to occur, we would be ready and could still complete the project on time.

Since the user will be interacting with our system through the user interface, this was the highest risk to project completion if there was a problem. This feature included risks in each of the following areas: financial matters, hardware, software, and time. There were problems which occurred in difficulty communicating with the other devices in our system, however we were eventually able to communicate with reasonable accuracy between the different components. To mitigate the risk of going out of budget, we looked at the different options to implement the feature, and the amount of time and difficulty that each solution had. As for the communication to different devices and the time that it might take to implement, we planned on starting this task early on so that we would have time to address any problems that showed up.

After the user interface, the controlling of wireless devices has the next highest risk in terms of impact. This feature allows the user to control devices such as the lights in their home, or opening and closing a door all while in their chair. Just like the user interface this feature required a large amount of time to set up and get working, so just like the user interface we planned on starting early on this part so that any problems could be handled early in the process. Because the user interface and the wireless device control require a lot of time and had the highest risk to project completion the part where these two features meet was a critical path for our project.

The object detection feature and the safety system were the lowest risks to our project completion. Because these two features required the use of sensor data they both share a financial risk of having to get better sensors if the initial testing found the sensors to be inadequate. This meant that we would have to spend more money to purchase more expensive sensors on top of the money that was already spent to buy the original sensors. To mitigate these risks, we researched the sensors before buying them so that we would have a good idea about the capabilities of the sensors. On top of the research we also decided to do initial testing on the sensors after buying just one of each type of sensor that we would need. This way if we found that one of them was not able to perform the way we thought, we would know early on before we bought a large amount of them.

After finding the different types of risks each component had we estimated the probabilities that the risk would happen and the impact that it would have on the project. Multiplying those two numbers we could get a number for the potential risk, and also better see the critical path of the project. The results of these numbers were put into a graph for individual features in the different section, as well as the entire project in the appendix.

References

[1] Kraus, Lewis. (2015). “2015 Disability Statistics Annual Report.” Durham, NH: University of New Hampshire.

[2] “National Spinal Cord Injury Association Resource Center.” University of Alabama. Web. 15 Sept. 2016

[3] “NINDS Cerebral Palsy Information Page.” National Institute of Health. Web. 15 Sept. 2016

[4] Herbert A. Everest, “Folding Propulsion Wheel Chair” U.S. Patent 2181420. Filed October 8, 1937.

[5] USA, Permobil. "Power & Electric Wheelchairs | Mobility | Permobil USA."*Power & Electric Wheelchairs | Mobility | Permobil USA*. N.p., n.d. Web. 17 Sept. 2016.

[6] Rascanu and R. Solea, "Electric wheelchair control for people with locomotor disabilities using eye movements," *System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on*, Sinaia, 2011, pp. 1-5.

[7] Parminder, Raina, Maureen O'Donnell, and Stephen Walter D. "The Health and Well-Being of Caregivers of Children With Cerebral Palsy." Table of Contents. American Academy of Pediatrics, 1 June 2005. Web. 12 Sept. 2016.

[8] Byrne, M. B., Hurley, D. A., Daly, L. and Cunningham, C. G. (2010), Health status of caregivers of children with cerebral palsy. Child: Care, Health and Development, 36: 696–702. doi:10.1111/j.1365-2214.2009.01047.x

[9] Sardana, Ramita, and K.B. Sumalatha. "Family Life of Caregivers: A Descriptive Study of Disruption of Family Activities, Leisures and Interaction of Caregivers of Children with Cerebral Palsy." Family Life of Caregivers: A Descriptive Study of Disruption of Family Activities, Leisures and Interaction of Caregivers of Children with Cerebral Palsy (2016): 1-8. Print.

[10]"Raspberry Pi," in Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/wp-content/uploads/2015/01/Pi2ModB1GB_-comp.jpeg. Accessed: Oct. 4, 2016.

[11] Jamil, M. Mahadi Abdul, and M. Shukri Ahmad. "A Pilot Study: Development of Home Automation System via Raspberry Pi." IEEE Xplore, 30 Mar. 2015. Web. 27 Sept. 2016.

[12] Rudolf, Simon, Adam Sporka J., and Ondrej Pol´acek. "Constructing On-screen Keyboards in Word Processing Environment for People with Motor Difficulties." IEEE, 5 Nov. 2014. Web. 22 Sept. 2016.

[13] Raspberry_Pi. "Raspberry Pi - Teach, Learn, and Make with Raspberry Pi."Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/ Accessed: 24 Sept. 2016.

[14] Arduino. Arduino USA. [Online]. Available: https://www.arduino.cc/ Accessed: 23 Sept. 2016.

[15] Goodrich, Ryan. 1 Oct. 2013. “Accelerometers: What They Are & How They Work.” Web.

[16]"Accelerometer MMA7361. Where to start?", Arduino forum [Online]. http://forum.arduino.cc/index.php?topic=66520.0.

[17] learn.sparkfun.com. “Gyroscope: What is a Gyroscope.” Web.

[18] Litomisky, Krystof. "Consumer RGB-D Cameras and Their Application." Consumer RGB-D Cameras and Their (n.d.): n. pag. Web

[19] Murakami, Soichiro, Manabu Shimakawa, Kimivasu Kivota, and Takashi Kato. "Study on Stairs Detection Using RGB-depth Images." 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS) (2014): n. page. Web.

[20] Artishuk,Vadim Law, Ryan, Schofield, Riley, Verstivskiy, Alex. Sacramento State University. April 2017.

[21] Artishuk, Vadim. Student of California State University: Sacramento 15 October 2016. Structural Breakdown of Graphical User Interface.

[22] Schofield, Riley. Student of California State University: Sacramento (10/15/2016). Structural Breakdown of Emergency Call System.

[23] Verstivsky, Alex. Student of California State University: Sacramento (10/15/2016). Structural Breakdown of Object Detection.
[24] Law, Ryan. Student of California State University: Sacramento (10/15/2016). Structural Breakdown of Wireless Automation of Household Appliances
[25] Artishuk, Vadim. Student of California State University: Sacramento (10/24/2016). Probability and Impact of Graphical User Interface Risks.
[26] Verstivskiy, Alex. Student of California State University: Sacramento (10/24/2016). Probability and Impact of Object Detection Risks.
[27] Schofield, Riley. Student of California State University: Sacramento (10/24/2016). Probability and Impact of Emergency Call System Risks.
[28] Law, Ryan. Student of California State University: Sacramento (10/24/2016). Probability and Impact of Wireless Automation of Household Appliances Risks.
[29] "Windows 10 IoT Core." Windows 10 IoT Core | Official Site | Windows IoT. N.p., n.d. Web. 29 Apr. 2017.

