https://doi.org/10.46335/IJIES.2024.9.3.9 **e-ISSN: 2456-3463**
Vol. 9, No. 3, 2024, PP. 44-50

International Journal of Innovations in Engineering and Science, www.ijies.net

Industrial Robotic Arm Using Raspberry PI

Rupali Dasarwar[1], Nayan Thakre[2], Kundan Tikale[3], Adarsh Dhoke[4], Tejas Randive[5]

[1]Professor, [2,3,4,5] B.Tech Students,

Department of Artificial Intelligence and Data Science, Wainganga College of Engineering and Management, Nagpur, Maharashtra, India.

rupalidasarwar20@gmail.com

***Received on**: 20 April, 2024* ***Revised on**: 15May, 2024* ***Published on**: 17 May, 2024*

***Abstract –** This project focuses on building a robotic arm using a Raspberry Pi to pick up objects from one set position and move them to another. The Raspberry Pi, a compact computer, serves as the brain controlling the robotic arm's movements. Through precise programming, we equip the Raspberry Pi-controlled robotic arm with the ability to detect objects, securely grasp them, and accurately relocate them. Additionally, the project integrates object detection algorithms to enhance the arm's functionality. The main objective is to create an affordable and accessible robotic arm solution tailored for tasks like object detection, picking, and placing. Through rigorous experimentation and testing, we showcase the Raspberry Pi-controlled robotic arm's reliability in executing these tasks with precision. The project underscores the potential of leveraging affordable technologies such as the Raspberry Pi to develop functional robotic systems across diverse applications, including manufacturing, education, and research. The Industrial Robotic Arm using Raspberry Pi represents a cutting-edge solution for efficient object manipulation in industrial settings. With a 360-degree base rotation capability and 6 degrees of freedom (6DOF) robotic arm, the system offers unmatched flexibility and precision in selecting and placing objects. Leveraging the Raspberry Pi's advanced computing capabilities, the robotic arm executes intricate movements with precise control. This feature makes it an ideal choice for applications that demand delicate handling of objects in various orientations.*

***Keywords-** Industrial robotic arm, cost-effective, automation, robotic arm, Raspberry Pi, object manipulation, 360-degree base rotation, 6 degrees of freedom (6DOF).*

I. INTRODUCTION

In recent years, advancements in single-board computers, such as the Raspberry Pi, have revolutionized the landscape of robotics by providing accessible and cost-effective computing platforms for a wide range of applications. The Raspberry Pi 4 Model B, in particular, has emerged as a popular choice among robotics enthusiasts and researchers due to its enhanced processing power, versatility, and robust community support. Leveraging the capabilities of the Raspberry Pi 4 Model B, this research focuses on the development of an Industrial Robotic Arm system equipped with seven MG995 motors, of which two enable 360-degree rotation and five facilitate 180-degree rotation. These motors, along with a dedicated motor controller board, form the backbone of the robotic arm, enabling precise and coordinated movement for object manipulation tasks.

The integration of the Raspberry Pi 4 Model B as the central control unit provides several advantages, including computational efficiency, scalability, and compatibility with a wide range of software libraries and development tools. By harnessing the computational prowess of the Raspberry Pi, the robotic arm can execute complex algorithms for object detection, path planning, and motion control, thereby enabling efficient and autonomous operation in industrial environments.

Moreover, the utilization of MG995 motors ensures robust and reliable performance, with the ability to handle varying payload sizes and perform intricate movements with precision. The inclusion of two 360-degree rotation motors further enhances the versatility of the robotic arm, allowing for unrestricted movement in all directions within its workspace.

Through this research endeavor, we aim to showcase the feasibility and effectiveness of employing the Raspberry Pi 4 Model B in conjunction with MG995 motors for the development of an Industrial Robotic Arm system. The proposed system offers a compelling solution for applications requiring object manipulation tasks in industrial settings, such as manufacturing, logistics, and automation.

