ParaPedal: A Practical Product for Paraplegic Piano Players

Benedict Eberhagen, Jason Knaebel, Andre Llanos, Rafael Molina

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This document records the design process that our team followed to develop the ParaPedal, an assistive device for paraplegic piano players. It includes the research undertaken to identify the problem and the solutions selected to address it. The solution is broken into hardware and software components. The hardware includes the electrical and mechanical design of a wireless, mouth-based user input device and a separate motor control unit that actuates the piano's pedals. The mouth-based control utilizes pressure sensors to allow the user to press pedals with their tongue, applying variable force. The software includes the algorithms to map user input to the required piano pedal pressure, as well as a web-based interface for system calibration.**

***Index Terms—Assistive Technology, Paraplegic, Piano Pedal, Servo Motor, ESP32, Wireless Control, Pressure Sensor, ESP-NOW*s.**

I. Introduction

The motivation for this project stems from a personal need within the design team. One of our group members, Benedict Eberhagen, is a piano player who is now in a wheelchair. Due to being incapable of using his legs, he is unable to use the piano's pedals, which are essential for properly playing many musical pieces. While not strictly necessary, the pedals significantly enhance the quality and sound of the music. This project, the ParaPedal, is motivated by the goal of helping him, and other individuals with similar physical disabilities, to intuitively and effectively activate the piano pedals and reach the full potential a piano has to offer. A few devices exist to solve this problem, often developed by universities or small companies. The University of Heidelberg, for example, developed two methods: one using head movements (tilt switches) and another using a ”chewing” motion on a teeth guard [1]. CanAssist at the University of Victoria also created a head-activated system using a wireless headband sensor [2]. Another company, Resonate, uses a simple mouth splint that activates a single pedal when bitten [3]. The key difference between these designs is the activation method. Our project seeks to create an intuitive, mouth-based system that provides dynamic control over all three pedals. Our project, the ParaPedal, aims to overcome the limitations of existing solutions by providing a wireless, mouth-based system that is intuitive, responsive, and allows for varying levels of force. The user will be able to operate all three piano pedals with varying amounts of force depending on their control input from a custom mouthpiece. This paper presents the complete design, from initial requirements to fabrication and testing. Section II details the system requirements. Section III and IV describe the hardware and software design, respectively. Section V details the implementation and testing results, and Section VI concludes the paper

II. System Requirements and Design

The design of the system was guided by a strict set of engineering requirements and practical constraints.

Based on the project goals, we established the following key performance metrics from our Engineering Specification table. The system must meet the requirements outlined in the requirements and specs table below.

TABLE I

Requirements		Specs
Maximum Input Power Consumption	≤ 20 Watts	15 Watts
Device Weight	5-10 Kilograms	9.163Kg
Input PCB Battery Runtime	≥ 1 Hour	> 1h30m
Output PCB Battery Runtime	≥ 1 Hour	> 2h
Communication Range	≥ 4 meters	>5m
Response Time	≤2 Second	500ms
Reliability of Motors When Sensor is Pressed	75-85%	99%
System Setup Time	≤ 2 Minutes	≤2minutes

Each requirement was carefully selected to balance performance, safety, and usability. The motor response time of 500–700 ms ensures smooth pedal motion without noticeable delay to the user. The power consumption limit of 20 W keeps the system efficient and prevents overheating during extended use.

Under system overall performance, maintaining a device weight between 5–10 kg guarantees portability without sacrificing stability. The battery runtime of at least one hour allows for full musical sessions without interruption, while the communication range of 4 m or more ensures consistent wireless operation between the controller and pedal units. The system also maintains a motor reliability of 99% upon sensor activation, with a total response time of 500 milliseconds from user input to pedal actuation. Finally, the setup time of less than two minutes ensures the system can be quickly deployed and adjusted before performances.