Glossary

Ambulatory Disability – An ambulatory disability is permanent physical disability in which a person is unable to walk for transportation, often needing assistance of a wheelchair.
Arduino – A microcontroller that can be used to control various electrical components.
C1-C8 – Nerves found in the cervical vertebrae that help classify the extent to which a person has injured them.
Camera Mouse- Head tracking software that translates head movement to cursor control on a screen.
Cerebral Palsy (CP) – A condition in which the brain is damaged, causing losses or impairment of motor functions.
Extremity – Refers to the arms, legs, hands, feet, and even head of an individual.
GUI- Stands for graphical user interface, it is used as a means for a person to communicate with a computer program through a graphic.
Gyroscope – An electrical device that measures the rotation of an object that it is attached to.
IR Sensor- Sensor that uses infrared
Quadriplegia – Refers to the disorder in which an individual is paralyzed in all four limbs.
Raspberry Pi- A small sized computer that can be considered a microcontroller.
Spinal Cord Injury (SCI) – Damage to the spinal cord that causes change in its function. This may be temporary or permanent.

Appendix A. User Manual

I. Introduction

This manual is to provide the information needed to operate the Interactive Mobility system. This includes a list of parts that are included and need to be checked off, as well as some system requirements procedures and applications. The system will already be put together when the package is received but will require some further set up to tailor to the user's preferences as well as connecting to the wifi. This system does not come with a mount to place on the wheelchair. The mount can be purchased based on whether or not you currently own a tablet or a computer. The ideal distance to place the tablet and or web cam is about 4ft away from the user's face or the extremity the user will use to control the mouse on the screen. The camera mouse software responds the best at that distance, as well as that distance being good for the user to see and access the system features.

II. Package Checklist

- 1 USB A to Mini B
- 1 USB A to Micro B
- 15 Jumper wires
- 3 pin to USB cable to power servo
- DC cord to power servo
- 20000maH battery
- USB splitter
- 2 XBEE modules
- Angle safety sensor
- 2 microcontrollers
- Pager
- Your own Tablet or Computer
- Your own Ethernet cable

1) Software Requirements

- Operating system: Windows 10 (needed to run the Windows IOT Dashboard, Windows IOT Remote Client)
- Your computer also needs Microsoft .NET Framework 4.0 or higher

2) Additional Applications needed to install

- Camera Mouse software (free) http://www.cameramouse.org/downloads.html
- Windows IOT Remote Client (free) https://www.microsoft.com/en-us/store/p/windows-iot-remote-client/9nblggh5mnxz
- Windows IOT Dashboard(free) https://developer.microsoft.com/en-us/windows/iot/docs/iotdashboard

3) Hardware Requirements

- Tablet with a webcam(if no webcam then you can buy one separate). Able to run Windows operating system.
- Computer with a webcam(Also webcam can be bought separately). Able to run Windows operating system.

III. Operation

This system relies on three different applications to be able to run and use the system's features. The Camera Mouse software is free and developed under the contract of Boston College. This software is used to track the head movement of the user and translate that into the mouse movement of the screen. This will be necessary for the user to be able to communicate to the system. The windows IOT Dashboard will be necessary for finding the IP address of the system that allow for the control of the system. Through this app you can find your network and connect to it. Once connected you can remote into the system through the Windows IOT Remote Client which allows for the User Interface to be displayed and communicated to.

IV. Getting Started

Download the Camera Mouse software, detailed and official instructions on how to use the software can be found on the Camera Mouse website. Next will be a guide on how to install the other applications and how to use them.

Procedure:

1. Initially download the Camera Mouse Software from the website. http://www.cameramouse.org/downloads.html
2. For an introduction to setting up the Camera Mouse for the specific user, modifying sensitivity, running on startup, modifying default settings, excluding parts of the screen etc. can be found on the Camera Mouse website http://www.cameramouse.org/
3. Download the Windows 10 IOT Dashboard application on your tablet or computer. https://developer.microsoft.com/en-us/windows/iot/docs/iotdashboard
4. Download the Windows IOT Remote Client. https://www.microsoft.com/en-us/store/p/windows-iot-remote-client/9nblggh5mnxz
5. Power on the system by plugging in the cable coming out from the raspberry pi to the iSmart USB port located on the battery. Connect the USB splitter to the green port located on the battery. There should be 2 USB cords connected to the USB splitter.

1) Connecting to your Home WIFI

6. Connect the Ethernet cable to the Ethernet port locacted inside the box. Once connected open the Windows IOT Dashboard application, click the My Devices tab on the left hand side and wait for the System to be recognized. It will auto populate on the screen. Click on the device that was found should say under Name "Interactive Mobility" or "minwinpc" as seen on figure A-1.

Figure A - 6: Windows IOT Dashboard [29]

Click on it and it will Open another window. Click the blue letters to which will open up system settings in your browser a way to set up your Wifi. Click on the networking tab as seen on figure A-1. Locate your Wifi and login based on the information given to you by your service provider.

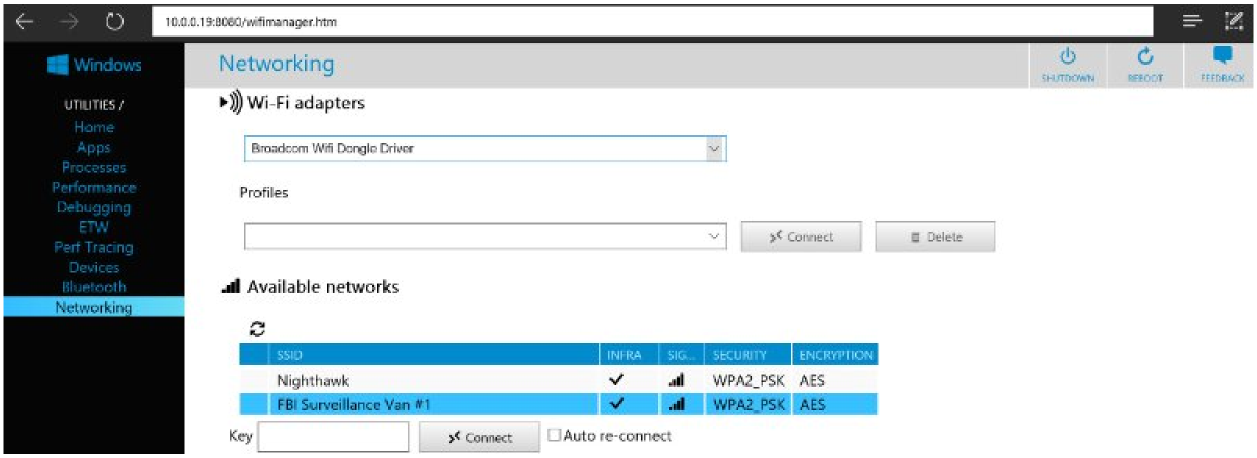

Figure A - 7: Windows IOT settings [29]

7. Once connected, open the Windows IOT Remote Client application and from the “selected devices” drop down menu you will find “Interactive Mobility” click on that and it will remote into the system. You may now begin to use the system

V. Using The System

8. Once you remote into the system through the Windows IOT Remote Client you may begin using the system and all of its features. There will be a main page and to get to the features of the project you click the house icon on the bottom left corner.