II. METHODS AND MATERIAL

2.1 Robotic Arm Design

The design process of the 6 Degree of Freedom (6DOF) robotic arm involved a meticulous approach to ensure its functionality, reach, and versatility. The arm was conceptualized to mimic the movements of a human arm, with six joints providing a wide range of motion in three-dimensional space.

2.1.1 Kinematic Analysis:

Before diving into the physical construction, a thorough kinematic analysis was performed to determine the optimal configuration of joints and linkages. This analysis included defining the range of motion required for each joint, considering workspace constraints, and calculating the inverse kinematics equations. The goal was to achieve maximum reach and dexterity while ensuring efficient and smooth movements.

2.1.2 Joint Configuration:

The robotic arm was designed with three rotational joints and three translational joints, each serving a specific purpose in the arm's operation:

- Rotational Joints:
  - Shoulder (Base) Joint: This joint provides rotation along the vertical axis, allowing the arm to move up and down.
  - Elbow Joint: Positioned between the shoulder and wrist, this joint provides rotational movement along the horizontal axis.
  - Wrist Rotation Joint: Located at the wrist, this joint enables the end effector to rotate around its own axis.
- Translational Joints:
  - Wrist (X, Y, Z) Joints: These three joints allow for linear movement in the x, y, and z directions, enabling precise positioning of the end effector.

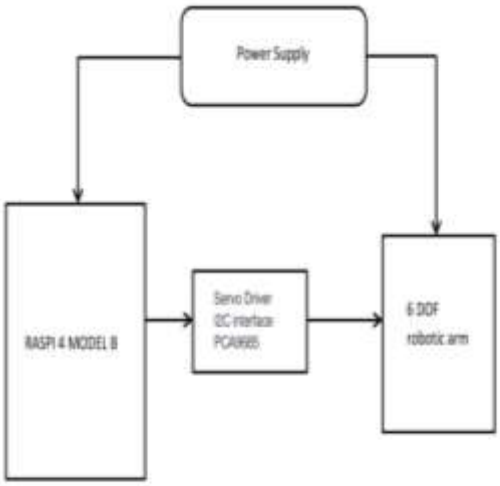


Figure 1: Block Diagram of Proposed System.

2.2 Components:

The prototype uses the following components:

1) Raspberry pi 4 Model B Microprocessor:

The Raspberry Pi 4 Model B is an ideal microprocessor for robotic arm applications due to its powerful features and versatility:

1. Processing Power: Equipped with a quad-core ARM Cortex-A72 processor running at 1.5GHz, the Raspberry Pi 4 delivers substantial processing power for real-time control of robotic arm movements, ensuring quick and precise calculations.
2. Memory and Storage: With options for 2GB, 4GB, or 8GB of LPDDR4 RAM and a microSD card slot for storage, the Raspberry Pi 4 efficiently handles complex algorithms and tasks, providing ample space for program codes and data storage.
3. GPIO Pins: The Raspberry Pi 4 offers a large

number of GPIO pins, crucial for interfacing with external components like motor drivers and sensors in robotic arm systems. These pins enable seamless control and monitoring of the arm's movements and interactions.

4. Operating System Support: Supporting various operating systems such as Raspberry Pi OS and Ubuntu, the Raspberry Pi 4 allows developers to choose the most suitable OS for their robotic arm projects. It provides access to a wide array of software libraries and tools.

5. Compact Size and Affordability: The Raspberry Pi 4's small form factor makes it easy to integrate into robotic arm designs without occupying much space. Its cost-effectiveness is beneficial for hobbyists, students, and small-scale industrial applications.

6. Community and Support: Benefit from a large and active Raspberry Pi community, providing extensive documentation, tutorials, and support. This ensures developers have resources for building and troubleshooting robotic arm projects based on the Raspberry Pi 4.

