;

Mini Project Report on
Gesture Controlled Robotic Arm Using MediaPipe
&
Robotic Arm Using Gui

By

Ansh Mukesh Kini (27)
Atharva Sharad Wakchure (54)
Jishnu Hareesh Kanachery (60)
Om Vinay Kothavale(62)

Subject: Mini Project – 1B

Under Guidance of:
Dr. Ashish Vanmali

Department of
Electronics & Telecommunication Engineering

Vidyavardhini's College of Engineering & Technology

University of Mumbai

2024-25

Vidyavardhini's College of Engineering & Technology

;

CERTIFICATE

This is to certify that following students

Ansh Mukesh Kini (27)
Atharva Sharad Wakchure (54)
Jishnu Hareesh Kanachery (60)
Om Vinay Kothavale(62)

have submitted mini project report – 1B entitled

Gesture Controlled Robotic Arm Using MediaPipe &
Robotic Arm Using GUI

S.E Sem IV Electronics & Telecommunication Engineering

academic year 2024-2025

Internal Guide : Dr. Ashish Vanmali ( )

Internal Examiner : _____ ( )

External Examiner : _____ ( )

Dr. Amrita Ruperee	Dr. Rakesh Himte
HOD, EXTC	Principal, VCET

;

Date: _____

Place: _____

;

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ansh Mukesh Kini (27)
Atharva Sharad Wakchure (54)
Jishnu Hareesh Kanachery (60)
Om Vinay Kothavale(62)

Date: _____

;

Acknowledgement

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this Mini Project 1B.We would like to thank our HOD, Dr Amrita Ruperee, our Management, our Principal, Dr Rakesh Himte and our respected professor and project guide Mr. Ashish Vanmali for being there constantly, and for letting us pick this topic which enabled us to study and learn about Arduino and Raspberry Pi based projects which otherwise we would have not been able to do .We would like to thank our guide Mr. Ashish Vanmali for their invaluable guidance, mentorship, expertise and unwavering support throughout the entire duration of the project. We would also like to extend our heartfelt thanks to Vidyavardhini's College of Engineering and Technology for providing the necessary resources and infrastructure that enabled us to carry out this project.

;

Abstract

This mini project focuses on the development of a multifunctional robotic arm system that can be controlled using both real-time hand gestures via computer vision and a graphical user interface (GUI). The system is designed to demonstrate the versatility and adaptability of robotic arm control using two parallel approaches. The first control method involves a gesture recognition system powered by OpenCV and MediaPipe in Python, which captures hand gestures through a webcam. The computer vision model detects hand landmarks, interprets predefined gestures such as open palm, fist, or directional pointing, and translates these into specific motion commands. These commands are then sent to an Arduino microcontroller, which actuates multiple servo motors attached to the robotic arm, enabling it to perform actions like gripping, lifting, and rotating. This approach provides an intuitive and contactless way to control the arm, enhancing the human-machine interaction experience.

In addition to gesture control, the project also incorporates a Raspberry Pi Pico-based robotic arm that is operated through a custom-built graphical user interface (GUI). The GUI, developed using Python libraries such as Tkinter or PyQt, allows users to manually control each joint of the robotic arm by interacting with sliders or buttons on the screen. The Raspberry Pi Pico receives the input from the GUI via USB serial communication or UART, processes it, and generates PWM signals to drive the servos. This dual-mode control system makes the project highly flexible — the GUI provides precise manual control and is especially useful for testing or when gesture input is not feasible, while the gesture mode offers an innovative, hands-free control method ideal for interactive demos and accessibility-focused applications.

The robotic arm used in both systems typically includes 4 to 6 degrees of freedom (DOF), allowing a range of human-like movements. The structure is built using lightweight materials like plastic or aluminum and powered by servo motors such as SG90 or MG996R. The integration of both gesture recognition and GUI control showcases advanced concepts in embedded systems, computer vision, electronics, and user interface design. This project finds potential applications in educational robotics, prosthetics research, assistive technologies, industrial automation, and remote-controlled operations. Overall, it demonstrates a holistic approach to robotic arm control by combining real-time gesture tracking with hardware programming and interactive design.

