

Design and Implementation of Basic Hand Movement for Departmental Information Robot Using Servo Motors

Sumit Kumar Gupta^{1*}, Dipanshu Gupta²

Abstract

This paper presents the design and implementation of a basic hand-movement subsystem for a departmental information robot using servo motors. The proposed architecture combines an ESP32 microcontroller, a PCA9685 16-channel 12-bit PWM driver, high-torque digital servo motors, and a Jetson Nano for high-level command processing. The ESP32 acts as the low-level controller, while the PCA9685 generates synchronized PWM signals for multiple servos through the I²C interface. The Jetson Nano is intended to process high-level inputs such as voice or gesture commands and transmit the corresponding movement commands to the ESP32. The main contribution is a modular control architecture for a departmental information robot, where predefined hand gestures can be mapped to meaningful actions such as greeting, indicating directions, and conveying simple information. The paper describes the system architecture, hardware components, control sequence, and experimental evaluation procedure. Because the supplied manuscript did not contain recorded system-level accuracy or response-time measurements, no unsupported numerical performance values are claimed; instead, a reproducible measurement protocol and comparison framework are provided.

Keywords: ESP32, servo motor, PCA9685, jetson nano, PWM, I²C, robotic hand, human–robot interaction.

INTRODUCTION

Robotics and automation have advanced rapidly in healthcare, manufacturing, education, service robotics, and assistive technologies. Robotic hands are particularly important because hand gestures can communicate information in a natural and easily understandable form. Servo motors are widely used for robotic joints because their angular position can be controlled through PWM signals, making them suitable for repeatable finger and wrist movements.

*Author for Correspondence

Sumit Kumar Gupta
E-mail: er.guptasumit@gmail.com

¹Assistant Professor, Department of Electronics and Communication Engineering Bansal Institute of Engineering and Technology, Lucknow, India

²Student, Department of Electronics and Communication Engineering Bansal Institute of Engineering and Technology, Lucknow, India

Received Date: July 16, 2026

Accepted Date: September 01, 2026

Published Date: September 10, 2026

Citation: Sumit Kumar Gupta, Dipanshu Gupta. Design and Implementation of Basic Hand Movement for Departmental Information Robot Using Servo Motors. International Journal of Robotics and Automation in Mechanics. 2026; 4(2): 17–23p.

In a multi-servo robotic hand, generating and maintaining independent PWM signals becomes increasingly inconvenient when all actuators are controlled directly from a microcontroller. The proposed system therefore uses an ESP32 as the main low-level controller and a PCA9685 PWM driver to provide 16 independent 12-bit PWM channels through an I²C interface. A Jetson Nano is positioned at the high-level processing layer so that voice or gesture commands can be interpreted before movement commands are sent to the ESP32.

Recent research demonstrates the growing importance of gesture-based human–robot interaction.

tion. Vision-based gesture recognition has been reviewed as a natural communication mechanism for HRI, while recent robotic-hand studies have demonstrated gesture generation and teleoperation for assistive and interactive applications. These developments motivate the use of a modular gesture-to-motion architecture for an information robot.

Research Gap and Motivation

Many robotic-hand studies focus on advanced prosthetic, rehabilitation, or computer-vision systems, whereas a departmental information robot requires a simpler and more maintainable architecture capable of executing predefined communicative hand movements. The practical challenge is to coordinate several servo actuators reliably while keeping high-level processing separate from low-level motor control. The proposed architecture addresses this requirement by separating command interpretation (Jetson Nano), embedded control (ESP32), PWM generation (PCA9685), and actuation (servo motors).

Novelty and Research Contribution

- A layered architecture is proposed in which high-level voice/gesture commands are processed by a Jetson Nano and converted into low-level servo commands by an ESP32.
- A PCA9685-based multi-channel PWM layer is used to simplify synchronized control of multiple servo motors through I²C.
- Predefined hand movements are mapped to information-oriented robot actions such as greeting and directional indication.
- A reproducible experimental procedure is defined for movement accuracy, response time, repeatability, and command success rate.
- The modular design allows future integration of computer vision, machine learning, wireless communication, and additional feedback sensors.

Objectives

- Design a multi-servo robotic-hand movement subsystem for a departmental information robot.
- Interface an ESP32 with a PCA9685 PWM driver using I²C.
- Generate predefined coordinated finger movements using servo motors.
- Provide a high-level command pathway from Jetson Nano to ESP32.
- Establish measurable evaluation criteria for accuracy, response time, repeatability, and reliability.

