A
MINI PROJECT REPORT
ON
PRECISION ROBOT ARM

Submitted By

Kalpesh Mayurdhvaj Narkhede	**T190553043**
Shubham Janardhan Patange	**T190553049**
Gaurav Prakash Patil	**T190553051**
Labhesh Vilas Patil	**T190553052**

Guided By

Dr Sagar V. Joshi

Prof. Mahesh G. Chinchole

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

NMVPM's and PCET's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

2023 - 2024

PRECISION ROBOT ARM

A Mini Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree of

Bachelor of Engineering

in

Electronics & Telecommunication Engineering

Submitted By

Mr. Kalpesh Mayurdhvaj Narkhede.	**T190553043**
Mr. Shubham Janardhan Patange.	**T190553049**
Mr. Gaurav Prakash Patil.	**T190553051**
Mr. Labhesh Vilas Patil.	**T190553052**

Guided By

Dr Sagar V. Joshi

Prof. Mahesh G. Chinchole

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

NMVPM's and PCET's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

2023 - 2024

NMVPM's & PCET's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the mini project entitled "**ROBOTIC ARM**" has been carried out **by Shubham Janardhan Patange, Kalpesh Mayurdhvaj Narkhede, Gaurav Prakash Patil, Labhesh Vilas Patil** under my guidance in partial fulfillment of the degree of Bachelor of Engineering in Electronics and Telecommunication Engineering of Nutan Maharashtra Institute Of Engineering & Technology affiliated to Savitribai Phule Pune University, Pune, during the academic year 2023-2024. To the best of my knowledge and belief this work has not been submitted elsewhere for the award of any other degree.

Prof. M. G. Chinchole **Mini-Project Guide**	**Dr. S. V. Joshi** **HOD E&TC** **Mini-Project Guide**	**Dr. V. V. Deotare** **Principal**

Prof.________________________

External Examiner

Date: Place: Talegaon Dabhade, Pune-410507

ACKNOWLEDGMENT

We express our gratitude to all those who contributed to the completion of this project. We extend our heartfelt thanks to **Prof. Mahesh Chinchole** and **Dr. Sagar Joshi** for his invaluable guidance and support throughout the project. We are also thankful to **Dr. Sagar Joshi**, HOD of Electronics & Telecommunication Engineering, for his encouragement and valuable insights. Additionally, we would like to extend our appreciation to **Dr. Vilas V. Deotare**, Principal of NMVPM's and PCET's Nutan Maharashtra Institute of Engineering & Technology, for his continuous support and encouragement.

Kalpesh Mayurdhvaj Narkhede

Shubham Janardhan Patange

Gaurav Prakash Patil

Labhesh Vilas Patil

ABSTRACT

Robotic arms have become indispensable in modern industrial applications due to their precision, versatility, and efficiency. This mini project focuses on the design, development, and implementation of a versatile robotic arm system aimed at enhancing precision handling and automation in industrial environments. The project aims to address the growing need for automation to improve efficiency and reduce reliance on manual labor in various industrial processes. The robotic arm system is designed to be cost-effective, adaptable, and capable of performing a wide range of tasks with optimal degrees of freedom (DOF). Intelligent control algorithms are integrated to enable autonomous operation and accurate positioning of the robotic arm, while safety features ensure worker protection in hazardous environments. The project encompasses hardware selection, algorithm development, software integration, and extensive testing to verify performance, reliability, and efficiency. The potential applications of the robotic arm system span across industrial automation, electronic assembly, material handling, precision machining, welding, quality inspection, medical assistance, laboratory automation, agriculture, and entertainment. Overall, the project aims to revolutionize industrial automation, enabling industries to achieve higher levels of productivity, safety, and competitiveness.

FIGURE INDEX

INDEX

CHAPTER 1
INTRODUCTION

Description

In today's rapidly evolving industrial landscape, automation plays a pivotal role in enhancing productivity, efficiency, and safety across various sectors. Robotic arms have emerged as indispensable tools in modern manufacturing, offering unmatched precision and versatility in handling intricate tasks. These robotic arms, equipped with advanced sensors and intelligent control systems, streamline production processes, reduce reliance on manual labor, and mitigate workplace hazards. As third-year ENTC engineering students, exploring the functionality and applications of robotic arms provides valuable insights into the transformative potential of automation in industrial settings.

Motivation

The motivation behind this project stems from the pressing need to address the limitations of existing robotic solutions in industrial automation. While traditional robotic arms offer significant advantages, they often lack the necessary dexterity, adaptability, and cost-effectiveness to tackle complex tasks effectively. Moreover, industries face constraints such as space limitations, budgetary considerations, and stringent safety regulations, hindering the seamless integration of advanced automation solutions.

Problem Description

Despite advancements in robotic technology, many industrial processes still require human intervention due to the limitations of current robotic solutions. Precision handling, hazardous material manipulation, and tasks requiring intricate manipulation often remain challenging for conventional robotic arms. This creates bottlenecks in achieving optimal productivity and safety standards in industrial operations.

Short Survey

A brief survey of existing literature reveals a diverse range of applications for robotic arms, spanning industrial automation, electronic assembly, material handling, precision machining, welding, quality inspection, medical assistance, laboratory automation, agriculture, and entertainment. However, there is a notable gap in cost-effective, versatile robotic arm solutions capable of addressing the evolving needs of modern industries.

1.1 Overview of the Project

The project focuses on the design, development, and implementation of a versatile robotic arm system tailored for precision handling and automation in industrial environments. Leveraging the latest advancements in robotics, sensor technology, and intelligent control algorithms, the project aims to address the growing demand for cost-effective, adaptable solutions capable of enhancing productivity, efficiency, and safety in diverse industrial applications.

The project encompasses a comprehensive approach, starting from conceptualization and design to prototyping, testing, and validation. Key aspects of the project include hardware selection, algorithm development, software integration, and performance optimization. Through systematic research, experimentation, and iterative refinement, the project aims to deliver a robust and reliable robotic arm system capable of meeting the complex requirements of modern industrial operations.

1.2 Problem Statement

The industrial landscape faces a critical challenge in achieving precise handling tasks due to limitations in existing robotic solutions. Traditional robotic arms lack the required flexibility, adaptability, and cost-effectiveness to address these needs effectively. As a result, industries struggle to automate tasks that require precise manipulation and hindering productivity. Developing a versatile, cost-effective robotic arm solution capable of precise handling tasks is essential to overcome these challenges and revolutionize industrial automation.

1.3 Need of Project

In the contemporary industrial landscape, there exists a compelling need for innovative solutions to address the evolving challenges faced by manufacturers. The advent of Industry 4.0 has ushered in an era of automation, where efficiency, precision, and adaptability are paramount for sustainable growth and competitiveness. However, traditional robotic solutions often fall short in meeting the diverse requirements of modern industrial applications, necessitating the development of advanced, cost-effective alternatives.

1.3.1 Enhancing Efficiency and Precision

The project responds to the imperative of enhancing efficiency and precision in industrial processes. By leveraging the capabilities of robotic arms, manufacturers can automate complex tasks that were previously reliant on manual labor, thereby streamlining production workflows and minimizing the risk of errors. The need for precision handling and manipulation in industries such as electronics, automotive, and aerospace underscores the importance of developing robotic arm systems capable of performing intricate tasks with unparalleled accuracy.

1.3.2 Mitigating Workplace Hazards

Moreover, the project aims to address concerns related to workplace safety by reducing human exposure to hazardous environments. Many industrial processes involve handling of toxic substances, heavy machinery, or repetitive tasks that pose risks to human operators. By deploying robotic arms for such tasks, industries can significantly mitigate these risks and ensure the well-being of their workforce, thereby aligning with regulatory requirements and fostering a culture of safety and responsibility.