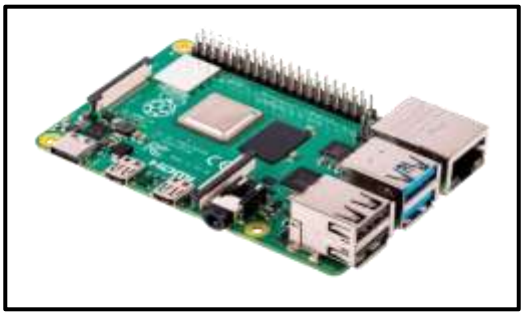


Figure 2: Raspberry pi 4 model B.

2) PCA9685 Servo Driver

The PCA9685 Servo Driver is a popular choice for controlling servo motors in robotic arm applications, especially when interfacing with a Raspberry Pi.

1. **Multiple Servo Control**: The PCA9685 can control up to 16 servo motors independently, which is often more than enough for most robotic arm designs. This allows for precise control over each joint of the robotic arm.
2. **I2C Interface**: The PCA9685 communicates with the Raspberry Pi via the I2C interface, which is a simple and widely used communication protocol. This makes it easy to connect and interface with the Raspberry Pi, reducing complexity in the overall system design.
3. **Hardware PWM Generation**: The PCA9685 generates hardware PWM (Pulse Width Modulation) signals, which provide accurate and jitter-free control of servo motors. This ensures smooth and precise movement of the robotic arm joints, essential for tasks requiring accuracy and precision.
4. **Configurable Frequency**: The PCA9685 allows the user to configure the PWM frequency, which can be adjusted according to the specific requirements of the servo motors used in the robotic arm. This flexibility ensures compatibility with a wide range of servo motors available in the market.
5. **Compact Size**: The PCA9685 is available in a compact package, making it suitable for integration into small-scale robotic arm designs. Its small footprint allows for space-efficient mounting on the Raspberry Pi or within the robotic arm structure.
6. **Open-Source Libraries**: There are many open-source libraries available for interfacing with the PCA9685 from the Raspberry Pi, making it easier for developers to get started with controlling servo motors for robotic arm projects. These libraries often provide high-level APIs that abstract the low-level details of communication with the PCA9685, simplifying software development.

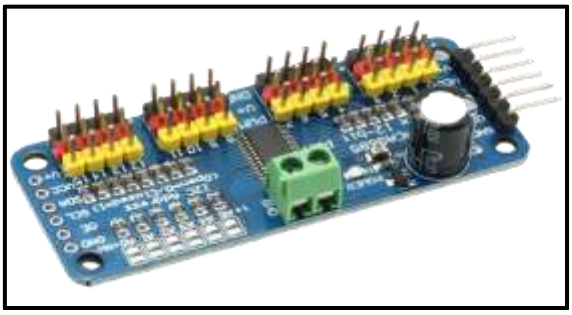


Figure 3: PCA9685 Servo Driver.

3) MG90S and MG995 metal gear motors:

The choice of MG90S and MG995 metal gear motors for a robotic arm controlled by a Raspberry Pi is driven by several key factors that contribute to the arm's functionality and performance. These motors are specifically selected for their torque, precision, and control capabilities, making them ideal for the complex movements required in a robotic arm system.

https://doi.org/10.46335/IJIES.2024.9.3.9 e-ISSN: 2456-3463
Vol. 9, No. 3, 2024, PP. 44-50

International Journal of Innovations in Engineering and Science, www.ijies.net

Figure 3: MG90S motor.

1. **Torque and Power:** The MG90S and MG995 motors are known for their high torque output, which is crucial for lifting and manipulating objects with varying weights. The 360-degree MG995 motors offer substantial torque for the base rotation of the robotic arm, allowing it to move smoothly in any direction without strain.
2. **Precision and Accuracy:** Both motor types are equipped with metal gears that provide precise and accurate movement control. This level of precision is essential for tasks such as picking up objects, placing them in specific locations, and adjusting the arm's orientation with precision.
3. **Durability and Reliability:** The metal gear construction of these motors ensures durability and longevity, even under continuous use. This reliability is crucial for industrial applications where the robotic arm needs to perform repetitive tasks with minimal downtime.
4. **Compatibility with Raspberry Pi:** The motors are compatible with the GPIO (General Purpose Input/Output) pins of the Raspberry Pi, making them easy to integrate into the arm's control system. The Raspberry Pi provides a versatile and programmable platform for controlling the motors, allowing for customized movement sequences and automation of tasks.
5. **Cost-Effectiveness:** MG90S and MG995 motors offer a balance of performance and affordability, making them suitable for a wide range of robotic arm projects. They provide excellent value for their capabilities, making them a popular choice among robotics enthusiasts and professionals alike.

