

Gesture-Controlled Input Device for Accessible Gaming Applications

Saketh Chodisetti¹, Ajay Bansal²

4/13/2026

¹Round Rock High School, Round Rock, Texas

²Arizona State University, Tempe, Arizona

Abstract

This research addresses the accessibility limitations that individuals diagnosed with Oligodactyly may encounter when using conventional gaming controllers. The objective of this research is to develop an alternative to conventional adaptive controllers that is a low-cost, wearable, gesture-based adaptive controller that offers patients with Oligodactyly greater inclusivity, while retaining usability. Conventional controllers typically require the function of multiple fingers, creating a barrier for gamers without full finger dexterity. The goal of this research is to design, implement, and evaluate a prototype of a cost-efficient, wearable, gesture-based adaptive controller that translates gesture movements and wrist flex into controller inputs. Using a microcontroller, accelerometer, flex sensor, and a tailored 3D-printed housing design, we developed a prototype that was efficient at emulating gaming inputs. Performance was evaluated using a subjective evaluation of basic gameplay and an objective evaluation using a generic typing test of just the controller inputs. The results indicate a relatively high accuracy and prove the prototype to have strong potential for an accessible alternative to traditional adaptive gaming controllers.

1. Introduction

This research investigates the development of an assistive gaming controller designed for people who face the condition of Oligodactyly, a condition where one is born without 5 digits per hand.(10)(11)(12) By utilizing motion sensors and an embedded system, the prototype aims to avoid finger usage in order to make it more accessible to people with this condition through both ease of use and price. As avid recreational gamers, we found it right to develop a controller that is accessible to more people since it occurred to us that individuals diagnosed with this condition do not have the same access to the same gaming facilities as we do.

The main problem addressed in this study is that most adaptive controllers are overly expensive and may still require multiple fingers to operate (5)(18), and regular controllers are aversive to use while having Oligodactyly on a daily basis. To address this issue, we designed a wearable device that reads wrist orientation and movement as controller inputs, enabling users to interact with the device with essentially only forearm and wrist movement. We also decided, to maintain its cost-effectiveness, as current adaptive controllers can easily exceed prices of upwards of 200\$(18), we would keep the production budget under 30\$ for everything included.

We included background information based on research we did before initial development, a basic overview of the design of our system, explained our methodology, and discussed the project's results and overall conclusion as well. We have also included many figures and diagrams that relate to our project.

2. Related Works

A recent study by Pittaccio developed a device by the name of Playcuff, which is a wearable videogame controller designed with children with motor disabilities in mind. At its core, the device uses inertial sensors and gesture thresholds and algorithms to recognize forearm movements and translates them into videogame inputs. Overall, their testing showed relatively high user feedback and testing accuracy. Similar to our present study, Playcuff provides an alternative solution to conventional buttons and replaces them with forearm movements, however our design prioritizes a lower-cost implementation only using an ESP32, accelerometer and flex-sensor and is specifically targeted towards those with Oligodactyly(4)

3. Background

Oligodactyly is a congenital condition characterized by the absence of one or more fingers or toes, typically present at birth and affecting one or both hands.(10)(11)(12) In some scenarios, oligodactyly occurs as just the condition; however, in rare scenarios, it may be associated with additional skeletal abnormalities.(10)(11)(12) Due to having fewer digits than typical, oligodactyly patients often face limitations when performing tasks that require dexterity and fine motor skills.(10)(11)(12)

Such limitations pose challenges when utilizing conventional gaming controllers. Modern gaming controllers for most video games often require simultaneous use of multiple buttons, triggers, and joysticks, making them difficult to operate for individuals with limited finger mobility. According to the Joseph Maley Foundation, adaptive gaming controllers have become increasingly important because they allow people with physical disabilities to participate wholly in recreational gaming and promote inclusion amongst gaming communities(5). However, many adaptive controllers still rely on external switches, customized hardware, and other specialized equipment.(5)(18)