1.3.3 Driving Innovation and Competitiveness

Furthermore, the project is driven by the imperative of driving innovation and competitiveness in the industrial sector. As markets become increasingly dynamic and competitive, manufacturers must continually strive to improve their processes, reduce costs, and differentiate their products. Robotic automation offers a pathway to innovation by unlocking new possibilities for process optimization, customization, and scalability. By investing in advanced robotic arm systems, industries can position themselves at the forefront of technological advancement and gain a competitive edge in the global marketplace.

1.3.4 Meeting Industry Demands

Lastly, the project aims to meet the specific demands and challenges faced by industries across various sectors. Whether it's precision machining in aerospace, material handling in logistics, or assembly line operations in electronics manufacturing, each industry has unique requirements that necessitate tailored solutions. The project seeks to develop a versatile robotic arm platform that can be customized and adapted to meet the diverse needs of different industries, thereby fostering widespread adoption and scalability.

1.4 Aim & Objective of Project

Aim:

The aim of this project is to design, develop, and implement a versatile robotic arm system capable of precision handling and automation in industrial environments.

Objectives:

1. Designing a robotic arm with optimal degrees of freedom (DOF) to ensure flexibility and adaptability for various industrial tasks.
2. Evaluating the practical applicability of the robotic arm in different industrial settings, including precision handling, material sorting, and assembly tasks.
3. Documenting the design, development, and testing processes to create comprehensive guidelines for future robotic arm projects and research endeavors.
4. Providing user-friendly interfaces and documentation to facilitate easy operation, maintenance, and troubleshooting of the robotic arm system by end-users and technicians.
5. Demonstrating the potential benefits of the robotic arm system in enhancing productivity, reducing labor costs, and improving safety standards in industrial operations.

1.5 Application Areas

The versatility and adaptability of the robotic arm system developed in this project make it suitable for a wide range of industrial applications across various sectors. By leveraging advanced robotics technology and intelligent control algorithms, the robotic arm can address diverse challenges and perform critical tasks with precision and efficiency. Some of the key application areas include:

1.5.1 Manufacturing and Assembly

In the manufacturing sector, the robotic arm can revolutionize assembly line operations by automating repetitive tasks, such as component placement, soldering, and quality inspection. Its ability to handle complex manipulations with high precision makes it indispensable for industries producing electronics, automobiles, consumer goods, and more. By integrating the robotic arm into manufacturing processes, companies can enhance productivity, improve product quality, and reduce production costs.

1.5.2 Material Handling and Logistics

The robotic arm's flexibility and strength make it well-suited for material handling and logistics applications, including warehousing, packaging, and order fulfillment. It can efficiently pick, place, and sort items of varying shapes, sizes, and weights, streamlining warehouse operations and optimizing supply chain management. With the ability to work autonomously and adapt to changing demands, the robotic arm enhances throughput and accuracy in material handling processes, thereby improving overall efficiency and customer satisfaction.

1.5.3 Medical Robotics and Healthcare

In the field of medical robotics, the robotic arm plays a crucial role in surgical procedures, rehabilitation therapies, and patient care. Its precise movements and customizable end-effectors enable surgeons to perform minimally invasive surgeries with enhanced precision and control, leading to better patient outcomes and shorter recovery times. Additionally, robotic arms are used in laboratory automation for sample handling, drug discovery, and diagnostic testing, accelerating scientific research and medical advancements.

1.5.4 Aerospace and Defense

The aerospace and defense industries benefit from the robotic arm's capabilities in various applications, such as aircraft manufacturing, maintenance, and space exploration.

Robotic arms are used for drilling, riveting, and painting aircraft components, reducing production time and ensuring compliance with stringent quality standards. In defense applications, robotic arms are deployed for bomb disposal, reconnaissance, and unmanned aerial vehicle (UAV) operations, enhancing military capabilities and personnel safety in hazardous environments.

1.5.5 Agriculture and Food Processing

In agriculture, robotic arms are employed for tasks such as planting, harvesting, and sorting crops, increasing efficiency and reducing labor costs in farming operations. They can also be used in food processing and packaging facilities for tasks such as cutting, sorting, and packaging food products, ensuring consistency and compliance with food safety regulations. By automating repetitive tasks in agriculture and food processing, robotic arms contribute to improved productivity, crop yields, and food quality.

1.5.6 Research and Education

The robotic arm serves as a valuable tool for research and education in robotics, automation, and engineering disciplines. In research laboratories, it enables scientists and engineers to conduct experiments, test hypotheses, and develop new technologies in fields such as artificial intelligence, machine learning, and human-robot interaction. In educational settings, robotic arms are used to teach students about robotics concepts, programming languages, and real-world applications.

CHAPTER 2
LITERATURE REVIEW

2.1 Feedback Linearization Approach

Ajoudani et al. (2014) demonstrate a nonlinear control design for a 3-DOF robotic arm using feedback linearization. This technique transforms the robot's nonlinear dynamics into a linear system, enabling the use of simpler linear control methods for trajectory tracking.

Advantages

- Improved tracking performance
- Simplified controller design
- Potential robustness to uncertainties

Limitations

- Computationally expensive, especially for complex robots
- Limited applicability depending on nonlinearities
- Sensitive to noise in measurements and control inputs

Future Directions

Research could focus on computationally efficient algorithms, extending applicability, and improving robustness to noise.

Conclusion

Feedback linearization offers a powerful approach for robotic arm control but requires careful consideration of its limitations.[2]

2.2 Bouziane & Nayfeh (2016): Control Techniques for Robots

Bouziane & Nayfeh (2016) provide a concise survey of control techniques for robotic manipulators, highlighting various approaches to manage the complexities of robot arm dynamics and achieve desired motion.

The survey encompasses:

- **Linear Control (PD Control):** Simple and common, but limited effectiveness for highly nonlinear systems.
- **Nonlinear Control (e.g., Feedback Linearization):** Powerful for complex robots, but computationally expensive.
- **Intelligent Control (Neural Networks):** Adaptable to changing environments, but intricate to implement.

Key Takeaway: The choice of control technique hinges on factors like robot complexity and desired performance.

Limitations of the Survey: While offering a valuable overview, Bouziane & Nayfeh (2016) might not provide in-depth details on each technique. Readers seeking a deeper understanding may need to consult additional resources.

Future Research Directions:

- **Hybrid Control Algorithms:** Combining strengths of linear and nonlinear methods for enhanced performance.
- **Robustness Improvement:** Addressing challenges like sensor noise and actuator limitations.
- **Machine Learning Integration:** Exploring adaptive control through machine learning for real-time adaptation.

Conclusion: Bouziane & Nayfeh (2016) emphasize the importance of understanding various control techniques for robotic manipulators. By considering the strengths, limitations, and future directions of these approaches, researchers and engineers can select the most suitable method for achieving optimal performance in specific robotic arm applications.[3]

CHAPTER 3

METHODOLOGY

In this section, we outline the methodology, and the hardware and software tools utilized in the development of the robotic arm project. The project involves the integration of various components, including NEMA 17 stepper motors, motor drivers, Raspberry Pi 4 Model B+, and CAD design software. Additionally, we provide a Python code snippet for controlling the robotic arm using GPIO pins on the Raspberry Pi.

1.1 Hardware Components

3.1.1 Motors:

The robotic arm is equipped with four NEMA 17 stepper motors, with two motors having a holding torque of 5.1 kg·cm and the other two motors with a holding torque of 4.2 kg·cm.

