**Proceedings of the 2025 Virginia Tech**

**Mechanical Engineering Senior Capstone Exposition**

**VTMECAP-2025**

**May 6, 2025, Blacksburg, Virginia, USA**

**VTMECAP-2025-205**

Assistive Robotic Aid for People with Duchenne Muscular DystrophY

**Brooke Harrington1, Cavan Moriarty1, Omar Abuljobain1, Justin Lee1, Andrew Baum1**

**1**Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA

Abstract

DMD is a genetic disorder commonly found in men that deteriorates muscle tissue as they age, which progressively reduces mobility and requires those affected to use motorized wheelchairs for daily travel [1]. In addition to motorized wheelchairs, young adults with DMD often need devices that assist with upper body mobility. People with DMD would value an affordable, easy to control assistive robot that mounts on their wheelchair and provides two high value functions: operating elevator and handicap door buttons and bringing a drink to the user’s face.

To provide the two high value functions, team 205 utilized an off-the-shelf robotic arm, an Adafruit Feather V2 microcontroller, a joystick, buttons, switches, and other key electronics to develop a prototype for an affordable assistive robotic. Additionally, common materials and manufacturing methods were used such as 3-D printing, CNC milling, and laser cutters, to manufacture a mounting/containment structure for the assistive device. A prototype was fabricated, tested, and revised to ensure the device could accomplish the high value functions.

Team testing showed that the prototype met the requirements developed while conceptualizing a design, as well as user requirements such as size, safety, and functionality. Additionally, the prototype is able to complete its functions on instruments complaint with standards (minimum door width, button height requirements). End user testing showed that the prototype could complete the two high value functions, while being easier to control than other options on the market. Though the prototype was able to complete the two high value functions, improvements can be made that elevate user experience, ease of operation of the device, and sophistication of device functionality.

Nomenclature

3D Three Dimensional

ADA Americans with Disabilities Act

ANSYS Analysis System

CNC Computer Numerical Control

DMD Duchenne Muscular Dystrophy

DOF Degrees of Freedom

ESP Electronic Stability Program

FBD Free Body Diagram

FEA Finite Element Analysis

FOS Factor of Safety

GPIO General Purpose Input-Output

JSON JavaScript Object Notation

LED Light Emitting Diode

PCB Printable Circuit Board

PVC Polyvinyl Chloride

PPMD Parent Project Muscular Dystrophy

ROM Range of Motion

SPST Single Pole Single Throw

TR Technical Requirement

UI User Interface

1. INTRODUCTION

PPMD is a nonprofit organization that has been leading the fight against DMD since 1994, whose leading goal has been to ensure afflicted families have access to the latest assistive technologies through investment in research and development of pharmaceuticals and medical devices [1]. To help reach this goal, PPMD has sponsored senior design team 205 to develop an affordable and easy-to-use robotic aid by modifying a low-cost robotic arm and mounting it to commonly used wheelchairs. Although assistive devices may take various forms, the most optimal design was determined to be a robotic arm because of its ability to perform two high value functions: pushing an elevator/handicap button and retrieving a drink. Current robotic arms on the market that directly address DMD are often too expensive, some exceeding sixty-thousand dollars, and are difficult to control with more than six DOF [2]. Therefore, team 205 will provide users with DMD the key functions they require at a substantially lower cost with a less complex method of operation via a lower DOF system.

To simplify the design process of an assistive robotic device that could achieve the high value functions, the system was divided into three main subsystems: mechanical, electronic, and software. The mechanical subsystem included designing a mounting assembly that attaches the robotic system to a Permobil wheelchair, the most common wheelchair used in the DMD community. Additionally, the mechanical subsystem involved designing a case for sensitive electronics that also acted as a platform for the low-cost robotic arm and UI. The electronics subsystem consists of the RoArm’s PCB, the Adafruit Feather and protoboard, the UI (switches, buttons, joystick and LED), and the battery. When assembled, these components allow power to be distributed and used for control logic of the robotic arm. Lastly, the software subsystem found on the Feather reads signals from the UI and completed all necessary logic to update the coordinates, which were then formatted and sent to the RoArm PCB to be used for motor control. The software was also used to optimize control of the robotic arm and provide programmable functions desired by the end user.

Additionally, the team had to meet specific requirements determined by members from PPMD. The first requirement stated that the motorized wheelchair with the robotic system attached shall fit through an ADA compliant door, which has a minimum clearance width of thirty-two inches [3]. Second, the attached robotic system shall be able to press an ADA compliant elevator button and handicap door button, which are positioned between 35 and 48 in from the ground [4]. Third, the robotic arm and case shall be water resistant in the event that the user is caught in a rainstorm. Lastly, the system shall be easily attached and detached from the motorized wheelchair for transportation.