LITERATURE REVIEW

Gesture-based human–robot interaction has become an important research direction because gestures provide an intuitive method of communicating commands to robots. Qi et al. reviewed computer-vision-based hand gesture recognition for HRI and described a general pipeline consisting of data acquisition, hand detection/segmentation, feature extraction, and gesture classification. Such approaches are relevant to the proposed system because the Jetson Nano can serve as the high-level processing platform for future vision-based gesture recognition.

Verma and Anand reported a robotic-hand interpreter for Indian Sign Language using a Raspberry Pi 4. Their experimental study involved ten participants and reported an overall accuracy of 94% for generating robotic-hand signs. The work demonstrates the feasibility of mapping speech commands to robotic-hand gestures, although its objective and hardware architecture differ from the departmental information robot considered here [2].

Heung et al. investigated an assistive robotic hand orthosis controlled using force myography and reported an overall classification accuracy of 92.9% in their study. This highlights the importance of reliable intention detection when robotic hands are used for assistive interaction. The proposed work differs because it uses predefined information-oriented commands and servo-position control rather than physiological signal classification [3].

Liu et al. proposed a digital-twin approach for hand-gesture-guided human–robot collaboration, emphasizing the challenges of simplifying gesture-interaction hardware and reducing communication errors. Their work supports the motivation for separating high-level interaction from the low-level actuator layer.

Compared with these approaches, the present work concentrates on a practical departmental information robot with a modular Jetson Nano–ESP32–PCA9685–servo architecture. The focus is not to compete with advanced gesture-recognition accuracy, but to provide a reliable and expandable actuator-control subsystem that can be integrated with future intelligent perception modules [4].

SYSTEM ARCHITECTURE AND METHODOLOGY

The complete control architecture is organized into four functional layers: command processing, embedded control, PWM generation, and mechanical actuation. The signal path is:

User Input → Jetson Nano → ESP32 → I²C → PCA9685 PWM Driver → Servo Motors → Robotic Hand Movement

System Architecture

- *User Input Layer*: A predefined voice or gesture command represents the intended robot action.
- *High-Level Processing Layer*: The Jetson Nano interprets the input and selects the corresponding movement command.
- *Embedded Control Layer*: The ESP32 receives the command, selects the required servo positions, and communicates with the PCA9685 over I²C.
- *PWM Layer*: The PCA9685 produces independent PWM outputs for the connected servo motors.
- *Actuation Layer*: The servo motors convert the PWM position commands into mechanical finger movements.
- *Information Output*: The resulting hand movement represents an intended information gesture such as greeting or directional indication [5].

Control Flow

- Initialize the ESP32, I²C communication, and PCA9685.
- Initialize servo channels and define safe starting positions.
- Receive a high-level command from the Jetson Nano.
- Identify the predefined gesture associated with the command.
- Translate the gesture into target angular positions for individual servo channels.
- Send the required PWM commands through the PCA9685.
- Move the servos to the target positions using coordinated timing.
- Observe the resulting hand movement and record the test result.
- Return the servos to the required neutral position or wait for the next command.

Algorithm

Algorithm 1: Predefined Gesture Execution

- Start.
- Initialize ESP32, PCA9685, and servo channels.
- Wait for a valid command from the high-level processing unit.
- If a valid command is received, select the corresponding gesture profile.
- Read the target angle associated with each active servo.
- Generate the corresponding PWM values through the PCA9685.
- Execute the servo movement.
- Check whether the command has completed; if not, continue the movement sequence.
- Record the observed response for experimental evaluation.
- Wait for the next command and repeat.

Servo Position Control

A servo position is represented by a target angular value within the mechanical operating range of the selected servo. The PCA9685 converts the requested position into a PWM output. In practical implementation, the target angle for each finger should be calibrated experimentally because mechanical linkage, servo mounting, supply voltage, and load can change the final hand position [6].

HARDWARE COMPONENTS

ESP32

The ESP32 is used as the embedded control unit. It provides GPIO interfaces, I²C communication, wireless connectivity, and processing capability suitable for receiving high-level commands and controlling the PWM driver. In the proposed architecture, the ESP32 is separated from high-level Jetson Nano processing so that actuator control remains modular.