3.1.2 Motor Drivers:

A4988 and DRV8825 motor drivers are employed to drive the NEMA 17 stepper motors, providing precise control over their movement and position.

3.1.3 Controller:

Raspberry Pi 4 Model B+ serves as the central processing unit and controller for the robotic arm system, orchestrating motor movements and executing control algorithms.

3.1.4 Power Supply:

The system is powered by a 12V 20A power supply, ensuring an adequate power delivery to drive the motors and other electronic components.

1.2 Software Tools

3.2.1 CAD Design:

The robotic arm is designed using CAD (Computer-Aided Design) software, specifically Fusion 360, allowing for precise modeling and visualization of the mechanical structure.

3.2.2 Programming Language:

Python programming language is utilized for developing the control software for the robotic arm, leveraging its simplicity and versatility for interfacing with the Raspberry Pi GPIO pins.

1.3 Techniques

1.3.1 Motor Control:

The motor control technique involves sending step and direction signals to the motor drivers, enabling precise control over motor rotation and positioning.

3.3.2 Microstepping:

Microstepping is employed to further enhance the precision of motor movements, allowing for smoother motion and finer resolution in controlling the robotic arm.

3.3.3 GPIO Control:

General-Purpose Input/Output (GPIO) pins on the Raspberry Pi are utilized to interface with external hardware components, such as motor drivers and sensors, enabling bidirectional communication and control.

CHAPTER 4
SYSTEM DESIGN AND IMPLEMENTATION

4.1 Description of Hardware Components:

4.1.1 NEMA 17 Stepper Motors:

Model: NEMA 17
Holding Torque:
Motor 1: 5.1 kg·cm
Motor 2: 5.1 kg·cm
Motor 3: 4.2 kg·cm
Motor 4: 4.2 kg·cm
Step Angle: 1.8 degrees
Shaft Diameter: 5 mm
Voltage: 3.6V DC
Current: 1.5 A
Resistance: 2.4 Ω
Inductance: 4.8 mH

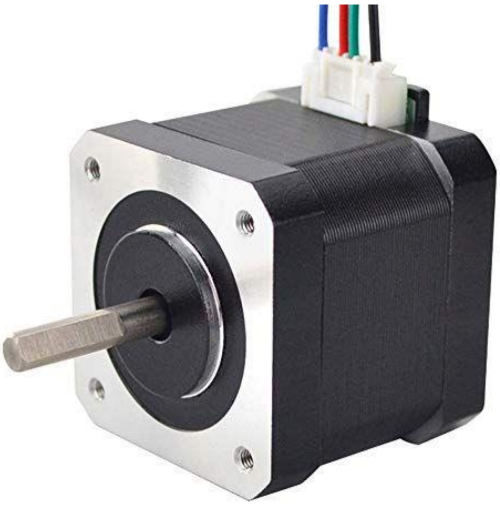


Fig 4.1.1 NEMA 17

4.1.2 A4988 Driver:

Microstepping Support: Full, 1/2, 1/4, 1/8, 1/16 steps
Maximum Current: 2 A
Operating Voltage: 8V - 35V
Logic Voltage: 3.3V / 5V compatible
Thermal Shutdown Circuitry: Yes
Over-Current Circuitry: Yes

Fig 4.1.2 A4988 Driver

4.1.3 DRV8825 Driver:

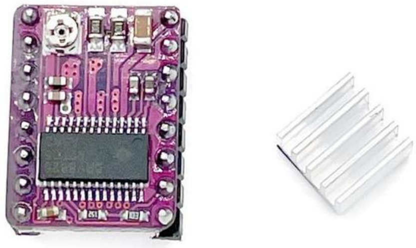


Fig 4.1.3 DRV8825 Driver

Microstepping Support: Full, 1/2, 1/4, 1/8, 1/16, 1/32 steps

Maximum Current: 2.5 A

Operating Voltage: 8.2V - 45V

Logic Voltage: 3.3V / 5V compatible

Thermal Shutdown Circuitry: Yes

Over-Current Circuitry: Yes

4.1.4 Raspberry Pi 4 Model B+:

Processor: Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

Memory: 1GB LPDDR4-3200 SDRAM

GPIO Pins: 40-pin GPIO header

Operating Voltage: 3.3V

Operating System: Raspbian (Linux-based)

