
ECE 480:

Accessible Voting Joystick

Michigan State University: College of Engineering



Resource Center
for Persons with
Disabilities

MICHIGAN STATE
UNIVERSITY

University Outreach
and Engagement
Usability/Accessibility
Research and Consulting

Final Report

April 24th, 2013

ECE 480 Team 5 Members

Yangyi Chen

Tyler Dennis

Graham Pence

Behdad Rashidian

Joy Yang

Prepared For

Dr. Timothy Grotjohn

Professor and Chairperson

Electrical and Computer Engineering

Executive Summary

According to the United States Census 6.7 million US citizens have difficulty grasping objects. These individuals are potential candidates for joystick controlled wheelchairs. An estimated 125,000 people in the U.S. use joystick controls for their power wheelchairs. Joysticks are employed due to the fact that various objects can be utilized a joystick's precision and control. A voting joystick has been designed for those with dexterity limitations that would like to exercise their right to vote.

Michigan State University Usability/Accessibility Research and Consulting and the Resource Center for Persons with Disabilities request a voting interface that is convenient to use for individuals with a wide variety of motor skill capabilities. This system needs to accommodate a variety of different motor needs. It has been determined that a force feedback joystick that can be interfaced with voting software on a secure personal computer will be constructed to satisfy the request of the sponsor. Force feedback will help produce a resistive force when moving to different selections of a ballot and will improve the overall accuracy and control of voting.

Acknowledgement

First and foremost, the authors would also like to thank Professor Timothy Grotjohn, Professor and Chairperson of the Electrical and Computer Engineering department at Michigan State University, for this great opportunity.

The authors would also like to acknowledge and extend our heartfelt gratitude to the following persons who have made the completion of this project possible:

- Stephen Blosser of the RCPD of Michigan State University for his vital assistance and inspiration of the research summarized here.
- Dr. Sarah Swierenga and Graham L. Pierce from the Usability/Accessibility Research and Consulting for passing their knowledge down to us and motivation throughout the development of this project.
- Dr. John Deller of the Michigan State University Electrical and Computer Engineering department for his constructive insight and continued support throughout the semester.
- Gregg Motter and Steve Zajac of the Michigan State University Electrical and Computer Engineering faculty and staff for their support and feedback.

This research was funded through a grant from the Information Technology & Innovation Foundation (ITIF), Accessible Voting Technology Initiative (AVTI) to Michigan State University (ITIF Subgrant No. 2013004; Prime Grant No. EAC110149B – U.S. Election Assistance Commission).
Principal Investigator: Dr. Sarah J. Swierenga, MSU, Usability /Accessibility Research and Consulting

Table of Contents

Executive Summary.....	1
Acknowledgement.....	2
1 Introduction and Background	5
1.1 Introduction.....	5
1.2 Background.....	6
1.3 Design Specifications.....	6
2 Research and Solution Approach	8
2.1 Decomposition of the Problem	8
2.2 House of Quality and Critical Customer Requirements	9
2.3 Risk Analysis	9
2.4 Estimated Budget	10
2.5 Project Timetable	11
3 Technical Description of Work Performed.....	12
3.1 Force Feedback	12
3.2 Contents of the Force Feedback Joystick.....	12
3.3 H-Bridge circuit.....	13
3.4 Actuator Motor Power Supply	15
3.5 Force Adjustment Software	16
3.6 FFShifter	17
3.7 AutoHotKey	19
3.8 Python Script.....	19
3.9 Python Script Pseudocode	19
3.10 Georgia Tech Ballot	20
4 Testing.....	23
4.1 Testing the Joystick	23
4.2 Interfacing the Joystick to the HTML Ballot	23
4.3 Testing the Adjustable Force Feedback	24
4.4 Testing with the Resource Center for Persons with Disabilities	24
4.5 Prototype Evaluation.....	25
5 Final Cost, Schedule, Summary, and Conclusions	26
5.1 Final Cost.....	26

5.2 Schedule	27
5.3 Future Enhancements	28
5.4 Conclusion	28
Appendix 1	30
Appendix 1.1 Yangyi Chen	30
Appendix 1.2 Tyler Dennis.....	31
Appendix 1.3 Graham Pence	32
Appendix 1.4 Behdad Rashidian.....	33
Appendix 1.5 Joy Yang.....	34
Appendix 2	35
Appendix 3	36
Appendix 3.1	36
Appendix 3.2	39

1 | Introduction and Background

1.1 | Introduction

The Usability/Accessibility Research and Consulting (herein after UARC) and Resource Center for Persons with Disabilities (herein after RCPD) of Michigan State University would like to develop and implement a dual-axis joystick on a computer system that will replace the manual interaction with a voting system. Such a device would allow many individuals with dexterity limitations and special needs to vote without significant frustration. It would also allow individuals with special needs to vote within a standard time frame when compared to an individual without special needs. Research has indicated that people with disabilities are less likely to vote than individuals who do not have disabilities (statistics show 7% of the disability population in the survey conducted are less likely in 2008 and 3% less likely in 2010, see Rosenberg). Voters would likely be more attracted to vote if a device was created that would allow all users to vote conveniently, comfortably, and in an efficient manner. To address those issues, Team 5 of the ECE 480 class will design a device that will attract more special needs individuals to the voting booth - a user-friendly voting machine to assist with special needs so that those individuals can cast a private ballot. Specifically, the design will be a dual-axis joystick with force feedback.

The proposed double axis joystick will be implemented with a force feedback control to enhance user interaction with a standard graphical user interface (GUI). While standard joystick devices are input-only, force feedback control utilizes the sense of touch in a user interface design to provide physical information to a user. For example, an application of a force feedback control in a joystick is a vibration when the joystick has "entered" information. Force feedback implemented in the proposed double axis joystick would provide physical sensations to a user for confirmation or reminder purposes.

To make the proposed double axis joystick compatible with any electronic voting machine system, the joystick must have USB connectivity. USB connectivity will replace the serial input currently used in various precincts across the country (a local precinct that uses this input is in East Lansing, MI). It will also allow any voting machine/computer that has USB access to use the joystick. A prototype of this joystick device will to be developed and tested by Design Day, April 26th, 2013.

1.2 | Background

Previous special needs voting machines have been proposed and implemented in voting booths in local precincts (precincts in East Lansing, as an example). One or two button voting devices have been commercialized and already implemented in voting booths. An example of a one or two button voting device is a rocker switch. The dual rocker switch has a silent option and an audio option. Swierenga and Pierce (2013) show that while using the silent option, a person without special needs will press the rocker switch buttons 872 times within 18 minutes while the audio option utilizes 1,200 button presses within 37 minutes for the standard ballot. 37 minutes for a user without disabilities is a significant amount of time to vote (without other factors considered, such as wait time, etc...). To determine the approximate time for persons with special needs to vote using this system, a factor of four should be multiplied to the time for a user without disabilities (Swierenga, Propst, Pierce, and & Sung, 2011).

Other devices that have been commercialized include features such as: screen-touches, sips, puffs, and other physical actions taken by the user to interact with the voting system. Individuals with dexterity limitations are physically limited by the number of interactions that they can initiate. For example, the size of a button, the location of a button, the number of presses required, etc... all contribute to the amount of strain that must be applied by users with disabilities. These factors must all be taken into consideration when designing a special needs voting machine.

