Project Report: 4-Servo Potentiometer Controlled Robotic Arm Using ESP32

Author: MuraleeKrishna VS
Designation: Student [Electronics and communication Engineering 3rd year]

Date: September 2025
Institution / Organization: Cochin University College of Engineering Kuttanadu

Statement of Inspiration:

This project was developed inspired by leading industry robotic arms including Epson Collaborative Robot, WidowX-250, Kinova Gen3, and RobotAnno 6DOF. These products motivated the design, modularity, and future upgrade paths of this prototype, bridging educational robotics and industry-ready automation solutions.

1. Abstract

This project presents the design and implementation of a low-cost **4-DOF robotic arm** using an **ESP32 microcontroller** and **SG90 servo motors**, aimed at providing an educational platform for learning basic robotic motion control. The system allows real-time manipulation of four servos using potentiometers, with a record-and-play feature implemented through push buttons to store and replay motion sequences, demonstrating the concept of teach-and-repeat robotics. Testing confirmed smooth servo movement, accurate response to potentiometer inputs, and reliable playback of recorded sequences, achieving a maximum reach of approximately 26 cm with 180° rotation per joint, limited by the torque of SG90 servos. This work is inspired by industrial robotic arms such as **Epson Collaborative Robot, WidowX-250, Kinova Gen3, and RobotAnno 6DOF**, and serves as a bridge between educational robotics and real-world automation. Future enhancements include upgrading to a **6-DOF system**, integrating **belt-driven actuation**, **stepper motors**, and **AI-based autonomous drawing**, aligning the project with industry-grade collaborative robotic systems.

2. Introduction

This project implements a **4-DOF servo control system** using **4 potentiometers** and an ESP32 microcontroller.
The servos can be controlled manually in real-time, and their motion can be **recorded, stored in memory, and replayed** on demand.
An additional **Clear button** allows erasing the recorded sequence to start fresh.

This project can be used as the foundation for **robotic arms, animatronics, or motion replay systems**.

3. Project Objectives

- Control four SG90 servo motors using four 10k potentiometers.
- Record servo positions when the **Record** button is pressed.
- Replay recorded motion sequence **only when the Play button is pressed**.
- Make the design modular for easy future upgrades.

4. Materials and Components

The following materials and components were used in this project:

Component	Quantity	Description
ESP32 Development Board	1	Dual-core Wi-Fi + Bluetooth microcontroller, generates PWM directly to control servos.
Servo Motors SG90	4	Micro servos with 180° rotation and ~1.8 kg·cm torque. (Plastic Gear, Positional Servo).
Potentiometers (10k Ohm)	4	Input devices for controlling servo angles
Robotic Arm Kit (Acrylic/3D Printed)	1	Mechanical structure of the arm
Breadboard	1	For prototyping and connections
Jumper Wires	Assorted	For electrical connections

Component	Quantity	Description
Power Supply (5V)	1	To power the ESP32 and servos
Push Buttons	3	For Starts/stops recording motion, Plays recorded motion, Clears recorded data

5. Working Principle

The project is based on the principle of **position control of servo motors** using **analog input signals** from potentiometers, processed by an **ESP32 microcontroller**.

1. **Analog Signal Acquisition:**
   Each of the four potentiometers outputs a variable voltage (0–3.3 V) based on its position. These voltages are read by the **ADC (Analog-to-Digital Converter)** pins of the ESP32.

2. **Signal Mapping:**
   The ESP32 firmware maps the potentiometer input range to a corresponding **servo angle** (0°–180°). This ensures smooth and precise control of each servo motor.

3. **Servo Control:**
   The mapped angle values are used to generate **PWM (Pulse Width Modulation) signals** that drive the four SG90 servo motors, positioning them to the desired angles.

4. **Record Function:**
   When the **Record button** is pressed, the ESP32 continuously samples potentiometer values and stores the corresponding servo angles in memory (array or buffer).

5. **Play Function:**
   When the **Play button** is pressed, the ESP32 retrieves the stored servo angle sequence and reproduces the motion by sending PWM signals in the same order, effectively replaying the previously recorded movement.