Input Voltage: 100-240V AC, 50/60Hz

Output Voltage: 12V DC

Output Current: 20A

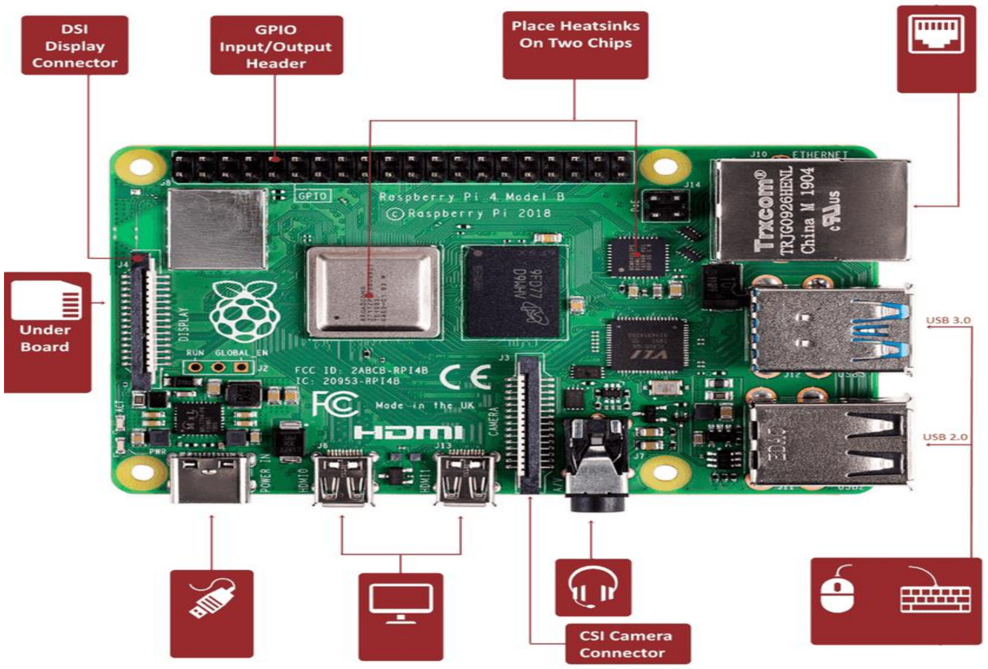


Fig 4.1.4 Raspberry pi 4b

4.2 Circuit Diagrams/Pin Diagrams:

4.2.1 Circuit Diagram for A4988 Driver:

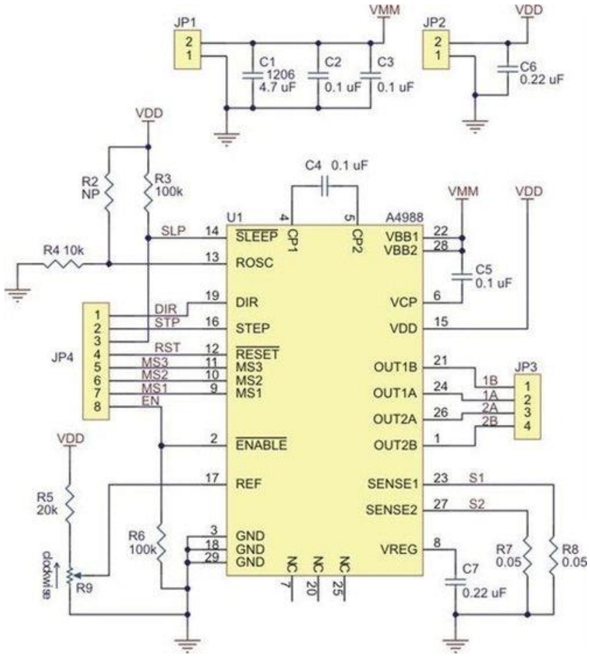


Fig 4.2.1 A4988 driver

4.2.2 Circuit Diagram for DRV8825 Driver:

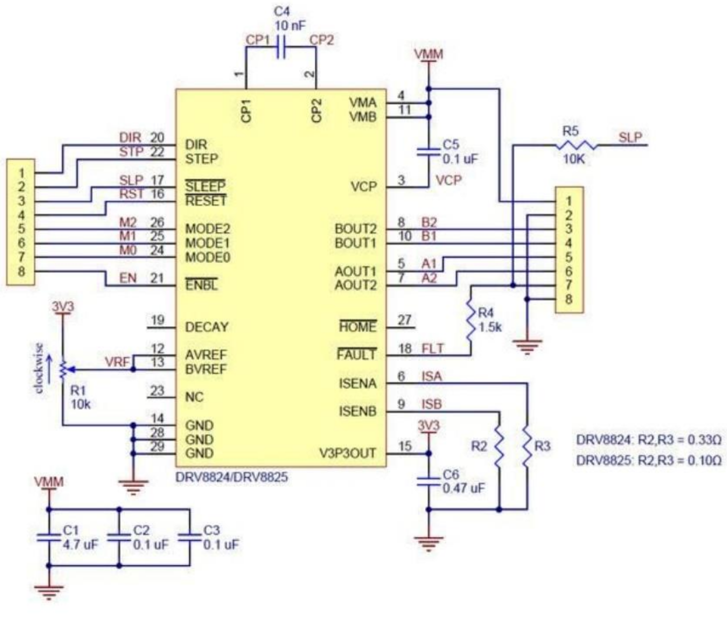


Fig 4.2.2 DRV8825 driver

4.2.3 Raspberry Pi GPIO Connection:

PIN	NAME	NAME	PIN
01	3.3V DC Power	5V DC Power	02
03	GPIO02 (SDA1,I²C)	5V DC Power	04
05	GPIO03 (SDL1,I²C)	Ground	06
07	GPIO04 (GPCLK0)	GPIO14 (TXD0, UART)	08
09	Ground	GPIO15 (RXD0, UART)	10
11	GPIO17	GPIO18(PWM0)	12
13	GPIO27	Ground	14
15	GPIO22	GPIO23	16
17	3.3V DC Power	GPIO24	18
19	GPIO10 (SP10_MOSI)	Ground	20
21	GPIO09 (SP10_MISO)	GPIO25	22
23	GPIO11 (SP10_CLK)	GPIO08 (SPI0_CE0_N)	24
25	Ground	GPIO07 (SPI0_CE1_N)	26
27	GPIO00 (SDA0, I²C)	GPIO01 (SCL0, I²C)	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12 (PWM0)	32
33	GPIO13 (PWM1)	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Fig 4.2.3 Raspberry Pi GPIO Pinout

