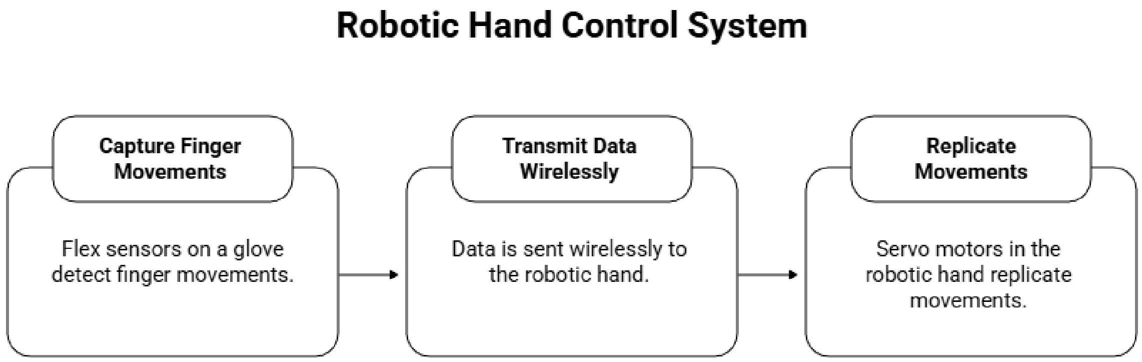
ROBOTIC ARM : EXOGENESIS

Introduction

In recent years, robotic systems have gained importance in medical, industrial, and educational fields. A robotic hand that mimics human finger movements can be useful in rehabilitation, prosthetics, automation, and research. This project focuses on designing and developing a wireless robotic hand using flex sensors and ESP32 microcontrollers. The system captures finger movements using flex sensors mounted on a glove and transmits the data wirelessly to a robotic hand. The robotic hand then replicates the same movements using servo motors.



Objectives of the Project

The main objectives of this project are:

- To design a wearable glove using flex sensors for motion detection.
- To transmit finger movement data wirelessly using ESP-NOW.
- To control servo motors for realistic finger movement.
- To build a low-cost and efficient robotic hand system.
- To ensure reliable and real-time performance.

System Overview

The system is divided into two main parts:

1. Sender Unit (Glove Side)
2. Receiver Unit (Robotic Hand Side)

1. Sender Unit

The sender unit consists of:

- ESP32 microcontroller
- Flex sensors
- Voltage divider circuit
- Power supply

It reads the resistance of flex sensors and converts it into digital values.

2.Receiver Unit

The receiver unit consists of:

- ESP32 microcontroller
- Servo motors
- Robotic hand structure
- Power supply

It receives wireless data and controls the servo motors accordingly.

Robotic Arm System Overview

Hardware Components

1 ESP32 Microcontroller

- Dual-core processor
- Built-in WiFi and Bluetooth
- Supports ESP-NOW protocol
- Low power consumption

2 Flex Sensors

- Resistance varies with bending
- Used for detecting finger movement
- Connected using voltage divider

3 Servo Motors

- Used to control finger movement
- Angle range: 0° to 180°
- Controlled using PWM signals

Software Requirements

- Arduino IDE
- ESP32 Board Package
- ESP32Servo Library
- ESP-NOW Library

Working Principle

1. Flex sensors change resistance when bent.
2. ESP32 reads analog values using ADC.
3. Resistance is calculated using voltage divider formula.
4. Noise is reduced using smoothing techniques.
5. Finger states (bent/flat) are determined using hysteresis logic.
6. Data is transmitted using ESP-NOW.
7. Receiver ESP32 receives data.
8. Servo motors rotate according to received values.
9. Robotic hand mimics human hand movement.

Code Implementation and Hardware Setup

1 Sender (Glove Unit) Code

The sender unit reads flex sensor values, processes them, and transmits data wirelessly using ESP-NOW.

Main Functions:

- Reads analog values from flex sensors
- Applies smoothing and hysteresis
- Calculates resistance
- Sends finger states to receiver

(Refer to project source code for complete implementation.)

2 Receiver (Robotic Hand Unit) Code

The receiver unit receives data and controls servo motors accordingly.

Main Functions:

- Initializes ESP-NOW
- Receives finger state data
- Controls servo motors using PWM
- Implements safety timeout

(Refer to project source code for complete implementation.)

