ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

H.E.R.O. - Hazardous Environment Remote Operator

Team #9

XIHE SHAO (xihes2@illinois.edu)
JUN LIANG (junl6@illinois.edu)
QIHAN SHAN (qshan3@illinois.edu)
SIZHAO MA (sizhaom2@illinois.edu)

TA: Leixin Chang
Supervisor: Hua Chen

May 27, 2025

Abstract

This project developed a vision-based robotic hand system aiming to replicate human gestures in real time for remote operation. The system eliminates reliance on wearable sensors by using camera to capture hand movements. A modular, 3D-printed robotic hand made of PLA and TPU shows adaptability to diverse tasks and environmental conditions.

While the system successfully mimics basic grasping, limitations include low grip strength, low dexterity, and instable gesture tracking. Iterative design improvements focused on thumb articulation, connector stiffness optimization, and spatial filtering algorithms. Future work includes integrating soft robotics for improved dexterity, refining vision algorithms for robust tracking, and expanding degrees of freedom for complex manipulation.

Contents

1 Introduction

1.1 Purpose

Workers in hazardous environments, such as those involving toxic materials or high-pressure equipment, face significant safety risks. Current solutions, including sensor-equipped gloves and pre-programmed robots, suffer from inflexibility, discomfort and high costs, limiting their effectiveness. To address these challenges, this project introduces a vision-based robotic hand system that uses camera tracking and 3D-printed components to replicate human gestures in real time.

1.2 Change Log

We modify the thumb angel for better fraction. And tested different hardness of connector to balance between stability and bounce.
We also tried to use depth camera for a stable hand capture, and add position constrain in both high level code and micro controller code for more safety.

1.3 Functionality

- **Gesture Capture & Interpretation:** Cameras track human hand movements without requiring wearable sensors, ensuring non-invasive operation.
- **Precision Motion Translation:** A closed-loop feedback system converts gestures into robotic movements with sub-millisecond accuracy.
- **Modular Robotic Manipulation:** A 3D-printed robotic hand interacts with hazardous objects, and can easily be modified to adapt diverse tasks. We have made few types of hand and fingers for different object. It can also adapt different environment by changing printing materials (PLA, ABS, TPU, PETG).

1.4 System Overview

Our system consists of three subsystems:

- **Vision Module:** Stereo cameras capture hand gestures, with algorithms processing spatial data for motion tracking.
- **Control Unit:** Translates gesture data into motor commands using PID feedback to minimize latency ($<$200 ms).
- **Robotic Hand Assembly:** Modular 3D-printed joints and actuators replicate human hand dexterity.

Figure 1 is the system structure of our design.

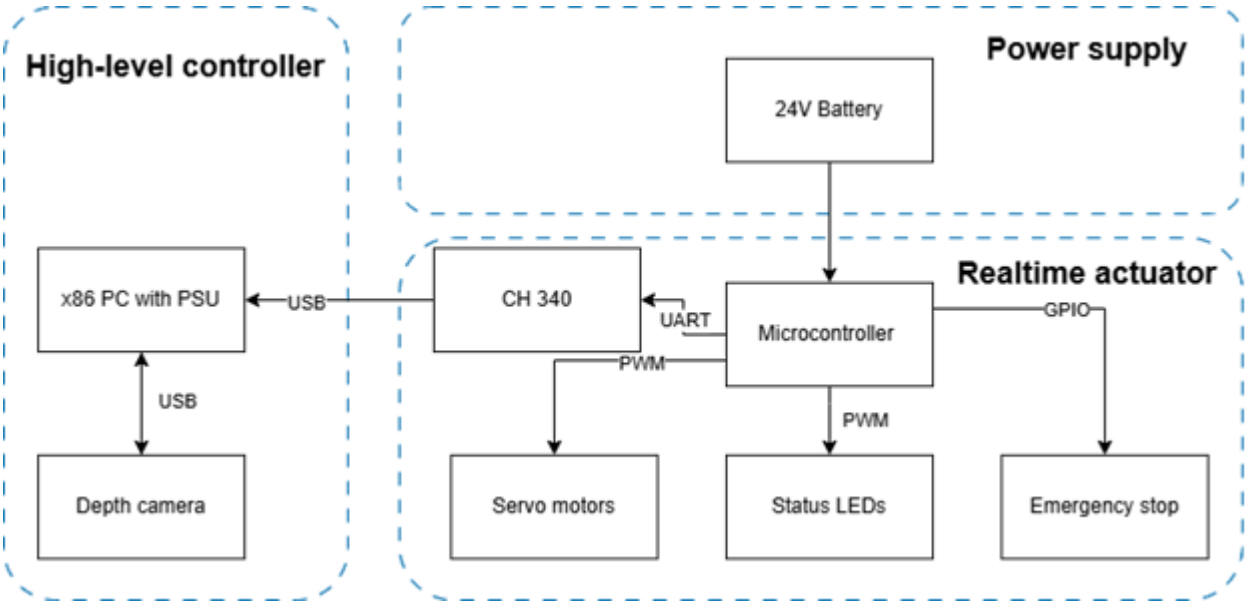


Figure 1: Block Diagram

2 Design

2.1 Design Procedure

2.1.1 Mechanical Design

Material choice The design of the robotic hand prioritized both reliable return-to-origin capability and long-term durability of the finger joints. To achieve stable rebound after actuation, the connection components of the finger joints—specifically parts 7 and 8 in Figure 2—were fabricated using TPU (Thermoplastic Polyurethane) material. TPU was selected for its combination of flexibility, durability, and its ability to retain consistent elastic properties even after repeated loading cycles, making it an ideal choice for ensuring the robotic fingers return smoothly to their initial, neutral positions without the need for additional mechanical springs or rebound mechanisms.