TABLE II

POWER BUDGET

User Input Module		
Major Component	Current / Voltage	Power
Pressure Sensor (RP-S5)	0.5 mA @ 3.3 V	1.65 mW
ESP32-C6 (Main Controller)	25 mA avg @ 3.3 V	82.5 mW
3.3 V Regulator Loss	Output 3.3 V @ 26 mA	10.4mW
Total (User Input)	**≈ 95mW**	

Output Module		
Major Component	Current / Voltage	Power
ESP32-WROOM (Main MCU)	100 mA @ 3.3 V	0.33 W
3.3 V Regulator – LD33V (LDO)	Input 8.0 V @ 100 mA → Output 3.3 V @ 100 mA	0.47 W(Loss)
Servo Motors (MG996R ×3) via 5 V buck(s)	5 V @ 1A (one servo at a time)	≈ 6.25 W (one servo at a time)
Battery Charger – CN3302 (charging)	-	≈Inactive during operation
Total		**≈ 10 W @ 8 V (≈ 3A)**

III. HARDWARE DESIGN

The hardware of the ParaPedal system is organized into two primary subsystems: the User Input Board and the Pedal Control/Output Board, each serving a distinct but complementary function. Together, they enable seamless, low-latency wireless communication between the performer's input and the actuation of the piano pedals.

The User Input Board is embedded within a custom mouthpiece that uses pressure sensors to detect tongue force. These signals are processed by an ESP32 microcontroller, which converts the analog readings into digital data and transmits them wirelessly using the ESP-NOW protocol. The mouthpiece assembly includes a miniature battery module, voltage regulator, and compact PCB designed for comfort and minimal weight, ensuring the user can perform comfortably. Calibration data, such as minimum and maximum tongue pressures, are stored locally and transmitted to the pedal unit to ensure consistent force between the user input and pedal movement.

The Pedal Control/Output Board, located near the piano pedals, receives the transmitted data from the mouthpiece and converts it into control signals for the actuators. This board utilizes a dedicated ESP32-C6 microcontroller paired with a servo driver circuit to command three Motors (MG996R), each corresponding to one piano pedal. A 3.3 V low-dropout regulator (MCP1700T-3302E) powers the control logic, while a 5 V, 3 A switching regulator (LM2596) supplies stable current to the motors. The system is powered by dual 18650 Li-ion batteries managed by a CN3302 charging IC, enabling simultaneous operation and charging with USB-A connectivity.

Both boards are integrated into compact enclosures fabricated through 3D printing. The User Input Board enclosure provides ergonomic mounting for the mouthpiece, while pedal control enclosure incorporates wire management.

Communication between subsystems is fully wireless, eliminating excess cabling around the piano and ensuring user mobility. The system maintains a communication range exceeding 4 m and a response latency below 2s, enabling near-real-time pedal actuation.

A block diagram with the overall architecture is shown in Fig. 1, illustrating the interaction between sensing, processing, and actuation components.