3 Software Setup Procedure

1. Install Arduino IDE.
2. Install ESP32 board package.
3. Install ESP32Servo library.
4. Connect ESP32 to computer.
5. Select correct COM port and board.
6. Upload receiver code first.
7. Note down MAC address.
8. Update sender code with receiver MAC.
9. Upload sender code.
10. Open Serial Monitor for debugging.

4 Hardware Connections

Flex Sensor Connections (Sender)

- One end of flex sensor → 3.3V
- Other end → Analog Pin (GPIO)
- Fixed resistor (200kΩ) → Ground
- Voltage divider output → ESP32 ADC pin

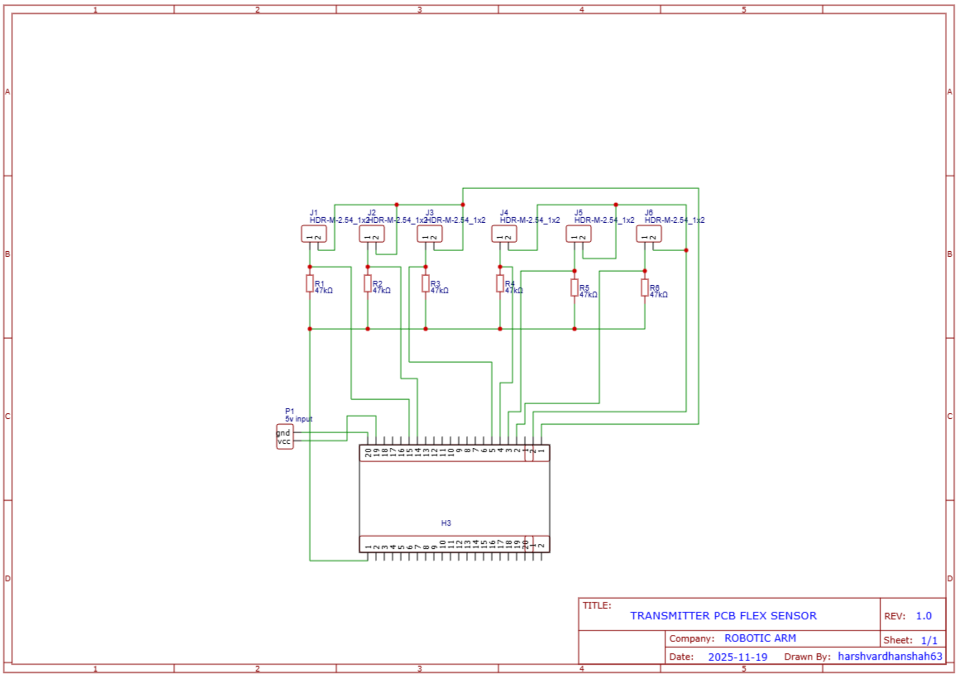
Servo Motor Connections (Receiver)

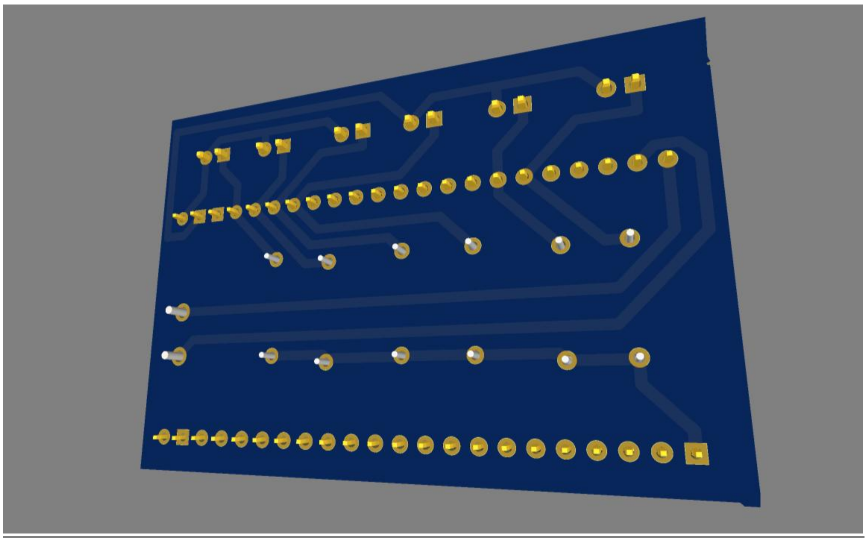
- Red wire → 5V external supply
- Brown/Black wire → Ground
- Yellow/White wire → ESP32 PWM pin
- Common ground between ESP32 and power supply

