3D-Printed Robotic Hand for Rehabilitation: Assembly and Control System

Catalin Pavel Movila
Student ID: S225535
Special Course Project (5 ECTS)
Supervisor: John Paulin Hansen

May 31, 2025

Abstract

This report documents the development of a 3D-printed robotic hand designed as a platform for exploring embedded system control, servo actuation, and gesture replication. The system consists of five continuous servo motors controlled by an ESP32 micro controller, programmed using the Arduino IDE and Python. While the primary goal is to gain hands-on experience in hardware-software integration, the project also investigates how such a system might be adapted for assistive or rehabilitation-oriented applications in the future.

1 Introduction

This project focuses on designing and controlling a 3D-printed robotic hand using continuous rotation servos and an ESP32 micro controller. The goal is to better understand the mechanics and electronics involved in gesture replication using embedded systems. The report also explains how the video tracking mechanism can be used to further expand the capabilities of the hand tool in the future.

The project was structured around several key objectives:

- **Assemble a 3D-printed robotic hand** based on a Parallax-style servo setup.
- **Understand the hardware architecture**, including servo mechanics, microcontroller usage, and control electronics.
- **Test system latency, responsiveness and accuracy** of gesture replication.
- **Explore the embodiment potential of the robotic hand** and its applicability in stroke rehabilitation.
- **Understand the integration of hand-tracking software** using MediaPipe and OpenCV.

2 Assembly of a 3D-Printed Robotic Hand

A primary goal was the mechanical assembly of a robotic hand based on a Parallax-style servo-driven design. At first, out of 15 available 3D-printed kits, five skin-colored hands were assembled. Each used five Tower Pro MG995 continuous servo motors to actuate individual fingers, connected via flexible wires routed through the printed joints. In the second part of the project, five other hands were assembled — the black-colored ones — which featured a better structural design and wire management.

