International Scientific Journal of Engineering and Management (ISJEM)
Volume: 04 Issue: 06 | June – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM04583

ARMPI 4DOF VISION ROBOTIC ARM

Nafiza[1], M Siva Kumar[2], T Premsai Viswanadh[3] G Sandeep Chaitanya[4],G Kalyan Kumar [5]

[1] *Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences*

[2]*Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences*

[3]*Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences*

[4]*Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences*

[5]*Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences*

---***---

Abstract - This paper presents the ArmPi 4DOF Vision Robotic Arm, a cost-effective, vision-guided robotic system capable of real-time pick-and-place tasks using Raspberry Pi and OpenCV. The system integrates a 4 Degrees of Freedom robotic arm with a Pi Camera for object detection based on shape and color. It achieves autonomous movement through Python-based control logic. Results show reliable operation in controlled environments with scope for enhancements like AI-based object recognition and closed-loop feedback.

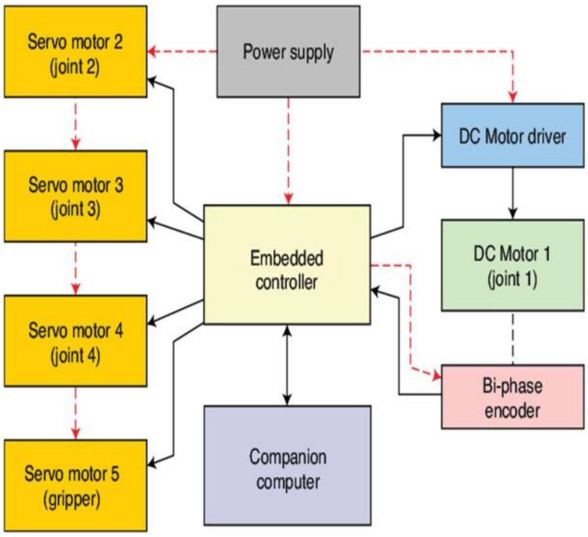
Key Words: Robotic Arm, Computer Vision, Raspberry Pi, 4DOF, OpenCV, Automation

1. INTRODUCTION

Industrial automation increasingly relies on robotic systems, yet most low-cost arms lack vision feedback. This project aims to bridge that gap with a 4DOF robotic arm integrating computer vision and embedded control via Raspberry Pi. The objective is to perform pick-and-place operations based on real-time visual detection using affordable components suitable for education, research, and light automation.

2. Methodology

The system uses a Raspberry Pi 4, PCA9685 servo driver, four MG996R servo motors, and a Pi Camera. Object detection is handled via OpenCV using HSV color filtering and contour analysis. Python-based control software links visual inputs to robotic movements. Hardware integration includes I2C-based servo control and real-time image processing.



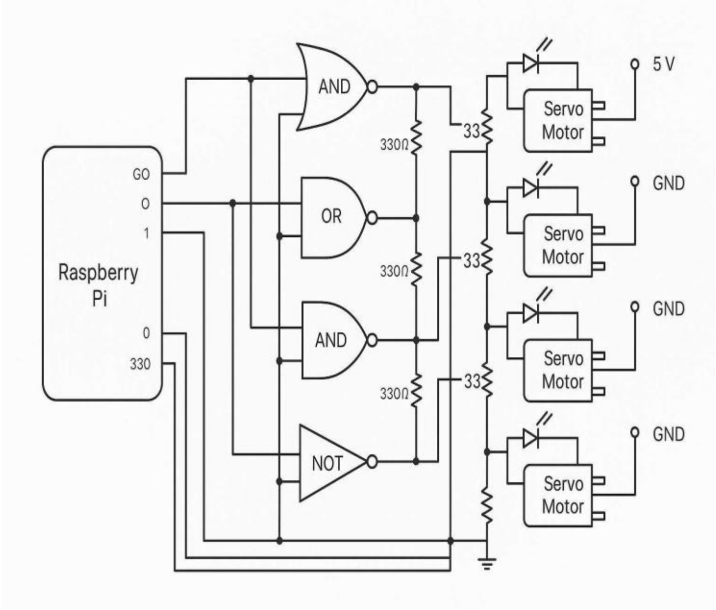
Block diagram of the ArmPi 4DOF Vision Robotic Arm system.

The diagram represents a control architecture for a robotic arm system comprising multiple servo and DC motors. An embedded controller serves as the central unit, interfacing with a companion computer for higher-level commands and data processing.

The system drives four servo motors (Joint 2, Joint 3, Joint 4, and a gripper) directly through the embedded controller, while a DC motor (Joint 1) is controlled via a dedicated DC motor driver. The DC motor's position feedback is provided by a bi-phase encoder, ensuring accurate motion control.

A power supply feeds the motors and control electronics, distributing power through the system as indicated by the red dashed lines. This structured design allows precise and coordinated control of multiple joints and the gripper in robotic applications.

International Scientific Journal of Engineering and Management (ISJEM)
Volume: 04 Issue: 06 | June – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM04583



Circuit diagram of the ArmPi 4DOF Vision Robotic Arm system.

This circuit design shows how a Raspberry Pi interfaces with four servo motors using AND, OR, and NOT gates to control the motors. It coordinates many actuators using digital electronics and embedded control.

GPIO pins on the Raspberry Pi are the main control signals in this configuration. GPIO pins send high (1) or low (0) logic levels based on Raspberry Pi program instructions. Different configurations of logic gates process input signals based on established logical requirements.

Circuit uses:

AND gates enable precision control where many conditions must be met for a servo to activate by producing a high output only when all inputs are high.

OR gates provide servo activation flexibility by producing a high output when any input is high.