Figure 4: MG995 Servo Motor

4) Raspberry Pi Camera:

The robotic arm designed in this project is equipped with a camera system to enable object detection and recognition capabilities. The camera, integrated with the Raspberry Pi-controlled robotic arm, plays a crucial role in enhancing its functionality and versatility.

Object detection is achieved through the camera's ability to capture real-time images of the arm's surroundings. The camera module, a vital component of the system, is strategically positioned to provide a wide field of view. This allows the robotic arm to scan its environment and identify objects within its reach.

Upon capturing an image, the Raspberry Pi employs sophisticated image processing algorithms to analyze the visual data. These algorithms utilize deep learning and computer vision techniques to identify objects based on their distinct features, shapes, and colors. The system is trained with a dataset containing various object classes, enabling it to classify and recognize a wide range of objects.
Once an object is detected and recognized, the robotic arm autonomously plans its movements to interact with the object. This includes calculating the optimal path for the arm to reach the target object, adjusting its grip based on the object's shape and size, and executing precise movements for object manipulation.

The integration of the camera system enables the robotic arm to perform tasks such as pick-and-place operations with remarkable accuracy and efficiency. Additionally, the system can adapt to dynamic environments by continuously scanning for new objects and adjusting its actions accordingly.
Overall, the camera-equipped robotic arm represents a significant advancement in automation technology. Its object detection and recognition capabilities make it suitable for a wide range of applications, including industrial automation, warehouse logistics, and smart manufacturing processes. This integration not only enhances the arm's operational capabilities but also opens up new possibilities for intelligent and adaptive robotic systems.

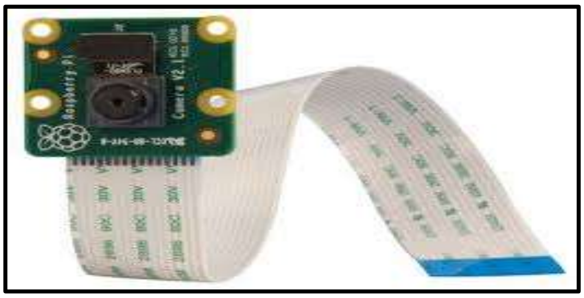


Figure 4: Raspberry pi camera.

2.3 Software Development:

2.3.1 Motion Control Algorithms:

A critical aspect of the software was the implementation of motion control algorithms to translate desired end effector positions into joint angles and velocities. The arm's movement commands were processed through PID (Proportional-Integral-Derivative) controllers to achieve smooth and accurate motion. These controllers were tuned to optimize performance based on the arm's dynamics and payload.

2.3.2 Inverse Kinematics Solver:

Given the desired end effector position in 3D space, an inverse kinematics solver was developed to compute the corresponding joint angles required to reach that position. This solver utilized geometric and trigonometric calculations to determine the joint configurations that achieved the desired end effector pose while considering the arm's constraints.

2.3.3 Object Detection and Recognition:

To enhance the arm's autonomy and enable tasks such as object manipulation, an object detection and recognition module was integrated. This module utilized computer vision algorithms, such as Convolutional Neural Networks (CNNs), to identify and locate objects within the arm's workspace. The module could detect objects of various shapes, sizes, and orientations, providing valuable input for the arm's decision-making process.