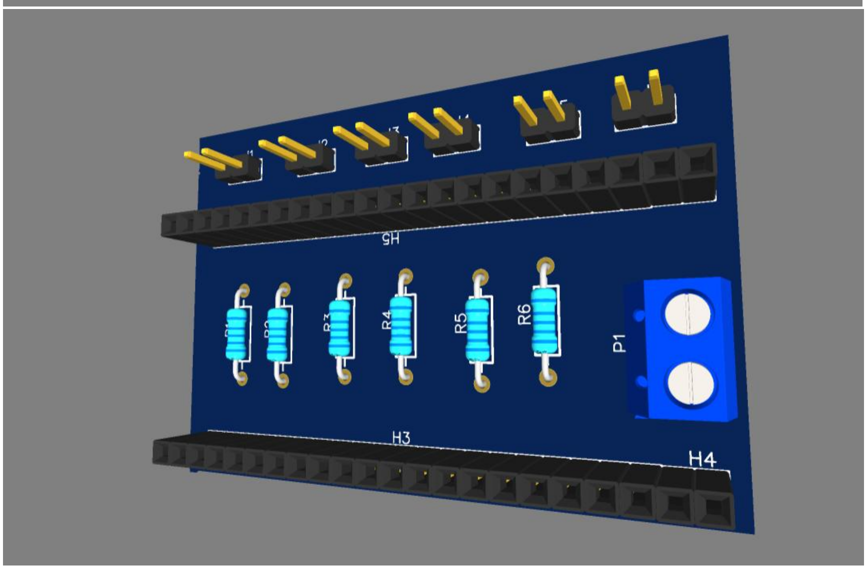
Power Supply

- ESP32 powered via USB or 5V regulator

- Servos powered using external 5V supply

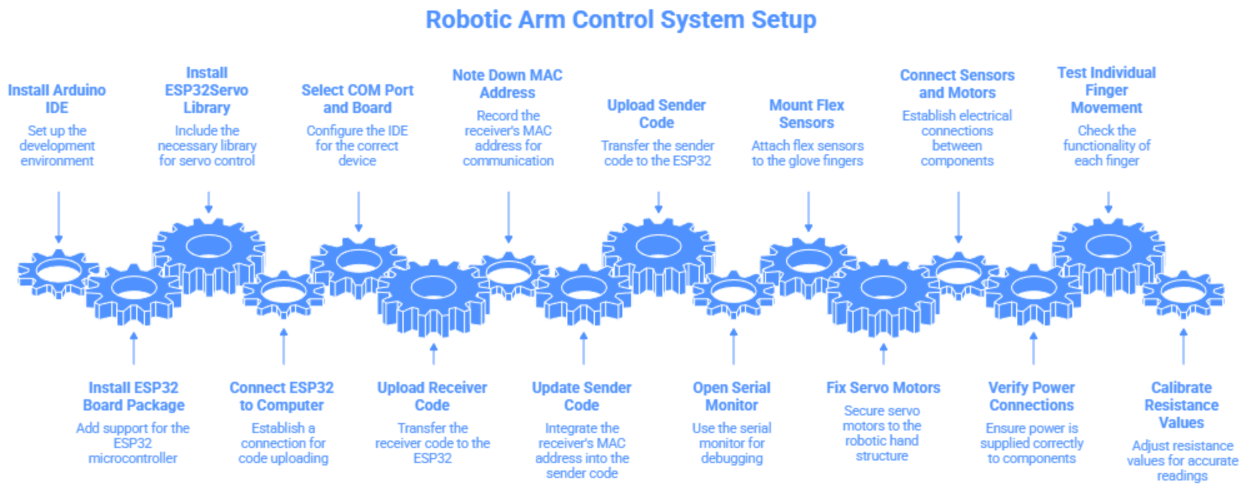




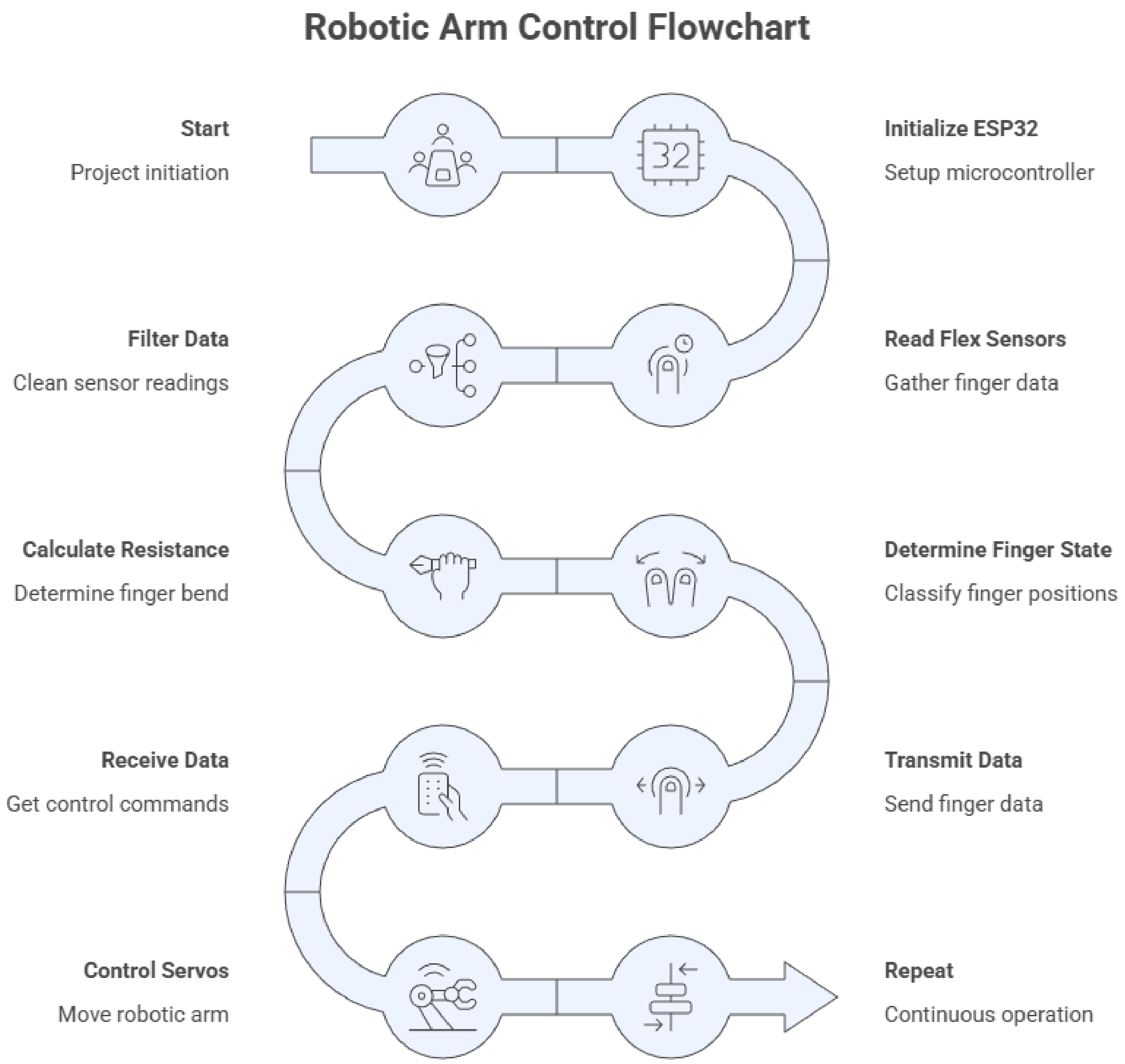


5 System Setup Steps

1. Mount flex sensors on glove fingers.
2. Fix servo motors on robotic hand structure.
3. Connect all sensors and motors.
4. Verify power connections.
5. Upload codes.
6. Test individual finger movement.
7. Calibrate resistance values.
8. Perform wireless testing.
9. Final system integration.



Flowchart



Communication Method (ESP-NOW)

ESP-NOW is a low-latency wireless communication protocol developed by Espressif. It enables peer-to-peer communication without using a router.

Features:

- Low delay
- Low power
- High reliability
- No internet required

Algorithms Used

1 Smoothing Algorithm

Moving average filter is used to reduce noise in sensor readings.