1. **MATERIALS AND METHODS**
   1. **Mechanical**
      1. **Manufacturing Processes**

Multiple manufacturing processes were required to fabricate the mounting and robotic system. For mounting, both the sliding and connecting plate were cut from an aluminum panel by a plasma cutter and CNC mill. Alternatively, acrylic panels were cut into their respective shapes using a laser cutter. Lastly, 3D-printing with ABS filament was used to create the required parts with varying infills and purposes.

3D-printed parts include the corner frame, battery case and cover, waterproofing well, cylindrical guide, end effector cover, pin, stopping plate, and armrest attachment. The battery case encapsulates the battery, which maintains its position at the lower half of the electronics housing by bolting to the bottom acrylic panel. The battery cover, waterproofing well, cylindrical guide, and end effector cover serve to waterproof exposed electronics by sheltering them or guiding water away from the electronics. The stopping plate was glued to the base servo-plate and provided a guide for a standoff. The standoff, bolted into the top of the electronics case, stopped the arm from rotating to undesired angles by colliding with the stopping plate, providing a mechanical safeguard as shown in Figure 1.

A close-up of a machine

AI-generated content may be incorrect.

**Figure 1:** Mechanical Safeguard

Lastly, the armrest is an optional attachment that can be put in place of the front acrylic panel to provide support to the user’s arm during operation.

* + 1. **Mounting**

A Unitrack is a linear slide bolted to the side of a Permobil wheelchair, allowing for many extensions and attachments to be added. The mounting system is designed for quick, direct attachment to the Unitrack; however, the team only had access to a Jazzy motorized wheelchair. Therefore, an additional aluminum L-channel was needed to attach the Unitrack to the Jazzy wheelchair. In the ideal case of a Permobil, the mounting assembly involves a total of four aluminum components as shown in Figure 2: sliding plate, diagonal rails, connecting plate, and French cleats. The sliding plate is the primary point of contact to the Permobil that allows horizontal adjustment of the robotic system. The sliding plate is connected to the Unitrack via wing bolts to allow for easy adjustment and attachment/detachment of the entire robotic system to/from the Permobil. Next, the lower portion of the diagonal rails are attached to the sliding plate via pan head screws at an approximate forty-five-degree angle. The upper portion of the diagonal rails are connected to the connecting plate via a set of three fourths inch diameter locknuts, socket head screws, and square nuts. The connecting plate can slide along the diagonal rails which allows for simultaneous vertical and horizontal adjustment of the robotic system. Lastly, the connecting plate is attached to a pair of French cleats while another set is attached to the electronics case such that the robotic system can be easily removed from the mounting system by simply lifting the electronics case off the French cleats. To prevent the robotic system from being “bumped” off the French cleats, there is a 3D-printed pin that slides through the connecting plate and electronics case.

A metal object with a black background

AI-generated content may be incorrect.

**Figure 2:** Mounting Assembly

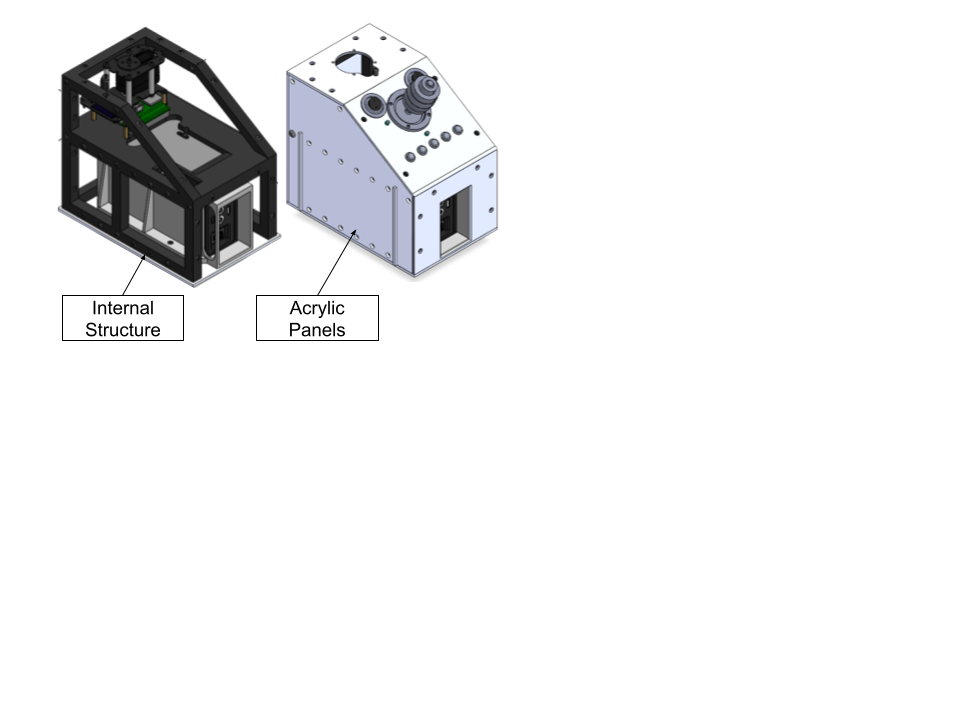
The process for removing the system off the wheelchair involves removing the key that secures the electronics case and lifting the case off the French cleats. Then, loosening the wing bolts on the sliding plate, and sliding the mounting structure off the Unitrack will completely remove all parts of the system from the wheelchair. With only these two processes, the system is completely removed, meeting the end user requirement of a quickly installed and detachable system.