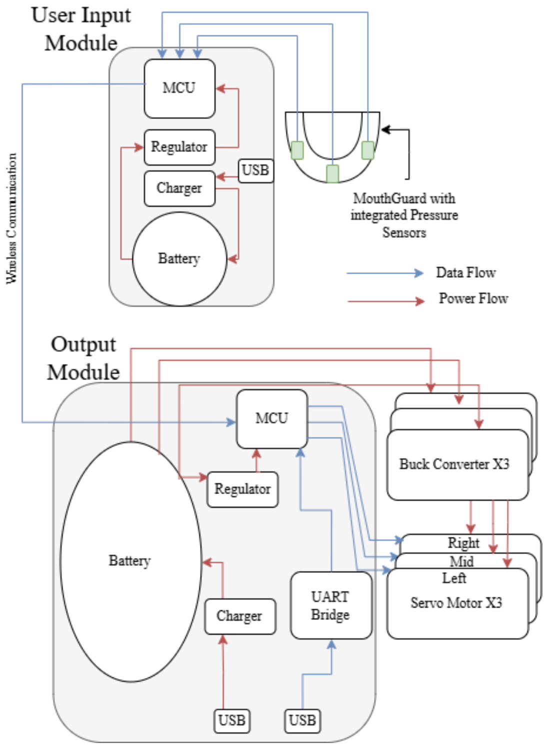


Fig.1. Overall system diagram

Fig.2. Mouthguard-based user input device

A. Mouthguard Pressure Sensor

The primary user interface consists of three Force-Sensitive Resistors (FSRs) integrated into a custom-formed mouthguard. An FSR is a robust polymer thick-film device that exhibits a nonlinear decrease in resistance in response to an increase in applied normal force. This characteristic is leveraged to provide dynamic, proportional control rather than a simple binary switch.

Each FSR is implemented within a voltage divider circuit, with the FSR paired in series with a fixed-value pull-down resistor. The midpoint of this divider produces an analog voltage that is directly proportional to the force applied by the user's tongue. This voltage is routed to an Analog-to-Digital Converter (ADC) input on the User Input Module's microcontroller for processing.

B. User Input Module

The User Input Module is a compact, battery-powered device responsible for acquiring sensor data and transmitting it wirelessly to the Pedal Control/Output Module. This module is the primary interface for the performer, and its design is therefore constrained by requirements for low weight, user comfort, and reliable, low-latency operation.

The MCU of this module is the ESP32-C6 microcontroller, a low-power system-on-a-chip (SoC). This specific MCU was selected over other options for its good blend of performance and power efficiency. Its single-core RISC-V architecture provides more than sufficient processing power to sample the three ADC channels and format the data packets. Importantly, the ESP32-C6 features integrated 2.4 GHz wireless connectivity, offering native support for the ESP-NOW protocol. This protocol is an enabling technology for the project, as it provides a direct, connectionless communication link, achieving significantly lower latency and power consumption than traditional Bluetooth or Wi-Fi protocols.

To meet the portability and runtime requirements, the module is powered by a 120 mAh LIR2450 rechargeable lithium-ion coin cell battery. This form factor was chosen as a deliberate engineering trade-off: it provides the minimal physical volume and weight necessary for a wearable device, while its 120 mAh capacity is sufficient to meet the one-hour operational runtime requirement. The low-power sleep and active modes of the ESP32-C6 are leveraged to maximize this runtime.

An integrated battery charger IC (mcp73831) is included on the module's custom PCB. This circuit manages the charging of the LIR2450 battery, providing a safe Constant-Current/Constant-Voltage (CC/CV) charge profile. This allows the user to easily recharge the device via a standard micro-USB port, which also serves as the programming and debugging interface for the microcontroller. The complete assembly including the MCU, the power management circuitry, and the three sensor voltage dividers—is integrated onto a compact, custom-designed printed circuit board (PCB) and housed in a small, lightweight, 3D-printed enclosure designed to be comfortably worn by the performer.

.

.

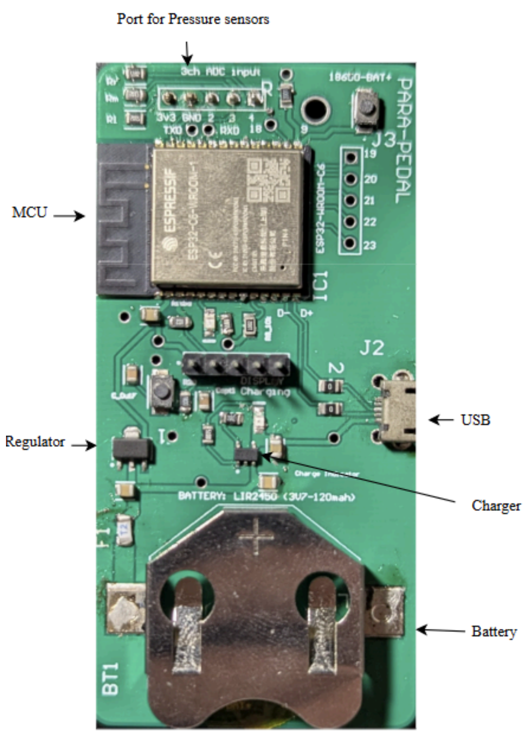


Fig.3. Input Board

C.Output Board

The Output Board is the system's central receiver and high-current actuation driver. Its primary functions are to receive wireless commands from the User Input Module and translate those commands into physical motion of the servo motors.