1.3 | Design Specifications

Various requirements must be met in order to create an effective accessible voting system. An interface needs to accommodate needs of special individuals with a wide variety of motor skill functions. This interface needs to be easily accessible and intuitive for anyone who uses it. Because the focus of this project is on special motor skill needs, Team 5 will not be focusing on people with unique sensory abilities, such as blind individuals. Given the range of special abilities, a wide variety of conditions is a challenge in the design. To take as an example, a system that works well for someone with cerebral palsy may not work as well with someone who has difficulty grasping their hands with force. The design will include a software component. Ideally, the time a special needs individual requires to complete a ballot using this system should be comparable to the time it takes for the typical voter to complete their ballot.

To find the design solution that best tailors the required specifications, a list of criteria by which each design solution can be assessed must be created. A first criterion is compatibility with existing voting software. It is imperative for the hardware to seamlessly interact with the voting software with which the design is tested. Team 5 will be utilizing the HTML ballot program voting software designed by the Georgia Institute of Technology. This software will be discussed later in the proposal. A second criterion is voting privacy. It is vital that the confidentiality of each individual's vote is ensured. A third criterion by which the design will be assessed is accuracy. It is crucial that the design ensures the accuracy of the user's vote. Another important aspect is the cost of developing and implementing the design. Due to the allotted \$500 budget maximum, it is essential that the design can be built using \$500 or less. Customization is another very important design trait. The joystick must work well with a wide variety of motor disabilities, and thus should be programmable and compatible to fit a variety of needs. Finally, the joystick needs to be durable. Some special skills users may exhibit exceptional strength that could damage a structurally weak design.

2 | Research and Solution Approach

2.1 | Decomposition of the Problem

Team 5 has designed a force feedback joystick with the task of making voting more accessible. It will take the place of voting machines that are available at voting polls currently. The intent of the joystick will help those of dexterity limitations, which will help users with control and stability when making selections on the ballot. A fast diagram utilizes the decomposition of the joystick beginning with the task, basic function, and lastly with secondary functions. The fast diagram follows a sequence shown from left to right of how one item is achieved by following the right most immediate object. Alternatively, the fast diagram may be followed from a right to left sequence. This sequence displays information of how a decision is made to complete an objective. From the fast diagram, the basic function of the accessible voting joystick is determined by the control of voting software interacting with the joystick. The use of this fast diagram, allowing the decomposition of the problem into subcomponents, can be undertaken individually to cultivate a conceptual design.

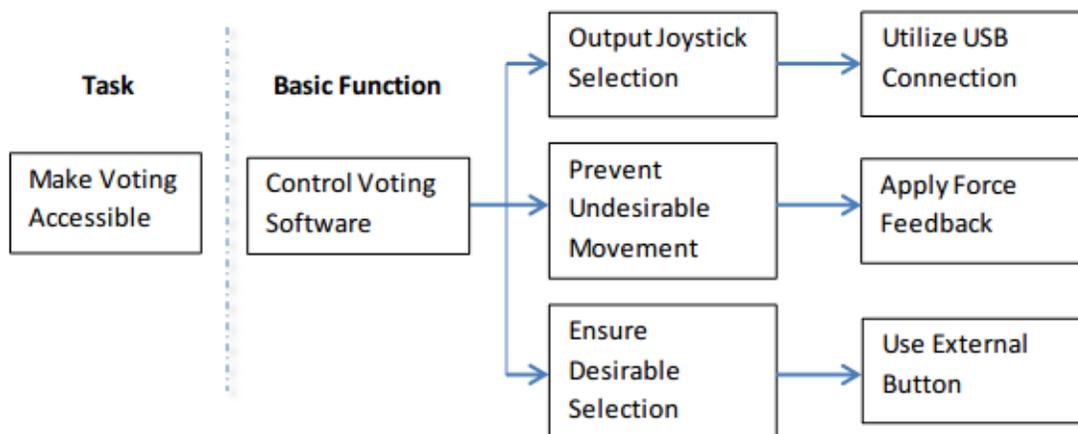


Figure 2.2.1: Fast Diagram

2.2 | House of Quality and Critical Customer Requirements

The basic and secondary functions were well defined, but the team wanted to finalize the best representation of our design based on the requirements of our sponsor. After talking with Stephen Blosser, he was open to a variety of different designs and we ended up with three final choices. The first two designs were different possibilities of a joystick, where the last design utilized voice recognition.

Using a decision matrix, the team was able to figure out the best representation for our project given the requirements for the sponsor. The most important factors taken in consideration were the voting software, voting privacy, accuracy to vote, cost to build, customization, and the durability of the joystick.

On the table from a scale of 1 to 3 and 1 being the best, the different choices were tallied up. The team picked to modify an existing joystick as the accuracy and cost were much more superior to that of building one and voice recognition is much more difficult to providing privacy issues.

1 = Worst 3 = Best	Compatibility with Existing Ballot Software	Voting Privacy	Accuracy	Cost	Customization	Durability	TOTAL
Modify Existing Joystick	3	3	3	3	2	3	17
Build Joystick	3	3	2	1	3	3	15
Voice Recognition	1	2	2	3	3	3	13

Figure 2.2.2: Fast Diagram

2.3 | Risk Analysis

A number of safety issues that were taken in consideration. High risk components included any circuitry or potential damage on the motors required to activate the force feedback. Medium risks included any

hardware components including the exterior housing of the base and the newly designed handle. Lastly the low risks included any software integration of the voting software or programming to the joystick.

Risk Summary
High Risk – Damage to Internal Electronic Components <ul style="list-style-type: none">➤ Internal hardware design of the joystick is critical. Any damage to the internal circuits or chips will delay our project schedule. If a component related to the force feedback function is damaged, the team’s cost will dramatically increase. In addition, any potential replacement of a component during the testing phase will be more complicated since the whole internal circuit may need to be broken down.
Medium Risk – Material Selection of the Joystick Body <ul style="list-style-type: none">➤ When building a desired joystick for people with special needs, the materials used to make the joystick body will require a special level of hardness. Any damage to the joystick body while testing will disadvantage the team by having to rebuild the joystick. Without the complete joystick, testing cannot be arranged.
Low Risk – Software Integration <ul style="list-style-type: none">➤ This project requires the joystick movement to adjust to the testing interface, meaning the input data analysis and the GUI have to be compatible. It will take time for the team to ensure the communications between hardware and software are effective and fully functioning.

Figure 2.2.3: Risk Analysis

2.4 | Estimated Budget

Over the course of the semester, a \$500 budget had to be carefully utilized for all parts and testing for the project. The majority of the expenses went toward the testing of joysticks, where the components inside the joystick were analyzed. These joysticks included a force feedback, hall effect (magnetic components), and basic return to center joysticks. The exterior housing modified for the joysticks were provided by RCPD and a few other mechanical parts such as the external button and handle totaled to \$50. The team also bought a micro controller Arduino Leonardo for programming purposes. Overall the estimated budget added to \$400 dollars with \$100 set away for any replacements items.

