**Project Number: 21311**

Mind Controlled Wheelchair

|  |  |
| --- | --- |
| **Cheyenne Dailey**  Computer Engineering | **Sidney Davis**  Electrical Engineer |

|  |  |
| --- | --- |
| **Ryan Radicello**  Computer Engineering | **Dawson Hilk**  Electrical Engineer |

# Abstract

This technical paper covers the integration of sensors onto an electric wheelchair to make it semi-autonomous, with minimal input from a device suited to a disabled individual. The team describes their use of the Intel RealSense 435 Depth Camera and T265 Tracking Camera, and how their initial plan to use a Raspberry Pi to perform Simultaneous Localization and Mapping (SLAM) evolved into using a laptop operating Linux to interpret basic sensor measurements. Furthermore, the process to use potentiometers to control the wheelchair’s joystick is described, along with the installation of two emergency stop buttons on the chair.

# Background

The Neurotechnology Exploration Team (NXT) Research Group at RIT is dedicated to using cybernetic techniques and physiological signals to create devices for those who rely on alternative methods. Adam Del Rosso is the project manager for an NXT initiative to create a wheelchair that responds to commands sent by a Brain Computer Interface (BCI). Upon realizing the need for more automation to relieve the effort required by the user, the wheelchair portion of the project was handed off as a Multidisciplinary Senior Design (MSD) project.

The MSD team intends to modify an electric wheelchair to accept limited commands from a BCI, as well as any input device best suited to the user’s disability. Furthermore, to compensate for the lack of command / user control, a method of machine vision has to be implemented, so the control over the heading of the wheelchair is a tandem effort between necessary decisions from the user and observing common sense decisions from the environment.

To stay within the scope of a 1-year project, NXT and the MSD team agreed that this device will only be tested indoors, on 1 floor of 1 building, and that limited setup instructions could be provided to the autonomous wheelchair software before beginning a run.

# Description of Design

The design of this project was broken down into two main sides, the software side, consisting of the camera control, communication, and mapping; and the hardware side, consisting of the joystick control and the emergency stop system.

The cost of the project was largely determined by the RealSense cameras. This cost was supplemented by the NXT budget as it costs a bit over three-hundred dollars for the pair. The MSD budget costs were also a little over three-hundred dollars for all other expenses. These included a wheelchair emulator to manually push and be able to test the system, a Raspberry Pi 4 and accessories, relays to handle the motor power in the emergency stop, cable extenders, printed circuit boards and components to control the joystick, and a bag to keep the laptop easily attached to the chair. The wheelchair itself was already owned and supplied by NXT.

## Software Design

When starting with the project, encouragement from our customer led to the decision to use Intel RealSense’s Depth Camera, D435, and Tracking Camera, T265, with their provided RealSense library. The team was under the assumption that these cameras were simple and easy to use with any operating system, ready for integration with full SLAM capabilities. This led most of the project with the goal to leverage SLAM and use it to make the wheelchair autonomous.

Under the above assumptions about the RealSense cameras and the availability of SLAM, the initial design was to connect with a Raspberry Pi 4 to have the processing power to handle both cameras and have the small form factor to integrate onto the wheelchair. Through much trial, error, and research, it was discovered that several assumptions were incorrect. Through testing and research, it was found that the T265 was no longer supported by RealSense and there was a critical error when interacting with a Pi that caused the communication with the camera to be cut off. After much deliberation it was decided that the best course of action was to transition to a laptop that was running a Linux OS, specifically Ubuntu 18.04, which was confirmed to connect to the T265 correctly.

With the processor adjustment, focus shifted to working with the cameras. The Intel RealSense D435 depth camera uses an infrared system to measure the distances. To satisfy the customer requirement of avoiding obstacles in the path of the chair in motion the pixels of the depth camera were divided into a grid of 16 segments. This allowed the system to know more accurately where an object was and what direction to turn the chair. Through this grid the steering control is determined via a proportional–integral–derivative controller (PID) and forward speed is based off of the steering amount. Both values are then sent to the Arduino to mechanically control the chair.