2 Hysteresis Algorithm

Two thresholds (R_LOW and R_HIGH) are used to prevent frequent switching.

3 Servo Control Algorithm

Finger states are mapped to servo angles for realistic movement.

Results and Observations

- The robotic hand accurately follows human finger movements.
- Wireless transmission is stable within 20–30 meters.
- System response time is less than 50 ms.
- Power consumption is moderate.
- System is reliable for continuous operation.

Applications

- Prosthetic hand control
- Rehabilitation systems
- Human–machine interaction
- Robotics research
- Educational projects

Advantages

- Low cost
- Wireless operation
- Real-time control
- Portable
- Easy to modify

Limitations

- Limited range
- Requires calibration
- Sensitive to noise

Future Scope

- Mobile app control
- AI-based gesture recognition

- Haptic feedback
- Cloud monitoring

References

1. ESP32 Datasheet – Espressif Systems
2. Arduino Official Documentation
3. ESP-NOW Programming Guide
4. Robotics and Sensor Technology Books
5. ESP32 Datasheet – Espressif Systems
6. Arduino Official Documentation
7. ESP-NOW Programming Guide
8. Robotics and Sensor Technology Books

Conclusion

This project successfully demonstrates the design and implementation of a wireless robotic hand using ESP32 and flex sensors. The system is efficient, low-cost, and suitable for academic and research purposes. It provides a strong foundation for advanced robotic and prosthetic applications.

YOUTUBE VIDEOS LINK TESTING AND PROTOTYPE

1. https://youtube.com/shorts/7Oa2iwueIwg?si=Fzh45fDLNvpiwW9M
2. https://youtube.com/shorts/HdQvleQe6Do?si=t_su_lGh6K9XxUNW

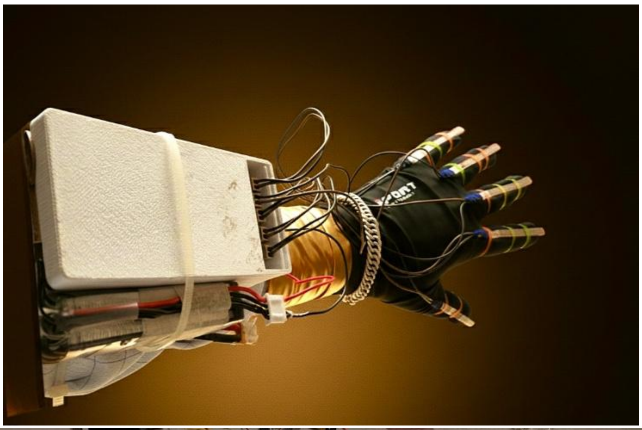
LINKEDIN POST

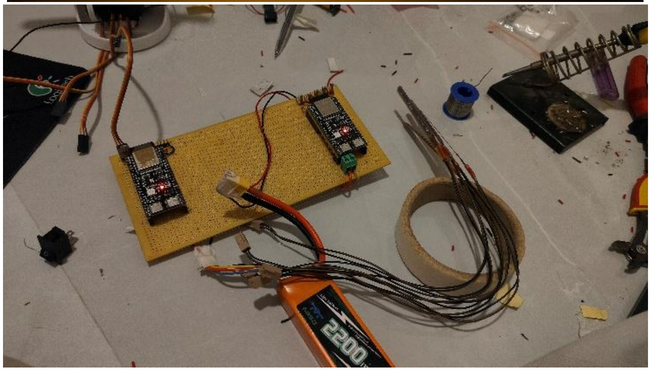
1. https://www.linkedin.com/posts/harshvardhanshah63_robotics-embeddedsystems-hardwareprojects-activity-7410461919540752384-7Hrv?utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAF4nhrcB4VpFJhVOthZDY2YUtMuMLE54Vss&utm_campaign=copy_link
2. https://www.linkedin.com/posts/harshvardhanshah63_roboticsdemo-mechatronics-esp32-activity-7410465693500022785-tMFq?utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAF4nhrcB4VpFJhVOthZDY2YUtMuMLE54Vss&utm_campaign=copy_link

GITHUB REPOSITORY

https://github.com/harshvardhanshah63/Robotic-Arm-Esp-Now-Flex-Sensor-

PRODUCT IMAGES




2200