;

PART A
Mini Project Report on

Gesture Controlled Robotic Arm Using MediaPipe

;

Content

;

List Of Figures

;

Chapter 1
Introduction

The Arduino-controlled robotic arm using Media Pipe for gesture recognition is an innovative project that bridges the fields of computer vision and robotics to create an intuitive and interactive control system. This project aims to build a smart robotic arm that can be operated using real-time hand gestures, eliminating the need for traditional remote controls or physical contact. The core concept revolves around using a webcam to capture the user's hand movements, which are processed using Media Pipe – a powerful machine learning framework developed by Google for real-time hand tracking and gesture recognition. The interpreted gestures are translated into commands that are sent to an Arduino Uno microcontroller, which in turn drives servo motors to control the joints of the robotic arm.

Gesture-controlled systems are becoming increasingly popular due to their potential in enhancing human-computer interaction. This project leverages that potential by providing a cost-effective, responsive, and portable solution for gesture-based robotic control. The use of Arduino for motion control offers reliability, ease of implementation, and wide community support. The system architecture consists of a camera module, a computer or Raspberry Pi running the MediaPipe-based Python script, serial communication with the Arduino, and a multi-DOF (Degree of Freedom) robotic arm powered by servo motors.

The main motivation behind this project is to develop a hands-free control mechanism for robotic systems, especially in applications where touch-based control is inconvenient or unsafe—such as in hazardous environments, assistive robotics for people with disabilities, or educational platforms for teaching robotics and programming. By integrating vision-based gesture recognition and embedded hardware control, the project demonstrates a practical and engaging use of modern technologies in robotics.

This project not only introduces students and enthusiasts to the fields of computer vision, embedded systems, and mechanical design but also encourages a multidisciplinary approach by combining software development, hardware interfacing, and real-time system integration. The result is a user-friendly, gesture-driven robotic arm system that opens doors to more accessible and advanced human-robot interaction.

;

Chapter 2

Literature Review

In the study by Sachin Kumar and Mahesh Sharma (2022) titled *"Hand Gesture Controlled Robotic Arm Using OpenCV and Arduino"*, the authors implemented a low-cost robotic arm that can be controlled via real-time hand gestures tracked using OpenCV. The system utilized a webcam to capture hand movements, processed the image using Python, and sent corresponding commands via serial communication to the Arduino. This paper demonstrated the feasibility and responsiveness of using vision-based interfaces in robotics without the need for physical controllers or gloves.

Another relevant paper, "Real-Time Hand Gesture Recognition using MediaPipe and its Application in Robotic Arm Control" by Ananya S. and Dinesh Kumar (2021), explored the use of Google's MediaPipe framework for efficient and lightweight hand tracking. MediaPipe offered more accurate and stable landmark detection compared to traditional OpenCV-based contour detection techniques. This improvement allowed for smoother and more reliable control of servo-driven robotic arms. The authors highlighted MediaPipe's advantages in terms of computational efficiency, real-time performance, and ease of implementation, making it ideal for embedded or real-time robotics projects.

A third study by M. Patel and R. Joshi (2020), titled *"Arduino-Based Gesture Controlled Robotic Arm Using Machine Vision,"* focused on combining machine vision with embedded control systems for robotics education. The researchers created a GUI and gesture control hybrid system for controlling robotic arms. They emphasized the importance of using microcontrollers like Arduino for servo control due to their real-time responsiveness and ease of use. Their research suggested that such systems are particularly beneficial in assistive robotics, helping individuals with mobility impairments perform daily activities with minimal physical effort.

These papers collectively show that gesture-controlled robotic arms, especially those using Media Pipe, represent a significant advancement in the field of human-machine interaction. The use of Media Pipe for real-time, marker less hand tracking enhances both the accuracy and user experience of robotic control systems. Combining this with Arduino's reliable actuation capabilities results in a powerful, low-cost, and accessible platform for developing educational tools, assistive devices, and automation systems.