The processor chosen is the ESP32-WROOM microcontroller. This System-on-a-Chip (SoC) was selected for its dual-core architecture, which is helpful with the system's real-time performance and user-friendly operation. This architecture permits a segregation of tasks: one core is dedicated exclusively to handling the reception and processing of ESP-NOW data packets, ensuring minimal latency between a user's command and the motor's response. Concurrently, the second core manages all non-real-time tasks, mainly the web-based interface that provides a user-friendly platform for system calibration and diagnostics. This separation ensures that web server activity, which can be processing-intensive, never interferes with or introduces jitter into the primary motor control loop.

The entire module is powered by a high-capacity system consisting of two 18650 lithium-ion batteries configured in series (2S). This provides a nominal bus voltage of 7.4V, which is an efficient input level for the downstream regulators. This battery chemistry was chosen for its high energy density and its ability to supply the high discharge currents demanded by the servo motors during actuation. An onboard CN3302 charging IC manages the battery pack,providing a complete Constant-Current/Constant-Voltage (CC/CV) charging profile and protecting the cells from overcharge or over-discharge conditions.

.

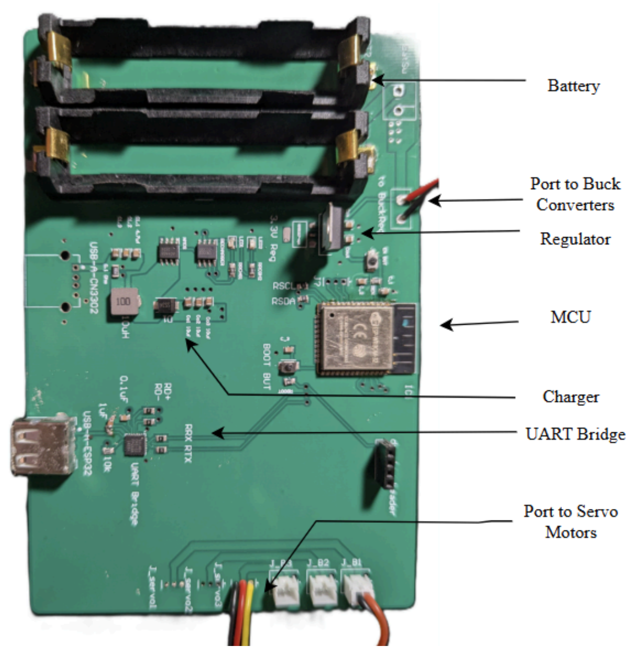


Fig.4. Output Board

D.Buck Converter

To generate the stable 5V high-current rail required by the actuators, three LM2596 switching regulators (buck converters) are employed. These converters are responsible for stepping down the nominal 7.4V from the dual 18650 battery pack. The selection of a switching regulator topology, as opposed to a linear regulator, is a critical design choice for this battery-powered application. It provides significantly higher power conversion efficiency maximizing the system's operational runtime.

The LM2596 was specifically chosen for two key reasons: its 3A output current capability and its practical package. The 3A rating provides sufficient overhead to handle the demanding, high-transient current draw of the MG996R servo motors, which can peak at approximately 2.5A during a stall condition. Additionally, its availability in an easily solderable package (TO-263) allows for high reliability and simplified assembly during the prototyping phase.

could affect the operation of the other motors.