Figure A - 8: Interactive Mobility system main page[20]

9. There will be another window that pops up that looks like this with the main page as the background. To begin using the system you will need to connect to the system first by clicking

the “Connect” button in the system section. There are different sections of the GUI that can control different things in the Home. The Temperature section controls a fan, turning it on and off with different speeds. The Door section allows for opening and closing doors. The lights section controls a light being able to turn on and off. The object detection shows if there is an object in the way. The system section will allow the user to turn off the system and turn on. If the user doesn’t want to use the GUI anymore or to avoid accidentally pushing something you can disconnect from the system which essentially puts the screen on a lockout mode until you connect back to the system.

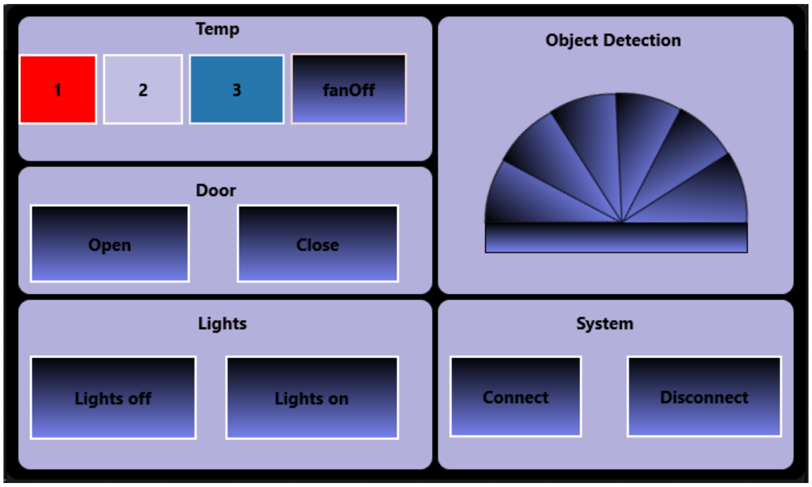

Figure A - 9: Interactive Mobility system home automation/object detection[20]

10. The safety feature will only show up in the event of an unsafe situation and the screen will look like in Figure A-5. If the user responds within a certain amount of time, an alert will be sent to the caregiver. If the user responds “NO” or doesn’t respond within a certain amount of time, then the alert will automatically send to the caregiver. Also if the angle goes way past a certain amount it will also automatically send out to the caregiver.

Figure A - 10: Safety Feature[20]

VI. Troubleshooting

If remoting into the system doesn't work through the Windows IOT Remote client application, try restarting the system by unplugging all wires from the battery and plugging them back in. Give it some time to start up and if the problem persists you may need to plug into the HDMI port located inside the box and see if the system was able to boot correctly. If you have determined that it booted correctly and the application is running by connecting the HDMI from the system to a monitor then there is a problem with the Wifi connection. Make sure that your home network is functioning correctly. Also make sure that the system is connected to the correct wifi, run through steps 5-6.

If the clicking the buttons are non-responsive, make sure to connect to the system in the system section of the application. Try disconnecting and the connecting. If that will not work you will have to reboot the system and connect to the system when it boots.

Appendix B. Hardware

The hardware concerning home automation is actually not very complicated. It consists of three components: automation of a lamp, a fan, and a common household door. The lamp and fan make use of relays to control to normal operations. The automation of a door is slightly more complicated as it involves using a solenoid and servo in parallel to open and close a door. The setup for the hardware required for home automation within our design is shown in the following block diagram.

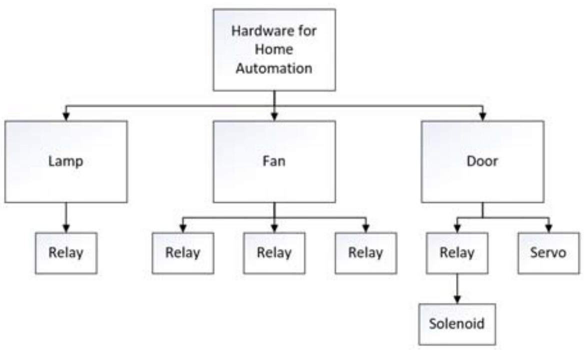

Figure B - 8: Hardware Regarding Home Automation – Block Diagram[20]

The basic circuitry for automating the control of a common household lamp is shown in figure B-2. A microcontroller is used to activate a relay, which in turn, determines whether or not the light is to turn on or off.

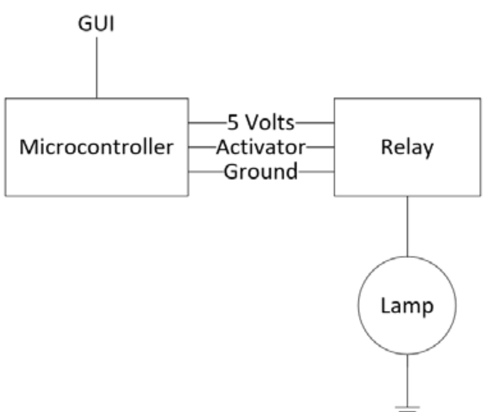

Figure B - 9: Basic Circuitry for Controlling a Household Lamp[20]

In order to control the settings of a common household fan, three relays were implemented into a simple circuit. When zero relays are on the fan is off. When the low relay turns on, the lamp is on low. When the low and medium relays are activated, the fan is on medium. When the low, medium, and high relays are activated, the fan is on high. The basic circuitry for controlling a common household lamp can be seen in the following figure.