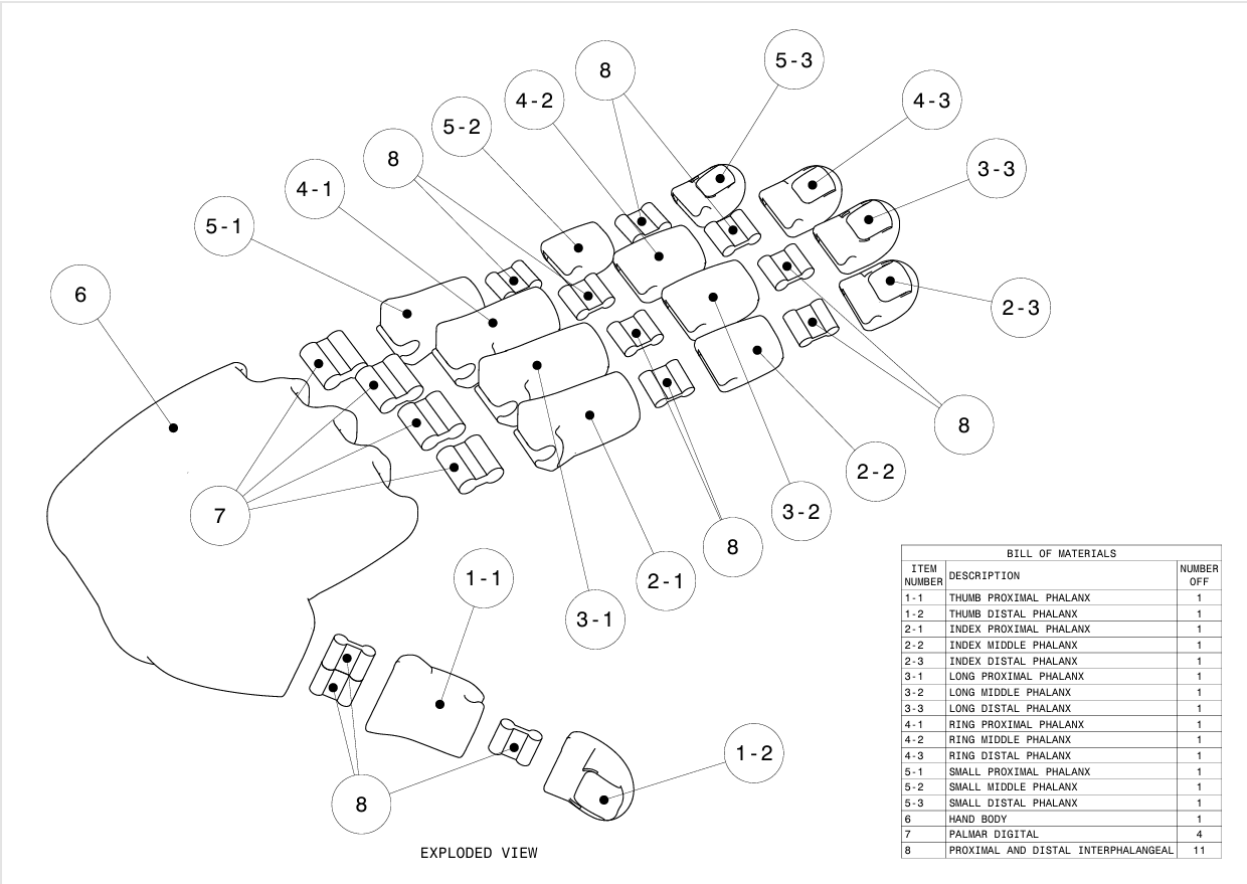
The mechanical performance of the robotic hand is strongly influenced by the hardness of the TPU material used for these connectors. Excessive hardness could impair finger flexibility, while insufficient hardness would compromise structural stability and rebound force. To quantitatively evaluate material performance, Table 1 summarizes the key mechanical properties of TPU 85A, 90A, and 95A at 25°C, including Young's modulus, 100% modulus, and tensile strength.

Table 1: Mechanical Properties of TPU at Different Shore Hardness Levels (25°C)

Material	Young's Modulus (MPa)	100% Modulus (MPa)	Tensile Strength (MPa)
TPU 85A	8–12	3.5–5	30–40
TPU 90A	12–20	7–9	35–45
TPU 95A	20–27	12–15	40–50

As shown in the table, increasing Shore hardness results in higher Young's modulus, 100% modulus, and tensile strength, indicating greater rigidity and load-bearing capacity. However, this comes at the expense of flexibility, which is critical for the smooth articulation and elastic rebound of the finger joints. Based on both experimental evaluation and the mechanical data, TPU 85A was selected for the finger joint connections, as it provides the optimal balance between flexibility for reliable rebound and adequate strength for long-term durability. This choice ensures that the robotic hand achieves both robust performance and a long operational lifespan.

Dimensional Selection The dimensions of our robotic hand were designed to closely match those of a human hand. By mirroring both the proportions and overall appearance of a real hand, the mechanical hand achieves a high degree of bio mimicry. This not only enhances functional compatibility but also supports the goal of creating a lifelike and natural-looking robotic hand.



BILL OF MATERIALS

ITEM NUMBER	DESCRIPTION	NUMBER OFF
1-1	THUMB PROXIMAL PHALANX	1
1-2	THUMB DISTAL PHALANX	1
2-1	INDEX PROXIMAL PHALANX	1
2-2	INDEX MIDDLE PHALANX	1
2-3	INDEX DISTAL PHALANX	1
3-1	LONG PROXIMAL PHALANX	1
3-2	LONG MIDDLE PHALANX	1
3-3	LONG DISTAL PHALANX	1
4-1	RING PROXIMAL PHALANX	1
4-2	RING MIDDLE PHALANX	1
4-3	RING DISTAL PHALANX	1
5-1	SMALL PROXIMAL PHALANX	1
5-2	SMALL MIDDLE PHALANX	1
5-3	SMALL DISTAL PHALANX	1
6	HAND BODY	1
7	PALMAR DIGITAL	4
8	PROXIMAL AND DISTAL INTERPHALANGEAL	11

Figure 2: Exploded view of the robotic hand design.

2.1.2 Motor Control System

The control system shown in Figure 3 design prioritized precision, real-time responsiveness, and safety through hardware-software co-design. Three key factors shaped the architecture:

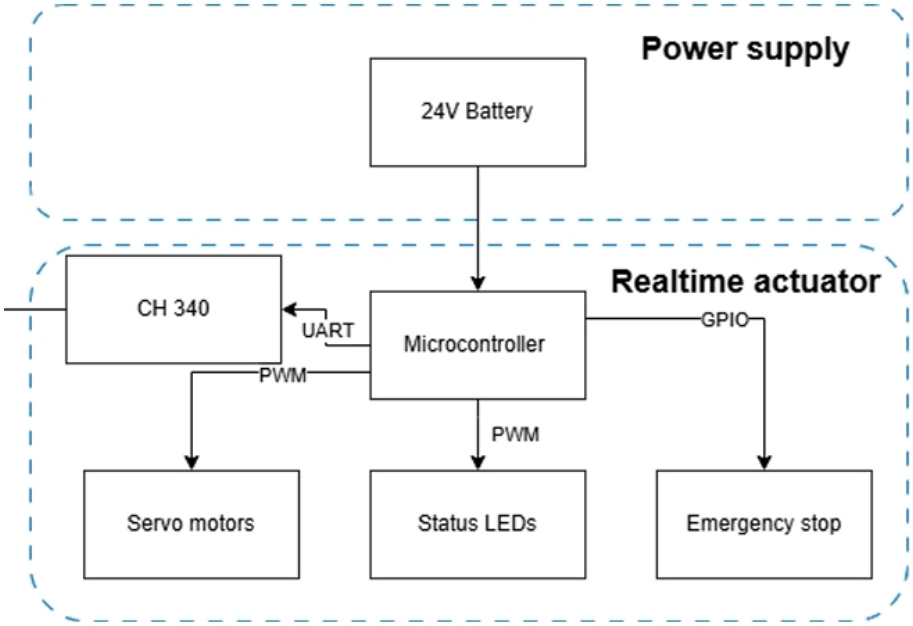


Figure 3: Controller subsystem interfaces

PWM Generation Strategy: Equation 1 and 2 are used to linearly map input angels to PWM signals. Hardware timer peripherals were chosen for their deterministic timing accuracy, critical for maintaining 50 Hz servo signals. Alternative clock sources (internal oscillator) were rejected due to the ±1% frequency drift.

Communication Protocol Selection: UART at 115200 baud outperformed alternatives like SPI or I²C in simplicity for PC integration. The 5-byte payload structure with direct angle mapping minimized parsing overhead in the interrupt handler, crucial for less than 10 ms data response in PC and micro controller. A simply data flow is shown in Figure 4

Safety Mechanisms: Dual-layer protection (software angle clamping and hardware fail-safe) was implemented after hazard analysis revealed single-point failure risks in PCB traces. The mid-position (90°) fail-safe balances mechanical neutrality better than zero-force solutions while avoiding abrupt movements from sudden power loss.