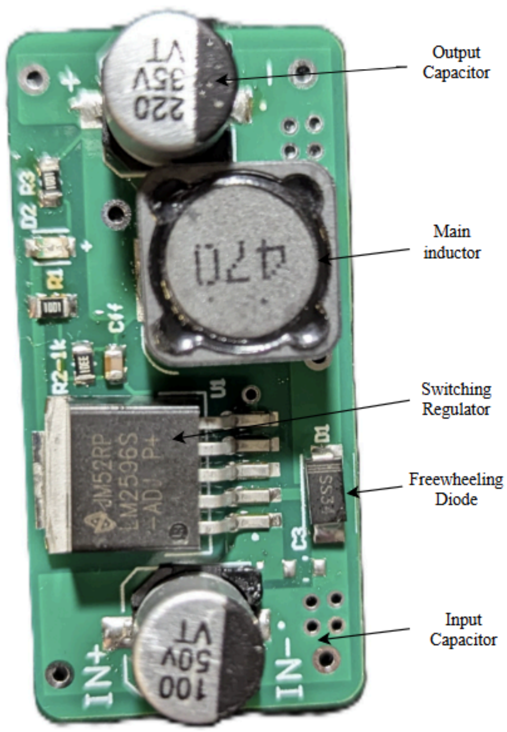


Fig.5. Buck Converter

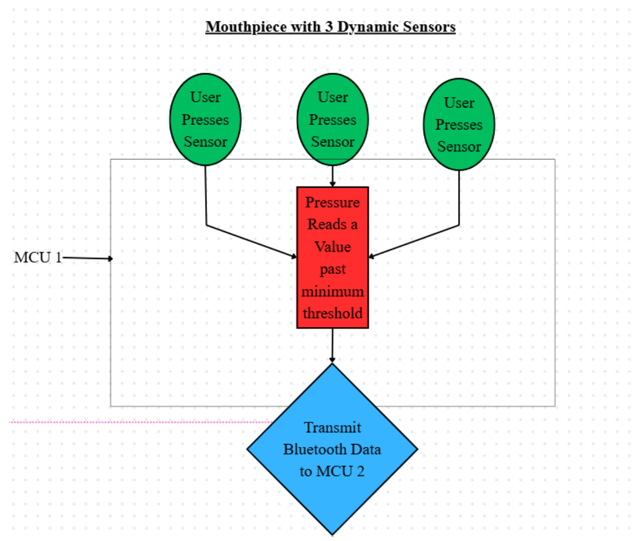
E.Mechanical implementation

The mechanical system translates the servo motors' rotational output into the linear depression of the piano's pedals. The three MG996R servo motors are mounted onto a rigid, 3D-printed frame. This chassis is designed to rest on the floor, aligning the servos with the pedals.

Custom-designed actuator arms (servo horns), also fabricated via 3D printing, are mounted to each servo. As a servo rotates, it pulls a string routed through a pulley under the piano pedal and eventually connects to a clamp-style connection on the pedal. This was chosen over a direct-drive system as there is limited clearance under the pedal. The 3D-printed enclosures for both modules provide wire management and secure mounting for all electronic components.

IV.Software Design

As talked about in section III Hardware design, our Parapedal is made out of two separate devices. The input device with the pressure sensors, and the output device with the servo motors are attached to the device. In order for the two devices to complete their tasks, we programmed our device as seen in the picture figure.