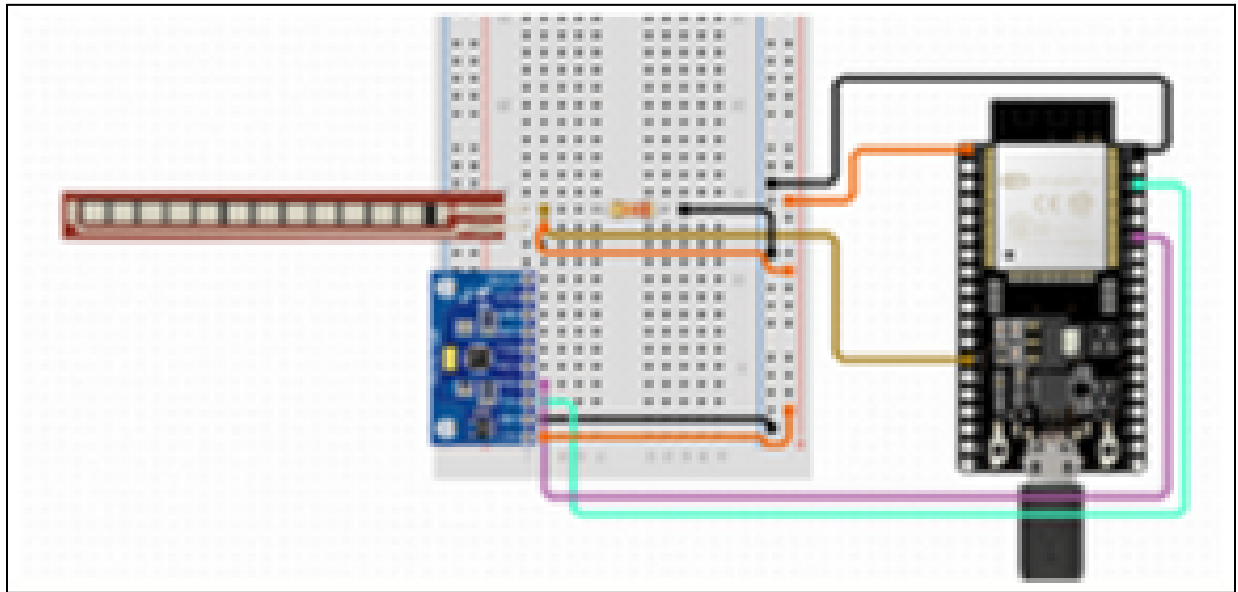
Recent work in accessible interaction design explores alternative input methods that reduce the dependence on finger dexterity. Verma's research on gesture-based wearable interfaces demonstrates how kinesthetic movements can be translated into input, allowing users to interact with technology through motor movements rather than precise finger mobility.(17) Gesture-based systems present a new promising approach for individuals who may have difficulty operating traditional controllers.(17)

Together, developments in adaptive gaming and wearable technology provide the foundation for the gesture-based gaming device in this research.