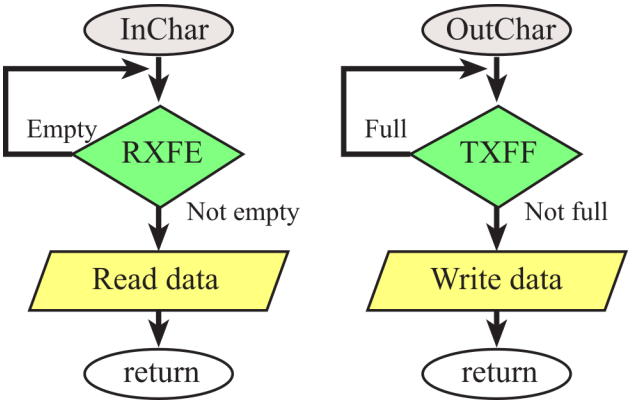


Figure 4: UART data processing flowchart

2.1.3 Visual Perception System

The visual perception system (shown in figure 5 is designed to accurately track hand movements in 3D space using a depth camera and computer vision algorithms. It captures synchronized RGB and depth data, detects hand landmarks in 2D using a lightweight neural network, and converts them into real-world 3D coordinates. This pipeline enables robust, low-latency hand tracking for applications like robotic control and human-computer interaction.

Sensor Selection: The system uses an Intel RealSense D405 depth camera [1] for high-precision 3D sensing. Compared to standard RGB cameras, the D405 provides active depth sensing using infrared patterns, ensuring reliable depth data even in low-light conditions. Its synchronized RGB-D output eliminates alignment issues between color and depth frames, while its compact size and USB 3.0 interface make it ideal for embedded systems. The depth accuracy (±2% within 0.3–3 meters) is critical for precise 3D hand tracking.

Visual Perception Algorithm: MediaPipe's hand landmark model [2] offers a real-time, lightweight solution for 2D hand tracking. Unlike heavier CNN-based models, MediaPipe achieves 20+ FPS on consumer CPU by optimizing for edge devices. It detects 21 hand landmarks with pixel-level accuracy and outputs normalized coordinates. The model's low computational cost allows seamless integration with depth processing, and its single-hand detection mode further reduces latency for our use case.

Pixel-to-3D Algorithm: The system converts MediaPipe's 2D landmarks into 3D world coordinates using depth data. Each landmark's (u,v) position in the RGB image is mapped to its corresponding depth value, then transformed via the camera's intrinsic parameters. This step includes depth validation to handle occlusions and noise, ensuring only physically plausible 3D points are generated. The final output is a vector of 3D hand landmarks, ready for robotic control or gesture recognition.

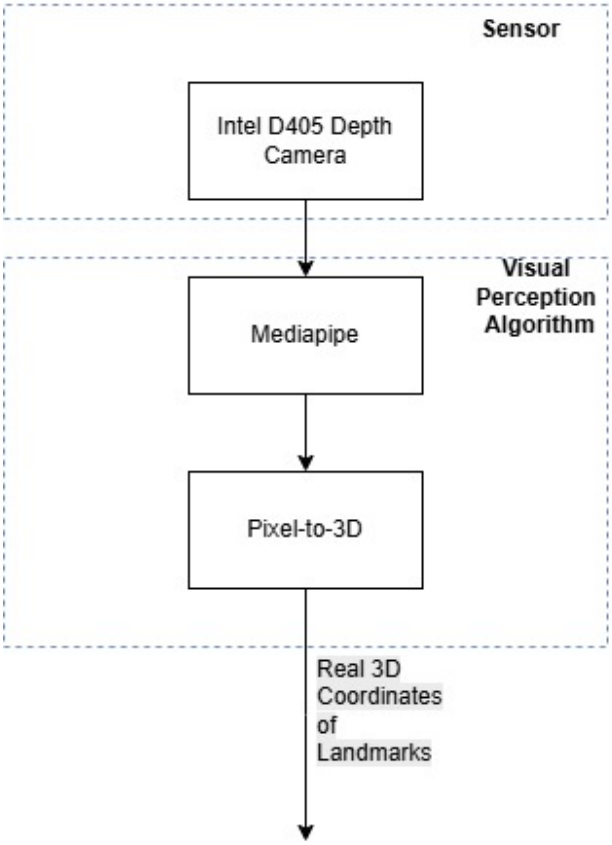


Figure 5: Visual Perception System

2.2 Design Details

2.2.1 Robotic Hand Mechanical Design

Reel design Converting the servo motor's rotational motion into the linear reciprocating motion of the tendon posed a significant design challenge. Initially, several mechanisms were explored, such as gear racks, cam systems, and linkage structures. However, these designs introduced structural complexity, reduced the mechanical stability of the robotic hand, and resulted in a bulky connection between the servo motor and the tendon.

To address these challenges, a more streamlined structure was developed, as shown in Figure 6. In this design, the tendon first wraps around a circular pulley via a guide pipe, completes one loop around the pulley, and then passes through holes with diameters of 2 mm and 5 mm. The end of the tendon is anchored at the 5 mm hole. The pulley ensures smooth guidance of the tendon while minimizing friction and wear. The pipe serves as a guiding channel for the tendon, preventing it from slipping off the pulley and ensuring consistent motion. The fixation point securely anchors the tendon, eliminating any risk of slippage during operation. This configuration ensures that the tension in the tendon is evenly distributed, reducing wear and improving the overall stability of the motion.

During installation, the servo motor is rotated to its maximum limit in one direction. The tendon emerging from the pipe is then tangentially aligned with the pulley's rotation and connected to the fixed position on the finger. By precisely controlling the rotation angle of the servo motor, accurate control over the tendon's movement length is achieved, which directly translates to the finger's motion. Additionally, the mounting interface allows for easy and secure installation of the servo motor, ensuring proper alignment with the tendon mechanism.

This design ensures both compactness and mechanical efficiency, addressing the challenges of motion conversion while maintaining the overall stability and reliability of the system. Furthermore, this approach minimizes the size of the tendon-motor connection structure, making it easier to integrate into the robotic hand without compromising its performance. By optimizing the interaction between the components, the system achieves high precision and durability, making it suitable for various applications.

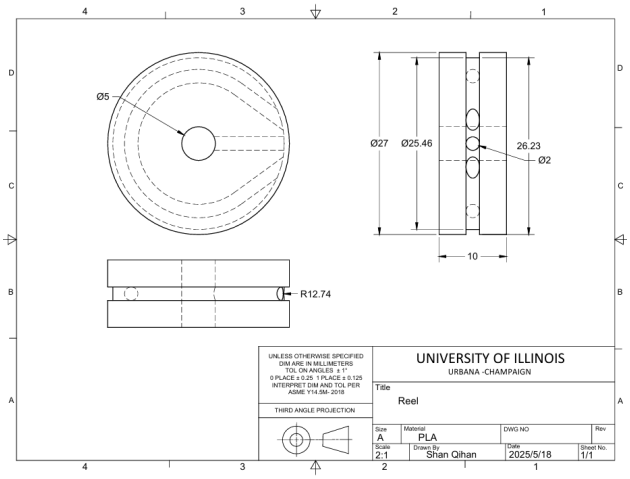


Figure 6: Engineering drawing of the reel.