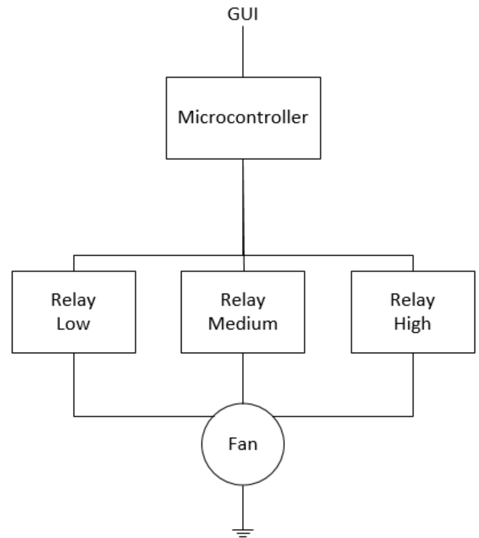

Figure B - 10: Circuitry for Controlling a Common Household Lamp[20]

This circuit is also rather simple, when the command to open the door is executed, the solenoid opens and the servo turns the door 90 degrees. When the user decides to close the door, the servo is once again activated, and turns the door 90 degrees in the other direction to close the door. The basic circuitry for controlling a common household door can be seen in the following figure.

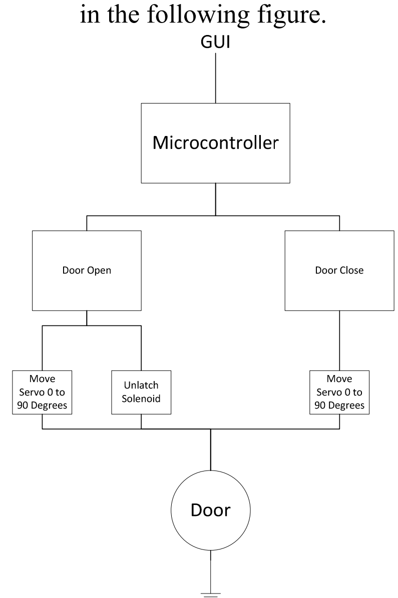

Figure B - 11: Circuitry for Controlling a Common Household Door[20]

The object detection system has few hardware components to make the system work. The goal is to mount the sensors onto the servo motor so that the sensors can scan in a 180-degree field of view in front of the wheelchair. The servo and sensors will be connected to a microcontroller that will process the data and be able to tell the user when there is something in the way. The different types of hardware and the wiring for the object detection system can be seen in the figures B-5and B-6.

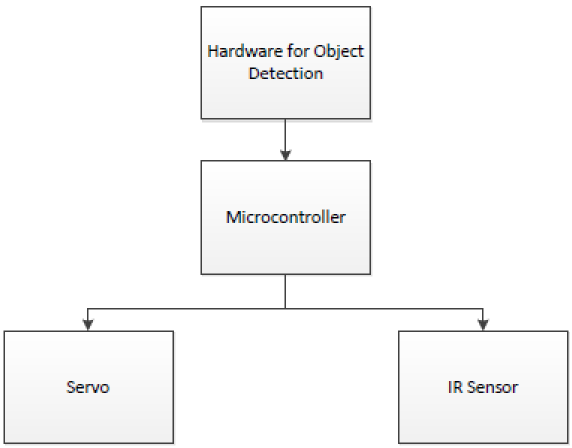

Figure B - 12: Hardware Regarding Object Detection – Block Diagram[20]

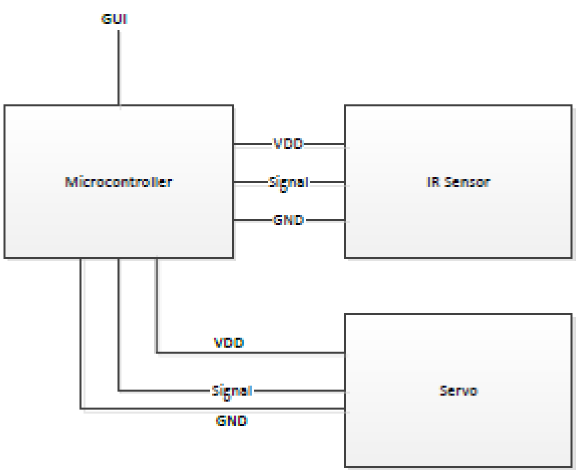

Figure B - 13: Circuitry for Object detection system[20]

For the emergency contact system, the only piece of hardware used is a MPU 6050 Gyroscope/Accelerometer sensor. This sensor will be mounted on a horizontal plane somewhere inside the wheelchair, most likely under the seat, where it will be able to measure the changes in the angular position of the wheelchair. The sensor is directly connected to the microcontroller through a power system and two input signals, as shown by the following figure.

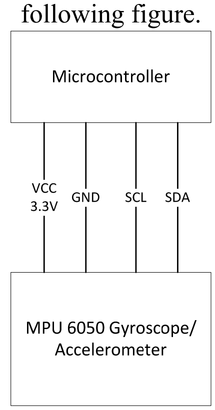

Figure B - 14: Electrical connections from the Gyroscope to the Microcontroller[20]

Appendix C. Software

The software of our design can be broken down into couple of different levels. The top level consists of the Graphical User interface. This is the software that the user will see and interact with. The underlying level will consist of the software used for implementing object detection, home automation, and a system to improve the user's safety. This software will be programmed into various microcontrollers, allowing the user to execute the individual features' specific functions. This software hierarchy is shown in the following block diagram.

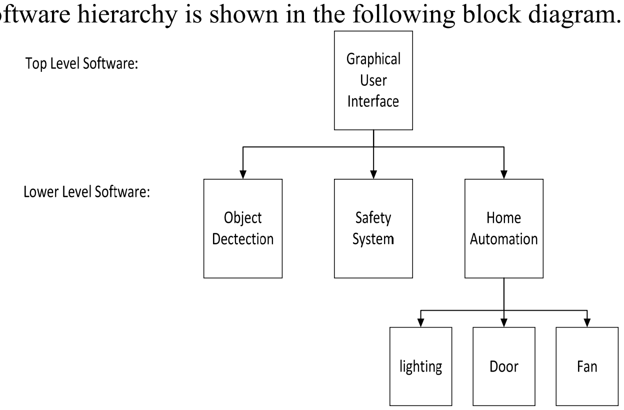

Figure C - 5: System Software - Block Diagram[20]

The GUI will be the means through which the user will communicate with the devices, either in terms of safety or leisure or necessity. The software will need to be relatively robust and iterative because the safety of the user is the most important focus of the project. The safety features need to be able to interrupt any tasks at any moment of the program and return to stable regular use if the user responds that he is in fact ok. If the user is unable to respond the GUI will use the established communications with the other microcontrollers to send out a stress call or signal or text or maybe even all of them. You will notice that in Figure C-1 the safety system located on the right hand side of the flowchart is connected to every section of the GUI. The safety features need to be accessed in every part of the process.