Testing on joysticks	\$300
Exterior Housing	Provided by RCPD
Mechanical parts	\$50
Arduino Leonardo	\$50

Figure 2.2.4: Estimated Budget

2.5 | Project Timetable

To make sure the team stayed on track, a project timetable was implemented with four design phases. Phase 1 included research about current joysticks on the market, voting polls, and the problems with past voting machines. Phase 2 started the product creation of our joystick, this led to ordering numerous parts and figuring out the best solution to our design. Phase 3 was the product development of the specific joystick chosen with the set modifications. Lastly phase 4 was product testing for the joystick; this includes all trouble shooting for haptic feedback and programming.

- Phase 1 – Research
 - Existing Design
 - Problems
- Phase 2 – Product Creation
 - Parts research
 - Ordering parts
- Phase 3 – Prototype Development
- Phase 4- Product Testing

The team completed a Gantt chart that can be located in Appendix 3.2 that depicts the team’s critical paths stated above.

3 | Technical Description of Work Performed

3.1 | Force Feedback

A major feature in the design of our joystick is the use of force feedback (also called haptic feedback). This relies on the sense of touch by applying forces, vibrations, or motions to the user. The earliest applications of force feedback have been used in flight control to simulate control of the outside surfaces or even for alerts for certain malfunctions. Often users with motoric disabilities have a hard time handling objects and may trigger unwanted movements. The team has devised that the best solution to this problem would be to implement force feedback on the joystick. There would be a resistive force implemented between the candidates on a ballot, preventing any unwanted movements. Overall this will provide users of motoric disabilities to have better accuracy and control of the desired selection.

3.2 | Contents of the Force Feedback Joystick

Unlike ordinary joysticks, force feedback joysticks have an onboard microprocessor, a couple of electrical motors and a belt system. The processor contains all the programming involved to control the joystick which is sent to the position sensor through electrical signals. The X-axis and Y-axis are controlled by each motor which activate the movement of the joystick through a pulley system. The belt's function is to transmit and amplify the force from the motor to the shaft.

- Synthetic polymers:
 - Poly-lactic acid
 - Poly-glycolic acid
 - Poly-caprolactone
 - Poly(dioxanone)
 - Poly-lactide-co-glycolide
- Magnesium based alloys
 - Mg
 - Zn
 - Li
 - Al
 - Ca

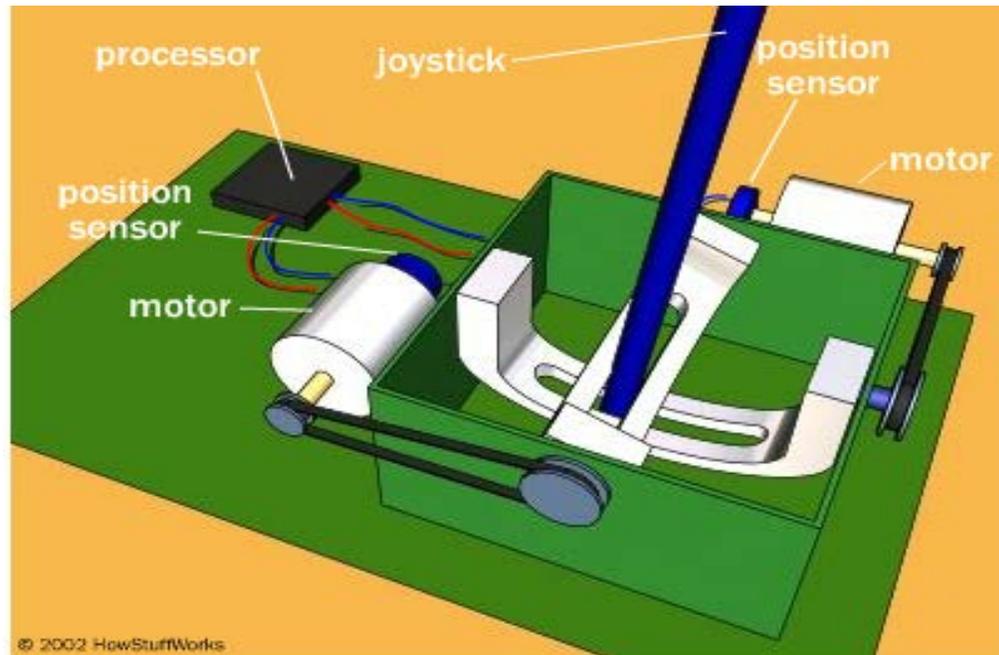


Figure 3.2.1: Force Feedback

3.3 | H-Bridge circuit

The H bridge circuit enables a voltage to be applied across a load in either direction. We need to use such a circuit for our force feedback motor to allow DC motors to run forwards and backwards. An H-bridge circuit is typically constructed using opposite polarity devices, such as PNP transistors or P-channel MOSFETs connected to the high voltage bus and NPN BJTs or N-channel MOSFETs connected to the low voltage bus.

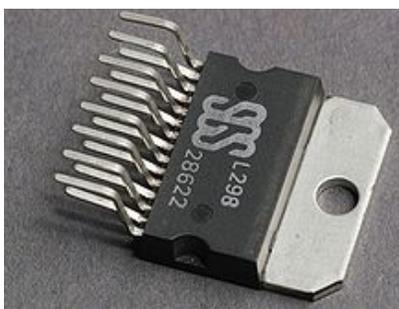


Figure 3.3.1: Dual H-Bridge Motor Driver

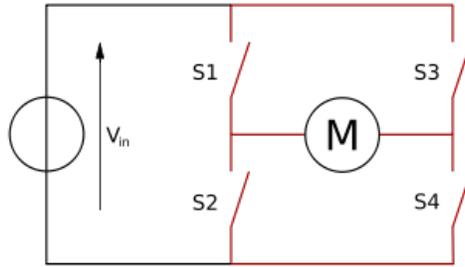


Figure 3.3.2: Structure of H-Bridge

There are different conditions that we have to consider for the H-Bridge circuit. When the switches S1 and S4 (shown in figure 2) are closed and S2 and S3 are open, we will have a positive voltage applied across the motor. By opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed. This will let us have reverse operation of the motor. The switches S1 and S2 should never be closed at the same time. It will cause a short circuit on the input voltage source. The same applies to the switches S3 and S4.

The following diagram summarizes the operation of the H-Bridge:

S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Shoot-through
0	0	1	1	Shoot-through
1	1	1	1	Shoot-through

Figure 3.3.3: H-Bridge switches

3.4 | Actuator Motor Power Supply

In order to choose the correct power supply, the amount of current and power required must be known. The force that raises the pin in the motor is directly related to the current flowing through the actuator motor. The AC/DC supply delivers 24Vdc to the main printed circuit board. The CPU receives the USB signals, checks the position of the X, Y, rudder and Throttle potentiometers, 12-switches, and controls the motor current.

The motor current control and power stages are all separate. Motor current is around +/-0.7A max, with 8.5 ohm motors. We could increase the current up to 1.4A. The circuits control the current through the motors instead of adjusting the PWM duty-cycle. Therefore, the motor current is precisely controlled, and not dependent on the motor internal resistance.