For navigation, the initial plan was to use the assumed SLAM capabilities stated to be provided by the Intel RealSense T265 Tracking Camera. This was set as a customer requirement from the very beginning. After thorough research and testing, it was found that given the time available and the experience/knowledge of the team that SLAM was unobtainable if a working product was to be delivered by the end of the term. Through much deliberation and discussion with the customer, it was decided to go with a node system on key locations or decision points with a weighted path system. From this node system, a start node and end node were given to a Dijkstra Algorithm to calculate the shortest path from start to finish and return it for execution. An adjacency list with weights is stored and read into the program on startup. This list allows Dijkstra's algorithm to find the best path to take and then direct the chair down said path to reach its destination. The system utilizes the T265 inertial measurement unit (IMU) to know how far the chair has moved and therefore whether it has reached a node and must change direction or not.

While cameras and mapping were essential, there was a need for communication protocols between the Arduino used for joystick control and the BCI client, designed by NXT, responsible for user input. The Arduino was designed to connect to a serial port on the laptop where messages were read and written to communicate. The server, created on the laptop, would write to the serial file, giving inputs for the X and Y magnitude from -100 to 100 that would be read by the Arduino and translated appropriately to control the joystick. The Arduino returned an acknowledgement message back to the main server stating whether the message was received successfully. The connection with the BCI was separate from the serial server, using ZeroMQ, an online concurrency framework that allows for socket connections. The zmq server was designed to create a socket with a PAIR configuration, allowing for uninterrupted communication between two sides. The server had one side of this, and the NXT team devised a client design that would connect via ethernet. This socket connection allowed for the transfer of JSON styled messages, with key/value pairs to be sent between each system. The BCI client would send a confirmation message that a connection was made with the server followed by a message stating the user’s chosen destination.

## Hardware Design

To control the motion of the wheelchair, the method used by a traditional user had to be fully understood. The joystick and accompanying electronics were removed from the arm of the provided wheelchair to learn what types of signals were sent from the joystick to the controller.

By using jumper wires to connect the joystick to a breadboard, and the breadboard to the controller, the voltages of each of the 8 pins were available to be probed. A labeled ground pin was found elsewhere on the controller to use as a reference, and then the voltage was recorded for the joystick in its rest position, forward position, reverse position, left position, and right position. As a result of this testing, the following table, and correlating functionalities was determined:

*Table 1: Original Joystick Measurements and Functionality*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pin | Rest (V) | FWD (V) | RVS (V) | Left (V) | Right (V) | Function |
| 1 | 2.5 | 2.5 | 2.5 | 2.5 | 2.5 | VDD |
| 2 | 0 | 0 | 0 | -1 | 1 | X |
| 3 | -2.5 | -2.5 | -2.5 | -2.5 | -2.5 | VSS |
| 4 | 0 | 1 | -1 | 0 | 0 | Y |
| 5 | 0 | 1 | -1 | 0 | 0 | Y |
| 6 | 0 | 0 | 0 | 0 | 0 | Ground |
| 7 | 0 | 0 | 0 | -1 | 1 | X |
| 8 | 0 | 0 | 0 | 0 | 0 | N/C |

The difference between ground and N/C was determined by probing the resistance compared to the reference point on the controller board. Ground reported as a short circuit, while N/C reported as an open circuit. Furthermore, the connectivity between the two X pins and two Y pins and determining whether they could be simplified to sharing the same pin, was tested experimentally using the first joystick emulator, described in a later portion of this document.