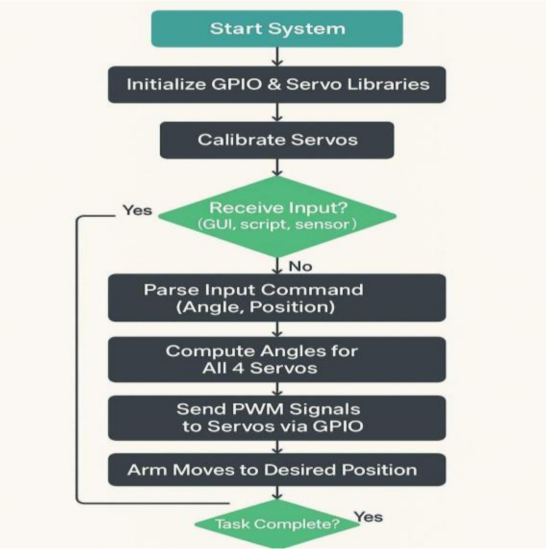
NOT gates (inverters) reverse input logic to provide complementary signals for control actions.

Each logic gate output is coupled to a 330Ω resistor before driving the servo motor. These resistors limit current, protecting the GPIO pins and servos from overcurrent, assuring system reliability and safety.

Dedicated 5V power supplies power the servo motors. For constant and reliable signal interpretation, the power supply's ground is linked to the Raspberry Pi's ground.

The Raspberry Pi can conditionally control numerous servo motors with this setup. Programming multiple GPIO pin states creates logical combinations through the gates to operate the servo motors in a specified sequence. In small-scale robots, automated mechanisms, and educational projects, this configuration is effective for controlling many outputs depending on logical requirements.

This arrangement shows a simple yet effective technique to merge digital logic with embedded computer to control several actuators, demonstrating digital electronics, logic operations, and microcontroller-based automation.



This flowchart represents the operational process of a **servo-controlled robotic arm system** using a Raspberry Pi:

1. The system starts by **initializing the GPIO and servo libraries**.
2. Servos are **calibrated** to ensure accurate positioning.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04583

3. The program waits to **receive input commands** (from a GUI, script, or sensor).
4. Once input is received, it **parses the command** to determine the required angle and position.
5. The system **computes the angles** for all four servos.
6. It then **sends PWM signals via GPIO pins** to position the servos accordingly.
7. The robotic arm **moves to the desired position**.
8. The system checks if the **task is complete**; if not, it waits for the next input.

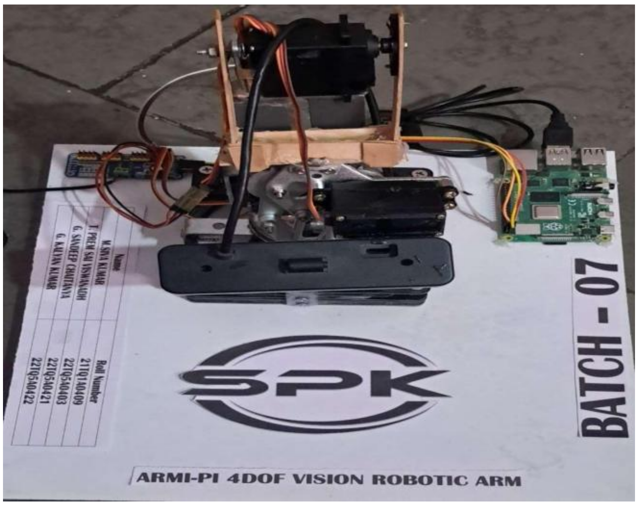
This control flow enables real-time, interactive operation of a servo-driven robotic system.

3. Key Features:

a. 4DOF robotic arm structure with servo-based joints
b. Pi Camera for real-time image capture
c. OpenCV-based object detection
d. Python-coded control logic for movement planning
e. Modular and scalable design

Results

The system was tested with colored blocks under controlled indoor lighting. It consistently detected and manipulated objects with 85–90% accuracy. Pick-and-place cycles averaged 4–6 seconds. The arm operated stably for 20+ minutes with no power issues, though performance declined under poor lighting or for similar-colored objects.



Key Observations:

i. Reliable object detection with strong color contrast
ii. Smooth servo transitions with minor jitter under load
iii. High repeatability in controlled settings

Conclusion

The project successfully demonstrated a low-cost, vision-based 4DOF robotic arm integrating real-time computer vision for object manipulation. The modular, scalable design is suitable for academic, DIY, and light industrial applications. Limitations include lighting sensitivity, absence of feedback control, and basic detection algorithms.

Future Scope:

1. AI-based object classification (YOLO, MobileNet)
2. Closed-loop servo control with feedback sensors
3. Depth sensing for 3D positioning
4. Adaptive gripper mechanisms
5. Wireless GUI-based control interface

References

1) OpenCV Documentation — https://docs.opencv.org
2) Raspberry Pi Documentation — https://www.raspberrypi.com/documentation
3) Adafruit PCA9685 Servo Driver Guide — https://learn.adafruit.com/16-channel-pwm-servo-driver
4) Patel, H., & Kumar, S. (2021). Vision-based Robotic Arm for Object Sorting. *Int. J. Robotics and Automation*, 36(4), 217–224.
5) Mohan, R., & Das, M. (2020). Design and Implementation of a 4DOF Robotic Arm. *Int. J. Engineering Research*, 8(1), 34–39.
6) Arduino & Raspberry Pi Forums — https://forum.arduino.cc, https://forums.raspberrypi.com
7) GitHub: https://github.com/adafruit/Adafruit-PCA9685, https://github.com/opencv/opencv
8) Zhang, Y., & Li, J. (2019). A Smart Pick-and-Place System Using Computer Vision and Raspberry Pi. *Journal of Automation and Control Engineering*, 7(2), 55–60.