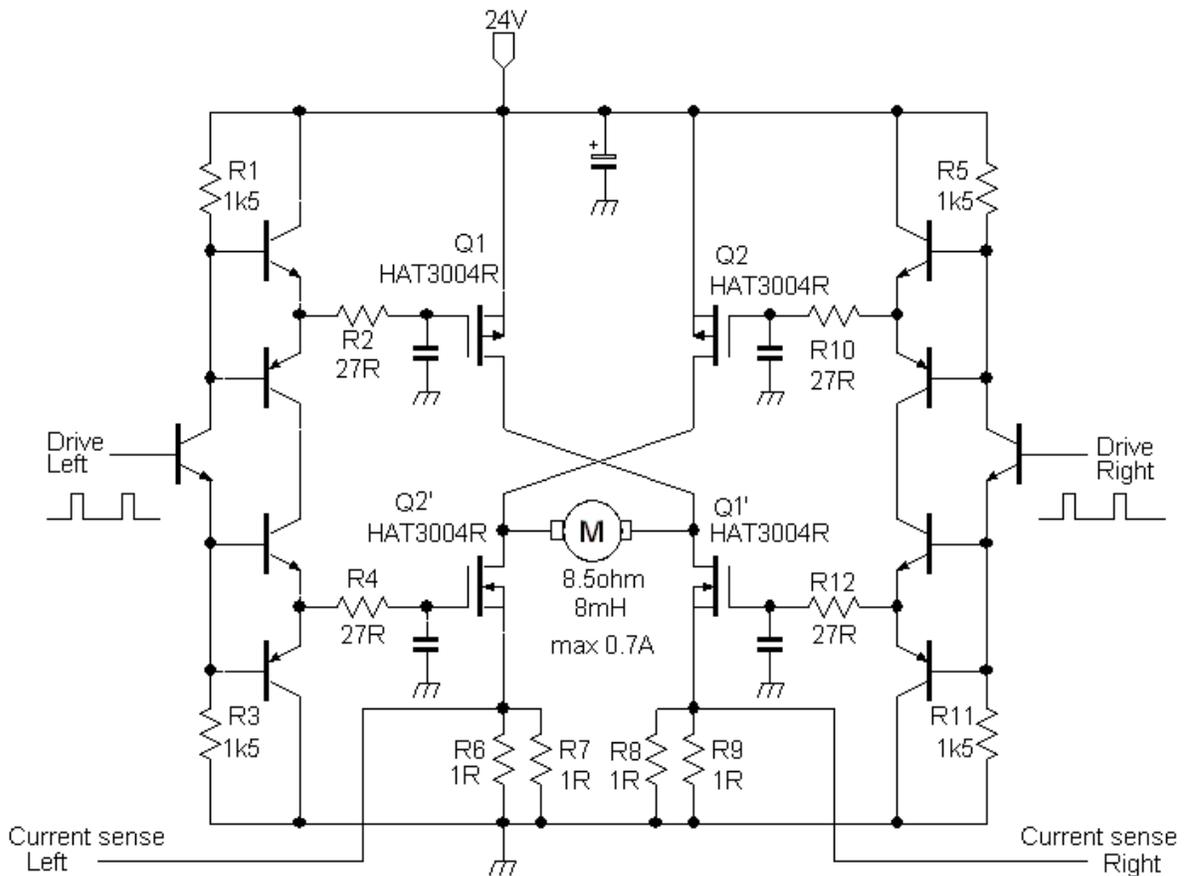


Figure 3.4.1: Actuator motor circuit

The above schematic is an H-Bridge based circuit which is introduced earlier. This design will help the team reach our goal of making force feedback for the joystick.

Team 5 needs to have 6 NPN BJT transistors that can act as a switch that will cut off current when the output is 0 V and switch on the external power supply when the output is 5 V. For this application the transistor will operate in the active region when the microcontroller output is 5 V, and it will operate in the cutoff region when the microcontroller output is 0 V. The transistor model being used is the 2N2222A NPN BJT.

Team 5 also needs to use 4 PNP BJT transistors that are sourcing the current. The PNP transistors have their emitter connected to the resistor and collector to the ground. Base is connected to the 1K Ω resistor and the emitter of the NPN BJT with the right-drive base.

In the schematic with MOSFETs switching that has conduction from GND(0V) to 24V via 2 MOSFETs, division of current can easily occur if there is a small delay in one of the MOSFET drives. We need to only have one side of the bridge active at the time for one motor current direction, the other side will be fully switched off, and the inductive current will flow through the other side MOSFET's body diodes. Disadvantage of this method is that the power dissipation in the MOSFET that is not active is actually quite high. This happens due to the higher voltage drop across the body diodes.

The voltages across the current sense resistors are processed with OPAMPs and work as the feedback of the force current loop. There are current sense resistors in each side of the bridge. By reducing the current sense resistors R7 and R9, the system will increase the voltage across the motors to keep the same feedback voltage, thus the current through the motors will increase.

3.5 | Force Adjustment Software

One of the critical features of the voting joystick was a quick and easy way to adjust the magnitude of the force feedback. In order to make the force adjustment feature easily accessible for all users, the team installed a force adjustment knob on the case of the joystick. When turned, the knob adjusts a potentiometer. This changes the signal that is sent to the computer. In order to translate this varying signal into a varying force, software was written. The software was written in Python, and requires two

supplementary pieces of software to run: FFShifter and AutoHotKey. Figure 3.5.1 below displays the knob that was installed to adjust the force of the joystick.



Figure 3.5.1: Force Adjustment Knob

3.6 | FFShifter

FFShifter is a free program that allows the user to easily apply force patterns to a force feedback joystick. Force boxes can be drawn onto the graphical user interface (GUI) in areas that correspond to the joystick position. The size of the box, location of the box, magnitude of the force, and type of force can all be easily adjusted using the GUI. FFShifter uses DirectX libraries to interact with the joystick and adjust the force feedback levels. DirectX is a collection of libraries that allow the programmer to interact with hardware at a low level using a high level programming language. The library that is used to interact with a joystick, or any other input device, is called DirectInput. The DirectInput library contains functions that allow the developer to adjust several aspects of a joystick. This includes input

ranges, force feedback magnitudes, and the specific functions of each button on the joystick. Figure 3.6.1 below shows the GUI of the FFShifter software.