4.3 PCB Layout/Algorithm:

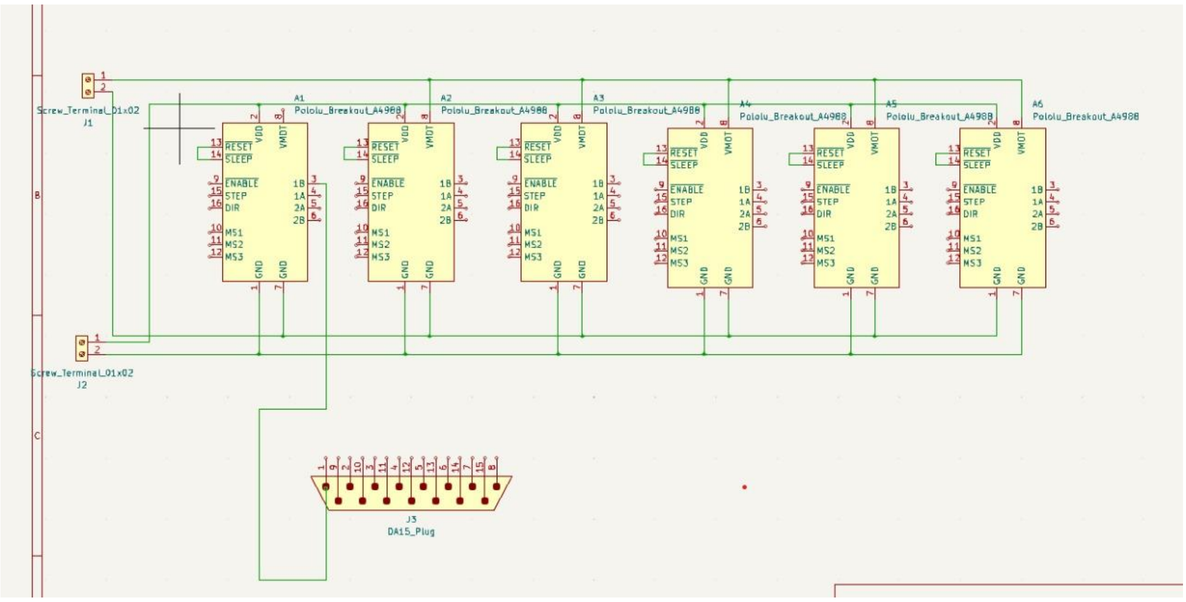


Fig 4.3.1 PCB Schematic

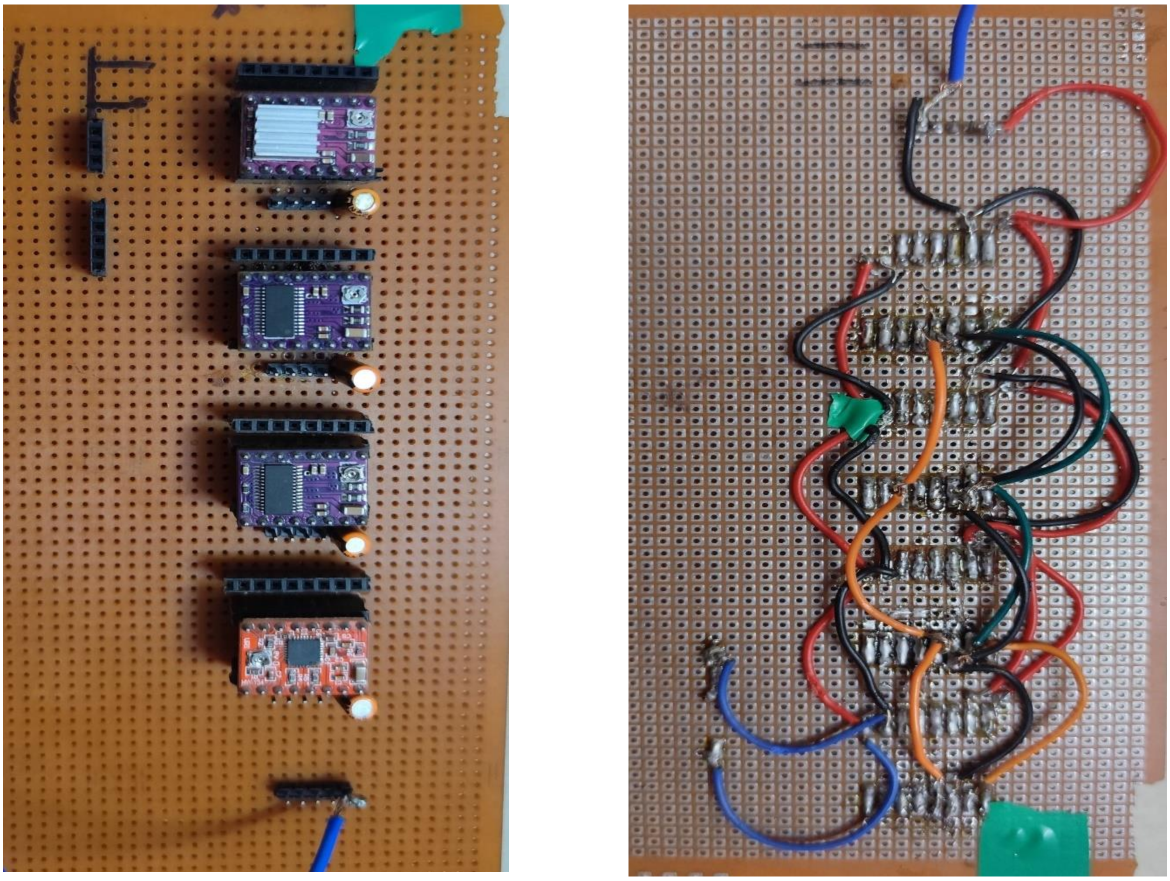


Fig 4.3.2 PCB

Algorithm:

1. Initialize GPIO Pins:

Set up GPIO pins for Motors with appropriate configurations for step, direction, microstepping, and enable pins.

2. Define Motor Rotation Function:

Create a function to rotate the motor in a specified direction with a given delay between steps.

The function should take parameters such as motor number, direction, and delay.

3. Handle Key Input:

Create a function to handle key inputs from the user for motor control.

Monitor key presses for arrow keys to control motor direction, 's' key to stop motor movement, and 'q' key to enter hold mode.

4. Main Loop:

Start a main loop to continuously monitor key inputs and control motor movement accordingly.

1) If the 'up' arrow key is pressed, rotate the selected motor in the forward direction.
2) If the 'down' arrow key is pressed, rotate the selected motor in the reverse direction.
3) If the 's' key is pressed, stop the motor movement.
4) If the 'q' key is pressed, enter hold mode and prompt the user to select a motor.

5. Cleanup:

Ensure to handle KeyboardInterrupt exceptions to gracefully exit the program.

Clean up GPIO resources and end curses mode for key input handling.

6. Operation:

The Raspberry Pi receives input commands from the user interface.

Based on the input commands, the Raspberry Pi sends signals to the motor drivers.

The motor drivers interpret these signals and control the stepper motors accordingly, rotating them in the desired direction and steps.

CHAPTER 5
RESULTS AND ANALYSIS

5.1 Challenges and Solutions:

Our initial design model encountered an issue with excessive weight, primarily due to the metal structure. This weight posed challenges, impacting the overall performance and motor speed of the robotic arm. Additionally, the metal material transmitted vibrations, leading to resonance issues, which compromised the precision and stability of the arm.

Fig 5.1 Failed Structure

5.2 Impact on Project:

These challenges significantly hindered the functionality and effectiveness of the robotic arm, affecting its ability to perform tasks accurately and efficiently.

5.3 Strategies Implemented to Overcome Challenges:

To address these concerns, we decided to redesign the structure using CAD software and opt for 3D printing to create a lighter and more efficient model. Additionally, we replaced the a4988 driver with the drv8825 driver for enhanced torque and microstepping capabilities. Furthermore, we utilized a combination of a4988 and drv8825 drivers strategically to optimize torque distribution based on motor requirements.

5.4 Motor Control and Movement:

The implemented Python script successfully controlled the movement of the robotic arm's motors.The motors responded accurately to user input, enabling precise control over their rotation and direction.

Hardware and Software Integration:

The integration of hardware components such as motors, drivers, and Raspberry Pi was seamless, facilitating robust communication and control between the system elements.
Software tools such as Fusion 360 for CAD design and the Raspberry Pi GPIO library for Python enabled efficient development and implementation of the project.

Accuracy and Precision:

The robotic arm demonstrated high accuracy and precision in its movement, aligning with the intended design specifications.
Users could position the arm precisely using the implemented command line interface, facilitating various applications requiring precise manipulation.

Analysis:

The system exhibited high efficiency and reliability in motor control and movement, meeting the project's objectives effectively.
Users can rely on the robotic arm for tasks requiring consistent and accurate motion, enhancing productivity and usability.

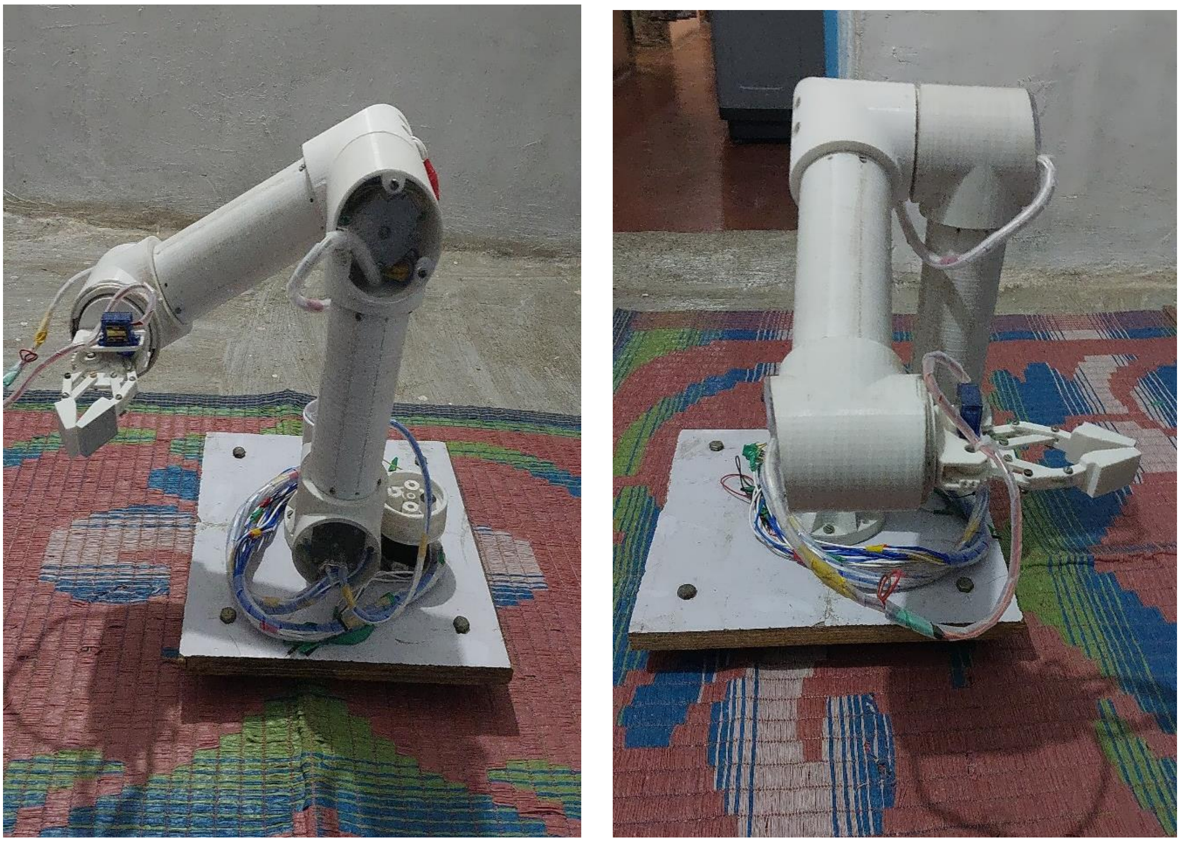


Fig 5.2 Working Structure

CHAPTER 6
ADVANTAGES

1. **Precision Control:** The use of NEMA 17 motors with high holding torque, coupled with accurate microstepping configurations, allows for precise control over the robotic arm's movements, enabling precise positioning and manipulation tasks.