Our user will access the home automation components of our design via the GUI. As the features will be running on a microcontroller, the functions are to be programmed into the microcontroller such that individual signals may be processed as individual inputs. Specific functions are then activated within the microcontrollers programming, causing the individual household tasks to be performed. The basic flow of this program can be seen above in Figure B-1

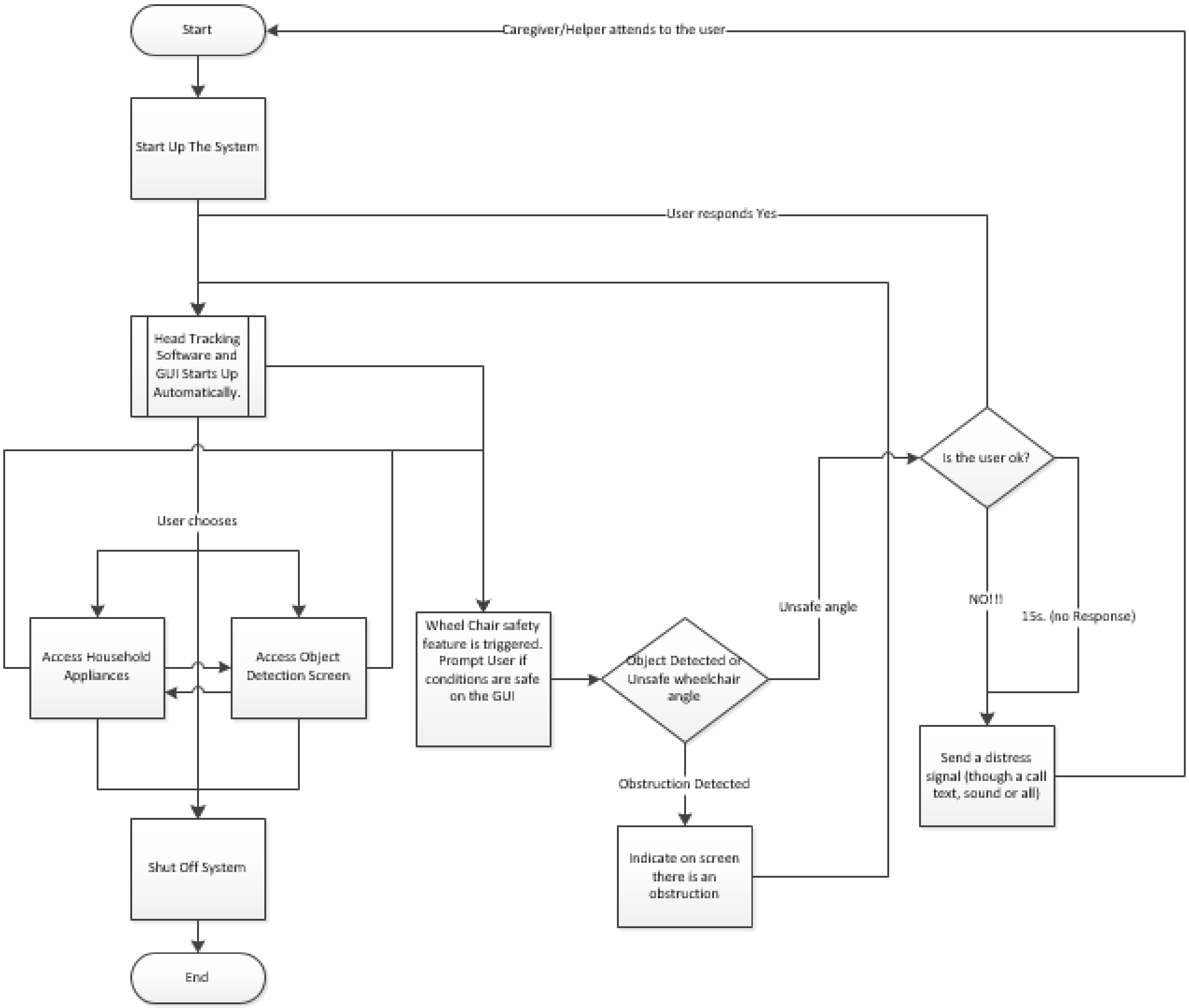

Figure C - 6: Device Control System Control[20]

The pseudocode for this design's home automation feature is shown below:

```
Input     lampOn, lampOff, fanOff, fanLow,
          fanMed, fanHigh, doorOpen, doorClose;

Output    Lamp, fanLow, fanMed, fanHigh, door;

main(){

    if(lampOn)
        Lamp = 1;
    if(lampOff)
        Lamp = 0;
    if(fanOff){
        fanLow = 0;
        fanMed = 0;
        fanHigh = 0;}
    if(fanLow){
        fanLow = 1;
        fanMed = 0;
        fanHigh = 0;}
    if(fanMed){
        fanLow = 1;
        fanMed =1;
        fanHigh = 0;}
    if(fanHigh){
        fanLow = 1;
        fanMed = 1;
        fanHigh = 1;}
    if(doorOpen)
        door = 1;
    if(doorClose)
        door = 0;
}
```

The figure below shows the normal operation for the object detection system. You can see that the system will be continuously running on startup unless it is shutdown or unplugged. This means that the system can continually send information to the user if it is able to find anything.

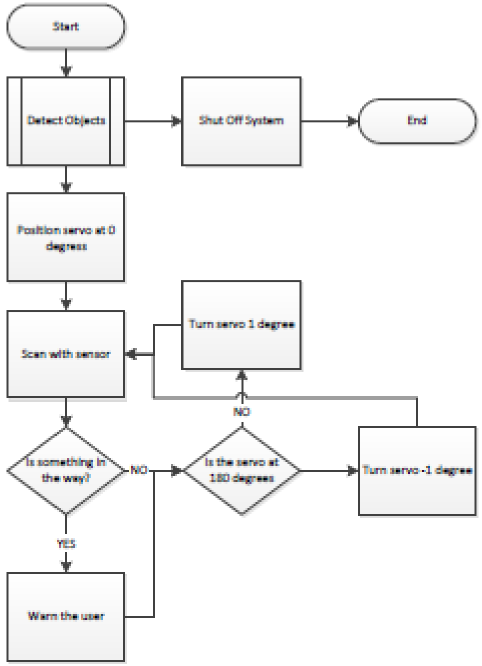

Figure C - 7: Object detection system flow[20]

The following figure shows the ideal flowchart for how the MPU 6050 sensor will work in detecting if the user has fallen past the designated safety angle. As you can see the angle is measured first and then input into the microcontroller before anything else occurs within the program. Once there is a measured angle in the program for the microcontroller to use it then compares the measured angle with the maximum safe angle. If the measured angle is equal to or greater than the safe angle then the program will increase a count variable to indicate that the user is past the safe point, and a delay of 1 second in the program will pass to allow the end count of 15 to have been at least 15 seconds. Once this is reached the microcontroller will send out a signal to the user interface, indicating that the sensor has detected the user has fallen and a prompt will ask if they wish to contact their caregiver.