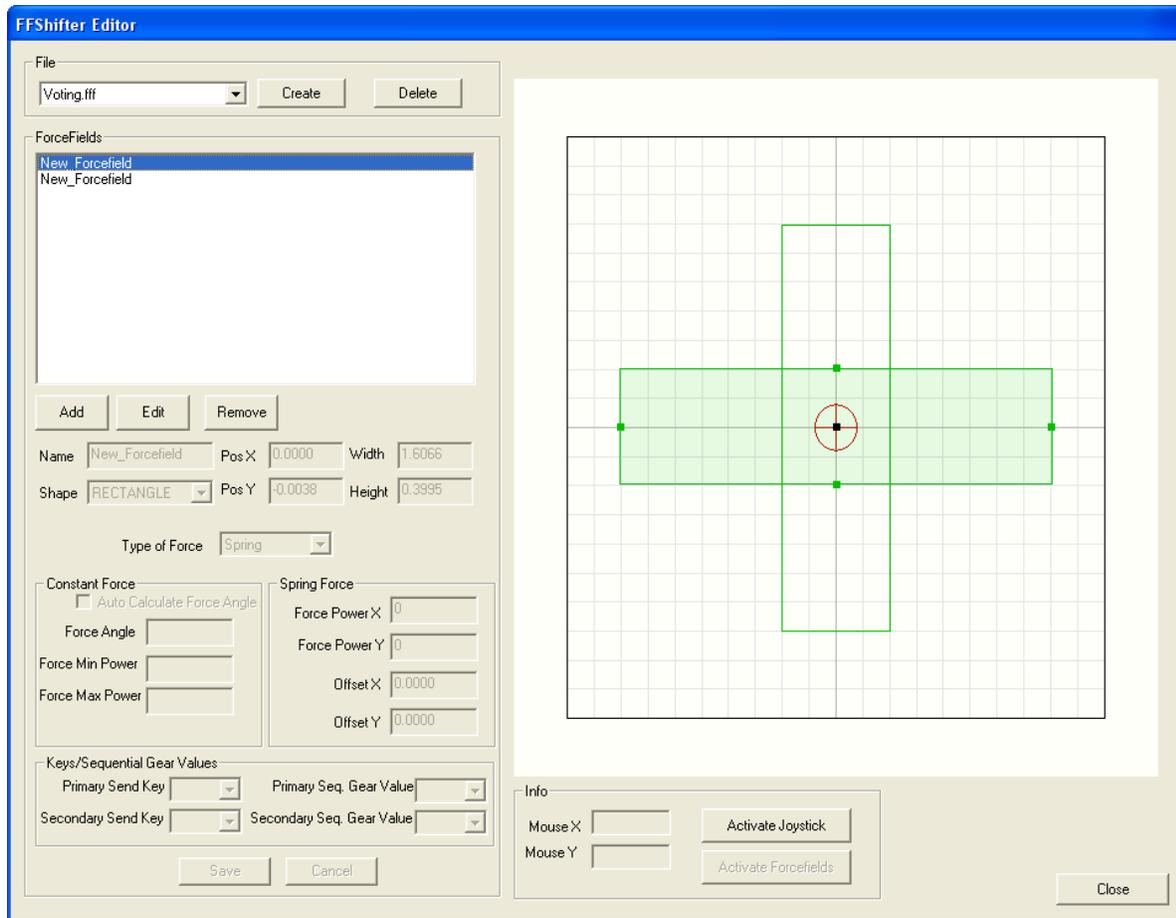


Figure 3.6.1: FFShifter GUI

The above GUI is the force editor of FFShifter. The user can draw the force boxes seen in the grid on the right. The red cursor in the center of the grid shows the current position of the joystick. The various parameters of each force can be manipulated using the text boxes on the left side of the GUI. The Force Power boxes adjust magnitudes of the forces. FFShifter allows for two different kinds of forces: constant forces and spring forces. Constant forces apply the same force to the joystick when the cursor is at any location within the force box. Spring forces apply a force that increases as the cursor moves further within the force box.

3.7 | AutoHotKey

AutoHotKey is free software that lets the user create executable files that simulate mouse clicks and keyboard presses when executed. These executable files are also called macros. This software enabled automatic interaction with FFShifter by running the macros from within the Python script.

3.8 | Python Script

The Python script is used to translate the position of the potentiometer into a corresponding force magnitude. Once this information is processed, the new force is applied to the joystick. In order to interface the joystick to the ballot using Python, a third party library needed to be installed. The team decided to use a library called Pygame. Pygame is a collection of Python modules developed for creating video games with the Python language. Pygame offers input modules for developing software that interacts with mice, keyboards, and joysticks. The joystick module allowed the team to detect the position of the force knob and equate its position with a specific force magnitude. The act of detecting this position is called joystick polling. The analog value polled from the potentiometer was translated to a value between -1 and 1 for each potentiometer position. The force magnitude parameter in FFShifter can range from 0 to 100. In order to translate the polled value into a value that can be implemented in FFShifter, the following method was applied to the value:

$$FFShifter\ Force = 50 * (Polled\ Force + 1)$$

The addition of 1 to the polled force ensured that the minimum FFShifter force was 0 (since the minimum polled force is -1). The multiplication of 50 scaled the range of 0 to 2 up to a range of 0 to 100 to ensure the maximum force corresponded to the true maximum force attainable with FFShifter. FFShifter reads force parameters from a text file. In order to apply the new force to the joystick, the text file needs to be opened, written to, saved, and closed. Additionally, FFShifter needs to be refreshed. To do this, the Python script runs a macro that interacts with the FFShifter GUI to refresh the program. Once the program is refreshed, the new force is applied to the joystick and can be felt by the user.

3.9 | Python Script Pseudocode

The pseudocode for the Python script is as follows.

Poll and store position of force knob

If position of force knob = stored position of force knob

```
{  
    Continue polling force knob position  
}  
Else if position of force knob != stored position of force knob  
{  
    Force = (Position value + 1) * 50  
    Write force to force field text file  
    Refresh FFShifter  
}
```

3.10 | Georgia Tech Ballot

Because the joystick is to be used as a voting device, it is crucial to interface the joystick with a voting ballot. The team's sponsors (MSU's UARC and RCPD) proposed using an HTML ballot that had been developed by a team at Georgia Tech. The ballot does not have an option for joystick input. However, this ballot does have support for a keyboard input. To work around this, the team decided to make the joystick emulate keystrokes. Each of the four directions (left, right, up, down) emulate the keystroke of arrow key in the corresponding direction. This is done using joystick-to-keyboard input conversion software. Figure 3.9.1 below shows the Georgia Tech ballot's many methods of input.