The ease of attachment for the mounting and robotic system was tested by timing the duration to attach the mounting system to the Unitrack, mounting electronics housing on French cleats, and placing the pin through the connecting plate slot and electronics housing. Ease of detachment was validated by sponsor testing, having a sponsor team member detach the system from the Permobil wheelchair without any prior instructions from the team. The stability of the robotic system in response to the mounting system was evaluated with root cause analysis as well as a sponsor assessment. The root cause analysis involved measuring the deflections of the diagonal rails when the robotic system was attached at different positions. Notably, this test was performed for the team’s Jazzy whilst the sponsor assessment was performed for the sponsor’s Permobil. The assessment involved visual inspection of the robotic systems response to static conditions and dynamic movement of the Permobil. While there were some deflections on the team’s Jazzy wheelchair, there were negligible deflections on the sponsor team’s Permobil wheelchair, which led the team to take no further action when addressing vibration, as it performed within expectations on the intended wheelchair.

* + 1. **Electronics Casing**

The electronics casing houses sensitive electronics, such as the lithium-ion battery, protoboard assembly, and ESP32, in addition to providing mounting points for the robotic arm and UI. The electronics housing is comprised of 3D-printed parts and laser cut acrylic plates.

As shown in Figure 3, the electronics casing has an internal structure that acts as the main point of connection to the robotic arm and acrylic panels. The robotic arm is bolted to the middle plate of the internal structure to a set of standoffs while the acrylic panels bolt to the sides of the internal structure with button head screws. These sides are lined with quarter-inch holes where nut inserts are heated and inserted to provide threading for the button head screws. Furthermore, the internal structure provides adequate space for sliding the protoboard into the pin headers of the ESP32 for a stable connection.



**Figure 3:** Electronics Casing

Since the system contains electronics and uses screws as fasteners, it is important to consider the maintenance process to ensure safe operation of the system. All physical components, including the acrylic panels and mounting structure, must be removed and inspected for physical and electric damage. Additionally, the electrical components must be inspected for loose wires, burns, or shorts, and fixed if necessary. Testing will ensure that maintenance activities do not result in system removal from the end user for an extended period, as the end user relies on the system for daily, repetitive tasks. A time limit of three hours was estimated to complete these maintenance activities and tested after full system assembly.

* + 1. **Robotic Arm**

The Waveshare RoArm-M2-S robotic arm was purchased from a third-party vendor and later modified to meet the team’s needs. Initially, the arm was selected as it satisfies the team payload capacity, accuracy at maximum reach, and price requirements. Choosing an arm that satisfied the requirements from the team’s analysis was important in developing a system capable of completing the two high value functions. Once acquired, minor modifications were made such as rotating the end effector ninety degrees as well as rotating the base of the arm ninety degrees to fit the axes of our coordinate system; effectiveness of these changes were verified through testing system operation.

The payload capacity of the end effector was tested by increasing the applied load at the end effector by 100 g each time the robotic arm successfully completed the drinking function; further details on drinking function can be found in Section 2.3. Alternatively, the accuracy of the robotic arm at maximum reach was tested by measuring the horizontal and vertical precision of the end effector position relative to a 35-inch and 48-inch mark. Additional details regarding the test for accuracy can be found in Section 3.2.

* + 1. **Water-resistant Design**

To aid in the water-resistant design of the system, a PVC sleeve, 3D printed end effector cover, 3D printed well, and 3D printed cylindrical sleeve guide were utilized. The end effector cover is meant to protect the end effector servo motor for a short duration in light rainfall. The sleeve acts as an elbow joint cover on its own, but when used with the cylindrical sleeve guide and well, can help protect the lower half of the system, ranging from the elbow servo motor, down to the base servo-motor inside the casing unit.

* 1. **Electrical**
     1. **Protoboard**

A generic protoboard was used as a connection point for all electronic components which included the ESP32, the Adafruit Feather V2, and the UI. The UI was soldered directly to the protoboard, while the Feather was inserted into corresponding headers that were soldered to the protoboard. Lastly, male headers were soldered to the protoboard and would plug into the female headers of the RoArm PCB, which allowed for serial communication between the Feather and RoArm, and power distribution.