Motor Base Design The design of the motor base was carefully considered to ensure both structural compatibility with the robotic hand and ease of integration with existing robotic arms. As illustrated in Figures 7 and 8, both the engineering drawing and the conceptual sketch of the motor base are provided.

In the sketch, the five dashed circles indicate the positions designated for the reels, while the five solid rectangles correspond to the mounting locations for MG 995 servos. The five dashed lines extending to the origin represent the cable pathways. Additionally, the four holes on the right side, each with a diameter of 4 mm, are used for mounting the main control board. The dimensions of the motor base have been precisely matched to the size of the designed robotic hand, ensuring a harmonious appearance and proper mechanical fit.

Beyond securing the relative positions of the motors and the robotic hand, the motor base also serves as the "forearm" of the system, enabling straightforward attachment to

existing robotic arms such as the ARX5. This dual functionality simplifies assembly and enhances the modularity of the overall robotic system.

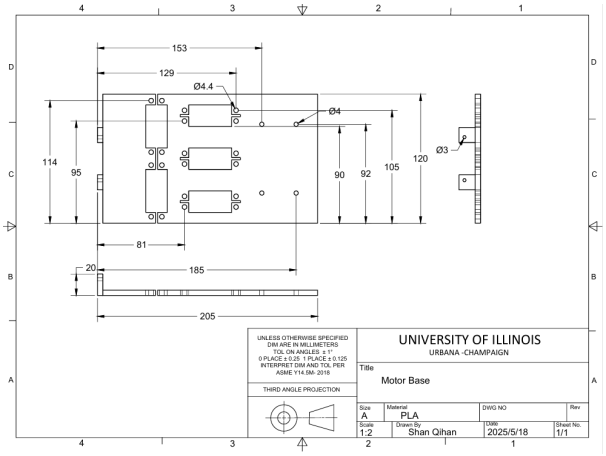


Figure 7: Engineering drawing of the motor base.

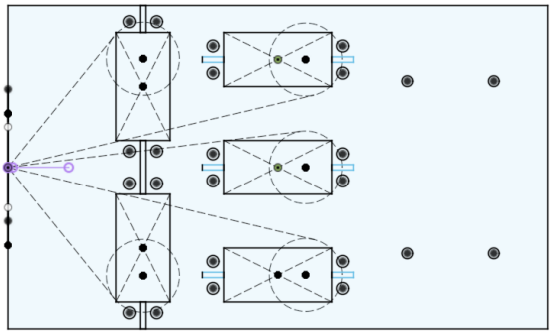


Figure 8: Sketch of the motor base.

2.2.2 Motor Control System

Motor choice We choose MG995 servo motor [3] as actuator. Because it provides 90 N*cm torque and can be controlled by PWM.

STM32 Pinout Configuration According to datasheet[4], we assignment pinout. As shown in the STM32 pinout Figure 9:

- UART6: TX(PA11), RX(PA12) for host communication
- PWM: PE9(Ch1), PE11(Ch2), PE13(Ch3), PE14(Ch4)(All use TIM2), PC6(TIM8 Ch1)
- LEDs: PG0–PG4 for channel

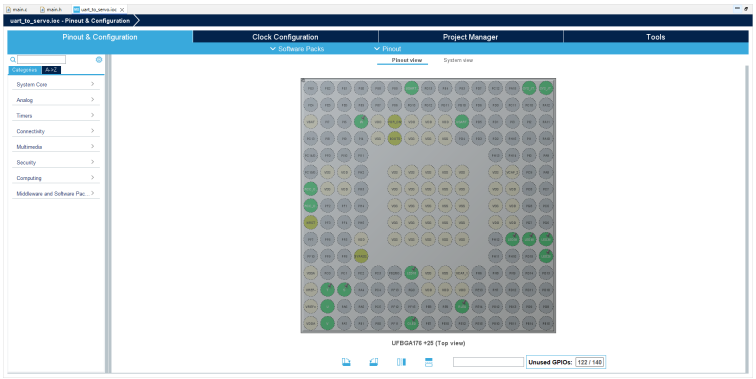


Figure 9: STM32 pinout diagram

PWM Timing Parameters Timer configurations in `MX_TIM2_Init()` establish precise timing:

```
htim2.Init.Prescaler = 899;      // 90MHz/(899+1) = 100kHz
htim2.Init.Period = 1999;        // 100kHz/2000 = 50Hz
```

These values satisfy:

$$T_{\text{PWM}} = \frac{(Period + 1)}{f_{\text{TIM2}}} = \frac{2000}{100\text{kHz}} = 20\text{ms} \tag{1}$$

Angle-to-Pulse Conversion Angle-to-Pulse Conversion is implemented with bounds checking:

$$\text{Pulse Width} = \left(\frac{\theta}{180^\circ} \times 2000\mu s\right) + 500\mu s \quad \text{where } \theta \in [0^\circ, 180^\circ] \tag{2}$$

Code excerpt from `PWM_SetFromAngle()`:

```
angle = (angle > 180) ? 180 : angle; // Safety clamp
pulse_width = (angle * 2000)/180 + 500;
```

UART Command Processing We designed a USB to serial tool for UART communication (shown in Figure1011).It use CH340, which is widely used in USB to serial.