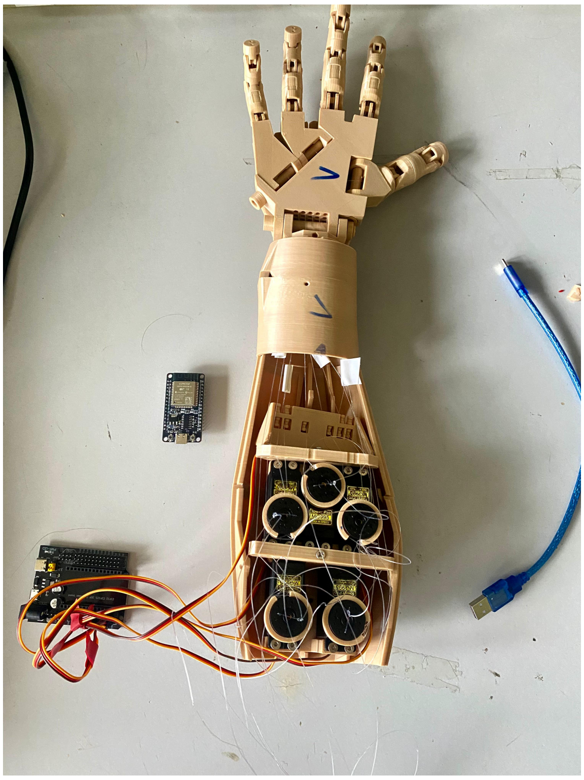

Figure 1: Overview of the skin-colored hand, the first to be assembled

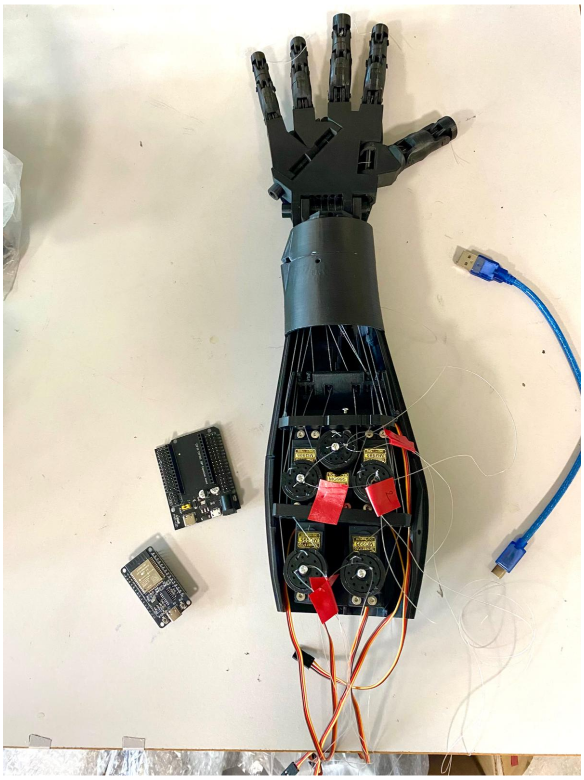

Figure 2: Overview of the black hands, with improved design

3 Understanding the Hardware Architecture

The robotic hand is operated using a combination of microcontrollers, expansion hardware, and actuators. Below is a breakdown of the components used in the project, along with short explanations and relevant illustrations.

- **ESP32 WROOM32 Microcontroller**
  A high-performance microcontroller from Espressif, with dual-core processing and built-in Wi-Fi and Bluetooth. Used to receive commands and generate PWM signals for the servos.

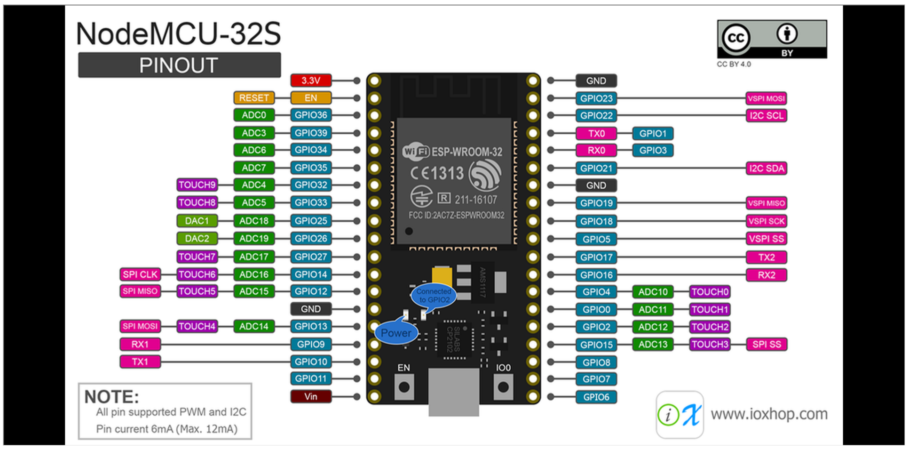

Figure 3: ESP32 architecture overview

- **ESP32S 30P Expansion Board**
  Provides easier access to GPIO pins and simplifies servo connections. It also helps manage USB and external power inputs.