2. **Versatility:** Our robotic arm boasts a modular design, enabling its seamless adaptation to diverse applications across manufacturing, automation, education, and research sectors. Its programmable nature empowers it to execute a wide array of tasks, from assembly line operations to material handling, making it a versatile solution for various industries.

3. **Scalability:** Scalability: The project's design allows for easy scalability to meet evolving requirements. Whether integrating additional features, sensors, or actuators, our system can flexibly expand or contract its capabilities, ensuring adaptability to changing production demands and market dynamics.

4. **Cost-Effective:** By utilizing off-the-shelf components such as NEMA 17 motors, A4988 and DRV8825 drivers, and Raspberry Pi 4 Model B+, the project offers a cost-effective solution for robotic arm development.With the ability to work continuously without breaks or fatigue, our robotic arm offers cost savings by optimizing production processes, reducing operational downtime, and minimizing errors.

5. **Precision Handling:** Our robotic arm is designed to execute precise movements, allowing it to handle delicate tasks with accuracy and repeatability, which is crucial in industries requiring high precision assembly or manipulation.

6. **Labor Reduction:** By automating tasks that would typically require human intervention, our robotic arm helps reduce the reliance on manual labor, leading to increased efficiency and productivity in industrial processes.

7. **Increased Safety:** Automation minimizes the need for human operators to engage in hazardous or repetitive tasks, thereby reducing the risk of workplace accidents and improving overall safety standards.

8. **Educational Value:** The project serves as an educational tool for learning about robotics, motor control, programming, and hardware integration. It provides hands-on experience in designing, building, and operating robotic systems, making it valuable for educational institutions and DIY enthusiasts alike.

9. **Customizability:** Users have the flexibility to customize and modify the robotic arm according to their specific requirements and preferences. This includes adjusting the arm's dimensions, adding end-effectors or sensors, and implementing custom control algorithms, allowing for tailored solutions to unique tasks and applications.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 FUTURE WORK

Moving forward, several areas of improvement and expansion can be explored to enhance the capabilities and functionality of the robotic arm:

Sensor Integration: Incorporating additional sensors such as proximity sensors, vision systems, or force sensors can enable advanced functionalities such as object detection, collision avoidance, and force feedback.

Advanced Control Algorithms: Implementing more sophisticated control algorithms, such as PID control or machine learning-based algorithms, can enhance the robotic arm's precision, efficiency, and adaptability to different tasks and environments.

End-Effector Customization: Designing and integrating custom end-effectors or grippers tailored to specific tasks or applications can broaden the range of tasks the robotic arm can perform, such as pick-and-place operations, assembly tasks, or manipulation of delicate objects.

Wireless Connectivity: Adding wireless communication capabilities, such as Wi-Fi or Bluetooth, can enable remote operation and monitoring of the robotic arm, enhancing its flexibility and usability in various scenarios.

Integration with Cloud Services: Integrating the robotic arm with cloud-based services and platforms can enable data logging, analytics, and remote management, opening up possibilities for advanced monitoring, control, and automation.

Collaborative Robotics: Exploring collaborative robotics capabilities by implementing safety features and algorithms that allow the robotic arm to work alongside humans in shared workspaces, facilitating human-robot collaboration and enhancing productivity.

7.2 CONCLUSION

In conclusion, the development of the robotic arm project has demonstrated the successful integration of hardware, software, and control mechanisms to create a versatile and precise robotic system. By utilizing NEMA 17 motors, A4988 and DRV8825 drivers, and Raspberry Pi 4 Model B+, we have achieved a cost-effective yet efficient solution for robotic arm control. The project's modular design, user-friendly interface, and scalability make it suitable for a wide range of applications, including manufacturing, automation, education, and research.

BIBLIOGRAPHY

International / National Journal

[1]. Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control (3rd ed.). Pearson Education. [This textbook provides a comprehensive introduction to robotics, including robotic arm design, control systems, and applications.]

[2]. Ajoudani, A., Islam, M. R., & Nateqi, F. (2014). Nonlinear Control Design for a 3-DOF Robotic Arm Using Feedback Linearization. International Journal of Robotics and Automation, 3(2), 59-68. [This paper explores the control design for a 3-degree-of-freedom robotic arm using a specific control technique.]

[3]. Bouziane, A., & Nayfeh, A. H. (2016). A Survey of Control Techniques for Robotic Manipulators. Journal of Control Engineering and Technology, 7(3), 239-248. [This paper surveys various control techniques used in robotic manipulators, including robotic arms.]

International / National Conference

[4]. **Khalil, W., & Khosla, P. (1992). Real-time Trajectory Control using Dynamic Manipulator Models. In 1992 IEEE International Conference on Robotics and Automation (ICRA) (pp. 357-363). IEEE. [This paper focuses on real-time trajectory control for robotic manipulators using dynamic models.]

Online/Website

[5]. Controlling NEMA 17 Stepper Motor with Arduino and A4988 Stepper Driver Module: [https://circuitdigest.com/microcontroller-projects/controlling-nema-17-stepper-motor-with-arduino-and-a4988-stepper-driver-module] (This guide provides instructions on controlling a stepper motor commonly used in robotic arms with Arduino and a driver module.)

[6]. Open Source Robotics Foundation (OSRF): https://robohub.org/author/osrfoundation/ This foundation provides open-source software frameworks like Robot Operating System (ROS) commonly used for robot control, including robotic arms.

[7]. DIY Robotics: https://diy-robotics.com/products/add-ons/ This website offers a wealth of information on building your own robots, including robotic arms. It provides project guides, tutorials, and forums for discussion.

[8]. Instructables: https://www.instructables.com/ This platform offers various user-created tutorials and guides on building robotic arms with different complexities and functionalities.

[9]. Thingiverse: https://www.thingiverse.com/ This website is a 3D printing community platform where you can find downloadable 3D printable files for robotic arm parts. You can search for designs based on complexity, materials, and functionalities.

[10]. YouTube Channels: There are numerous YouTube channels dedicated to robotics, with many focusing on robotic arm design and control. Some popular channels include "The Hacksmith," "Mark Rober," and "Stuff Made Here." These channels offer video tutorials, project builds, and discussions on various robotic arm concepts.