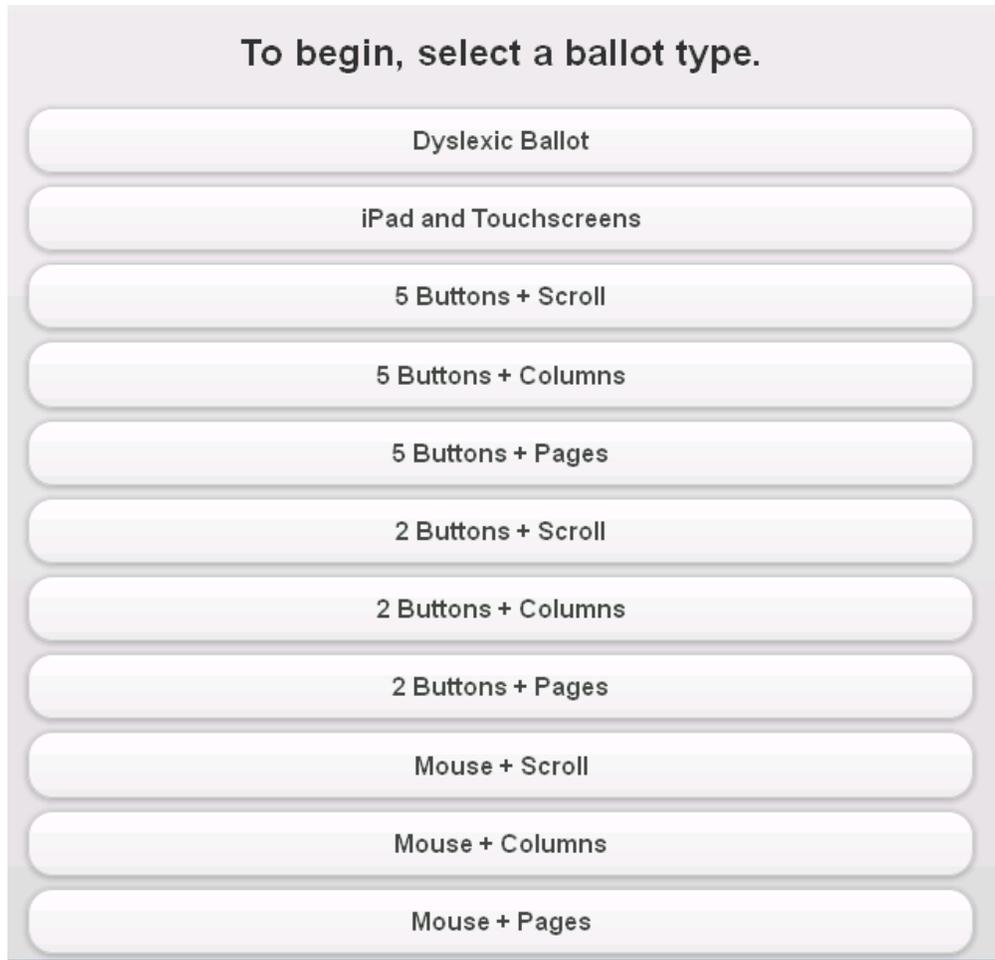


Figure 3.9.1: Georgia Tech Ballot Input Options

The ballot software offers three methods of keyboard input: 5 Buttons + Scroll, 5 Buttons + Columns, and 5 Buttons + Pages. Each method is described below.

5 Buttons + Scroll

Allows the use of 5 keyboard buttons (up arrow, down arrow, left arrow, right arrow, and the enter button). Also, all election candidates are located in a single column that scrolls up and down when the visible bottom of the menu is reached.

5 Buttons + Columns

All election candidates for any given position are located on one page, but they are separated into columns if enough candidates are present.

5 Buttons + Pages

Only contains one column of candidates per page. If one column isn't sufficient to display all of the candidates, the rest will be displayed on other pages.

Any of these input methods can be used with the joystick. The user can use whichever style he or she prefers.

4| Testing

4.1 | Testing the Joystick

Extensive testing of the voting joystick was performed. As a group, the joystick was tested using several pieces of software. The hardware was tested on a basic level using joystick-to-keyboard software and the FFShifter software. The joystick-to-keyboard software was used to test whether the computer read basic input commands sent by the joystick. The team mapped key presses to the joystick axes, potentiometer, and selection button. Using the Windows control panel, the team was able to confirm that the hardware was interacting with the computer properly by viewing the key presses that were present when the joystick inputs were sent. Once basic functionality was confirmed, the team was able to test the force feedback features of the joystick using FFShifter. FFShifter allowed the team to quickly draw force feedback patterns into the GUI. After the force patterns were drawn in, the team was able to confirm the functionality of the DC motors that control the force feedback.

4.2 | Interfacing the Joystick to the HTML Ballot

Once the basic functionality of the joystick was achieved, the team began testing the joystick with the HTML ballot. Because the HTML ballot didn't provide direct support for a joystick, it was imperative to work around this issue. The team decided to use the joystick-to-keyboard software, in conjunction with the HTML ballot, to navigate the ballot with the joystick. The ballot offered three different options for keyboard input support; it was important for the team to evaluate each one and weigh the pros and cons of each. The benefits and disadvantages of each input method are discussed as follows.

5 Buttons + Scroll

The main benefit of this method is that it allows for all the candidates in each group on the ballot to be listed on a single page. This eliminates the confusion caused by separating a candidate group into multiple pages. Due to the fact that the left and right joystick movements change pages, it's much simpler to have all of the candidates in one column to encourage only vertical selection movements.

5 Buttons + Columns

Similar to the previous method, '5 Buttons + Columns' allows the voter to see all candidates of each group on a single page. However, this method is also confusing because left and right inputs on the joystick change pages, not columns. This would result in voters accidentally changing from one candidate group to another instead of changing columns.

5 Buttons + Pages

This input method eliminates the confusion of switching columns that the previous method was plagued by. However, it is difficult to differentiate between switching from one candidate group to another and switching between two columns of candidates within the same group.

After much testing, the team decided that the '5 Buttons + Scroll' method is the most intuitive of the three. With this method, the user can choose any of the three that they deem most comfortable.

4.3 | Testing the Adjustable Force Feedback

The software for adjusting the force feedback of the joystick was written using the software Python. One of the most important commands for testing software in Python is the print function. The print function allowed the team to view the forces applied to the force feedback files before they were actually applied to the joystick. In combination with FFSifter, this allowed the team to fine-tune the force feedback pattern until it was appropriate for voting.

4.4 | Testing with the Resource Center for Persons with Disabilities

The RCPD specializes in researching ways to improve the accessibility of common tasks for individuals with disabilities. The RCPD frequently works with individuals of varying abilities to accommodate their needs. The team was fortunate enough to be able to speak with a couple of individuals who assist in testing within the RCPD organization.



Figure 4.4.1: Branden testing joystick

4.5 | Prototype Evaluation

The team has decided to deem the prototype success. The joystick interfaces quite seamlessly with the ballot, and the user has full control of the magnitude of the force feedback. Regardless, there are still several improvements that could be made to the design.

5 | Final Cost, Schedule, Summary, and Conclusions

5.1 Final Cost

Design team 5 was allocated a \$500 budget by the Electrical and Computer department of the College of Engineering. The team was able to finish the project without exceeding the budget. The list below gives a detailed usage of the budget. Most of parts were ordered from Amazon.com, and a few others were ordered from other vendors. Some commercial joysticks and an Arduino board were purchased since the team was trying to build a couple prototypes to see which of them would work better. These prototypes contained different functions, such as a detent function and force feedback function. Mr. Stephen Blosser, of the RCPD department, provided much mechanical supports to team 5. Because of his help, there were no mechanical costs.

Aside from the electronic components, Team 5 used software to reprogram the modified joystick. The software, such as FFSifter, AutoHotKey, Python Script, etc, was also free of cost to download. Due to the zero cost of the software, the design became inexpensive and achievable. The cost for all the parts ordered totaled \$318.87, within the \$500 budget.