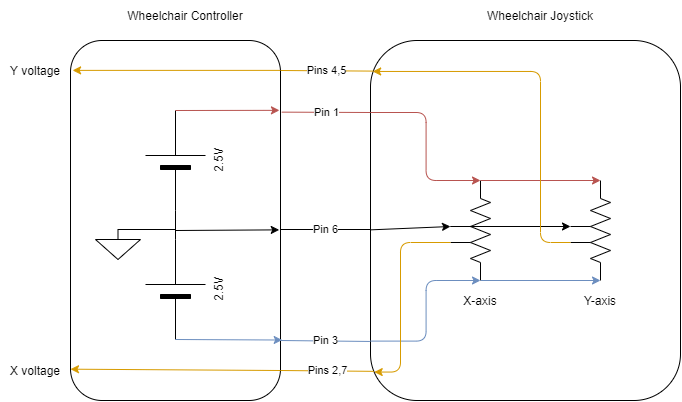
Steering the chair was important but so was stopping the chair in the case of an emergency, leading to the creation of the emergency stop design. The plan from the beginning was to cut off power to the motors only when either one of the buttons were pressed. Before we did this, we needed to figure out how to mount the buttons to the chair and come up with a schematic for how the components were to be connected. Luckily, there was a rail system on the sides of the chair that allowed for a button to be screwed to. The second button placement was talked about a little more. It was decided that the most convenient place for it to go was on the back right of the armrest. There was a 6m screw hole that we could use to fit the button with. This button had to fit an Arduino in it because it was the best place for it. It as 3d printed with these variables in mind. As for creating the schematic and how we planned to stop the chair. We decided on using a NO DPST (Normally Open Double Pole Single Throw) Relay in between the motor and the motor controller. For deciding to use the relay. It would allow us to use a smaller voltage and current to control the motor because without the relay, the buttons we had did not have a high enough amperage rating to match the motor. Buttons that could match that high rating were extremely expensive. Therefore, a 12V coil relay was chosen. We used a 24V to 12V converter from the battery, which is 24V, to get the relay coil voltage. The plan was to splice the power wire to the motor and hook that into the relay, but that would be a permanent fixture for the chair. Instead, an adapter of sorts was made instead that used quick disconnects to connect the wires that do not change into their original connections, and then also used quick disconnects to connect the motor to the relay and then the relay to the controller. This all fit nicely where the motors connected to the controller, and the relay was velcroed in a safe spot-on top of the controller.

# Supporting Feasibility Evidence

## Software Feasibility

To complete the mapping an adjacency list was created that correlated node numbers to locations and keep track of the distances between each that connected to another node. Dijkstra’s algorithm is used to find the best path from the start node to the end node by traversing the graph and tells the system how far to travel and to which node ID in the graph. The system then tells the Arduino to push the chair forward and the T265 IMU will determine when the target node has been reached. While this happens the D435 starts measuring in front of the chair with its infrared sensors to determine if there is something in the path, and if so, how to turn around the object and still make it to the target node. Each of these components were tested via walking through the mapped area with the cameras before doing the same with all connected. The map and pathfinding were validated by selecting locations and following the path given to be sure it matched what was expected and reached the correct location. The object detection was tested by doing a much shorter distance and watching a display of where the system saw objects in the grid created by the depth camera. The turn control was connected to the Arduino and LEDs were wired to display the turn and forward strength that the system was directing to the chair. All of these were then run simultaneously and walked through the mapped area to verify correct pathing, object detection, and steering control.

## Hardware Feasibility



*Figure 1: Observed Schematic of the existing Wheelchair Joystick/Controller Relationship*

Based on the measurements in Table 1, the schematic above was derived. The joystick consists of two center-tapped potentiometers, and the wipers on the potentiometers are determined by the position of the joystick. A corresponding voltage is returned to the controller that is providing the positive and negative rails to both elements. Using this design, an equivalent circuit to the joystick was implemented on a breadboard using two 10k potentiometers and 2 equal resistors to make a voltage divider between the two rails to represent the center tap.

When connecting this breadboard to the wheelchair controller, the wheelchair screen showed no errors, and upon twisting the potentiometers the chair began to move, validating the schematic in Figure 1. The motors turned slower than anticipated, but this was later determined to be due to the position of the seat of the chair, which lowers the max velocity if the user is not reclined back past an internal threshold.