A diagram of a circuit board

AI-generated content may be incorrect.

**Figure 4:** Protoboard Layout with Feather

A circuit board with wires

AI-generated content may be incorrect.

**Figure 5:** Protoboard with Solder Connections to Switches, Buttons, and Joystick

* + 1. **User Interface**

The UI contains a three-axis, two-button, Ruffy controls hall effect joystick, five momentary buttons, two SPST switches, and one white LED. Two LEDS beside the switches were removed due to electrical complications that could not be resolved before project completion. The switches were used to enable system and joystick control, and the joystick was used for manual manipulation of the robotic arm. The buttons were used for programming saved locations for later recall, and the LED was used to convey system status to the user. All UI subsystems were mounted on the diagonal plate of the electronics casing, where they would be accessible to the person in the motorized wheelchair as seen in Figure 6.

A white and black device with buttons and knobs

AI-generated content may be incorrect.

**Figure 6:** User Interface (Left – Switches, Right - Joystick, Bottom – Buttons, LED above 3rd button)

* + 1. **Battery**

A 38,400 mAh battery was used to power the robotic arm and all electronics used in the system. The battery is contained in the base of the electronics case and has an LED display showing capacity. The battery also has a switch for powering on and off the system, and a USB port for other attachments. 12 V is routed from the battery to the ESP32 on the robotic arm, which then powers the Feather and protoboard, then the protoboard distributes 3.3 V and 5 V to systems on the UI.

To determine the required battery size, the time at max output and time at minimum output per day were estimated and multiplied by the amperage of these operations then summed to determine the necessary capacity for a battery to last 1 day. Using the frequency of buttons as 30 times a day with a time of 15 sec and the frequency of drink as 30 times per day with a time of 20 secs, max current I of 5 A and min current Imin as 0.2 A, the required battery capacity was determined to be 6200 mAh.

|  |  |
| --- | --- |
|  | (1) |
|  |  |
|  | (2) |
|  | (3) |
|  |  |
|  |  |

The battery selection provides adequate capacity (6.2 times estimated value), which accounts for any differences between estimated values and actual values for the tasks above, and variance in use from different end users.

* 1. **Software**
     1. **Waveshare ESP32**

The Waveshare RoArm comes equipped with a PCB that contains motor drivers and handles serial communication. This PCB is sold with preinstalled code that can be modified; however, the team opted out of modifying the RoArm code since one of the project goals was to be easily replicable. The code on the RoArm PCB reads a JSON object that contains position data from the serial communication (transfer-TX and receive-RX) pins, which the RoArm then deserializes and turns into motor control. Since the team did not modify this code, an Adafruit Feather V2 microcontroller was used for all logic and coordinate updates.

* + 1. **Adafruit Feather V2**

The Adafruit Feather V2 is an ESP32 based microcontroller equipped with GPIO pins that are used for digital and analog communication with other electronic components. Communication between the Feather and the RoArm PCB uses the “RX” and “TX” pins as seen in Figure 4, which are wired inversely between the boards. The GPIO pins on the Feather are used to take inputs from the switches, joystick, and buttons, and output to the LED. The code on the Feather uses these readings to update the coordinates or call specific functions such as the automatic drink function. Specifically, the joystick is used to manually update the coordinates of the end effector, and the buttons are used to recall specific coordinates that have been previously programmed by the user. Once the user wants to move the end effector, the coordinates are formatted into a JSON object that contains the numeric identifier for the type of control, the cartesian coordinates of the end effector destination, and the speed at which the arm should move. An example is provided below. This data is sent across the serial communication pins, where the RoArm uses the numeric identifier and position data to calculate motor movements. The numeric identifier used for all code on the feather was “1041” or “104”, which specifies that the data contained in the object are x, y, z, and end effector pitch, t, positions (104 allows you to specify a “spd” variable in the object, where 1041 does not, both produce the same linear movement between current and goal points).

The following code is an example of what is written in the Feather:

StaticJsonDocument<200> curposdoc;

curposdoc["T"] = 104;

curposdoc["x"] = curxpos;

curposdoc["y"] = curypos;

curposdoc["z"] = curzpos;

curposdoc["t"] = curtpos;

curposdoc["spd"] = 0.2;

serializeJson(curposdoc, Serial1);

Serial1.println();

What the RoArm receives from the Feather:

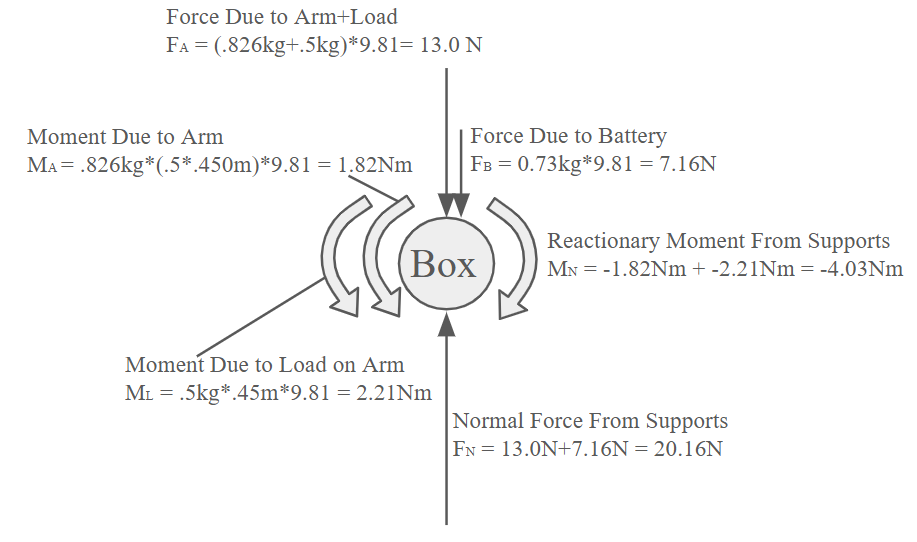
{“T”: “104”, “x”: curxpos, “y”: curypos, “z”: curzpos, “t”: curzpos, “spd” = 0.2}

These objects that contain the position data are continuously updated in the Feather and sent to the RoArm to move the end effector position. Optimizations in the code that sync the transfer speed (between the Feather and RoArm) and the coordinate update speed ensure that equal increments of coordinates are sent with each data transfer.

1. **RESULTS AND DISCUSSION**
   1. **Results of FEA**
      1. **Free Body Diagram**

To estimate the mechanical load experienced by the case and mounting structure, a FBD was developed using estimated forces and moments generated during arm operation, which includes the weight of the arm, case, battery, and load, as well as the moments generated by the load and arm and full extension. The model assumed an extreme use test scenario in which the arm is fully extended with a payload at the end effector. This condition was selected to ensure the support structure can withstand maximum mechanical stress.

The total moment applied to the mounting assembly was found to be 4.03 Nm, acting in the counterclockwise direction. The corresponding force at the mounting point, excluding the weight of the case, was determined to be 20.16 N. The contribution of dynamic human interaction was considered negligible in the FBD but was evaluated separately for safety margins. All forces were applied at the interface between the arm base and the vertical support tracks. The calculated forces were then used as the input for subsequent finite element analyses to verify the structural integrity of the electronics casing and mounting assembly under expected operating conditions.



**Figure 7:** Free Body Diagram of System

* + 1. **Engineering Analysis**

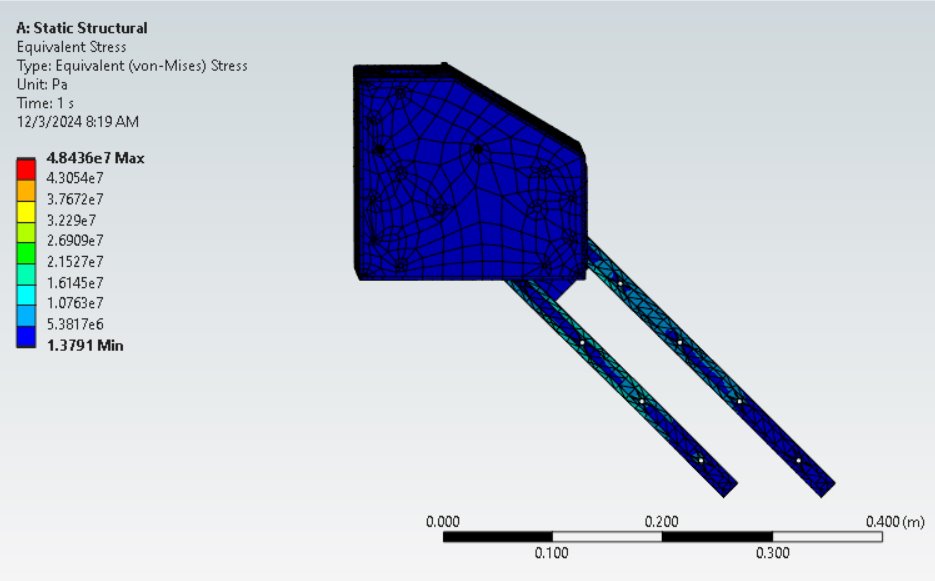
Following the force and moment estimations from the FBD, a structural FEA was conducted using ANSYS to evaluate stress concentrations and deflection behavior in the electronics casing and mounting assembly. The materials used in the construction were primarily 6061-T6 aluminum alloy, selected for its high strength-to-weight ratio and availability. The alloy has a yield strength of 275.8 MPa, which served as the basis for FOS calculations. Under the maximum loading conditions, the highest observed stress concentration was 48.44 MPa, occurring in the T-track that mounts to the vertical rails as seen in Figure 9. This yielded a conservative FOS of 5.69, which indicated the design was capable of withstanding operational forces without material failure.