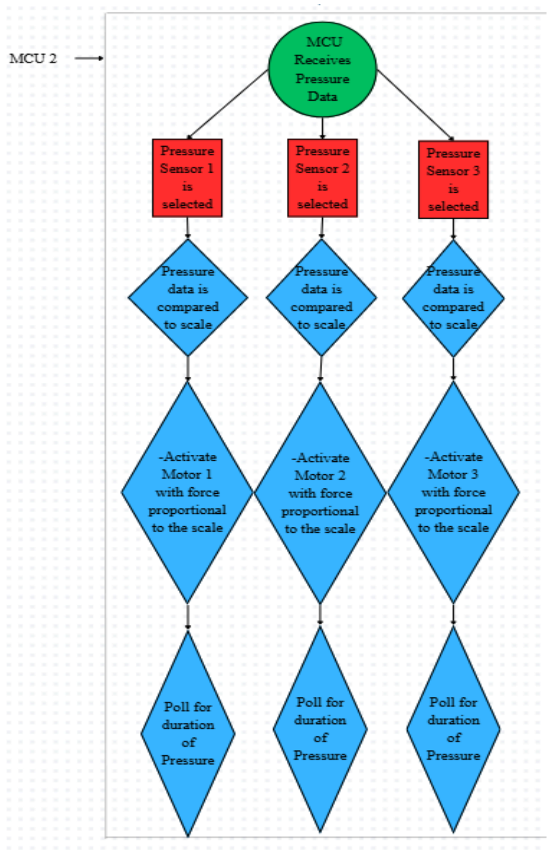


Fig. 6. Overview of pedal actuation code.

For the input device, we used an ESP-32 that reads pressure sensor data from three ADC channels and transmits its analog data wirelessly. It initializes Wi-Fi and sets up ESP-NOW communication to send data packets to a predefined receiver MAC address. The device samples each sensor 32 times for averaging, supports ADC calibration for voltage conversion, and streams data at 10 Hz while printing readings to the console and transmitting via ESP-NOW with timestamps and unit indicators.

The output device also uses ESP32 and receives the sensor data from the input device through ESP-NOW. It then converts it into PWM signals to use for the servo motors. The device maps received ADC values to PWM duty cycles using calibrated ranges in the setup process. It then uses this information to drive the servo motors accordingly. Basically, based on the calibrated ranges, the output will adjust the degree of the servo motors based on the pressure exerted on the pressure sensor. The output device also provides a web interface. This website provides a look into the system for the average user. It shows the pressure exerted on the pressure sensor as well as the change in degrees of the servo motor. This gives the user real-time data on the device's functionality.

As part of the device, we have a setup mode in order to properly calibrate the device to the user's needs and preferences. This mode is activated on the website through a press of the setup mode button on the site. The calibration code implements an automated routine that dynamically maps sensor input ranges to PWM outputs. When activated via the website button, the system samples each of the three pressure sensors for 5 seconds, recording the minimum and maximum values detected during that period. These measured ranges are then stored and used to linearly map incoming ADC values to corresponding PWM duty cycles, ensuring the full range of sensor input is properly translated to the complete servo movement range.

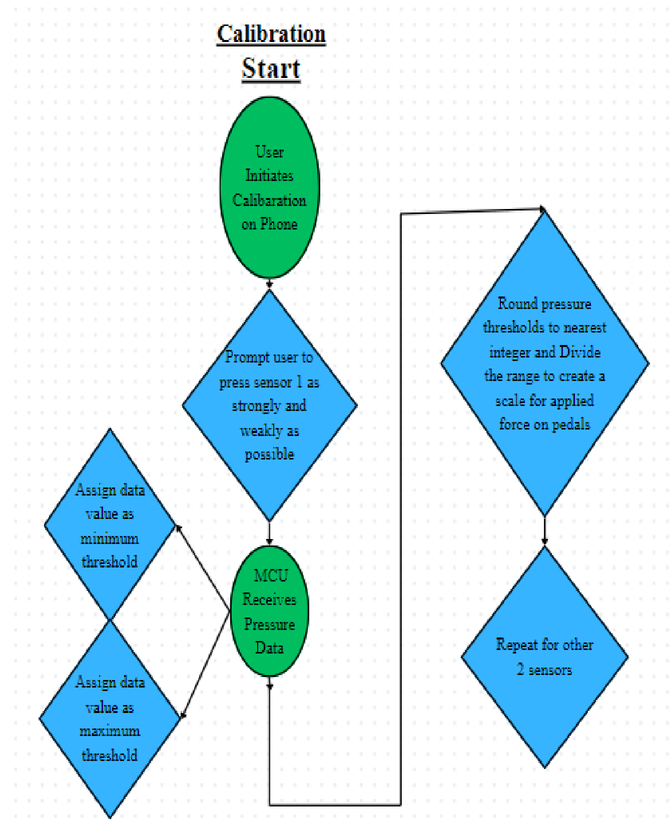


Fig. 7.Overview of input calibration code.