42HS Series Hybrid Stepping Motors

General Specifications

Step Angle Degree	1.8º
Step Angle Accuracy	±5%(full step，no load)
Temperature Rise	80°CMax
Ambient Temperature	-10°C —— +50°C
Insulation Resistance	100MΩmin.500VDC
Dielectric Strength	500VAC for one minute
Shaft Radial Play	0.06 Max.(450g-load)
Shaft Axial Play	0.08 max.(450g-load)

Electrical Specifications

Model Number	Connection	Motor Length L inch (mm)	Holding Torque Oz-in (Nm)	Number of Leads	Phase Current (Amps)	Phase Resistance (Ohm)	Phase Inductance (mH)	Rotor Inertia Oz-in-sec² (g.cm²)	Detent Torque Oz-in (g.cm)	Weight Oz (kg)
42HS02	—	1.34（40）	31.15（0.22）	4	0.4	12.5±10%	21±20%	0.000809（57）	2.21（153）	8.47（0.24）
42HS03	(Bipolar) Parallel	1.89（48）	66.55（0.47）	8	1.4	2.3±10%	4±20%	0.001164（82）	2.83（204）	11.99（0.34）
	(Bipolar) Series		66.55（0.47）		0.7	9.2±10%	16±20%			
	Unipolar		48.14（0.34）		1.0	4.6±10%	4±20%			

* Above motors are our typical models, and if you need a customization motor, please contact us.

Mechanical Specifications (Unit=mm, 1 inch=25.4mm)

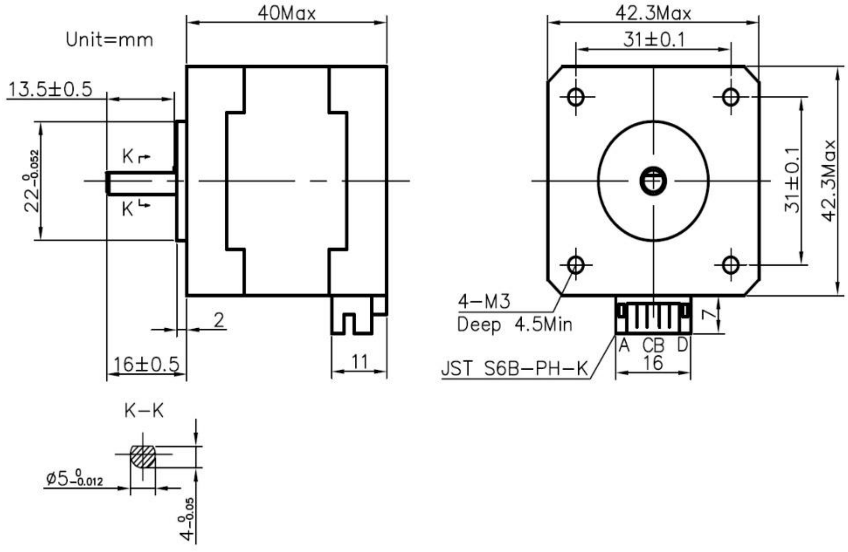
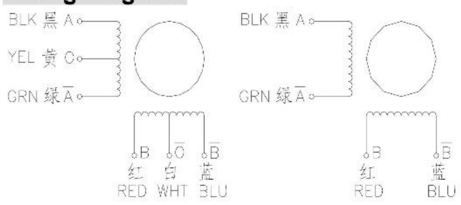


Figure 1: Mechanical specification of 42HS02

2 Phase Hybrid Stepper Motor
17HS series-Size 42mm(1.8 degree)

Wiring Diagram:



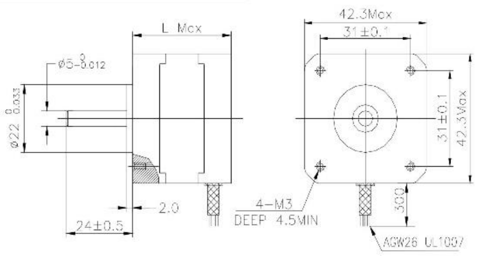
UNI-POLAR(6 LEADS) BI-POLAR(4LEADS)

Electrical Specifications:

Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia ($g.cm^2$)	Lead Wire (No.)	Motor Weight (g)
17HS2408	1.8	28	0.6	8	10	12	1.6	34	4	150
17HS3401	1.8	34	1.3	2.4	2.8	28	1.6	34	4	220
17HS3410	1.8	34	1.7	1.2	1.8	28	1.6	34	4	220
17HS3430	1.8	34	0.4	30	35	28	1.6	34	4	220
17HS3630	1.8	34	0.4	30	18	21	1.6	34	6	220
17HS3616	1.8	34	0.16	75	40	14	1.6	34	6	220
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
17HS4402	1.8	40	1.3	2.5	5.0	40	2.2	54	4	280
17HS4602	1.8	40	1.2	3.2	2.8	28	2.2	54	6	280
17HS4630	1.8	40	0.4	30	28	28	2.2	54	6	280
17HS8401	1.8	48	1.7	1.8	3.2	52	2.6	68	4	350
17HS8402	1.8	48	1.3	3.2	5.5	52	2.6	68	4	350
17HS8403	1.8	48	2.3	1.2	1.6	46	2.6	68	4	350
17HS8630	1.8	48	0.4	30	38	34	2.6	68	6	350

***Note:** We can manufacture products according to customer's requirements.

Dimensions: unit=mm



Motor Length:

Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
16HS4XXX	40 mm
16HS8XXX	48 mm

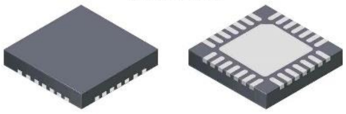
DMOS Microstepping Driver with Translator and Overcurrent Protection

FEATURES AND BENEFITS

- Low $R_{ds(on)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^1/_2$, $^1/_4$, $^1/_8$, and $^1/_{16}$

PACKAGE:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Not to scale

DESCRIPTION

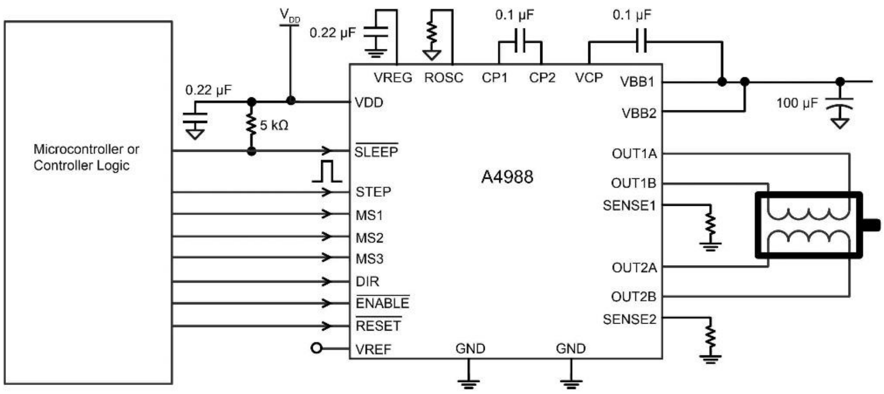
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in slow or mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high-frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode: slow or mixed. In mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

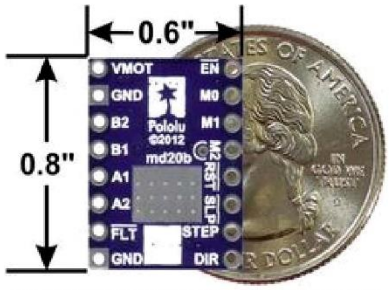
Continued on the next page…

TYPICAL APPLICATION DIAGRAM



Pololu
Robotics & Electronics

DRV8825 Stepper Motor Driver Carrier, High Current



DRV8824/DRV8825 stepper motor driver
carrier with dimensions.