Deformation analysis revealed a maximum deflection of 1.62 mm at the edge of the case. This displacement was well within the design tolerance and did not interfere with arm accuracy, which had a resolution requirement of ±10 mm at full extension. Additionally, rotational and positional accuracy for each joint was analyzed based on encoder specifications. The final cumulative error was estimated to be <0.8 mm, confirming that the structural deflection would not compromise system precision. These results validated the mechanical robustness of the mounting architecture and confirmed that the system can maintain both strength and accuracy throughout normal use.

A colorful wireframe of a cube

AI-generated content may be incorrect.

**Figure 8:** Ansys Static Structural Analysis of Deformation



**Figure 9**: Ansys Static Structural Analysis of Stress

* 1. **Results of Test Plans**
     1. **Installation Test**

An installation test was conducted to determine the ease of installation in accordance with TR 3.0.0: ​The robotic aid shall be installed in less than 1 hr. The test was performed by timing the assembling of the mounting system to the wheelchair Unitrack, placing the French cleats of the electronics casing onto the French cleats of the mounting system, and finally securing the electronics casing to the mounting assembly with the 3D-printed pin. Lower times indicated an easier installation process. The limit for passing this test estimated at 1 hr, and the test was completed in under 1 min at 58.52 secs. This overestimate of installation time was due to multiple design simplifications that improved the installation process by decreasing the amount of assembly required to completely install the system.

* + 1. **Maintenance Test**

A maintenance test was completed to evaluate the time it takes to perform standard maintenance activities, which is important since removal of the system from the user for an extended period of time (3 hrs) is non-ideal, as the user needs the system for daily tasks (TR 3.1.0). Maintenance activities included inspection of all physical structures of the system, inspection of all wire connections and heat shrink, and inspection of the PCB and protoboard for any burnt components. Any damage or broken components that were fixed were included in the test time. The limit for passing was 3 hrs, the expected value was estimated at 1 hr, and the actual value was 22 mins and 53 secs. The system passed the test with a minor electrical tape fix, and it is recommended that a retest be completed after significant use or after six months.

* + 1. **End Effector Max. Load Capacity Test**

The end effector max. load capacity test addresses TR 1.1.4: the arm should be able to bring a drink weighing 470 g to the user, which verifies the robotic arm’s ability to meet the key function of retrieving a drink to the user. This test was performed by increasing the weight of the drink by 100 g every time the robotic arm successfully completed the drinking function. For each execution of the drinking function, a visual inspection of the system’s performance and stability determined whether the robotic arm performed satisfactorily or if the maximum weight limit had been reached.

The results determined that the robotic arm is able to hold a weight of 500 g when extended to its maximum and a weight of 600 g when at close range. Thus, the robotic arm exceeds expectations, meets TR 1.1.4, and is able to successfully retrieve a drink. The weight of the drink includes the combined weight of the liquid, container, and portable cupholder.

* + 1. **Accuracy Test**

An accuracy test was performed to ensure the system was accurate to 10 mm at full extension, stated as requirement TR 1.3.2. This requirement was developed from the ADA button size and height standards, stating the minimum diameter of an elevator button to be ¾ in, and the height from ground to be 35-48 in [4]. This test utilized a tape measure, calipers, the assembled system on a motorized wheelchair, a whiteboard imitating a button wall, tape, and a whiteboard marker to mark the necessary points. At the lowest incremental movement translated from the joystick’s X and Y axis, one unit was moved right from the initial mark set at 35 in from ground. A second point was marked, and the end-effector was once again incrementally moved one unit upward in the Z direction. Once three points were marked, the same procedure was executed 48 in from ground.

Initial results ranged from 5 mm to 14 mm, implying the system can still press ADA complaint buttons; however, it fails the team’s accuracy requirement of 10 mm.

To resolve the initial failure, the speed of position incrementation in the code was changed to 40 ms, such that the incrementation of the position matched the data transfer rate. Syncing these tasks resulted in an accuracy improvement, now ranging from 4 mm to 8 mm, thereby passing the test.