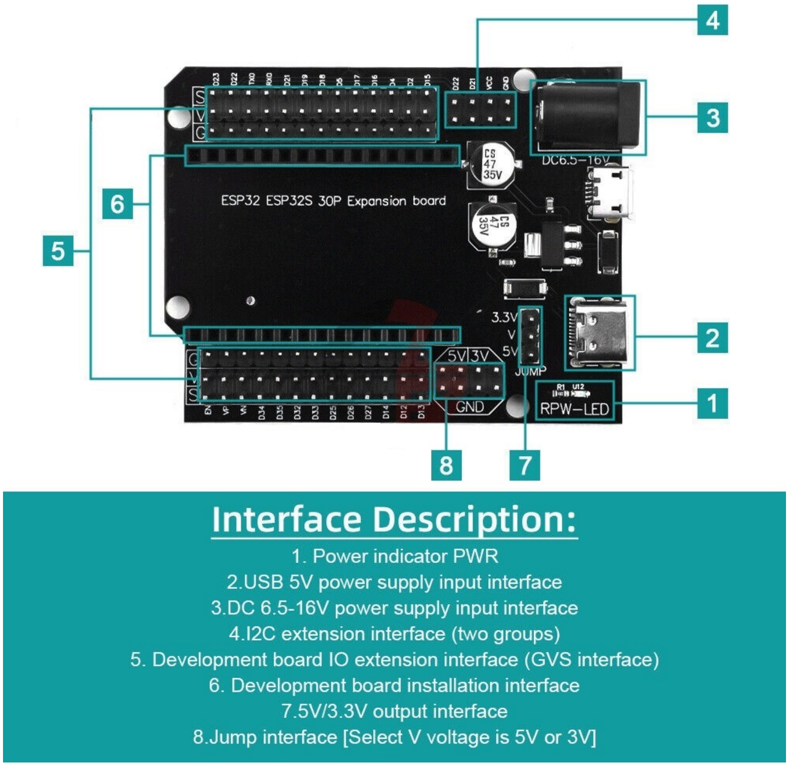

Figure 4: ESP32S 30P expansion board

- **Tower Pro MG995 Continuous Rotation Servos**
  High-torque continuous rotation servos used for finger actuation. Lacking position feedback, they rely on timed PWM signals for control.

  Each MG995 can draw up to 2.5 A under stall conditions, but normal operation draws far less current. Since fingers don't move under full load simultaneously, a 5V 3A supply is usually sufficient. For more intense tasks, a 5V 4–5A supply is recommended.

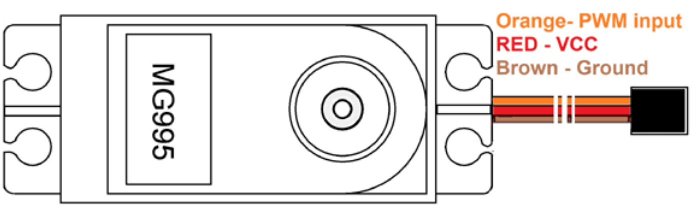

Figure 5: MG995 servo pinout and wiring diagram

- **Programming and Control Interface**
  The ESP32 was programmed using Arduino IDE. Python scripts sent commands via USB serial. These were translated into PWM control signals for the servos.

4 System Testing: Latency and Responsiveness

System testing evolved throughout the project as the design and control methods were improved. The process can be divided into two main stages:

Initial Phase – Skin-Colored Hands

During the early phase, I assembled five skin-colored 3D-printed hands. These initial tests exposed several mechanical issues. The thread used to drive finger movement was slippery against the plastic, and the way it was tied created uneven tension. The servo wheels had some play, and the alignment of the wire loops caused unpredictable motion. As a result, fingers would sometimes clench but fail to return, or stall mid-motion due to accumulated resistance.

This phase helped identify critical design flaws and limitations in my initial approach. A video demonstrating the problems encountered during this stage is available here:

Video: `https://youtube.com/shorts/0d-_u7yJpnQ?feature=share`

Final Phase – Black Hands and Improved Setup

Later in the project, I switched to a set of five black 3D-printed hands with better geometry and cable routing options. Applying lessons from the first phase, I glued the servo wheels in place to reduce wobble, adjusted the routing holes to minimize tension, and tested each servo individually before performing multi-finger gestures. These changes resulted in a more stable, responsive setup.

However, challenges still occurred when attempting to control all five servos simultaneously. Occasionally, the serial connection to the ESP32 would drop — especially when external power was used. This suggests a need for better power management or signal isolation.

A second video showcases the improved mechanical structure and gesture performance:

Video: `https://youtube.com/shorts/MI0_rE01ikc`

5 Final Status of the Results

At the conclusion of the project, five robotic hands were assembled and evaluated. Each kit reflects improvements and also limitations of the system which needs further development. **One common aspect of the black hands is the the pinky finger** does not clench entirely. The first joint of the finger has more friction than all the others(I pressume a printing problem which needs to be sand down to make room). Below is a summary of the final status for each hand:

- **Hand 1:** The first one presented in the video and simulation behaves the best, it has a a good response time due to the tightness of the wire.
- **Hand 2:** The second has persistent issues with the index finger where it looses tension and the wire drops from the servo. I reattached the wires 5 times on the servo
- **Hand 3:** The third one does not have the ring finger wire. In the assembly process I accidentally removed it. I did not have enough time to place another one. The rest of the hand does not show any kind of irregularities.
- **Hand 4:** The most stiffest hand out of these 5. The first articulation of the thumb is blocked. This is a design problem from the printing process. I will need to either sand it down or oil the surfaces.
- **Hand 5:** The last one does not show any outstanding issues.