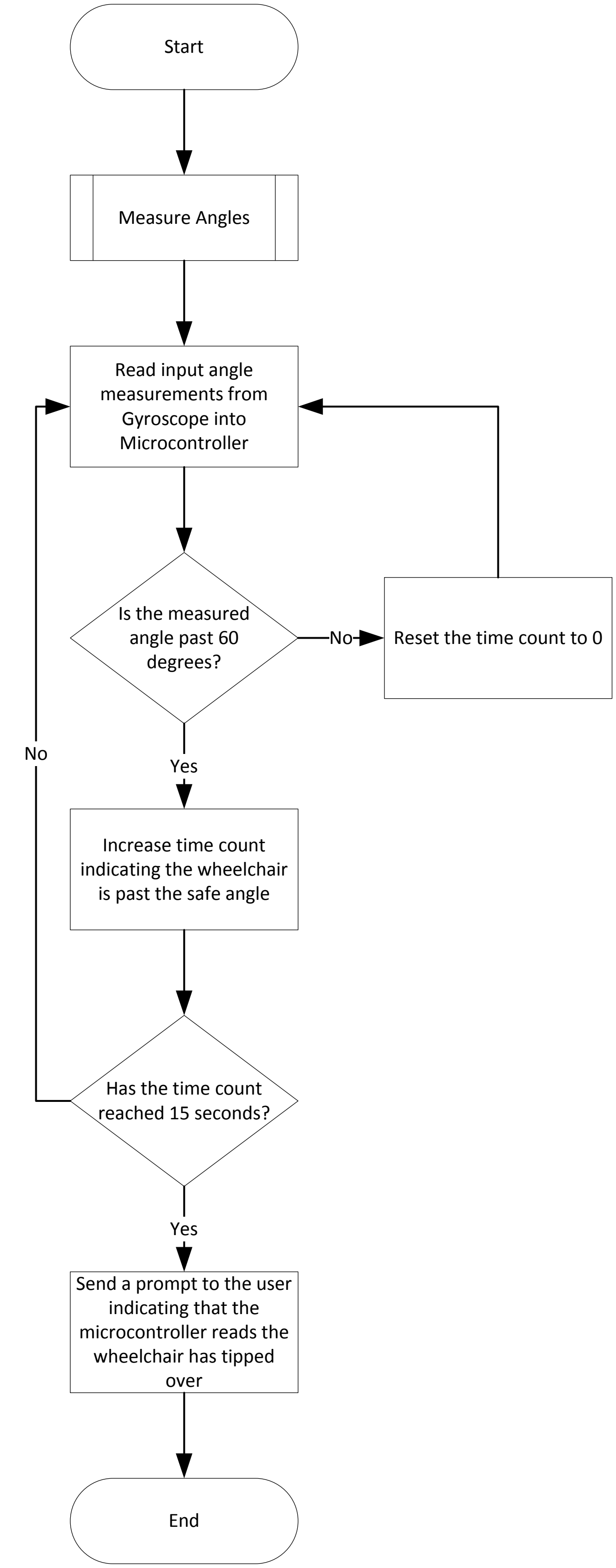

Figure C - 8: Gyroscope measurement and count system flow[20]

Appendix D. Mechanical

A mechanism was implemented to open and close a door using a combination of a servo and a solenoid, both to operate as the automated devices that will control the movement of the door. A solenoid will operate as the latching mechanism, while the servo is to move the door to and from the opened and closed positions. Since the door will extend out from its initial position, in comparison to the servo, a sliding mechanism has been implemented to account for this minor change in distance. This mechanism can be seen in its entirety below in Figure D-1.

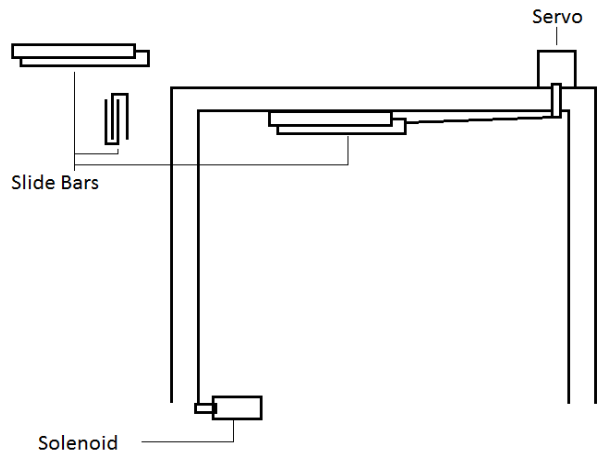

Figure D - 3: Mechanical Components Associated with Automated Door[20]