4. System Design

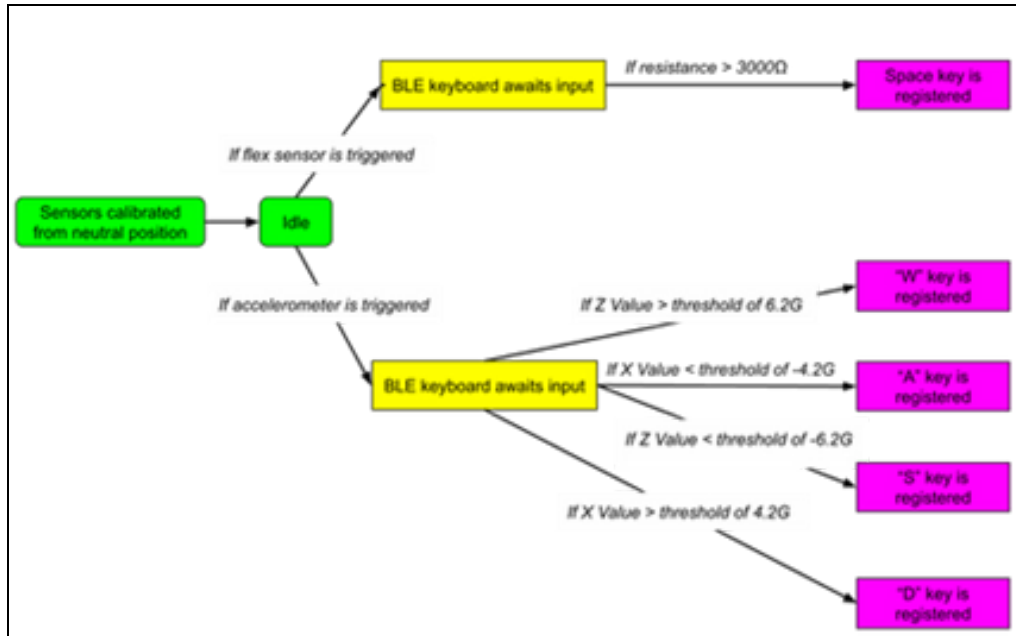
Our system consists of three primary electronic components. It utilizes an ESP32 DevKitC as the microcontroller, a BMA220 as the accelerometer, and an unbranded generic flex sensor.(1) To begin, we made a simplified electronics diagram showing an example of how the circuit would look on a breadboard, as shown in Figure 1.

Figure 1: Simplified Electronics Diagram



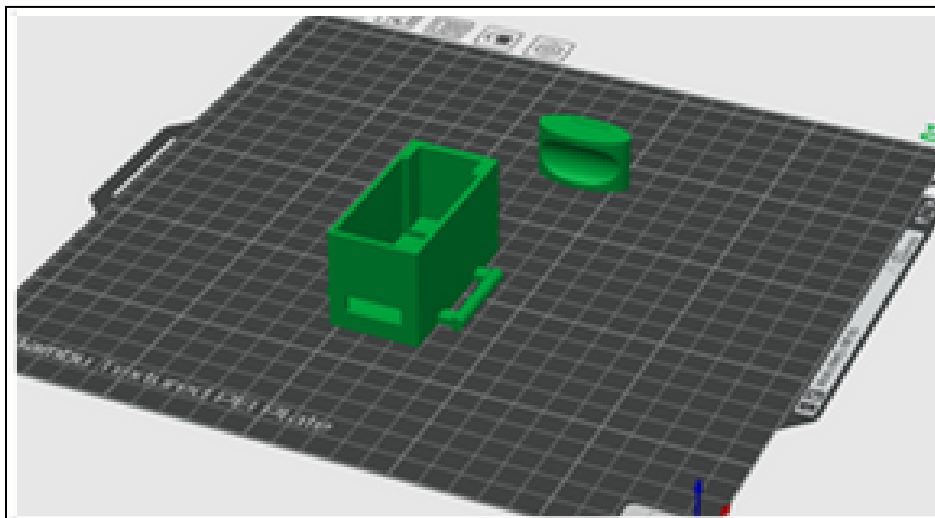
Then, building on Dominic Matthews' adaptive controller thesis (8), we utilized minimal input systems for widespread utility by using simple thresholds and inputs as shown in Figure 2. Essentially, at first, we did not do any hard-coding, but we made a simple diagram of how the system would work. For reference, the BLE keyboard is the specific Python library we utilized to recognize input.(6)At first, the accelerometer and flex sensor should be recalibrated every time the system boots up, while assuming that the user is wearing the system in a comfortable, neutral position. Then, from an idle position, the system awaits input. The BLE keyboard library is able to check if a specific threshold is reached at any time and is able to perform the corresponding input.(16) For example, if the flex sensor reaches a certain resistance, or if the accelerometer reaches a certain 3-dimensional threshold on an x, y, or z axis.