* + 1. **Restricted Motion Software Safeguards**

To ensure safe operation around the user, particularly preventing unintended contact with the user's arm or torso, the team implemented software-based spatial restrictions into the robotic arm’s control system. This safeguard was developed in response to requirement TR 5.2.0, which specifies that the system must include physical or software constraints that prevent the robot from reaching undesired locations. The Feather was programmed to actively monitor the robot’s spatial coordinates—specifically the x, y, and z positions—using magnetic encoder feedback from the Waveshare RoArm motors. The code defines a restricted region in the operating environment, primarily an angular zone ±30° on either side of the UI as seen in Figure 10, where the end effector is prohibited from entering. This safety zone was determined by ergonomic analysis and sponsor feedback, ensuring the robot would never encroach on the user’s hand space during operation.

The restricted motion logic is embedded in the coordinate update routine. If an input command—via joystick or recalled motion function—would move the end effector into a forbidden zone, the command is overridden, and the motion is aborted before transmission to the RoArm system. During the test, the software safeguard was verified across minimum, medium, and maximum speed settings. At all tested velocities and configurations, the robot’s movement remained strictly within allowed regions. As a result, the software safeguard not only fulfilled TR 5.2.0 but also made the reliance on a mechanical limit a redundancy rather than a necessity. The success of this test highlights the effectiveness of embedded software control in maintaining operational safety, especially for users with limited mobility who rely on consistent, predictable system behavior.

Diagram of a diagram of a mechanical system

AI-generated content may be incorrect.

**Figure 10:** Diagram of Restricted Motion Test

* 1. **High Value Functions**
     1. **Automatic Water Function**

An automatic water function was provided in the Feather code, which allows the user to program a “drink” location on the wheelchair and a “user” location where they want the drink to remain for use. The user will additionally program two “safe” points that the system will path through to ensure proper clearance between the drink and the case. Holding the second button on the UI for five seconds will enter “programming” mode for this button, where the user will program these four points. Then pushing the second button momentarily will run the function, moving to the positions saved by the user. Additional logic, such as when to open the end effector and how to move between points, is done within the code and is not programmed by the user. To change this behavior, the user must change the function in the code, which is provided online in a GitHub repository. Currently, the function is operable and works for specific mounting locations of the drink; however, testing on the intended wheelchair was not possible due to mounting interferences with other necessary equipment. Since each person's wheelchair is different, it is ideal to have the cupholder at roughly the same height as the case due to robotic arm capabilities, but more research should be done on an ideal location.

* + - 1. **Code Progress**

The code is broken into three parts: manual joystick operation, button operations, and safety checks. Manual joystick operation reads the values of the potentiometers in the joystick and compares against limits to increment the local coordinates, which are formatted into a JSON object and sent to the robot.

The button operations use a general button reading function, which passes the button variables into the function, and calls the corresponding programming or recall function for the robot depending on how long the button is pushed for. This framework can be used for additional functions, or to change existing functions, as the movement functions are independent of the button reading framework. Button operation is further discussed in 3.3.2.1.

Lastly, spatial safety features have been implemented to prevent the robot from moving erratically. There are limits on the end effectors maximum and minimum distance from the origin, which prevent the robot from swinging too quickly and hitting other objects or people. There is also a restricted area that spans 60° above the UI as discussed in 3.2.5, which prevents the end effector from hitting the user’s hand while operating the joystick. The last safety measure defines the coordinates of the case and does not allow the end effector to move into this area, so the system does not break itself. Together, these safety features define the operating space of the robot and prevent it from harmfully interacting with itself and its environment.

* + 1. **Elevator Button Pressing**
       1. **Ready Positions**

One key goal for the system was the ability to store and retrieve position data throughout power cycles. The Arduino “preferences” library was used, which stores data to non-volatile memory using a preferences namespace and keyword. By holding a specific button down for five seconds, the user can enter the “programming” mode and move the arm to the position they would like to store. By pushing the same button momentarily, the user will now store this location and exit programming mode. To recall this position at any time, the user can momentarily push this button, and the arm will move to this stored position. Three buttons are compatible with this operation, firstly, button one programs a “ready” position that is ideally used to move the arm up and in front of the case. Button three and four are used to program two different stow positions. Together, the user can save three unique positions, and one function (automatic drink function), which stores four positions.

* + - 1. **Accuracy**

Accuracy of the system was a high priority when making design choices, which was reflected during testing through focus on end effector accuracy and spatial safety. The accuracy of our system is determined by three factors: the code, the servo accuracy, and the structural stability of the assembled system. By syncing the coordinate update speed with the data transfer speed, user inputs have been converted to single increments in position data, thereby optimizing the system by preventing overshoot of desired position. Additionally, the distance the joystick is moved from its neutral position is utilized for variable speed control of the robotic arm, where pushing the joystick further from the default position results in quicker movement of the arm.

The Waveshare RoArm was selected because of its servo motor accuracy, advertised to have a resolution of 0.088, translating to a single (base) motor accuracy of ±0.78 mm. As system complexity increases with more motors, the accuracy will likely decrease, but tests show the value remains within the team’s specified accuracy for performing high value functions.