6. **Real-Time Control:**
   During manual mode (no play button pressed), the servos respond instantly to potentiometer movements, allowing direct, real-time control.

The following table summarizes the electrical connections:

Component	Pin	ESP32 Pin
Servo 1 (Base)	Signal	GPIO 18
Servo 2 (Shoulder)	Signal	GPIO 19
Servo 3 (Elbow)	Signal	GPIO 21
Servo 4 (Gripper)	Signal	GPIO 22
Potentiometer 1	Wiper	GPIO 32
Potentiometer 2	Wiper	GPIO 33
Potentiometer 3	Wiper	GPIO 34
Potentiometer 4	Wiper	GPIO 35
Record Button	Signal	GPIO 25
Play Button	Signal	GPIO 26
Clear Button	Signal	GPIO 27

6. Software Implementation

- Code written in **Arduino IDE** using `ESP32Servo` library.
- Uses **analogRead()** to get potentiometer position (0–4095) and maps it to servo angle (0–180°).
- Stores recorded angles in a **2D array** `recordedAngles[stepIndex][servoIndex]`.
- Playback simply loops through stored angles and writes them to servos.
- `delay(20)` ensures smooth motion (50 Hz update rate).

7. Key Features

- **4-Channel Servo Control** – Fully independent.
- **Real-Time Potentiometer Control** – Instant feedback.
- **Record & Playback** – Store complex motion patterns.
- **Clear Button** – Reset memory without recording new motion.

- **Simple & Expandable** – Can be extended to more servos or saved to EEPROM/SPIFFS for permanent storage.

8. Limitations

- **Reach:** Mechanical reach limited to ~26 cm by link length.
- **Torque:** SG90 servos have limited torque (~1.8 kg·cm) and can lift only light loads.
- **Angle:** Each joint is limited to 180° rotation.
- **Volatile Memory:** Recorded sequence is erased after power cycle (no persistent storage yet).

9. Applications

- **Robotic Arms** – Teach & repeat movements.
- **Animatronics** – Replay character motions.
- **Industrial Training** – Demonstrate motion sequences.
- **DIY Automation Projects** – Record and repeat mechanical tasks.

10. Future Improvements

- **EEPROM/SPIFFS Storage:** Save motions permanently.
- **PC GUI / Mobile Control:** Record and play motions via computer/smartphone.
- **Wireless Control:** Add Wi-Fi/Bluetooth control options.
- **Speed Scaling:** Allow slower/faster playback of recorded motions.
- **Motion Smoothing:** Interpolation between recorded points for smooth operation.
- **Mechanical Upgrade:** Extend reach beyond 26 cm with longer links.
- **6-DOF Upgrade:** Add more servos for complex tasks.
- **Belt-Driven System:** Use belts to reduce backlash.
- **Stepper Motor Upgrade:** Replace SG90s with stepper motors for higher torque and precision.
- **Trajectory Planning & IK:** Advanced motion control for precise path following.
- **AI-Powered Drawing:** Train the arm to replicate images on paper or walls.
- **Autonomous Mode:** Implement AI to allow self-learning and decision-making and decision-making.

11. Conclusion

This project successfully demonstrates a **4-DOF motion control system** using ESP32, with features for **manual control, recording, playback, and clearing memory**.It

provides a strong foundation for **robotics and automation projects** where repeatable motion is required.

12. Appendices

12.1. Industry Inspirations

- **Epson's Upcoming 6 kg Payload, 900 mm Reach Collaborative Robot**
- **WidowX-250 6DOF Arm (Trossen Robotics)**
- **Kinova Gen3 6DOF Manipulator Arm**
- **RobotAnno 6DOF Industrial Cobot Series**

12.2. Schematic Diagram

[A schematic diagram illustrating the electrical connections between the ESP32, servo motors, and potentiometers] Image 4DOF arm [Working Video](#)

12.3. Source Code

The full source code for the ESP32 can be found in the attached file:
Source Code 4DOF ARM