Figure 2: Software State Workflow



Following this, we designed a comfortable brace that would both attach to the user's carpal region and house the system as a whole. To do this, we used a CAD software called TinkerCad(15) to create a rigid, simple housing to hold the microcontroller and accelerometer, and another brace extension to stabilize the flex sensor, as shown in Figure 3. We then 3-D printed the brace with TPU filament, as it is relatively comfortable to wear compared to PLA.(3)(6)

Figure 3: CAD of ESP32 Case and Flex Sensor Brace



We then did hard-coding to assign various wrist flex and positions to map to specific inputs. All relative to the calibrated neutral position, a forward arm position would correspond to the “W” key on a keyboard, a right tilt of the forearm would correspond to the “D” key, a left tilt would correspond to the “A” key, the upwards or 90 degree position would correspond to “S”, and if the flex sensor reached its threshold it would correspond to a spacebar input.

5. Methodology

To adjust the thresholds to maximize input registration efficiency and accuracy, we first assigned a random value to each input's respective dimension. Accelerometer values were mapped to directional inputs based on orientation ranges, and the flex sensor value was configured to detect the slightest change in resistance.

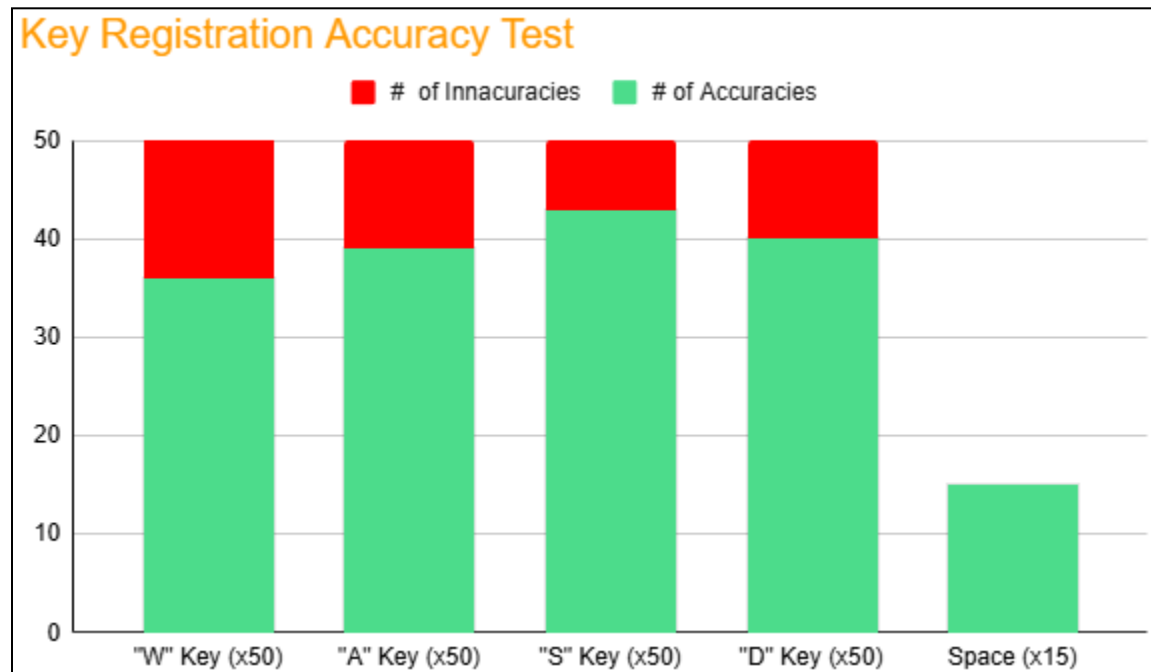
Initially, it was difficult to use the directional inputs detected by the accelerometer, as we had previously used proportional threshold values for every dimension. As we began to experiment with different values, we noticed that up and down movements of the forearm required a larger threshold, as they were more sensitive to movement, and rotational movement of the forearm left and right required a proportionally much smaller threshold, as it was much harder to rotate than move the forearm up and down. Meanwhile, we had to drastically increase the flex sensor threshold as it was detecting every minute change in resistance.