The purpose of our code is to form a complete wireless sensing and actuation system where the input device streams pressure sensor measurements and the output device translates them into proportional servo movements. The system includes calibration capabilities to map sensor ranges to PWM outputs, and remote monitoring through a web interface, creating an integrated solution for wireless sensor-to-motor control.

During the coding process, we had to make a variety of decisions that influenced the software's design choices. To start off, we needed to decide what kind of environment we wanted to use. Our two main options we were interested in using were FreeRTOS and Arduino. Although they can be used together, we decided to only use Arduino during this experiment due to its strong hardware support and, unlike FreeRTOS, stronger community of libraries and lower overhead. With low overhead being particularly important, we need our device to run as fast as possible.

Our next decision we had to make was what code we wanted to use. We were planning on using either Python or C++. This choice was pretty straightforward. As we wanted our code to be fast and require a smaller system, C++ was the obvious choice for our Parapedal device.

Lastly, there are multiple communication tools we have access to. Wifi, Bluetooth, and ESP-NOW. Wifi and Bluetooth are great for handling and transmitting complex protocols and the ability to interact with multiple different devices at the same time. However, as mentioned before, our main goal is low latency, smaller overhead, and lower power consumption. In all these categories, ESP-NOW was the better choice, even if it struggled with complex protocols and was our obvious primary choice. Due to the lower protocol handling capabilities, we adjusted our code to best fit ESP-NOW's requirements, which functioned well during our testing.

V. Testing and Results

Testing was conducted in an iterative manner. The major unknowns were the communication between the devices, the reading of the pressure sensor and the servo supplying the necessary force to the pedal for actuation.

Initial testing made use of development boards to prove the concept with minimal design overhead.
The tests using the development boards was successful so we continued with the PCB and software development,

Major Component Testing

The buck converter was tested by supplying power through an adjustable DC power supply set to 8V and an adjustable load. The adjustable load began at 2A, 0.5A above the servo motors' usual current draw (1.5A). We then slowly increased the Load Current up to 3A, which is the maximum rated amperage for the servos. The output voltage remained within the acceptable range. As a motor load may be less reliable than the DC load, the system was tested with a load on the motor and which performed under the motor load effectively as expected.

The user input was tested for signal range, battery runtime, and recharge time.

Signal reliability was confirmed by increasing the distance between the output board and input board, The system remained connected at 4m which is enough distance to confirm its viability, but also exceeded expectations substantially during excess range testing. During excess range testing, we kept the output board stationary while walking the input board out of the Senior Design 2 Lab in Engineering Building 1 on campus, and increased the distance down the hall until signal was lost. The boards maintained successful communication until a distance of roughly 20m through the walls of the building, proving that the boards can communicate at a range equal to/exceeding five times our system requirements in open space.

For the user input board's battery runtime testing, we connected the output board to a DC power supply to create a constant "battery charge" for the system, while running the input board on only the battery (120 mAh LIR2450 rechargeable coin cell battery). The User Input Board stayed in constant data transmission with the Output Board to consistently drain the battery. Starting at 3.927V (not fully charged), and dropping to 3.686V over the course of one hour, The battery proved fully capable of running for greater than or equal to one hour, as set by our requirements.
The Charging functionality was tested for the input board by using a battery that had a low voltage 3.6V and observing the voltage rise over time when the USB cable was connected.