The overall E stop system had to be compact with very minimal wire exposure. This was to be something that you could hardly tell was there, except for the two buttons. The left rail button used four hex nuts that slid into the rails. The bottom of the button case was screwed into those nuts at the desirable position along the rail. The second button was 3d printed to fit the Arduino along with the top half of the button case which included the button part. The case had a whole in the bottom to be screwed to the armrest. It also had holes for the estop wires coming in and out, a hole for the wheelchair switch that controls whether the wheelchair is in manual or semi-autonomous mode, and holes for the Arduino wires. This all fit in a box that is about 4in x 4in x 4 in. The relay coil voltage required 12V, but the chair gave off 24V. To get this 12V, a 24V to 12V converter was needed. There was an open positive and negative terminal on the controller that the converter was hooked into to change the voltage to 12V. From this converter, the wires went to both E-stop buttons in series, as to shut off the power when a button was pressed, and then into the relay and back to the converter to complete the circuit. This was tested once these parts were implemented to see if the relay was switching on and off once a button or both buttons were pressed. The wheelchair previously had two black connecters from the motor that supplied power from the controller to the motor. These were disconnected and instead between the two, we used quick disconnects to connect the two again, but instead the power wires were diverted into the NO DPST relay and then back. This allowed the power to the motors to be cut as soon as an E-stop button was pressed. The wires and disconnects were able to fit under the plastic cover of the wheelchair and thus were hidden very well. The relay was velcroed securely to the top of the controller. The DC converter was able to be placed inside an existing component on the chair.

# Results, conclusion, and recommendations

This project initially had a budget of $1,000 total. Half of that came from RIT MSD funds, and the other half came from NXT’s budget. In total about $700 of it was spent. The RealSense camera bundle (tracking and depth camera) came from the NXT budget and all other purchases came from the MSD budget.

## Software Results

Mapping proved to be a success with the project. Tests with the emulator proved that the maps found the shortest path and that the nodes could be recognized based on distances. The one caveat to the design is the manual measurement of the distances using the T265 IMU. This created a potential point of failure as it was not always guaranteed that the chair would start in the same location. Keeping consistent for all our testing however, proved that the mapping solution to be accurate. The path planning for the system was handled by traversing the map using Dijkstra’s shortest path algorithm. This allowed for the system to know how to get from the starting point to the ending point with the least total distance covered, not just the least distance from the current node to the next map node. The adjacency list to create the map and tell the chair what direction to turn at each intersection based on the does before and after it were all successful. The chair was able to navigate following the mapped path and tell the user what direction to turn the chair. As the time on the project ran out, we would have liked to have been able to manipulate the gyroscope or pose data of the T265 to automatically turn the chair to face the correct direction.

For steering, thresholding was used with the location of an object in the depth frame along with the PID that was created. During testing and the demonstration, it was found that the design could drive the chair down the intended space avoiding most objects; however, it was not the cleanest driving. The PID designed was a quick method to get the baseline needed for correcting once a turn was made. This design did not receive the proper testing needed prior to demonstration that was required to have smooth steering. The result was over-correction and bouncing between walls. While this still achieved the drive goal of unassisted movement, an issue was run into in different hallways when doorways or corners were reached. This was either due to the PID design or to the thresholding done based on the depth camera. There was not sufficient time remaining to flush out and resolve the problem.

Testing was completed to confirm that each connection, serial and socket, could be made with messages successfully passed between the devices. The first test was with the serial connection for the Arduino. Messages containing x direction and y direction magnitudes were sent from the main program to the Arduino and translated into the proper joystick voltages. Testing proved successful by a handoff test where the program sent inputs to the Arduino without user assistance. One caveat found with the Arduino was that when it was first connected, there would be a hard fault that would prevent the program from running. After terminating the program and restarting it, this was no longer an issue.