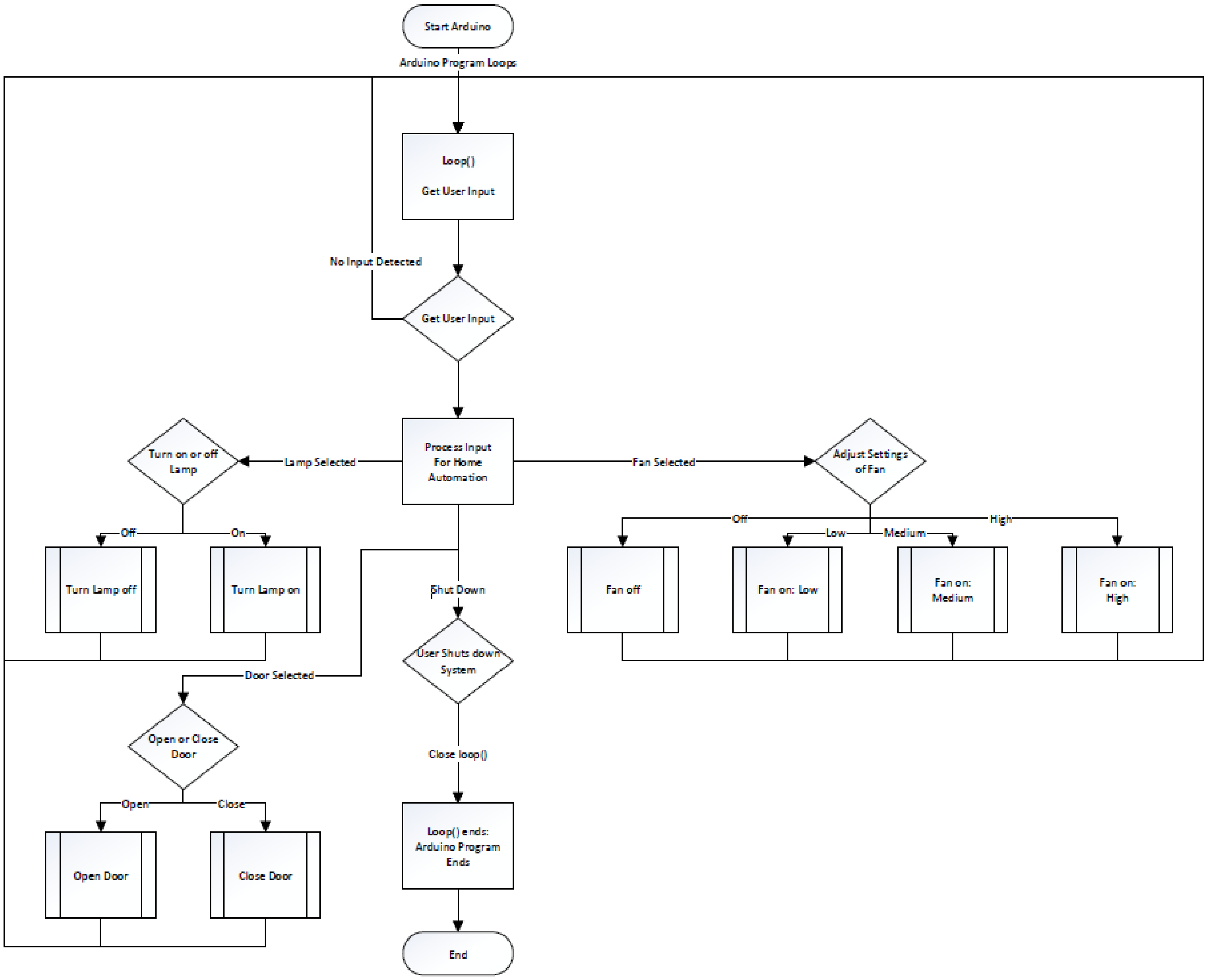

Figure D - 4: System Flow for Home Automation Software[20]

Appendix E. Vendor Contacts

Jefferey Rondoni- Donated his old wheelchair to help with our project.

Appendix F. Resumes

Riley Schofield

Education:

BS, Electrical Engineering, California State University, Sacramento — May 2017

Hartnell College, Salinas, CA — 2011 – 2014

A.S. Math	Fall 2014
A.S. Physics	Fall 2014
A.S. Engineering	Fall 2014

Relevant coursework:

Digital Control Systems	Intro to Feedback Systems	Signals & Systems
Intro to Logic Design	Intro to Scientific Programming	Applied Electromagnetics
Intro to Microprocessors	Network Analysis/(Lab)	Engineering Graphics/CAD
Electronics I/(Lab)	Electro-mechanic Conversion	Modern Communications

Skills:

Computer: Android IDE, C, Python, Excel programming, AutoCAD, B2Spice Circuit Design, MATLAB, Linux, Virtual Machines, Solid Edge, BASIC Stamp, Visual 2016 lighting software

Engineering: Electrical troubleshooting, Hardware and program troubleshooting, Basic Circuit building and analysis, Oscilloscopes, Multimeters, power supplies, Arduino, Vernier Logger Pro, Lab Pro interface, data analysis in Excel, Vernier Graphical Analysis

Experience:

Intern — **Aurum Consulting Engineers** — Summer 2016

Worked on placement of electrical components in multiple design plans using AutoCAD software. Began exploring using a program for developing Title 24 documentation. Also began using a program called Visual 2016, to figure out the amount of lumens in a particular area based on the different lights that are being used. Worked on a project which required the development of a new parking lot for a business in the area.

Intern — **National Science Foundation** — **Hartnell/UCSC Alternative Energies Projects** — Summer 2013-14

Built and improved a Robot meant to autonomously clean solar panels to increase their efficiency. Developing ideas on how best to go about achieving this goal. Delegating tasks to fellow interns as well as managing my own tasks. Researching and understanding code for different pieces of hardware. Building and troubleshooting circuits for different aspects of the robot. Demonstrating and explaining our product to grade school students at the end of the summer. Presenting a poster on the different steps we went through to accomplish our project, and also the different problems we encountered and how we overcame them. Testing with different type of wheels on a solar panel. Discovering new problems and redesigning the robot to help overcome them. Researching solutions to hardware problems. Worked on designing a remote control system using Bluetooth with android. Coming up with ideas for future aspects to input into the robot.

Intern — **Naval-Post Graduate School** — Summer 2012

Worked on testing a new Wi-Fi device for the Lego Mindstorm Brick. Normally the GUI software would have a block to use, but since the device was new they had not yet developed a programmed block. Typed coding was necessary to get the device set up and working to talk to and control the Mindstorm Brick.

Project Experience:

Autonomous Solar Panel Cleaning Robot — **Group Lead** — Summer 2013

This was the inaugural summer for coming up with a design for the robot. Came up with several ideas for a system to keep the robot on the solar panel carports. Built and adjusted a chassis and motor circuit system for the robot to run. Developed a presentation poster to share with other interns and the public at a general end of summer session.

Autonomous Solar Panel Cleaning Robot — **Group Lead** — Summer 2014

Analyzed the robot and built a test rig to tilt a solar panel to test the effectiveness of the wheels on being able to climb the solar panel. Discovered a possible reason for a problem we were having with the robot going in a straight line without adjusting the code. Presented a few changes made to the robot and discussed some possible future ideas that could be applied to help reach the ultimate goal.

Activities:

Member, IEEE, PES

Communications Officer, Hartnell Physics Club — 2012/2013

Aleksandr Verstivskiy

Objective Seeking a position in computer engineering or related field to apply my education as well as develop new skills through a practical approach to continue learning.