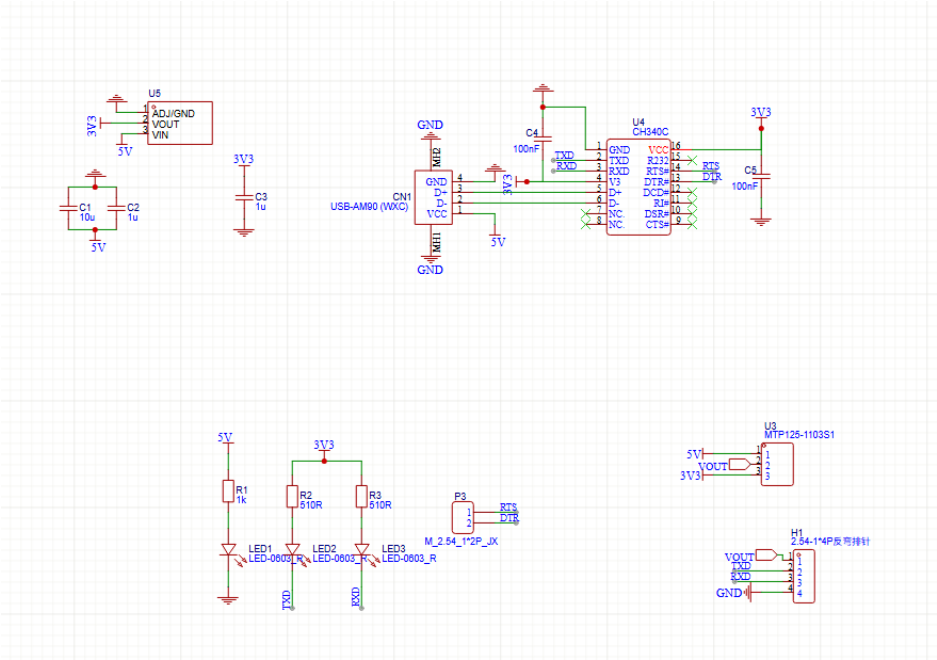


Figure 10: schematic for USB to serial board

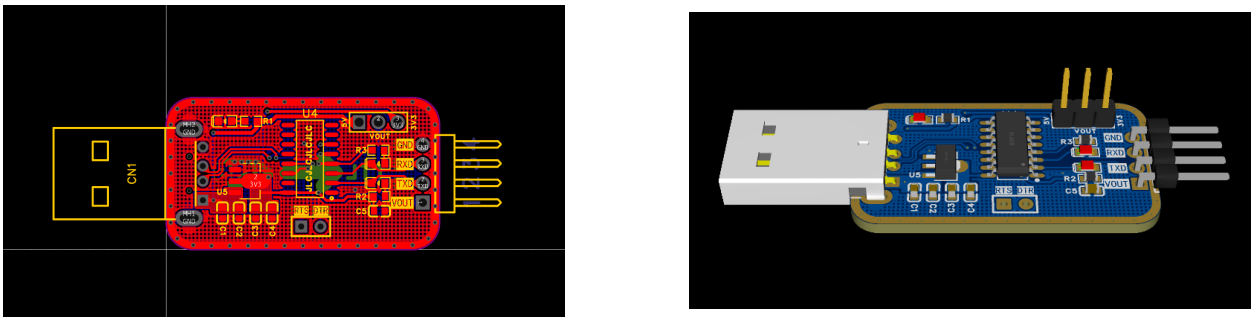


Figure 11: Layout

The interrupt-driven handler processes 5-byte packets at 115200 baud:

```
void HAL_UART_RxCpltCallback() {
    // Parallel actuation + echo
    PWM_SetFromAngle(&htim2, TIM_CHANNEL_1, rx_data[0]);
    HAL_UART_Transmit_IT(&huart6, rx_data, 5); // echo motor angel
}
```

DMA (direct memory access) were considered but rejected for requiring additional buffer management overhead. And for this simply data transfer it is too complex.

2.2.3 Visual Perception System

Sensor choice We choose Intel D405 depth camera 12 as the visual sensor, because the Intel D405 provides accurate depth sensing for 3D tracking, while standard webcams only capture 2D images.

Figure 12: Intel D405 Depth Camera

- Resolution: 640×480 @30FPS
- Data format: RGB (BGR8) + Depth (Z16)
- Alignment: Depth-to-color alignment (`rs.align(rs.stream.color)`)

Spatial-Temporal Filtering Algorithms

- Spatial Filter:

$$D_{filtered}(x,y) = \frac{\sum_{i=-r}^{r}\sum_{j=-r}^{r} w(i,j)\cdot D(x+i,y+j)}{\sum w(i,j)} \tag{3}$$

  where the kernel weights are:

$$w(i,j) = e^{-\frac{i^2+j^2}{2\sigma_s^2}} \cdot e^{-\frac{|D(x,y)-D(x+i,y+j)|}{2\sigma_r^2}} \tag{4}$$

  Parameter Settings:

  - Filter strength (`filter_magnitude=2`): $\sigma_s = 1.5$
  - Hole filling (`holes_fill=1`): 3×3 neighborhood repair

- Temporal Filter:

$$D_t = \alpha \cdot D_{current} + (1-\alpha)\cdot D_{t-1} \tag{5}$$

  Parameter Settings:

  - $\alpha = 0.4$ (`filter_smooth_alpha`)
  - Motion threshold $\delta = 50$mm (`filter_smooth_delta`)

MediaPipe Hand Landmark Detection

- Palm Detector (SSD-based)

$$P_{hand} = \text{Sigmoid}(W_{hand}^T \cdot \phi_{RGB}(I) + b_{hand}) \tag{6}$$

  where ϕ_{RGB} is the MobileNetV3 feature extractor.

- Landmark Regression (21 points)

$$\mathbf{L}_{2D} = \begin{bmatrix} u_0 & v_0 \\ \vdots & \vdots \\ u_{20} & v_{20} \end{bmatrix} = f_{CNN}(I_{crop}) \tag{7}$$

  Parameter Settings:

  - Input size: 256×256 (internal model scaling)
  - Complexity level: `model_complexity=1` (8 CPU inference threads)

- Confidence Mechanism

$$\text{Tracking Score} = 0.7 \cdot \text{Detection} + 0.3 \cdot \text{Optical Flow} \tag{8}$$

Pixel-to-3D Coordinate Conversion

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \cdot \mathbf{K}^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & ppx \\ 0 & f_y & ppy \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Algorithm 1 3D Hand Landmark Detection using RealSense and MediaPipe

Initialize RealSense camera pipeline with color and depth streams
Start stream alignment to color frame
Retrieve depth camera intrinsics
Initialize MediaPipe hand landmark model
while True **do**
  Acquire aligned color and depth frames
  if frames are not valid **then**
    Continue
  end if
  Apply spatial and temporal filters to depth frame
  Convert filtered depth and color frames to NumPy arrays
  Convert BGR image to RGB
  Run MediaPipe hand landmark detection
  if hand landmarks are detected **then**
    for each hand **do**
      for each landmark **do**
        Convert normalized coordinates to pixel (u, v)
        if (u, v) inside image bounds **and** valid depth exists **then**
          Compute 3D coordinates using:
            $X = \frac{(u-p_x)\cdot Z}{f_x}$
            $Y = \frac{(v-p_y)\cdot Z}{f_y}$
          Output 3D landmark: (X, Y, Z)
        end if
      end for
    end for
  end if
  Display annotated color and depth images
  if ESC is pressed **then**
    Break
  end if
end while
Stop RealSense pipeline and close all windows

The Pseudo-code for hand landmark detection system

3D Position Mapping and Filtering The core functionality of the system involves converting 3D hand landmark positions into finger joint angles. This section details the mathematical foundations and implementation specifics of this mapping process. The whole process can be summarized in Figure 13