;

Chapter 3

Problem Definition

In today's rapidly advancing technological landscape, there is an increasing demand for intuitive, contactless, and efficient methods to control robotic systems, especially in areas such as prosthetics, assistive technologies, industrial automation, and remote operations. Traditional robotic control methods often rely on complex hardware interfaces such as joysticks, remote controls, or specialized wearables, which can be bulky, expensive, or non-intuitive for common users. Furthermore, these methods may not be accessible to individuals with physical limitations. Existing gesture recognition systems using hardware like accelerometers or color markers often face limitations in accuracy, real-time responsiveness, and ease of integration. Additionally, many such systems require significant computational resources or lack flexibility for customization and deployment on low-cost platforms.

The problem lies in the absence of a low-cost, vision-based robotic arm control solution that is easy to implement, responsive in real-time, and flexible enough to adapt to various tasks and environments using only a standard webcam and hand gestures.

Solution Statement

To design and implement a cost-effective and flexible gesture-controlled robotic arm using the MediaPipe framework for hand tracking, integrated with an Arduino microcontroller for motor control. The system will capture and interpret real-time hand gestures through a webcam using MediaPipe, translate those gestures into specific commands, and send them to the robotic arm via serial communication. The project aims to provide an interactive, contactless, and accessible robotic control method that can be applied in assistive technologies, education, and prototyping.

;