We also operationally defined how we would test the system while considering the factors of our original objective.(2) Expectedly, our objective of developing a low-cost controller that was purely gesture-based meant that accuracy would be imperfect, but we wanted it to be over 60% and be relatively consistent.

6. Results And Discussion

After fine-tuning thresholds as much as possible, we tested only the accuracy by recording the number of correct inputs from the total trials. Figure 4 shows a bar chart with the number of correct trials out of 50 for each directional input and 15 trials for the flex sensor input.

Figure 4: Key Registration Accuracy Test



From the key registration accuracy test in Figure 4, it is evident that the space key via flex sensor was the most accurate by far, likely due to its simple carpal motion.(13) Meanwhile, the least accurate input was the “W” key. This is likely because the forward forearm motion is the hardest motion to actually perform consistently, as it is forward relative to your neutral position. It requires extra energy to reach the threshold. Meanwhile, the “A” and “D” keys were almost on par, as they are both rotational motions, and the user’s flexibility is a minor factor in the accuracy of these inputs. Lastly, the “S” key registration was the 2nd most accurate after the flex sensor input, likely due to the fact that it is simply making an approximate 90-degree angle between your forearm and shoulder. Overall, when testing the accuracy of each keystroke, inputs from the accelerometer were mostly accurate with some discrepancies, while input from the flex sensor was completely accurate due to its simplicity, and with the space key arguably being one of the most important inputs in most games, this proved adequate.

In Figure 5, we tested both the accuracy and the efficiency of the adaptive controller through a generic typing test that we customized to only have the characters that our prototype is enabled to type: “W, “A”, “S”, “D”, and the space bar.

Figure 5: Customized Typing Test Results



We specifically used the typing test website, MonkeyType, as it is easily customizable and is easy to analyze accuracy and performance results.(9) From the customized typing test in Figure 5, we were able to achieve our desired accuracy of above 60%. Using data from Figure 4, it was mostly the directional inputs that were inaccurate, as the flex sensor input maintained 100% accuracy. Within 17 seconds, we were able to accurately achieve an input speed of 7 words per minute, which is approximately 1.17 words per second. The performance of gaming peripherals is measured in CPS, or clicks per second, and although this score is not on par with the renowned devices of the assortment of gaming technologies, it is fast enough to be considered operable for basic games. However, we also faced a problem of input latency, which also severely reduced our CPS, as the library we used, the BLE keyboard, wasn't able to register more than 2 inputs at a time. This is called ghosting, where rapid, overlapping inputs cause a failure to register certain key combinations.(7) Overall, although the input was slow due to ghosting, the accuracy was relatively high for the most part, and the speed was fast enough to be usable in basic gaming applications such as SlowRoads, an online driving simulator that utilizes basic controller inputs, and can be improved with further calibration and prototype testing.(14)

7. Conclusion and Future Work

The development and testing of this accessibility controller revealed promising outcomes in gesture-based key registration, with consistently high accuracy across most inputs. The system was able to achieve high overall accuracy, reliable gesture detection, and functional input registration. While refinement is needed due to issues with ghosting, accuracy was high overall. Ultimately, the results affirm the controller's precision and responsiveness, offering a promising alternative to conventional adaptive controllers and underscoring its potential to provide inclusive input for Oligodactyly patients.(5)(17) Considering the free TPU filament and access to a 3-D printer, the price of the microcontroller (ESP32 DevKitC), which was approximately 15\$, the price of the accelerometer (BMA220), which was approximately 3\$, and the price of the unbranded, generic flex sensor, which was less than 7\$, the total development price of this adaptive controller remained under 30\$, meeting the objective of developing a cost-effective controller.

Future improvements to this project might include expanding input to emulate a full controller or keyboard with improved accuracy in the future and conducting user testing with

individuals with various disabilities to refine ergonomics and increase inclusivity. These advancements can further increase usability and make the device suitable for real-world applications.