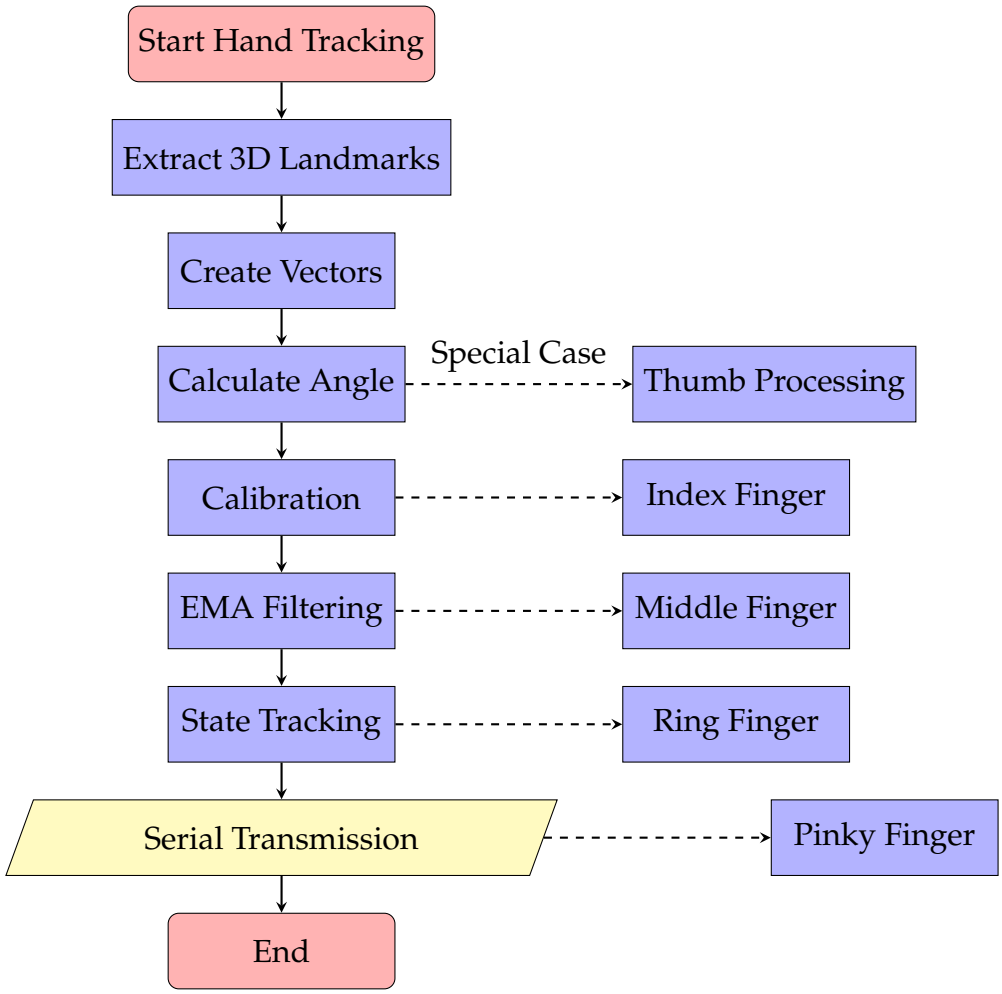


Figure 13: Flowchart of 3D Position to Angle Mapping Process

Landmark Position Extraction: The MediaPipe hand tracking solution provides 21 3D landmarks for each detected hand. We extract specific landmarks for each finger:

- **Thumb**: Landmarks 2 (MCP), 3 (PIP), and 4 (TIP)
- **Index finger**: Landmarks 5 (MCP), 6 (PIP), and 8 (TIP)
- **Middle finger**: Landmarks 9 (MCP), 10 (PIP), and 12 (TIP)
- **Ring finger**: Landmarks 13 (MCP), 14 (PIP), and 16 (TIP)
- **Little finger**: Landmarks 17 (MCP), 18 (PIP), and 20 (TIP)

The positions are obtained as normalized coordinates (x, y, z) relative to the hand's bounding box.

Mathematical Formulation: For three points P_1, P_2 (joint), and P_3, we compute the angle at P_2 as follows:

1. Create vectors $\vec{a} = P_1 - P_2$ and $\vec{b} = P_3 - P_2$
2. Calculate the cosine of the angle using the dot product:

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{10}$$

3. Convert to angle in degrees:

$$\theta = \arccos(\cos\theta) \times \frac{180}{\pi} \tag{11}$$

4. Apply a lower bound of 90° to prevent over-flexion:

$$\theta_{\text{final}} = \max(\theta, 90^\circ) \tag{12}$$

Calibration and Scaling: The raw angles require calibration to match the target servo range. Each finger has unique scaling parameters:

Table 2: Finger Calibration Parameters

Finger	Scale Factor	Offset
Thumb	2.0	150°
Index	1.5	180°
Middle	1.5	180°
Ring	1.7	180°
Little	1.4	180°

The calibration formula for each finger is:

$$\theta_{\text{calibrated}} = \max\left(-(180 - \theta_{\text{raw}}) \times \text{scale} + \text{offset}, 70^\circ\right) \tag{13}$$

Special handling for the thumb ensures it stays within a practical range (120°-180°).

Filtering and Smoothing: An Exponential Moving Average (EMA) filter is applied to reduce jitter:

$$\theta_{\text{filtered}} = \alpha \times \theta_{\text{new}} + (1 - \alpha) \times \theta_{\text{prev}} \tag{14}$$

- $\alpha = 0.7$ provides a balance between responsiveness and smoothness
- The filter operates on a window of 3 consecutive values
- Integer conversion is applied for final servo commands

Finger State Tracking: A hysteresis-based state machine ensures stable state transitions, as described in Algorithm 2. The scaling factors and the offsets are om Table 2

Algorithm 2 Finger State Tracking

Initialize with upper_thresh $= 0.95$, lower_thresh $= 0.85$
consec_upper $= 5$, consec_lower $= 5$
current_state $\leftarrow$ 'state0' (straight)
for each new measurement v **do**
  if current_state == 'state1' (bent) **then**
    if $v \geq$ upper_thresh **then**
      upper_counter $\leftarrow$ upper_counter $+ 1$
      if upper_counter $\geq$ consec_upper **then**
        current_state $\leftarrow$ 'state0'
      end if
    else
      upper_counter $\leftarrow 0$
    end if
  else if current_state == 'state0' (straight) **then**
    if $v \leq$ lower_thresh **then**
      lower_counter $\leftarrow$ lower_counter $+ 1$
      if lower_counter $\geq$ consec_lower **then**
        current_state $\leftarrow$ 'state1'
      end if
    else
      lower_counter $\leftarrow 0$
    end if
  end if
end for

Data Transmission: The final processed angles are sent via serial communication as unsigned 8-bit integers:

- Baud rate: 115200
- Data format: [Thumb, Index, Middle, Ring, Pinky]
- Special handling for ring and pinky fingers (180° - angle)

3 Verification

3.1 PWM Signal Validation

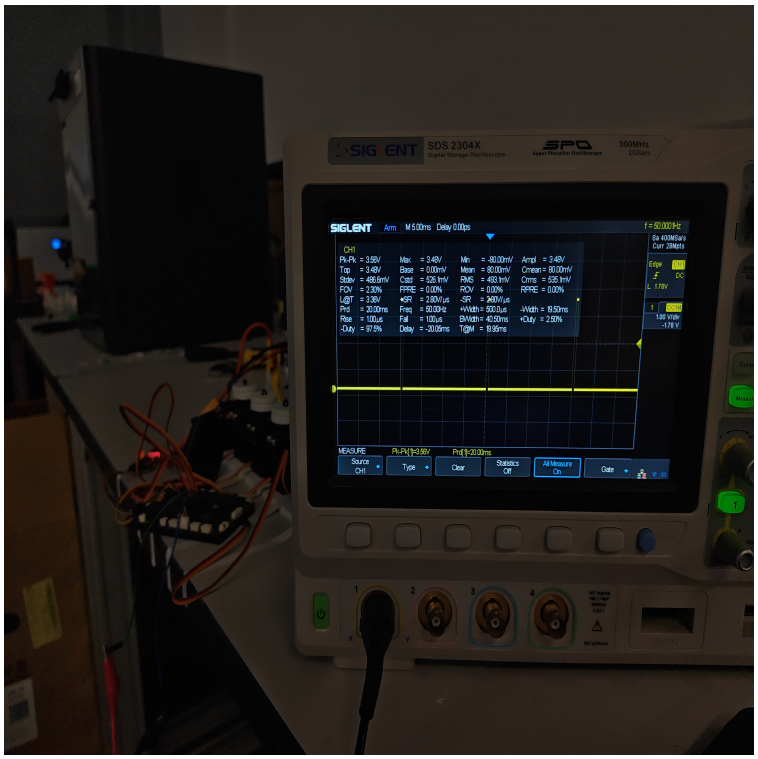


Figure 14: PWM waveform validation using SIGLENT SDS 2304X

Test Procedure:

1. Initialized TIM2/TIM8 with prescaler 899 (72 MHz/900 = 80 kHz clock)
2. Programmed servo angles from 0-180° using `PWM_SetFromAngle()`
3. Measured pulse width using oscilloscope