Chapter 4

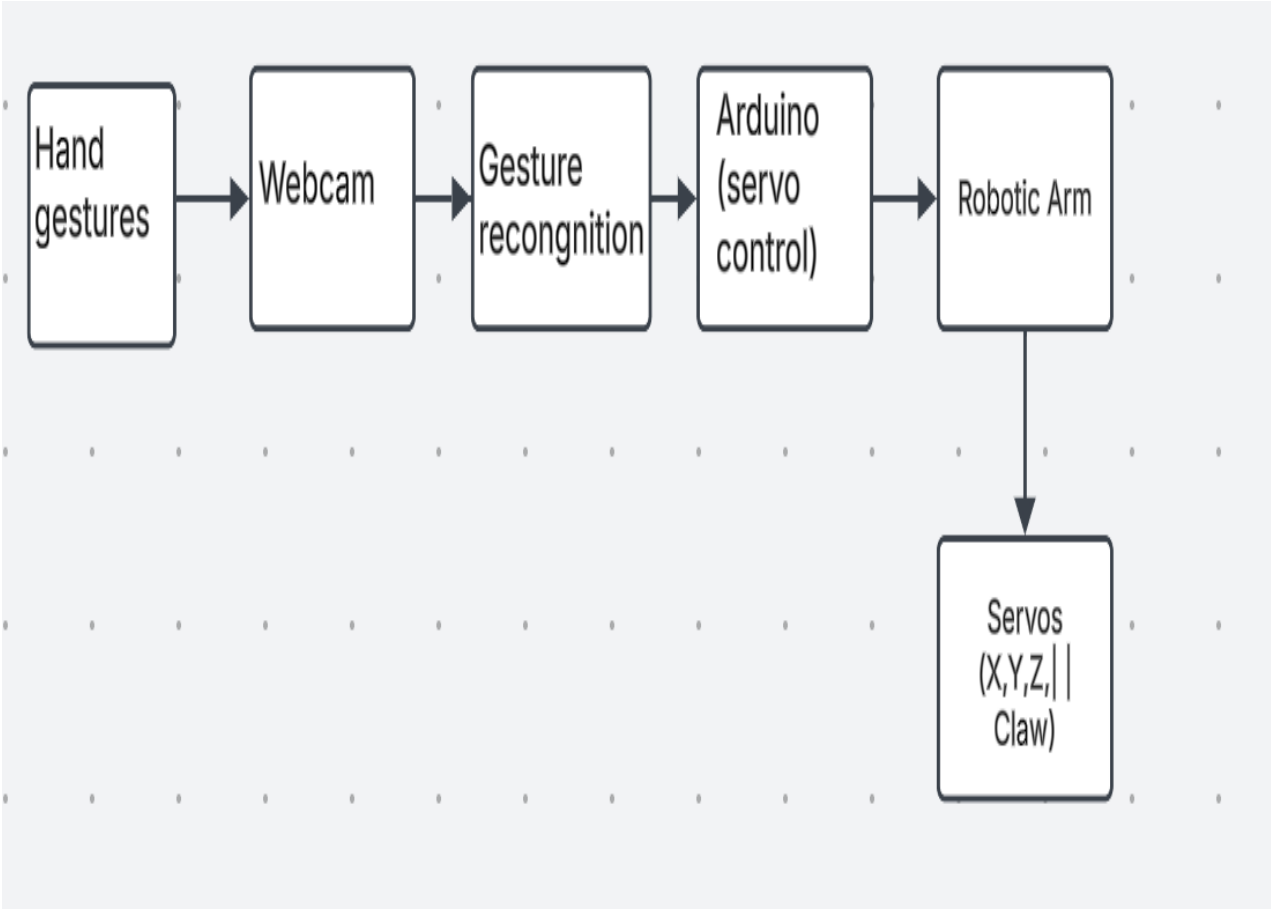
Proposed Method/Approach/Algorithm

The Gesture-Controlled Robotic Arm using MediaPipe and Arduino operates by integrating two primary components: a gesture recognition system and a robotic arm control system. The first component captures and interprets hand gestures, while the second controls the robotic arm based on the recognized gestures. To detect hand gestures, the MediaPipe framework is employed, utilizing a webcam or video capture device. MediaPipe's real-time hand tracking model processes the video feed to identify hand positions and gestures such as open hand, fist, or pointing finger. Once the gesture is detected, the Arduino Uno is used to control the robotic arm. The arm is made up of multiple servos that are responsible for controlling the joints of the arm, such as lifting, rotating, or extending. These servos are controlled through PWM signals sent from the Arduino board, which is connected to the system.

The system's hardware setup involves a USB webcam or an external camera to capture real-time video of hand gestures. This video feed is processed by OpenCV and MediaPipe to detect and classify hand gestures. Once the gesture is recognized, it is sent to the Arduino controller via serial communication. The Arduino is programmed to interpret these gestures and convert them into commands that control the servos, thereby moving the robotic arm in the desired manner. The software development phase includes using OpenCV for video capture and processing, while MediaPipe performs the hand gesture recognition. Arduino IDE is used for programming the Arduino Uno to control the robotic arm. The communication between the computer (running the gesture recognition software) and the Arduino is established using a serial USB connection, ensuring real-time control of the robotic arm based on the recognized gestures.

;

Block diagram of robotic arm



;

. # Chapter 5

Implementation Methodology

The hardware design of the robotic arm involves selecting and configuring the components for optimal performance and reliability. It involves: -

5.1 Circuit Diagram

Fig 5.1 illustrates the connections and electrical components used in the robotic arm is essential. This diagram provides a detailed overview of how the hardware components connect with each other.