The last factor that effects accuracy is system structural stability, which is a concern for end effector accuracy as the weight shifts or adjusts, causing deflection. Initial FEA showed that deformation under load is not significant enough to affect the end effector’s location, which was verified when installed on the intended Permobil wheelchair through inspection.

Proper design of these three factors minimized errors in accuracy, which was observed when the team and end user were able to successfully navigate to and press a button.

* + - 1. **End Effector Modification**

Several modifications to the end effector were made so that the two functions were better performed, including rotating the end effector ninety-degrees, glueing a rubber pad to the tip of the end effector, and 3D-printing a cover. Previously, the end effector was oriented to open about the z-axis of a standard cartesian coordinate system. The end effector was rotated so that it opened about the y-axis. Opening about the y-axis resulted in better gripping the horizontal handle of the portable cupholder, allowing it to better perform the drinking function. It was observed that the end effector tended to slip when pressing a button, so a rubber tip was glued to the end effector for better grip and continuous pressure on button surfaces. Lastly, the 3D-printed cover was added to the servo that actuated the end effector, shielding from possible splashing or sudden rain.

* 1. **Open-Source Planning**
     1. **GitHub/Google Drive**

All files related to the purchase, design, and construction of the project are available on a GitHub repository as part of PPMDs goal to have the system be available for hobbyist and future teams to work on and improve. The repository also includes a manufacturing manual that provides an estimate for third party manufacturing based off the team’s design choices, in case the user does not have their own 3D printer, CNC mill, or laser cutter. Additionally, the repository includes assembly and operations manual, an itemized bill of materials, and CAD files that are necessary for system construction.

In addition to the GitHub repository, a google drive has been created that contains images of the project, videos of testing by the team and end user from PPMD, and tutorials videos made to aid the operations manual when describing programming and recalling locations.

The hope of PPMD is that making this project open source will allow the community to continuously improve the design and functions after the end of this project and add new features that would be beneficial to a person with DMD. Additionally, the GitHub repository will be used for future teams if the project continues or is revamped in years to come.

* + 1. **Bill of Materials**

Throughout the duration of the project, a multitude of materials were purchased for initial testing, backup materials if something breaks, and final production materials which summed to a total cost of $2114.45. This value represents the total amount spent by the team, including shipping and taxes, but excluding manufacturing costs as all necessary parts were made in house. The total cost to replicate the system is $1147.62 when considering direct materials and manufacturing costs through third parties and excluding shipping and taxes. Overall, both cost estimates are less than half of the team’s budget of $5000, and a fraction of the cost of other market solutions, meeting the teams goal of developing an affordable system that is available to a larger group of people with DMD.

1. **CONCLUSION**

The system can complete the two high value functions desired by the end user: pushing elevator buttons and bringing a drink to the user. Additionally, the system has safety measures put in place to prevent the robot from harming the user or those around the system. Though the system meets the standards defined by the team, validated through testing, refinements and improvements can be made that elevate the end user experience and provide better assistance for everyday tasks.

A machine with a black bag on the side

AI-generated content may be incorrect.

**Figure 11:** System Assembled on Jazzy Wheelchair

**ACKNOWLEDGEMENTS**

The team extends gratitude to Mr. Keith Van Houten and PPMD for sponsoring the project, as well as the PAAC Committee from PPMD for working with the team and providing valuable feedback to ensure a quality prototype.

Many thanks to Mr. Stephen Moyer and Mr. Jim Hess for advising team 205 through the duration of the project, and for providing crucial insight and guidance when the team was struggling.

A special thanks to Colin Werth, for allowing the team to visit and get feedback from Colin’s design testing, and for continuously engaging with the team through the duration of the project. Thank you to Mr. and Mrs. Werth as well, for allowing the team into their home and for their hospitality during the visit.

Lastly, a thanks to Mr. Richard Henegar and Ms. Kristie Yelinek for helping the team secure a test wheelchair, and for guidance when the team was unsure about certain technologies used for the project.

**REFERENCES**

[1] “About Duchenne” (2020), Parent Project Muscular Dystrophy. <https://www.parentprojectmd.org/about-duchenne/> (Accessed Dec. 10, 2024).

[2] ROBOTS. “JACO” *IEEE*, <https://robotsguide.com/robots/jaco> (accessed Apr. 28, 2025).

[3] U.S. Access Board. 2024. “Chapter 4: Entrances, Doors, and Gates.” *United States Access Board*, <https://www.access-board.gov/ada/guides/chapter-4-entrances-doors-and-gates/>. (Accessed May 6, 2025).

[4] U.S. Access Board. 2024. “Chapter 4: Elevators and Platform Lifts.” *United States Access Board*, <https://www.access-board.gov/ada/guides/chapter-4-elevators-and-platform-lifts/>. (Accessed May 6, 2025).