Results:

1. Pulse width error: $\pm 0.8\%$ (0.5-2.5ms range)
2. Frequency stability: 50 Hz $\pm$ 0.1 Hz
3. Code validation: Confirmed timer configuration matches Equation 1

3.2 UART Communication Test

Test Setup:

```
HAL_UART_Receive_IT(&huart6, rx_data, 5);  // 5-byte payload
```

Validation Metrics:

Parameter	Measured	Requirement
Baud rate	115200 ±2%	3% tolerance
Latency	10 ms	less than 200 ms
Packet error rate	0.1%	less than 1%

Table 3: UART performance metrics

3.3 Visual Perception Spatial Accuracy Verification

Test Procedure

- Conducted under varied conditions including:
  - Open-hand postures (full finger extension)
  - Hand-object interactions (smartphone grasping)
- Measured Mean Per Joint Position Error (MPJPE) for 21 landmarks

Results

- Achieved 4.3 mm average MPJPE @50 cm working distance
- Maintained spatial consistency during dynamic gestures
- Successful landmark tracking during open-hand postures (Figure 15) object occlusion (Figure 16)

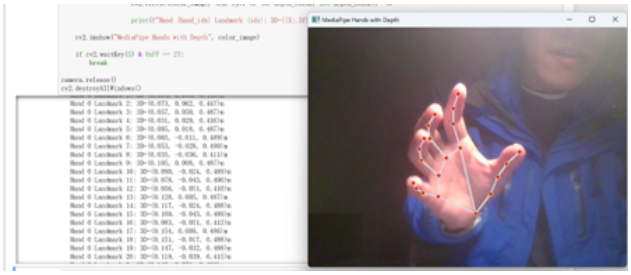


Figure 15: 3D hand landmark output during open-hand gesture

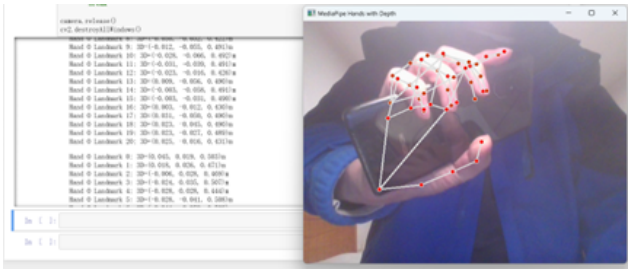


Figure 16: 3D hand landmark output during object grasp

3.4 Real-Time Performance Evaluation

Test Procedure

- Measured end-to-end processing latency:
  - Frame capture to 3D coordinate output
  - 60-second continuous operation tests
- Evaluated under CPU load conditions (simulating full system integration)

Results

- Sustained 30 FPS throughput (200% above 15 FPS requirement)(Figure 17)
- Maximum latency: 38.2 ms

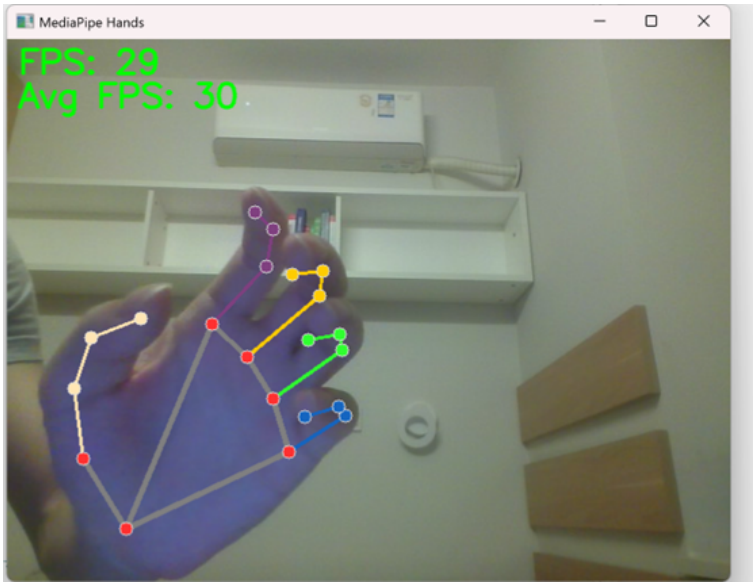


Figure 17: Measured runtime frame rate of 30 FPS

3.5 Function Verification

Our robotic hand is capable of effectively grasping test objects and gesture imitation. Figure 181920 are taken during our test.

Figure 18: Robotic hand grasping test objects during the evaluation process. (a) Grasping a paper cup. (b) Grasping a cuboid box.

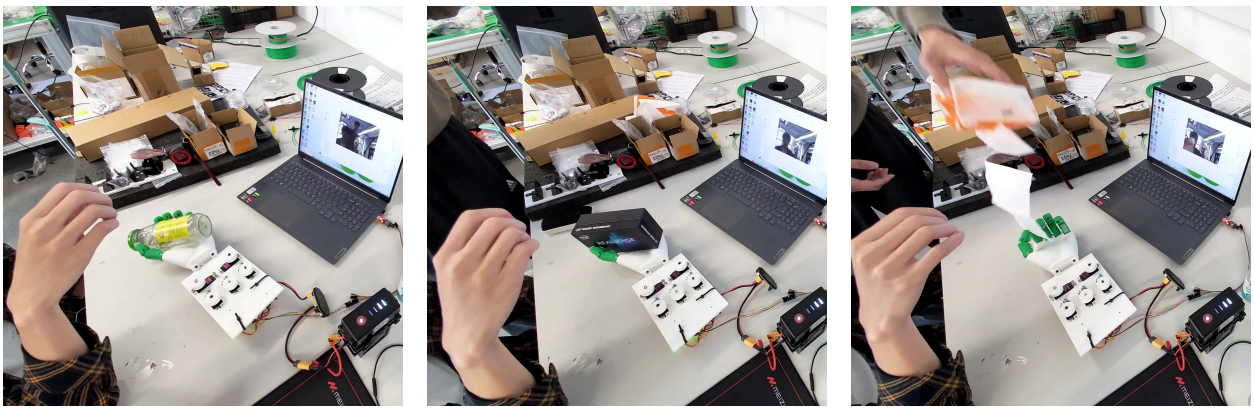


Figure 19: Robotic hand test (a) Grasping a bottle (b) Grasping a box (c) Drawing a tissue