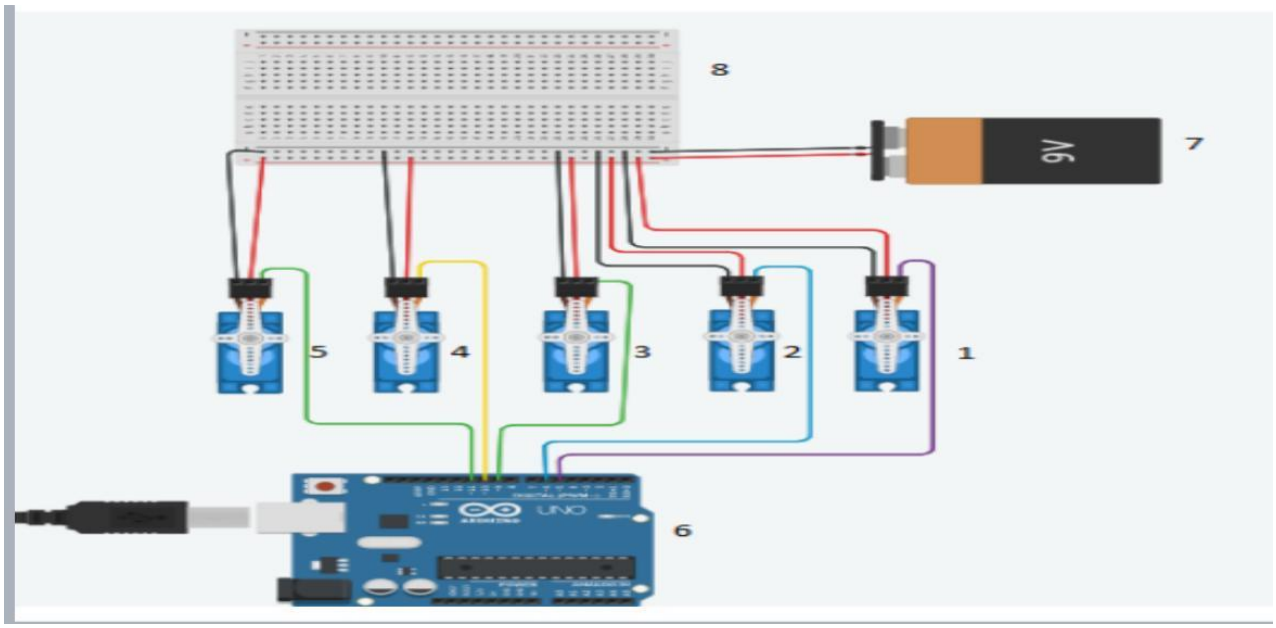


Fig 5.1.: Schematic Diagram of robotic arm controlled using mediapipe

;

5.2 Hardware / Software Requirement

The following components are very used implement robotic arm

5.2.1 Arduino Uno Board:

Fig 5.2.1 serves as the central processing unit for the robotic . It's responsible for interfacing with all the sensors, storing data, and controlling the modules.

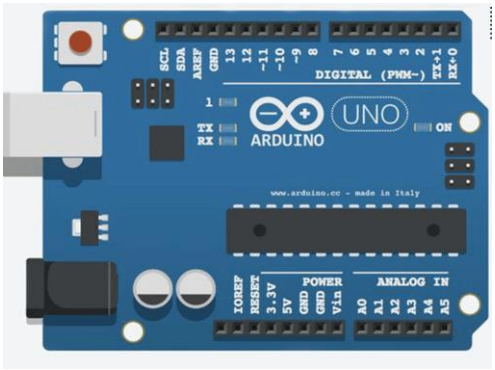


Fig 5.2.1: Figure of Arduino Uno Board

5.2.2 Mg99r servo motor

Fig 5.2.2: Figure of servo motor

5.2.3 Web Cam:

Fig 5.2.3 Web cam can be used to sense the gesture using real time image processing

Fig 5.2.3: Figure of web cam

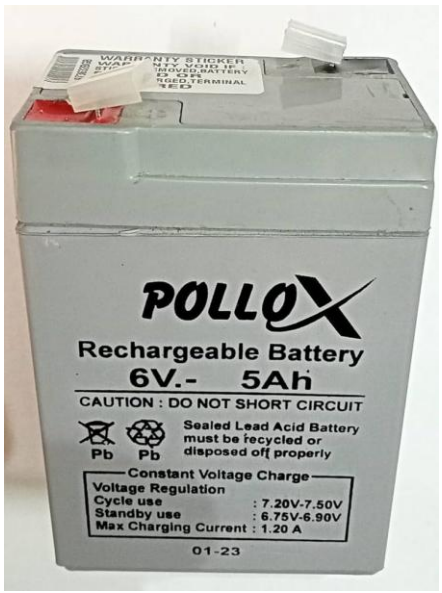
;

5.2.4 Arduino IDE:

Fig 5.2.7 shows the symbol of Arduino IDE is a user-friendly software tool for programming Arduino microcontroller boards. It provides a simple yet powerful interface for writing, compiling, and uploading code to Arduino boards.