Items	Costs
Logitech Joystick	\$29.99
Thrustmaster T-16000M Flight Stick	\$44.26
Arduino Leonardo	\$24.95
Joystick Shield Kit	\$12.95
Arcade Joystick - Short Handle	\$14.95
Big Dome Push Button - Red	\$9.95
Shipping	\$13.54
Total	\$149.90

Figure 5.1.1: Ordered parts cost

Figure 5.1.2: Cost for final design

Items	Costs
Microsoft Sidewinder Force Feedback 2 Joystick	\$158.99
MSU Golf Ball 3 Pack	\$11.49
Shipping	\$7.49
Total	\$177.97

5.2 | Schedule

Due to the many solutions that could be produced, Team 5 experienced a short setback in the brainstorming stage. It took a period of time for the team to make a decision regarding the final design at the beginning of this semester. Early in this process, the team researched the possibilities of each approach to achieve project's objections. During this process, the team discussed our design solutions with Mr. Stephen Blosser and Dr. Sarah Swierenga weekly. With their help, the team eliminated designs that were too expensive and complicated to achieve within four months or that would not fully satisfy the project objections. Throughout these discussions and the elimination process, the team made a final design decision on week six of the semester. After the final design was introduced, the team moved to the next stage and started researching for a programmable joystick that could fulfill the design requirements. The programmable joystick was delayed in shipping and came two weeks after it was ordered. The team came out with a detailed design while waiting for the ordered joystick. The joystick was getting built at week 9.

By following a precise schedule, the joystick remodeling was finished in advance. During this process, Mr. Blosser assisted with the remodeling the joystick body within two weeks. With his help, this period of joystick remodeling was dramatically shortened. In the last couple of weeks, the team smoothly followed the schedules of testing the performance of the joystick. Throughout the semester, the team was able to finish the project by following the initial schedule.

5.3 | Future Enhancements

The force feedback joystick provides users a new way to vote at an affordable price. Although the performance of the joystick satisfies the project requirements currently, some improvements are necessary in order to make this assistive technology into the commercial market in the future.

Mounting of the joystick to any given object will be one essential consideration. Even though the joystick body was made of iron and is much heavier compared to the joysticks in the market, displacements happened while Branden was testing it. In order to avoid the usage limitation of the joystick by fixing it on a table or a chair, a portable and a removable setup to prevent the displacement of the joystick body while using it will need to be developed.

There are also some enhancements that should be implemented in computer programming. One such enhancement would be an application note written for the joystick function which is available online for users for a free download. This application note could be either a PDF document or software that can be pre-installed at every voting computer. This application note could give users a detailed instruction regarding how to use the joystick to vote efficiently. Audio function could also be added to help users experiencing reading disabilities. Another enhancement would be developing a GUI to display the force amount while adjusting it forthwith. This interface could provide the user a visual force level instead of feeling it by moving the handle. The GUI will give a confirmation if the force levels are accidentally switched.

5.4 | Conclusion

Team 5 was given an objective to build a joystick that could help special needs individuals to vote efficiently. Design requirements included easy accessibility regarding joystick handle movements, an interface that displays the joystick movements and haptic function that alerts users while selecting. The team accomplished this objective by following a Gantt chart made in the beginning of the semester and stayed within the \$500 budget.

The team finished this assigned project with some notable accomplishments. The first key achievement of the team's design was the successful implementation of the force feedback function. Haptic function was added regularly in the previous semester designs. This semester, team 5 not only implemented the force feedback function, but also made the force amount adjustable for users by simply switching a

knob on the joystick body. Moreover, the force amount can be displayed on an HTML file while simultaneously changing it.

Another successful accomplishment is the combination of the arrow key functionality with the joystick movements. By making the joystick function with arrow keys on a keyboard, the mouse was limited as it could only be moved vertically. This way, individuals with a wide variety of motor skill capabilities would not be limited to move the joystick in one specific direction. Instead, they can move the joystick handle in any direction forward or backward to move the mouse vertically line by line. This function made the voting process easier and more efficient. Other functions, such as a detented design, also optimized the performance of the joystick.

The performance of the joystick has met the requirement for this project; however, with a larger budget and more allotted time, the improvements described in the last section can be explored in the future.

Appendix | 1

Appendix | 1.1 **Yangyi Chen**

Yangyi Chen was the lab coordinator for ECE480 Design Team 5. Her role included but was not limited to ordering the parts needed while building and testing the joystick, and keeping the lab equipment organized.

On the electrical side of this project, Yangyi mainly focused on the hardware aspects of the team's design. During the semester, Yangyi devoted a lot of energy in researching the configuration of the force feedback system used in the design, such as motors, force transducers and FSRs. Initially, the team had several design to approach the objective. Her research results played an essential role in narrowing down the design selections and helped the team to choose the most efficient one. Moreover, Yangyi pursued several designs to build a joystick with achieving force feedback function before the team made the final decision about the design.

Yangyi also participated the testing of the joystick outputs. She worked with Graham Pence and Stephen Blosser, the project sponsor, to go through the internal circuit board of the joystick. After a careful observation, a logical and theoretical analysis, and several times of testing, they found the way to ease the mistakenly output of the selection button.

Another key objective of Yangyi's contribution was designing the appearance of the joystick housing. This process involves fully consideration of the rearrangement of the electrical components inside the joystick body, precisely measurement of distance between each nail hole, and carefully installation of the electrical components. Through discussing and sharing ideas with Stephen Blosser, the final design of the joystick hosing came out and it was perfectly fulfilled the team's needs.



Appendix | 1.2 **Tyler Dennis**

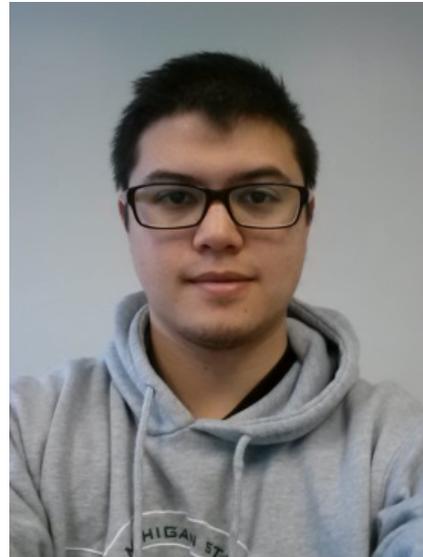
Tyler's worked on the project as a software developer. He was responsible for researching, writing, and testing the software that is responsible for interfacing the joystick with the voting ballot. Although his background is mainly in electrical engineering, he took two semesters of programming courses at MSU. His experience with the Python programming language was invaluable in creating the main software for the joystick. Tyler also worked with Stephen Blosser to install the knob on the joystick that the software interacted with to change the force feedback of the joystick.



Once the software was developed and functional, Tyler worked with Behdad Rashidian to test and refine the software. The two worked together to create a force feedback control that was responsive and easy to use. This involved many hours software development and debugging.

Appendix | 1.3 Graham Pence

Throughout the design process, Graham Pence worked on the research, testing, and the implementation of hardware components on the joystick. Initially, there were various routes that the design for this project could be executed. The teams sponsor, Stephen Blosser was very flexible in any mechanical or electrical portions of the design as long as the requirements were taken care of. Graham researched information on everyday joysticks on the market, which varied from Hall Effect (magnetic) or haptic feedback joysticks. He



understood the complete makeup of a joystick as an application note was written on how to build one, which is located on the team’s website. Graham concluded that a joystick with the features: force feedback, automatic return to center, and usb interface were required to complete the requirements of the sponsor.