Education Bachelor of Science, Computer Engineering *Expected May 2017*
California State University, Sacramento
GPA: 3.4

Projects

Database:
Collaborated with a group to create a database using Java and MySQL for a division of California Parks and Recreation as a class project. The database allowed the user to store attendance numbers, create graphs, and create monthly notes.

Data path:
Using Synopsys's VCS tool I designed and programmed a single cycle five stage pipelined data path in Verilog that performed some instructions similar to a MIPS based instruction set architecture.

Voice Recognition:
Interfaced an Arduino and Raspberry PI to listen to verbal commands that would perform actions such as simulate the press of button on a remote and also launch an application on the PI using a Python script.

Wheelchair System:
Together with my senior design team, we created a system to give more independence to people in wheelchair who have no functionality in their arms and legs. With the system the user could control household appliances using head tracking, display warnings about objects that were detected, and contact a caregiver during an emergency.

Skills

Programming: Java, C, Verilog, VHDL, x86 Assembly, SQL
Software: Multisim, Quartus II, jGRASP, PuTTY, Xilinx ISE, PSpice, VMware, Cadence Virtuoso, Synopsys VCS
Hardware: Function Generator, Oscilloscope, Digital/Analog Multimeter, Logic Analyzer, Arduino, Microchip, Raspberry PI, Parallax Propeller, de0-nano FPGA, Spartan 3E FPGA
Leadership: Lead a team through the SDLC. Responsible for creating an agenda for team meetings and making sure deadlines were met on top of my assigned programming tasks.
Multilingual: English and Russian

Related Courses

Computer Software Engineering
Advanced Logic Design
CMOS & VLSI
Operating System Principals
Computer Networks and Internet
Network Analysis
Advanced Computer Organization

Accomplishments

Member: Tau Beta Pi
Dean's Honor List: Fall 2012, Spring 2013, Spring 2014, Spring 2015

Vadim Artishuk

Objective Improve my skills in a computer engineering environment through experience and learning.

Education Bachelor of Science, Computer Engineering — Expected May 2017
California State University, Sacramento GPA 3.3

Projects

Assist individuals that are limited only to head movement access technology (Senior Project):

Using head tracking software, my team and I were able to develop a system that enables the user to access some common household appliances through a graphical user interface. This system also includes safety features such as object detection as well as a detection system to check if the user has tipped over in his wheelchair. This is a two semester long project, the first semester is design and implementation and the second is project refinement and testing.

Using Speech Recognition to Control Devices:

The purpose of this project is to use speech recognition to simulate the control of devices such as an IR remote controller, a personal computer and a LCD. The inspiration for the project came from my group member who knew a person with cerebral palsy, so voice commands provided a lot more aid.

Short demo can be found at: https://youtu.be/FC6C8UaRvrg

Skills

Multilingual: English, Ukrainian, and Russian.

Technical skills: Java (proficient), C (proficient), x86 Assembly (proficient),HTML/CSS (beginner), Verilog (proficient), VHDL (proficient),MIPS Assembly (proficient), C# (beginner), XAML (beginner), Python (beginner).

Software Applications: Quartus II, Eclipse, Xilinx, Cadence Virtuoso, Synopsys VCS, PSpice, JGrasp, Visual Studios.

Personal Skills: Strong work ethic, Communication skills, Time management, Strong team player, Adaptability.

Work Experience

Assistant Mechanic: 2009-Present

Assist my dad in his trucking business with both mechanical and cosmetic reconstruction including minor side jobs in repairing vehicles. I do a majority of electrical work that comes in; including reading electrical wiring diagrams, testing and debugging engine fault codes, rewiring and fixing SRS airbag faults.

Office Assistant/Internship Century 21 Real Estate: 8/2011-1/2012

Organized customer paperwork, scan, update and backup client paperwork on an online database.

Volunteer and Community

Grace Trinity Church:

Assisted with the reconstruction of my church. Manage the technical and sound team, tasked with troubleshooting sound interference, sound loss, system and machine failure, and adjust sound to different environments.

Ryan Law

Objective Seeking a position as an engineering/manager in initiatives that utilize state-of-the-art, software and/or hardware components with a creative, technology-driven organization in an environment that encourages innovative thinking, recognition, and career development. Customer interaction is a plus.

Education Bachelor of Science, Computer Engineering — Expected August 2017
California State University, Sacramento — GPA 3.4

Projects

Assist individuals that are limited only to head movement to access technology (Senior Design):

Using head tracking software, my team and I were able to develop a system that enables the user to wirelessly access common household appliances through a graphical user interface. This system includes safety features such as object detection and a system which monitors to see if the user has tipped over in their wheelchair. This is a two semester long project, the first semester focused on design and implementation while the second and current semester focuses on design refinement and testing.

Using Speech Recognition to Control Devices:

The purpose of this project is to use speech recognition to simulate the control of devices such as an IR remote controller, a personal computer and a LCD. I was inspired to work on this project in order to develop a communication system for my friend with cerebral palsy, as he has trouble speaking.

Skills

Technical skills: Java (proficient), C (proficient), x86 Assembly (proficient),Verilog (proficient), VHDL (proficient),MIPS Assembly (proficient), C# (beginner), Python (beginner), IEEE 502.15.4 Protocol (proficient)

Software Applications: Quartus II, Eclipse, Xilinx, Cadence Virtuoso, Synopsys VCS, PSpice, JGrasp, MASM

Personal Skills: Strong work ethic, Communication skills, Time management, Strong team player, Adaptability

Work Experience

Direct Support Professional — **Strategies to Empower People** — **November 2014-September 2016**

Strategies to Empower People aims to help the disabled live independent, successful lives. Duties were to provide support in home, work, transportation, and physical and recreational activities. Skills in communication are necessary as much of the work involves verbal translation. Since a disabled person may be unable to communicate their needs verbally, this skill is vital to understanding the disabled person's wants and needs. Other skills that this job requires are trust and accountability, as disabled people are indeed trusting the Direct Support Professionals with their lives.

Team Leader — **Bridges After School Program** — **April 2012-September 2013**

Bridges After School Program provides a safe and friendly environment in which children are able to attend after school until the time that their parents are able to pick them up. One Team leader is responsible for a group of about 20 children. Responsibilities included managing the children in a classroom, teaching them academic information, and maintaining order during recreation. A team leader must be able to solve problems with no prior knowledge of the situation. They must be understanding of other people needs and concern, while at the same time knowing when to exercise authority to keep children behaved and on-task.