2.3.4 Communication and Networking:

The software architecture included provisions for communication and networking capabilities, allowing the robotic arm to interface with external devices or systems. This included wireless connectivity options such as Wi-Fi or Bluetooth, enabling remote control, data logging, and integration with other automation systems.

2.3.5 Simulation and Testing:

Prior to deployment, extensive simulation and testing were conducted using software tools such as ROS (Robot Operating System) and Gazebo. These simulations allowed for virtual testing of the arm's functionalities, validation of control algorithms, and refinement of motion trajectories. Additionally, hardware-in-the-loop (HIL) testing was performed to verify the software's compatibility with the actual robotic arm hardware.

2.4 Object Detection Integration:

2.4.1 Object Detection System Setup:

For enhanced functionality and versatility, an object detection system was integrated into the 6DOF robotic arm using a USB camera module. The integration aimed to enable the robotic arm to detect objects in its environment, identify their positions, and perform tasks such as object picking and placing autonomously.

2.4.2 OpenCV Library Integration:

The OpenCV (Open Source Computer Vision) library, a powerful open-source computer vision and machine learning software library, was utilized for real-time image processing and object detection algorithms. The library provides a wide range of tools and functions for image manipulation, feature extraction, and object recognition.

2.4.3 Object Detection Workflow:

The workflow for object detection integration in the robotic arm system consisted of several key steps:

1. Camera Initialization:

- The USB camera module was connected to the Raspberry Pi 4.
- The camera was initialized using OpenCV to capture real-time video frames.

2. Frame Processing:

- Python scripts were developed to capture video frames from the camera feed.
- Each frame was processed to extract relevant features and detect objects within the frame.

https://doi.org/10.46335/IJIES.2024.9.3.9 e-ISSN: 2456-3463
Vol. 9, No. 3, 2024, PP. 44-50

3. Object Detection Algorithm:

- An object detection algorithm was implemented using pre-trained deep learning models available in the OpenCV library, such as YOLO (You Only Look Once).
- The algorithm analysed the processed frames to identify objects of interest, such as specific shapes, colours, or patterns.

4. Object Position Estimation:

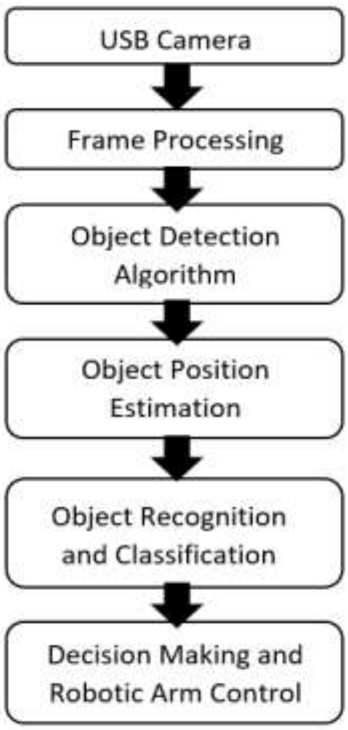
- Upon detecting objects, the algorithm estimated their positions and orientations within the camera frame.
- This information was translated into coordinates relative to the robotic arm's workspace.

5. Object Recognition and Classification:

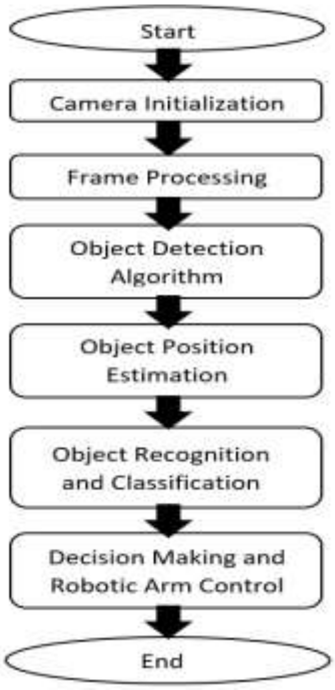
- Additional machine learning algorithms, such as support vector machines (SVM) or convolutional neural networks (CNN), were employed for object recognition and classification.
- The system could identify different types of objects based on their visual characteristics.