The BCI socket connection was tested next, using a simple python client program that would connect to the ZeroMQ server, and pass the two messages. Testing was done to ensure that the correct messages were received, and that the information could be parsed from the JSON message format. One issue that occurred with the socket design was creating a non-blocking receive that could allow for the user to issue stop and start commands whenever they pleased. Given the remaining time and the debugging necessary to resolve this issue, it was accepted that allowing for these messages would not be achievable.

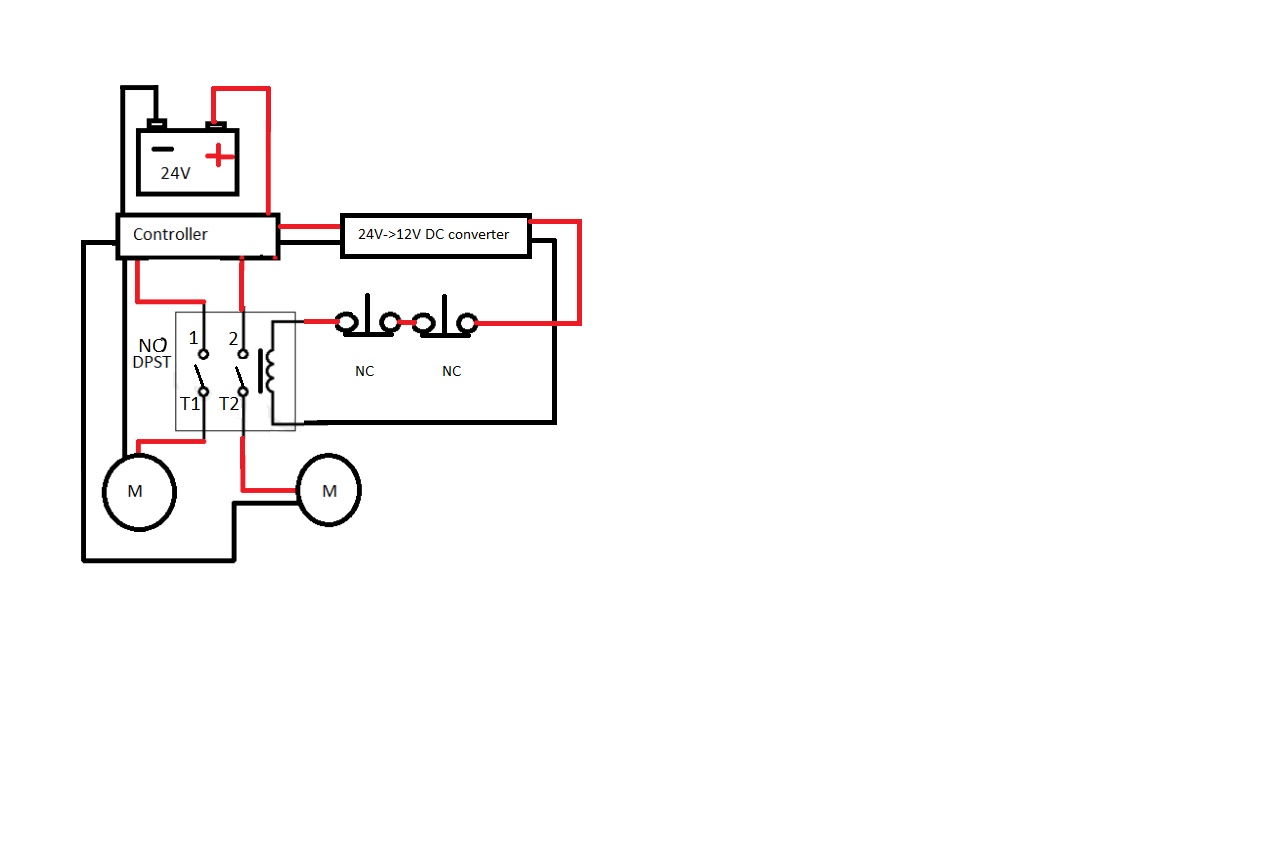
## Hardware Results

|  |  |
| --- | --- |
| *Figure 2: Design out of Case* | *Figure 3: Design in Case on Wheelchair Arm* |