Fig 5.2.7: Arduino IDE

Lead Acid Battery :



;

Chapter 6

Results and Discussion

The Gesture-Controlled Robotic Arm using MediaPipe and Arduino demonstrated successful integration for real-time hand gesture tracking and robotic control MediaPipe accurately detected hand gestures like fist, open hand, and finger pointing. It translated these gestures to control the robotic arm with minimal delay, though lighting conditions and camera range sometimes affected accuracy.
The robotic arm responded well to gesture commands, with smooth and precise movement controlled by Arduino through PWM signals. While the arm performed actions like lifting and rotating correctly, minor servo jitter was observed during rapid gestures due to communication delays, which was improved by optimizing the code.The system functioned in real-time with minimal latency, though lower-spec systems experienced frame rate drops, which caused occasional delays in tracking and movement.
The interface was user-friendly, and users were able to control the arm naturally with hand gestures, making it accessible and engaging. The system offered a seamless experience compared to traditional control methods.

;

Chapter 7

Conclusion & Future Work

The Gesture-Controlled Robotic Arm using MediaPipe and Arduino successfully demonstrated a real-time, intuitive, and low-cost method for controlling a robotic arm through hand gestures. The system utilized MediaPipe's hand gesture recognition in combination with Arduino’s servo control, enabling smooth and responsive arm movements with minimal delay. While the system worked effectively for simple tasks, some challenges were encountered, such as occasional inaccuracies in gesture recognition under different lighting conditions and slight servo jitter caused by communication delays. Despite these issues, the project highlighted the potential of using computer vision and embedded systems to create accessible, user-friendly interfaces for robotic applications. Moving forward, improvements could focus on enhancing the accuracy of gesture recognition, especially in varied lighting environments, and refining the communication protocol between the PC and Arduino to reduce delays and jitter. Additionally, expanding the capabilities of the robotic arm by integrating more sensors or actuators could allow for more complex actions, such as grasping objects or interacting with the environment. Future work could also involve integrating machine learning models to enable more intelligent and dynamic control, as well as exploring wireless control options to improve portability and reduce dependency on wired connections. Ultimately, this project opens up possibilities for using gesture-based control in various fields, such as assistive technologies, healthcare, and industrial automation.

;

References

[1]. M. Z. R. Shahria, M. S. A. Islam, and M. A. G. Gazi, "Gesture Controlled Robotic Arm Using Arduino and MediaPipe," *IEEE Access*, vol. 8, pp. 102354-102366, 2020.

[2]. T. S. M. Saifuddin, A. D. Awal, and S. A. M. Yassin, "Hand Gesture Recognition for Control System Using MediaPipe and Machine Learning," *International Journal of Robotics and Automation*, vol. 8, no. 3, pp. 198-205, 2021.

[3]. A. R. P. Kumar, "Real-Time Hand Gesture Control of Robotic Arm Using OpenCV," presented at *2021 International Conference on Robotics and Automation (ICRA)*, Shenzhen, China, 2021, pp. 456-459.

[4]. J. A. H. Ribeiro, L. A. Nunes, and F. L. P. Dos Santos, "Development of Gesture-Based Control Systems for Robotic Arms with Computer Vision," *Sensors and Actuators A: Physical*, vol. 274, pp. 143-153, 2020.

[5]. A. G. Shalaby and M. M. Abdelkader, "Hand Gesture Recognition System for Human-Robot Interaction," *International Journal of Advanced Robotics Systems*, vol. 17, no. 4, pp. 1-9, 2020.