Once a design was implemented, he did some preliminary testing on the motors to determine a suitable operation voltage. There were also an addition to an exterior button and force adjustable potentiometer added requiring a lot of trouble shooting to design. The exterior button was automatically being triggered without activation. Using a digital multi-meter to test the voltage at different areas of the circuit, it was required that the addition of 10k Ω resistor would bring down unwanted current. Graham also determined that a 17.5 k Ω potentiometer had a high enough resistance to bring down the current in the circuit and allow the knob to work efficiently. Lastly he helped design the dimensions and design for the exterior housing for all the required electrical components.

Appendix | 1.4 **Behdad Rashidian**

Throughout the design process, Behdad played a large role on the software side of the development of the joystick. He was responsible for developing an intuitive graphic user interface for the joystick. His background in programming with C++, python, Visual Studio and java was valuable and was useful for creating software interface that will eventually be the most important part of the software design of this project. This involved large amount of research and experimentation in order to get it to work.



Behdad's attention to the little details in creating the interface was vital and he made sure that the program works perfectly with the least amount of code. Moreover he paid attention to the runtime of the program and chose algorithms that would work best. Most of the code that he wrote had $O(n)$ runtime. This helped the software to work more efficiently. While spending long hours developing the GUI, Behdad also assisted Tyler to test the programs and made sure that knob (force feedback potentiometer) works flawlessly with the other parts of the joystick.

He played a large role in developing the H-Bridge circuit that works with motors directions and functionality of the force feedback. His researches and calculations on H-Bridge circuit was valuable and helped the circuit to be done in a short time.

Appendix | 1.5 Joy Yang

Joy Yang was the presentation prep of Team 5. Her role included not was not limited to the designing and developing of all power points, presentations, posters, etc... for Team 5.

Joy Yang's technical role in the project mainly focused on the mechanical designing aspect of the team's design. As a member of the team, Joy focused on researching the many types of joysticks that could be used for the implementation of the project. Working with Yangyi Chen, Joy helped the team focus on a few particular joysticks to use for the project.



Joy diligently worked with the rest of the team and Stephen Blosser with the software implementation of the team.

Regarding the team's mechanical design, Joy and Yangyi Chen carefully inspected the electrical components of the interior of the joystick in order to help design a hard case housing for the joystick. Each square inch was measured with care to ensure that the joystick and the handle would have sufficient space to move effectively, while also effectively minimizing surface area for the packaging of the housing.

Appendix | 2

Works Cited:

"Accessible Voting Technology Initiative Designing Accessible Elections." *Accessible Voting Technology Initiative RSS*. N.p., n.d. Web. <<http://elections.itif.org/projects/grants/round-2-funded-projects/michigan-state-university/>>.

"AutoHotKey." *AutoHotKey*. N.p., n.d. Web. <<http://www.autohotkey.com/>>.

"DirectInput." *MSDN*. N.p., n.d. Web. <<http://msdn.microsoft.com/en-us/library/windows/desktop/ee416842%28v=vs.85%29.aspx>>.

"H Bridge." *Wikipedia*. Wikimedia Foundation, 19 Apr. 2013. Web. <http://en.wikipedia.org/wiki/H_bridge>.

"Haptic Technology." *Wikipedia*. Wikimedia Foundation, 21 Apr. 2013. Web. <http://en.wikipedia.org/wiki/Haptic_technology>.

"How Joysticks Work." *HowStuffWorks*. N.p., n.d. Web. <<http://www.howstuffworks.com/joystick8.htm>>.

"JoyToKey English Version." *JoyToKey English Version*. N.p., n.d. Web. <[http://www.electracode.com/4/joy2key/JoyToKey English Version.htm](http://www.electracode.com/4/joy2key/JoyToKey%20English%20Version.htm)>.

Swierenga, S. J., & Pierce, G. L. "Accessible Voting Systems Usability Measures." Presentation at the International Technology and Persons with Disabilities Conference. Los Angeles, CA: CSU Northridge, 28 Feb. 2013.

"Using Force Feedback to Enhance Human Performance in Graphical User Interfaces." *Using Force Feedback to Enhance Human Performance in Graphical*. N.p., n.d. Web. <http://www.sigchi.org/chi96/proceedings/shortpap/Rosenberg/rl_txt.htm>.

Appendix | 3

Appendix | 3.1

Code:

```
import pygame
```

```
import os
```

```
import time
```

```
#Detects any joysticks connected via USB and initializes them
```

```
pygame.joystick.init()
```

```
joysticks = [pygame.joystick.Joystick(x) for x in range(pygame.joystick.get_count())]
```

```
joystick = pygame.joystick.Joystick(0)
```

```
joystick.init()
```

```
pygame.display.init()
```

```
#Creates arrays that are used to detect edited lines in force files
```

```
force_lines = (32,33,65,66)
```

```
x_line = (32,65)
```

```
#Initializes all of the complementary programs for the joystick
```

```
def start():
```

```
os.chdir('C:\\Documents and Settings\\admin\\Desktop\\Voting Joystick\\Force Feedback')
```

```
    os.startfile('FFShifter.exe')
```

```
    os.startfile('Ballot.htm')
```

```
    os.chdir('C:\\Documents and Settings\\admin\\Desktop\\Voting Joystick\\Force  
Feedback\\jtk374en (1)\\jtk374en')
```

```
    os.startfile('JoyToKey.exe')
```

```
    os.startfile('J2KMin.exe')
```

```
    os.chdir('C:\\Documents and Settings\\admin\\Desktop\\Voting Joystick\\Force  
Feedback\\macros')
```

```
    os.startfile('FFShifterSetup.exe')
```

```
os.chdir('C:\\Documents and Settings\\admin\\Desktop\\Voting Joystick\\Force
Feedback\\forcefields')
os.startfile('MouseMove.exe')
```

#Writes new forces to the force text file in order to change force feedback

```
def write_forces(force):
    os.chdir('C:\\Documents and Settings\\admin\\Desktop\\Voting Joystick\\Force
    Feedback\\forcefields')
    force = int((force + 1)*5000)
    i = 1
    ls = []
    new_file = open('Voting.FFF','w')
    old_file = open('Voting2.FFF','r')
    for line in old_file.readlines():
        if i not in force_lines:
            new_file.write(line)
            i = i + 1

        else:
            if i not in x_line:
                lst = line.split(' ')
                lst[2] = 'Y=' + str(force) + '\n'
                new_line = ''.join(lst)
                new_file.write(new_line)
                i = i + 1

            else:
                lst = line.split(' ')
                lst[2] = 'X=' + str(force) + '\n'
                new_line = ''.join(lst)
                new_file.write(new_line)
                i = i + 1
```

```
os.system("FFShifterToggle.exe")
old_file.close()
new_file.close()
```

#Calls the start function in order to initiate the program

```
start()
```

```
os.startfile('Click.exe')
```

```
while 1:
```

#Polls the joystick in order to detect the potentiometer position

```
pygame.event.pump()
```

```
force = joystick.get_axis(2)
```

```
while force == joystick.get_axis(2):
```

```
    pygame.event.pump()
```

#Calls the write_forces function to change the force feedback text file

```
write_forces(force)
```

```
os.startfile('ForceRefresh.exe')
```

```
os.startfile('ActivateBallot.exe')
```

Appendix | 3.2