As a last resort, when V2 was producing the correct voltages but the chair refused to accept it as input, I fell back on the physical potentiometers I used to verify the joystick implementation in section 2.3. Connecting these potentiometers to a protoboard with resistors acting as the center-taps mimics the same circuitry as before, so attaching servos to the heads of the potentiometers provides electrical control from the Arduino, making the system go from electrical, to mechanical, back to electrical. The servos, potentiometers, and protoboard are given a case that can screw into the right arm of the wheelchair, midway between the pre-existing Arduino case and the wheelchair joystick.

The servos do not always return exactly to the center position, depending on whether it is rotating clockwise towards the center vs rotating counterclockwise towards the center. This causes an error on boot for the wheelchair if not perfectly centered, but a quick fix is to set the potentiometers off center in the Arduino Code on boot, return to center, and then try powering on the wheelchair. If this does not work, set the potentiometers to rotate in the opposite directions on boot, then return to center.

For more information about the Joystick Controller PCB, please refer to “P21311 Joystick Controller Documentation.docx”



*Figure 4: Emergency Stop Schematic*

There were some things on the estop system that did not initially work as planned. The converter that was on the chair from the last team, did not work at all. This was replaced with a new DC-DC converter. The original design of the system included a NC DPST relay so that the motors would always be connected unless the E-stop buttons were hit in which case the relay would get power and flip open the Motor connections. After careful deliberations with the team, for safety purposes and practice, it was changed to be a NO DPST relay. It will now only run when power is supplied to the relay and so when a button is hit, the relay coil will lose voltage and flip open the motor connections. This electrical schematic for the final design and prototype is shown in Fig. 4. Once all the components were in place, which are the two buttons, the DC converter, and the relay adapter, the chair E-stop system was tested repeatedly. We tested for how abrupt the stop would be at different stops and how far the chair rolled after a button was pressed. These values were under what our Engineering requirements were. It was therefore labeled as a success. There were some small things the team would do differently when creating this system. The first is always having someone double checking the cad prints. Some prints were not done exactly right the first time due to some stupid mistakes. This could have saved a lot of time if done right the first time. The only other thing that would be done differently are the quick disconnects connecting the motors, relay, and battery. Some of these disconnects were “Macgyvered'' to fit the overly sized pins on the chair’s motor connection port. The quick disconnects fit well currently, but they are probably the only type of their kind in their current state. We would have liked to do a little more digging on the internet to find the right sized quick disconnects. The future work of this subsystem can vary wildly or not at all. This entire system is designed so that it can be replaced or removed if someone deems it necessary. Nothing was done so that it was permanently attached to the chair.

# Acknowledgments

Thank you, to our customer, Adam Del Rosso from the Neurotechnological Exploration Team at RIT. NXT provided an electric wheelchair, and sponsored purchasing components critical to the completion of the objective. Adam also dedicated an impressive weekly commitment to meeting with the team and offering his technical expertise and experience towards our common goals.

Thank you, to our guide, Charles Hacker from L3 Harris. He assisted the team with interpreting project management requirements and understanding the generic systems in place to help us with our current project, as well as how to apply the same systems towards our future projects. He also proved valiant at the generation and research of alternative ideas whenever we discovered our methods were not going to work.

Thank you to the Multi-Disciplinary Robotics Club (MDRC) at RIT, for allowing us to use your club room and tools as a workspace. Having a private space with dedicated tools and equipment allowed for a great boost in efficiency while prototyping.

Thank you to Multidisciplinary Senior Design, for supplementing our funding and providing us with the resources, modules, and feedback to organize and complete our project.

Thank you to Carlos Barrios, for your advice on integrated circuit selection.