For the output board's battery runtime testing, we had the output board run on solely battery, while the board actuated all three servo motors in sequence consistently, to ensure the battery was constantly having to supply power to a load. During this testing, the two-cell battery pack's starting voltage was 8.2 V, ending at 7.9V one hour later, proving the product meets and exceeds our 1 hour minimum run time requirement under all stress conditions the device will be under.
The charger for the output board was tested by likewise noticing a voltage rise over time when the USB A cable was connected.

For Calibration Testing,The system's user calibration routine, managed via the web interface, was validated. The test was initiated by activating the "setup mode" button on the device's webpage. During the subsequent 5-second automated sampling period, the user applied minimal and maximal tongue pressure to each of the three sensors, Upon completion of the sequence, the system successfully stored the measured ADC value ranges as the new operational thresholds. To confirm, post-calibration inputs were applied to the mouthguard. The servo motors were observed to actuate with a full and proportional range of motion in direct response to the newly calibrated pressure inputs, confirming the calibration process was successful.

For comfort and ease-of-use testing, Benedict, who is in a wheelchair, transported the system and installed it within the two-minute timeframe established in our system requirements. The input board was light enough for the mouthguard to mount to his clothing without creating any discomfort from downward pulling on the mouthguard. This proved our portability, ease of use, and installation time requirements met.

For device weight testing, we weighed our final components to ensure the weight stays under 10kg, as established in the product's requirements. With the motor/motor system weighing 1007 g, the user output board and regulator array weighing 134 g, and the user input board weighing 22 g, the major components' weight specs definitely stay within the system requirements. However, with the variable weights we use for the system to prevent the lifting effect, the system's overall maximum weight can reach up to 9.163kg.

The initial motor testing for the final product was performed as an integrated system test, without the piano pedals attached. We connected the mouthpiece assembly to the User Input Board, the Output Board to the regulator array and the servo motors, along with the regulator array to the servo motors. We then established successful communication between the User Input Board and began inputting signals into the system through the resistive pressure sensors inside the mouthpiece. The server motors responded as expected, actuating between zero and one-hundred-eighty degrees proportionately with the level of pressure applied to the resistive pressure sensors. This behavior proved that the motors are functioning as expected, while drawing power within their 15W maximum power rating.

For final product functionality testing, we tested similar to the initial motor testing, but with the piano pedals attached. We connected the mouthpiece assembly to the User Input Board, the Output Board to the regulator array and the servo motors, along with the regulator array to the servo motors. The motors were then connected via pulley system to the three pedals for pedal actuation. 8kg weights were also applied to the system to prevent the lifting effect. The servo motors responded as expected, actuating between zero and one-hundred-eighty degrees proportionately with the level of pressure applied to the resistive pressure sensors. This final test confirmed that our product performs as expected and meets all requirements set.

VI. Conclusion

In conclusion, the parapedal prototyping was a success. We have developed a device for paraplegic piano players that can read a pressure sensor input from the user that transmits wirelessly to an output device in order to depress a piano pedal. Through scrutinous parts selection, software development, and hardware interfacing resulting in electronic servo motor control, all that is left is to integrate two more input pressure sensors and output servo motors, and parallelizing them through code, along with the physical engineering of the mouthpiece, PCBs and hardware housing units.

Acknowledgement

The authors wish to acknowledge the assistance of professor Arthur Weeks and Chung-yong Chan.

References

[1] Steingraeber & Söhne. "Steingraeber Pedal Systems for Wheelchair Users." Steingraeber, www.steingraeber.de/en/innovationen/pedal-devices-for-pianists-in-wheelchairs/. Accessed 25 May 2025.

[2] CanAssist. "Head-Activated Piano Pedal." CanAssist, University of Victoria, www.canassist.ca/EN/main/programs/our-technologies/museum/piano-arts.html. Accessed 25 May 2025.

[3] Resonate Adaptive. "Resonate. Make Music Accessible." Resonate Adaptive, www.resonateadaptive.com/. Accessed 25 May 2025.

[4] "ChatGPT." *OpenAI*, 31 Oct. 2025, chat.openai.com.