5.1 Responsiveness

Servo response to commands was near-instant for single movements, but synchronization across all five fingers was less consistent under load. **The connection stopped for 2 seconds after inserting a command for all the 5 servos in the Serial Monitor.**

5.2 Limitations

- MG995 servos lack position feedback. They will only take time and speed in consideration and the speed is dependent on the setup, the friction and load of the serov.
- **Wiring problem.** After a lot of uses, the wires tend to slip from the servo and need to be placed back which takes time.
- Mechanical wear can appear after a lot of usages. PLA is not the best material for long term use.

5.3 Accuracy

All hands replicated the movements using the same 2 impulses at the same written speed values (10/180 for a movement and 170/180 for the opposite direction) Some of the fingers do not need the same amount of time at that speed in order to achieve the desired position. The first hand required impulses of 600ms, 1.2 seconds runtime on each direction in order to fully clench and fully.
In sum, all of the fingers required between 1 and 1.2 seconds of runtime between fully clenched and full release.

6 Further Developments

In order to use the robotic hands at their full capacity, the following improvements should be considered in the future:

- **Secure wires:** A more reliable method is needed to fasten the wires to the servo motor. Currently, the knots tend to slip after multiple uses, which requires frequent adjustment and consumes time.
- **Reduce joint friction:** Joint resistance creates unnecessary load on the servos. Sanding and cleaning the internal surfaces of the 3D-printed joints improve finger movement and accuracy.
- **Use a stronger power supply:** A regulated 5V power supply with higher current (at least 3–5A) is recommended to ensure all servos operate at full capacity without voltage drops or disconnections.
- **Consider positional servos:** Replacing continuous rotation servos with positional (angle-based) servos would allow more precise control over finger angles.

7 Potential Hand Tracking Integration

In the later stages of the project, I explored the theoretical use of computer vision techniques MediaPipe and OpenCV — for understanding concepts of movement tracking. Although this was not implemented during the current phase, it represents a clear next step in advancing the functionality of the prototype.

MediaPipe, developed by Google, provides real-time hand tracking through a standard RGB webcam. It detects 21 landmarks on a human hand, including each finger joint and fingertip. By extracting this landmark data, it's possible compute the finger bending, extending, and overall postures. Combined with **OpenCV**, a library for computer vision processing gesture recognition pipelines can be developed to classify these poses or translate them into real-time control signals.

For example:

- When MediaPipe detects a closed hand (all finger landmarks folded), a Python script could send a "close" command to all five servos.
- When only the index finger is extended, the script could trigger motion for a single servo on the robotic hand.
- Smooth transitions could be approximated by combining time-based servo rotation with dynamic pose detection.

8 Rehabilitation Potential

While the robotic hand developed in this project was not designed for direct clinical use, its structure and control system make it a desirable candidate for exploring assistive applications.

Gesture replication and remote control offer potential in rehabilitation exercises where repetitive, guided motion is needed. With further development, the system could assist in visualizing or simulating hand movements for patients who have limited mobility or need recovery.

9 Conclusion

The project successfully delivered a functional 3D-printed robotic hand capable of executing basic gestures. Key skills were developed in hardware integration, servo calibration, and system testing. The result is a flexible, extensible platform for future development in gesture-based robotics and potential assistive applications.

References

- Parallax Robotic Hand Kit Documentation
- Espressif ESP32-WROOM-32 Datasheet
- Tower Pro MG995 Servo Specifications
- MediaPipe: `https://mediapipe.dev/`
- OpenCV: `https://docs.opencv.org/`