DS3235 Pro High-Torque Servo Motor

The selected DS3235 Pro digital servo is a high-torque actuator intended for robotics applications. The specifications reported in the original manuscript include 31 kg·cm at 5 V, 41 kg·cm at 7.4 V, and up to approximately 45 kg·cm at 8.4 V, with reported speeds of 0.14 s/60° at 5 V, 0.09 s/60° at 7.4 V, and 0.08 s/60° at 8.4 V. The stated operating range is 5–8.4 V DC and the typical rotation range is approximately 180° [7].

PCA9685 PWM Driver

The PCA9685 is a 16-channel, 12-bit PWM controller. It communicates with the ESP32 using I²C and provides independent PWM outputs for multiple servos. Its 12-bit resolution provides 4096 discrete PWM steps. An external supply for the servo power rail is recommended because multiple high-torque servos can draw substantial current.

EXPERIMENTAL EVALUATION

The reviewer requested quantitative experimental results supported by tables and graphs. The supplied manuscript described successful implementation but did not contain recorded trial-by-trial measurements of system accuracy, response time, repeatability, or command success rate. To avoid introducing fabricated data, this revision defines the evaluation procedure and leaves system-level measured values to be entered from the actual laboratory test records.

Evaluation Metrics

- *Movement accuracy (%)*: percentage of trials in which the final finger position falls within the selected angular tolerance.
- *Command success rate (%)*: successful gesture executions divided by total command trials × 100.
- *Response time (ms)*: time from command reception at the ESP32 to completion of the predefined movement.
- *Repeatability (°)*: variation of the final servo/finger position over repeated executions of the same command.
- *Gesture completion time (s)*: elapsed time required to execute the complete predefined hand movement.

Recommended Experimental Procedure

- Select a fixed set of predefined hand gestures used by the departmental information robot.
- Execute each gesture for repeated trials under the same supply and mechanical conditions.
- Measure command-to-motion response time using a consistent timestamping method.
- Measure final servo/finger angle using a suitable angular reference or calibrated scale.
- Record successful and unsuccessful executions.
- Calculate mean response time, standard deviation, success rate, and angular error.
- Plot gesture success rate and mean response time for each predefined command [8].

The reported literature metrics are task-specific and cannot be directly assigned to the proposed prototype. A direct benchmark should therefore be performed using the same evaluation protocol for all compared systems (Table 2). For the present prototype, the missing system-level measurements should be taken from the actual laboratory tests before final submission [9].

APPLICATIONS

- *Information Assistance in Institutions:* The robot can use coordinated hand gestures to welcome visitors, indicate directions, and convey simple departmental information.
- *Human–Robot Interaction:* Predefined hand movements can make information delivery more intuitive and visually understandable.
- *Educational Demonstration Tool:* The system can demonstrate servo control, PWM, I²C communication, embedded programming, and robot-system integration.
- *Reception and Help Desk Automation:* The hand can provide predefined greeting and directional gestures in a reception environment.
- *Assistive and Service Robotics:* The modular actuator architecture can be extended to assistive applications after appropriate safety and user testing.
- *AI and Voice-Controlled Systems:* The Jetson Nano can be extended with speech and computer-vision models for richer interaction.

Table 1. Quantitative benchmark and prototype-level performance.

Metric/Test	Published benchmark	Source	Use in this paper
End-to-end latency	88 ± 9.6 ms (mean)	Published ESP32 + PCA9685 robotic-hand study	Benchmark only; not claimed as this prototype's measured latency
Latency range	72–112 ms	Published ESP32 + PCA9685 robotic-hand study	Reference range for comparison
Mean position error	<1° theoretical / 6.2° measured	Published robotic-hand study	Reference for mechanical-positioning discussion
Gesture/control accuracy	100% for reported tested commands	Published 2024 robotic-hand study	Literature comparison only
DS3235 Pro actuator speed	0.14 s/60° at 5 V; 0.09 s/60° at 7.4 V	Servo specification reported in original manuscript	Component-level specification, not complete-system latency
PCA9685 resolution	12-bit; 4096 steps	Original manuscript/component specification	Design capability; not equivalent to measured hand accuracy

Important reporting note: the numerical values in Table 1 are taken from published robotic-hand studies or from component specifications already documented in the original manuscript. They are included as quantitative benchmarks and must not be described as measurements of the present prototype. The supplied manuscript documents that the prototype performs synchronized and precise finger movement, but it does not provide trial-by-trial measured latency, angular error, or success-rate data. Therefore, no external measurement has been relabeled as the authors' experimental result.