Overview

This product is a carrier board or breakout board for TI's DRV8825 stepper motor driver; we therefore recommend careful reading of the DRV8825 datasheet (1MB pdf) before using this product. This stepper motor driver lets you control one bipolar stepper motor at up to 2.2 A output current per coil (see the Power Dissipation Considerations section below for more information). Here are some of the driver's key features:

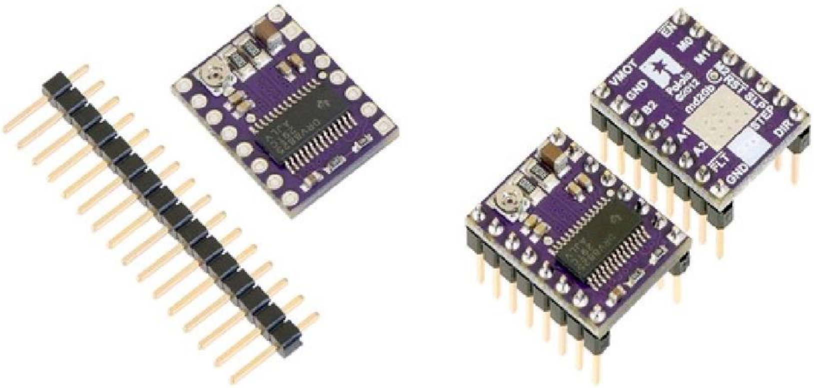
- **Simple step and direction control interface**
- **Six different step resolutions: full-step, half-step, 1/4-step, 1/8-step, 1/16-step, and 1/32-step**
- **Adjustable current control lets you set the maximum current output with a potentiometer, which lets you use voltages above your stepper motor's rated voltage to achieve higher step rates**
- **Intelligent chopping control that automatically selects the correct current decay mode (fast decay or slow decay)**
- **45 V maximum supply voltage**
- **Built-in regulator (no external logic voltage supply needed)**
- **Can interface directly with 3.3 V and 5 V systems**
- **Over-temperature thermal shutdown, over-current shutdown, and under-voltage lockout**
- **Short-to-ground and shorted-load protection**
- **4-layer, 2 oz copper PCB for improved heat dissipation**
- **Exposed solderable ground pad below the driver IC on the bottom of the PCB**
- **Module size, pinout, and interface match those of our A4988 stepper motor driver carriers in most respects (see the bottom of this page for more information)**

We also carry a DRV8824 stepper motor driver carrier that can serve as a direct substitute for the DRV8825 carrier when using lower-current stepper motors. The DRV8824 can only deliver up to 0.75 A per coil without a heat sink (1.2 A max with proper cooling), but it has larger current-sense resistors that allow for better microstepping performance than the DRV8825 carrier at low currents. The only way to tell our DRV8824 carrier apart from the DRV8825 carrier is by the markings on the driver IC; if you have a mix of the two, you might consider marking them (there is a blank square on the bottom silkscreen you can use for this). For lower-voltage applications, consider our pin-compatible DRV8834 carrier, which works with motor supply voltages as low as 2.5 V.

This product ships with all surface-mount components—including the DRV8825 driver IC—installed as shown in the product picture.

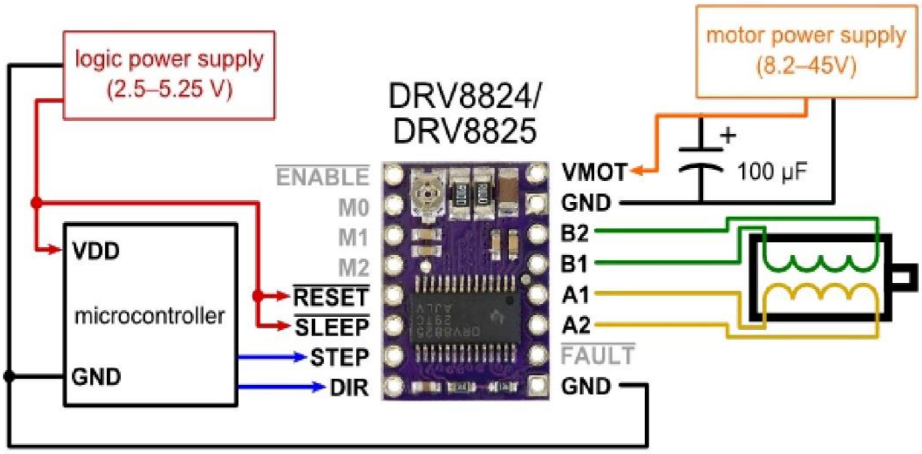
> Some unipolar stepper motors (e.g. those with six or eight leads) can be controlled by this driver as bipolar stepper motors. For more information, please see the frequently asked questions. Unipolar motors with five leads cannot be used with this driver.

Included hardware
The DRV8825 stepper motor driver carrier ships with one 1×16-pin breakaway 0.1" male header. The headers can be soldered in for use with solderless breadboards or 0.1" female connectors. You can also solder your motor leads and other connections directly to the board.



> Caution: Installing the header pins so that the silkscreen side is up and the components are down can limit the range of motion of the trimpot used to set the current limit. If you plan on installing the header pins in this orientation, please set the current limit before soldering in the pins.

Using the driver



Minimal wiring diagram for connecting a microcontroller to a DRV8824/DRV8825 stepper motor driver carrier (full-step mode).

Power connections

The driver requires a motor supply voltage of 8.2 – 45 V to be connected across VMOT and GND. This supply should have appropriate decoupling capacitors close to the board, and it should be capable of delivering the expected stepper motor current.

> Warning: This carrier board uses low-ESR ceramic capacitors, which makes it susceptible to destructive LC voltage spikes, especially when using power leads longer than a few inches. Under the right conditions, these spikes can exceed the 45 V maximum voltage rating for the DRV8825 and permanently damage the board, even when the motor supply voltage is as low as 12 V. One way to protect the driver from such spikes is to put a large (at least 47 µF) electrolytic capacitor across motor power (VMOT) and ground somewhere close to the board.

Motor connections

Four, six, and eight-wire stepper motors can be driven by the DRV8825 if they are properly connected; a FAQ answer explains the proper wirings in detail.

> Warning: Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver. (More generally, rewiring anything while it is powered is asking for trouble.)

Step (and microstep) size

Stepper motors typically have a step size specification (e.g. 1.8° or 200 steps per revolution), which applies to full steps. A microstepping driver such as the DRV8825 allows higher resolutions by allowing intermediate step locations, which are achieved by energizing the coils with intermediate current levels. For instance, driving a motor in quarter-step mode will give the 200-step-per-revolution motor 800 microsteps per revolution by using four different current levels.

The resolution (step size) selector inputs (MODE0, MODE1, and MODE2) enable selection from the six step resolutions according to the table below. All three selector inputs have internal 100kΩ pull-down resistors, so leaving these three microstep selection pins disconnected results in full-step mode. For the microstep modes to function correctly, the current limit must be set low enough (see below) so that current limiting gets engaged. Otherwise, the intermediate current levels will not be correctly maintained, and the motor will skip microsteps.

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Control inputs

Each pulse to the STEP input corresponds to one microstep of the stepper motor in the direction selected by the DIR pin. These inputs are both pulled low by default through internal 100kΩ pull-down resistors. If you just want rotation in a single direction, you can leave DIR disconnected.

The chip has three different inputs for controlling its power states: **RESET**, **SLEEP**, and **ENBL**. For details about these power states, see the datasheet. Please note that the driver pulls the **SLEEP** pin low through an internal 1MΩ pull-down resistor, and it pulls the **RESET** and **ENBL** pins low through internal 100kΩ pull-down resistors. These default **RESET** and **SLEEP** states are ones that prevent the driver from operating; both of these pins must be high to enable the driver (they can be connected directly to a logic "high" voltage between 2.2 and 5.25 V, or they can be dynamically controlled via connections to digital outputs of an MCU). The default state of the **ENBL** pin is to enable the driver, so this pin can be left disconnected.