6. Decision-Making and Robotic Arm Control:

- The detected object's position and classification results were fed into the robotic arm control system.
- Based on predefined task commands or user inputs, the robotic arm made decisions on object manipulation tasks, such as picking, placing, or sorting.



2.4.4 Block Diagram



III. Results and Discussion

3.1 Robotic Arm Performance Evaluation:
Upon completion of the 6 Degree of Freedom (6DOF) robotic arm using the Raspberry Pi 4 as the control unit, a series of tests and evaluations were conducted to assess its performance in various tasks. The robotic arm demonstrated remarkable precision, agility, and versatility in its movements, showcasing its capabilities in object manipulation, positioning, and payload handling.

3.2 Positioning Accuracy:
The robotic arm exhibited high positioning accuracy, achieving target coordinates within millimeter-level precision. Through the implementation of inverse kinematics algorithms, the arm smoothly navigated to specified end effector positions in 3D space. Test scenarios included reaching for predefined objects at different heights and distances, with the arm consistently hitting the target coordinates with minimal deviation.

3.3 Payload Handling Capacity:
The payload handling capabilities of the robotic arm were evaluated by testing its ability to lift and transport objects of varying weights. The arm successfully lifted

https://doi.org/10.46335/IJIES.2024.9.3.9 **e-ISSN: 2456-3463**
Vol. 9, No. 3, 2024, PP. 44-50

International Journal of Innovations in Engineering and Science, www.ijies.net

objects ranging from lightweight items such as pens and small tools to heavier objects such as books and small boxes. The servo motors, specifically the MG995 servo motors used for translational joints, demonstrated sufficient torque and stability to handle the payloads without significant sagging or instability.

3.4 Object Manipulation Tasks:

Integration with an object detection system further enhanced the robotic arm's functionality. Using a USB camera module and OpenCV for real-time image processing, the arm was able to detect and identify objects within its workspace. Once detected, the arm autonomously moved to grasp and lift the objects, showcasing its ability to perform pick-and-place tasks with efficiency and accuracy.

3.5 Versatility and Flexibility:

One of the standout features of the 6DOF robotic arm was its versatility in performing a wide range of tasks. The combination of rotational and translational joints allowed for complex movements and orientations, enabling the arm to reach objects from various angles and positions. This flexibility makes the robotic arm suitable for applications such as assembly line tasks, material handling in warehouses, and educational purposes in robotics labs and classrooms.

3.6 Future Improvements and Applications:

Moving forward, there are several avenues for further improvement and expansion of the robotic arm system. Integration with advanced machine learning algorithms for enhanced object recognition and decision-making capabilities could elevate its autonomy and adaptability. Additionally, the addition of end effectors, such as grippers or suction cups, would enable the arm to handle a broader range of objects and materials.

IV. CONCLUSION

In conclusion, the development of a 6DOF robotic arm using the Raspberry Pi 4 as the control unit showcases the feasibility and effectiveness of using this platform for robotic applications. The Raspberry Pi 4's processing power, memory, GPIO pins, and support for various operating systems make it a versatile and reliable choice for controlling sophisticated robotic systems. Future work may involve integrating vision systems for object detection and implementing advanced control algorithms for enhanced performance.

REFERENCES

[1] *Krishnaraj Rao N S, Avinash, Rama Moorthy H, Karthik K, Sudesh Rao, Santosh S, "**An Automated Robotic Arm: A Machine Learning Approach**", in 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC).*

[2] *Mahmoud Abdelaal , "**A Study of Robot Control Programing for an Industrial Robotic Arm** ", in Computers and Systems Dept., Electronics Research Institute, Giza, Egypt .*

[3] *Ms.Rasika Yenorkar, Prof. Dr.U. M.Chaskar, "**GUI Based Pick and Place Robotic Arm for Multipurpose Industrial Applications**", in Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3.*

[4] *https://www.waitingforfriday.com/?p=156*