Acknowledgements

We would like to thank Professor Patrick Benfield and Professor Ajay Bansal for their help and guidance throughout this project and for providing essential resources and dedication towards our research.

References

1. Admin. (2020, June 11). Flex Sensor & Arduino Interfacing - Measure Bend/Resistance. How to Electronics. <https://how2electronics.com/interfacing-flex-sensor-with-arduino/>
2. Dixon, T. (2016, October 24). Operational Definitions. IB Psychology. <https://www.themantic-education.com/ibpsych/2016/10/24/operational-definitions-2/>
3. Exploring Exotic 3D Printing Materials Used in Prosthetics - Shapeways Blog. (2019). Shapeways.com. <https://www.shapeways.com/blog/exploring-exotic-3d-printing-materials-used-prosthetic-s?srsltid=AfmBOorzRvWEhncdin8m4mok6SyyknKO8vvG-t9yQJ0arUKHXeBtVAr>
4. Lazzari, F., Romanò, J., Nossa, R., Meloni, S., Garavaglia, L., Diella, E., Valoriani, M., Fedeli, F., Porro, M., Biffi, E., & Pittaccio, S. (2026). Design and Preliminary Evaluation of a Smart Orthotic Videogame Controller Dedicated to Children. *Annals of Biomedical Engineering*, 54(7), 2346–2363. <https://doi.org/10.1007/s10439-026-04035-7>
5. Maley, G. (2024, March 13). Accessibility in video gaming: adaptive controllers. Joseph Maley Foundation. <https://www.josephmaley.org/accessibility-in-video-gaming-adaptive-controllers/>
6. MarkForged. (n.d.). Your TPU 3D Printing Questions Answered. Markforged.com. <https://markforged.com/resources/blog/your-tpu-3d-printing-questions-answered>
7. Mash, S. (2022). What is Keyboard Ghosting? Hp.com; HP. <https://www.hp.com/us-en/shop/tech-takes/what-is-keyboard-ghosting>
8. Matthews, D. (2024, January 1). DIY Adaptive Gaming : Game Control for People with Upper Limb Disabilities.
9. Miodec. (2020, May 15). Monkeytype. Monkeytype.com. <https://monkeytype.com/>
10. *Oligodactyly*. (n.d.). <https://www.nicklauschildrens.org/conditions/oligodactyly>
11. *Oligodactyly: Causes, symptoms, and treatment*. (n.d.). <https://www.medicoverhospitals.in/diseases/oligodactyly/>
12. Oligodactyly - Pediatrics - Orthobullets. (n.d.). [Www.orthobullets.com. https://www.orthobullets.com/pediatrics/4079/oligodactyly](https://www.orthobullets.com/pediatrics/4079/oligodactyly)
13. Padmore, C., Nishiwaki, M., Gammon, B., Langohr, G., Johnson, J., & Stoesser, H. (2016). Biomechanical Evaluation of Carpal Kinematics during Simulated Wrist Motion. *Journal of Wrist Surgery*, 06(02), 113–119. <https://doi.org/10.1055/s-0036-1588025>
14. Slow Roads. (2026). Slowroads.io. <https://slowroads.io/#A2-7c60e525@49.13>
15. Tinkercad. (2019). Tinkercad. Tinkercad. <https://www.tinkercad.com/>
16. T-vK. (2021, September 10). GitHub - T-vK/ESP32-BLE-Keyboard: Bluetooth LE Keyboard library for the ESP32 (Arduino IDE compatible). GitHub. <https://github.com/T-vK/ESP32-BLE-Keyboard>
17. Verma, S. (2024, May 6). Shreedhar Verma – IXD@Pratt. <https://ixd.prattsi.org/author/sverma6/>
18. Xbox Accessories & Controllers | Xbox. (2025). Xbox.com. <https://www.xbox.com/en-us/accessories?xr=shellnav#controllers>