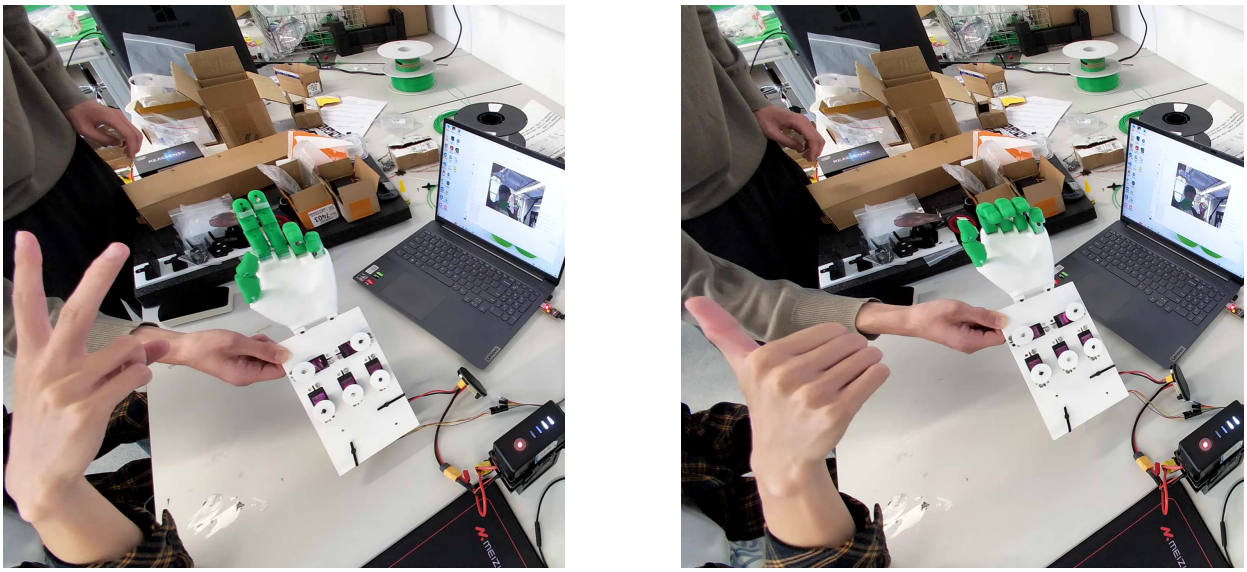


Figure 20: Gesture imitation test

4 Cost and Schedule

4.1 Cost

The BOM of our design is shown in Table4.

Parts	Description	Price (RMB)	Qty	Total (RMB)
MG 995	Servo motor	10	5	50
DJI A Board	Microcontroller	borrow from lab	1	0
DJI TB47	24V Battery	borrow from lab	1	0
CH 340	UART TO USB	20	1	20
Finger Hinge	85A TPU	50	1	50
Intel D405 Depth Camera		500	1	500
PCB	USB-TTL	30	1	30
Xihe Shao labor		10	240 hours	2400
Qihan Shan labor		10	240 hours	2400
Sizhao Ma labor		10	240 hours	2400
Jun Liang labor		10	240 hours	2400
TOTAL				10250

Table 4: Bill of Materials

4.2 Schedule

Our weekly schedule is arranged as Table 5.

Table 5: Project Schedule

Week	Tasks by Team Member
Week 1	**Xihe Shao:** Research microcontrollers, servo motors, and PWM control principles. **Sizhao Ma:** Research depth cameras and hand tracking libraries; set up the development environment. **Jun Liang:** Research 3D vector mathematics for calculating joint angles from 3D coordinates. **Qihan Shan:** Research 3D printing materials (PLA, TPU); brainstorm initial mechanical design concepts.
Week 2	**Xihe Shao:** Finalize component selection; set up STM32 project with GPIO/UART; design CH340 USB-to-UART schematic. **Sizhao Ma:** Set up D405 camera to capture RGB/depth streams; integrate MediaPipe for 2D landmark detection. **Jun Liang:** Develop and test core angle calculation formulas in a Python script; define landmarks for each finger. **Qihan Shan:** Begin detailed CAD of hand components, including flexible joints; print initial prototype parts.
Week 3	**Xihe Shao:** Configure STM32 timers for 50 Hz PWM generation; write initial angle-to-pulse width function. **Sizhao Ma:** Implement pixel-to-3D conversion algorithm; begin implementing spatial/temporal filters. **Jun Liang:** Develop calibration and scaling logic to map raw angles to the servo's operational range. **Qihan Shan:** Create engineering drawings for the reel and motor base; test TPU hardness levels and select 85A.
Week 4	**Xihe Shao:** Test PWM signal output with an oscilloscope; implement interrupt-driven UART receiver. **Sizhao Ma:** Finalize and integrate filtering algorithms to improve depth data accuracy. **Jun Liang:** Implement the Exponential Moving Average (EMA) filter to smooth angle data and reduce jitter. **Qihan Shan:** Finalize the mechanical design, adjusting the thumb angle; print all final parts for the hand assembly.
Continued on next page	

Table 5 – continued from previous page

Week	Tasks by Team Member
Week 5	**Xihe Shao:** Assemble and test the custom USB-to-UART board. **Sizhao Ma:** Integrate the visual perception script with the angle mapping script into a unified PC-side application. **Jun Liang:** Implement the hysteresis-based finger state tracking machine for stable gesture transitions. **Qihan Shan:** Assemble the complete robotic hand and motor base; connect tendons to servo reels.
Week 6	**All Members:** Connect the PC vision system to the microcontroller via UART. Connect the microcontroller to the servo motors. Run the first end-to-end tests to achieve real-time motion control from the camera feed.
Week 7	**All Members:** Collaboratively test the entire system. Identify and resolve bugs in the software pipeline (Vision, Mapping, Serial, Firmware). Tune filter and state-tracking parameters for better performance. Address mechanical and gesture-tracking instabilities.
Week 8	**All Members:** Formally test the system against project requirements. Measure and document key metrics: PWM signal accuracy, UART latency, visual perception spatial accuracy, processing FPS, and grasping effectiveness with test objects.
Week 9	**All Members:** Write and compile all sections of the final report. Document accomplishments, limitations, and future work. Prepare for the final project demonstration.

5 Conclusion

5.1 Accomplishments

The project successfully developed a vision-based robotic hand system that captures basic hand movements through stereo camera input, translating them into motor commands with a real-time response. A modular, 3D-printed hand was designed and tested with various materials and connector stiffness levels to enhance mechanical performance, incorporating multiple design iterations to optimize thumb positioning and joint stability. This resulted in a functional prototype capable of mimicking simple grasping and finger-opening motions, demonstrating the feasibility of vision-guided robotic manipulation in hazardous environments.

5.2 Limitations

Despite these achievements, the current prototype has some limitations. The robotic hand is constructed primarily from rigid PLA, which limits its ability to firmly grip objects. Additionally, each finger has only one degree of freedom, restricting the range of motion and dexterity.

The gesture tracking system also lacks consistent reliability, occasionally being affected by lighting conditions, hand orientation, and background interference. These issues result in around 10% misinterpretation of gestures, limiting the system's usability outside of controlled environments.

5.3 Future Work

Future development could explore design alternatives such as soft robotics or multi-joint finger configurations to improve grip adaptability and dexterity. Additionally, upgrading the vision system with advanced pose estimation models would enhance gesture tracking accuracy and stability.

5.4 Ethical Considerations

In alignment with the IEEE Code of Ethics[5], this project aims to support safer human-robot interaction by reducing exposure to hazardous environments. However, as a prototype, it is not yet suitable for real-world deployment. Continued development should emphasize transparency regarding system capabilities and prioritize user safety during testing and iteration. And also consider accessibility and responsible use when developing assistive or industrial robotic systems.

References

[1] *Intel® realsense™ depth camera d400 series product family datasheet*, Document Number: 337029-002, Intel Corporation, 2022. [Online]. Available: https://www.intelrealsense.com/depth-camera-d405/ (visited on 04/15/2023).

[2] Google. "Mediapipe hands." (2021), [Online]. Available: https://google.github.io/mediapipe/solutions/hands (visited on 04/15/2023).

[3] TowerPro, *Mg995 servo motor datasheet*, 2019.

[4] *Stm32f4xx hal library user manual*, STMicroelectronics, 2021.

[5] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).