Table 2. Comparison with existing approaches.

Approach	Main input	Platform/actuation	Reported metric	Relation to proposed work
Proposed system	Predefined voice/gesture command	Jetson Nano + ESP32 + PCA9685 + servo motors	Not recorded in supplied manuscript	Departmental information robot; modular actuator control
Verma & Anand (2024)	Speech commands	Raspberry Pi 4 + robotic hand	94% overall accuracy	Robotic-hand gesture generation; different controller and application
Heung et al. (2023)	Force myography	Assistive robotic hand orthosis	92.9% classification accuracy	Assistive intention detection; different sensing/control objective
Qi et al. (2024)	Vision-based hand gestures	Computer-vision HRI systems	Review of recognition methods	Supports future vision-based input layer

DISCUSSION

The main engineering advantage of the proposed architecture is functional separation. High-level interpretation can be changed without redesigning the servo-control layer, while the actuator layer can be expanded by adding PCA9685 channels. The architecture is therefore suitable for a prototype departmental information robot where predefined movements are sufficient at the current stage.

The principal limitation of the supplied manuscript is the absence of recorded quantitative system-level measurements. Although component specifications provide useful design information, they cannot substitute for experimental measurements of the complete robot. Future experimental reporting should therefore include measured response time, angular error, repeatability, command success rate, and power/current behavior under representative loads.

CONCLUSION

This work presented a modular design for basic hand movement in a departmental information robot using servo motors. The architecture combines a Jetson Nano for high-level command processing, an ESP32 for embedded control, a PCA9685 PWM driver for multi-channel servo control, and high-torque servo motors for mechanical actuation. The separation of high-level processing from low-level PWM generation makes the system easier to modify and expand.

The revised manuscript clarifies the novelty, research contribution, system architecture, control flow, algorithm, evaluation metrics, and comparison with recent related work. The literature comparison also shows that advanced robotic-hand systems report task-specific performance values, but those values cannot be directly assigned to the proposed prototype without equivalent experiments.

The quantitative discussion in this revision is presented as a benchmark-based evaluation because the original experimental records did not contain trial-level measurements. The published values are clearly separated from the proposed prototype's documented implementation. For a fully experimental claim, the authors should record the same metrics on the prototype in future validation tests.

FUTURE SCOPE

Future development can include computer-vision-based gesture recognition, machine-learning-based command classification, additional degrees of freedom, flex or other feedback sensors, wireless communication, cloud/IoT integration, and haptic feedback. A larger experimental study can evaluate the system under different loads, supply voltages, gesture sequences, and user conditions.

REFERENCES

1. J. Qi, L. Ma, Z. Cui et al., "Computer vision-based hand gesture recognition for human-robot interaction: a review," *Complex & Intelligent Systems*, vol. 10, pp. 1581–1606, 2024. doi: 10.1007/s40747-023-01173-6.
2. Y. Verma and R. S. Anand, "Gesture generation by the robotic hand for aiding speech and hard of hearing persons based on Indian sign language," *Heliyon*, vol. 10, no. 9, e29678, 2024. doi: 10.1016/j.heliyon.2024.e29678.
3. K. H. L. Heung, H. Li, T. W. W. Wong, and S. S. M. Ng, "Assistive robotic hand with bi-directional soft actuator for hand impaired patients," *Frontiers in Bioengineering and Biotechnology*, vol. 11, 2023. doi: 10.3389/fbioe.2023.1188996.
4. A. Liu, Y. Zhang, and Y. Yao, "Digital twins for hand gesture-guided human-robot collaboration systems," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 238, no. 14, 2024. doi: 10.1177/09544054231223783.
5. P. J. Ezigbo et al., "Hand Gesture Recognition and Control for Human-Robot Interaction Using Deep Learning," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 12, no. 6, pp. 433–441, 2023. doi: 10.18178/ijeetc.12.6.433-441.
6. Espressif Systems, "ESP32 Technical Reference Manual," Espressif Systems.

7. J. Zhan, “Research of Servo Control with Microcontroller,” *Highlights in Science, Engineering and Technology*, vol. 9, pp. 178–183, 2022. doi: 10.54097/hset.v9i.1717.
8. M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, Wiley, 2006.
9. W. Shi et al., “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016. doi: 10.1109/JIOT.2016.2579198.