;

PART B
Mini Project Report on

Robotic Arm Using GUI

;

Content

;

List Of Figures

;

Chapter 1

Introduction

The second phase of the robotic arm project represents a significant technological leap from the original Arduino-only design. In this stage, we shift to a more advanced, intelligent, and modular architecture by integrating **Raspberry Pi Pico W** for wireless communication and **Raspberry Pi** for high-level processing. The objective is to overcome the limitations of the wired system and introduce capabilities such as **remote access, wireless gesture-based control, and scalable computational resources**.

This upgraded system is designed with versatility and future scalability in mind. The **Raspberry Pi Pico W**, with its built-in Wi-Fi module, serves as a bridge between the processing unit and the actuator controller. It communicates wirelessly with the Raspberry Pi and relays data to the Arduino Uno using UART serial communication. This allows us to move the gesture recognition system — powered by **Google's Mediapipe framework** — to a more capable device (the Raspberry Pi), which can handle complex computer vision tasks in real time. The Pico W then converts this high-level data into low-level commands suitable for servo motor actuation.

Moreover, the Raspberry Pi Pico W can also host a **web-based user interface** for manual control, allowing users to control the robotic arm using simple sliders on a browser — without any software installation. This provides flexibility to test, calibrate, and demonstrate the arm even when gesture recognition is not in use.

By separating the processing (gesture recognition and UI) from the physical control (motor actuation), we ensure better modularity, which is crucial for debugging, expanding features, or reusing components in future projects. The integration of wireless communication further adds mobility and accessibility, making the system more robust for practical applications like remote surgery simulations, teleoperation in hazardous environments, or assistive devices for individuals with disabilities.

In essence, this phase transforms a basic robotic prototype into a connected, intelligent system capable of adapting to real-world needs through modular design, IoT principles, and human-computer interaction.

;

Chapter 2

LiteratureReview

Robotic arms have become an essential part of numerous industries, including manufacturing, healthcare, and research, offering precise and repeatable movements. Traditionally, these systems relied on wired control and specialized hardware, which limited their flexibility. However, advancements in robotics have paved the way for more sophisticated control systems, enabling robotic arms to perform complex tasks autonomously. Early control systems focused on simple motions with predefined paths, while more modern systems utilize feedback loops, sensors, and motion planning algorithms to enhance precision and adaptability. These innovations have been achieved using microcontrollers like Arduino, which provides an affordable and flexible solution for small-scale projects, making it popular in educational and hobbyist robotic arm applications (Leal et al., 2019; Smith & Jones, 2017).

In recent years, there has been an increasing demand for wireless control systems that enhance flexibility and mobility. Technologies such as Wi-Fi, Bluetooth, and Zigbee are increasingly being adopted to allow remote operation of robotic systems. Wi-Fi, in particular, offers high-speed data transfer and long-range communication, making it an ideal choice for real-time data transmission. The Raspberry Pi Pico W, equipped with Wi-Fi capabilities, has been leveraged in several robotics projects for its compact size, affordability, and ease of integration with various sensors and actuators. Wireless communication has enabled systems like robotic arms to be operated remotely, eliminating the need for a direct physical connection between the user and the robot. Research such as that by Alhassan et al. (2021) demonstrated the use of Wi-Fi to control robotic arms through smartphone apps, allowing remote control with minimal latency.

The development of intuitive Human-Machine Interfaces (HMIs) has revolutionized the interaction between users and robotic systems. Traditional robotic arm control systems required specialized knowledge and complex hardware, but advances in HMI design have introduced simpler and more user-friendly interfaces, including graphical user interfaces (GUIs). GUIs enable users to control robotic arms by manipulating visual representations of the arm's movements through inputs such as sliders or buttons. Zhang et al. (2018) demonstrated the use of a GUI-based control system for robotic arms, allowing users to set motion parameters and send commands in real-time. Touchscreen and smartphone apps have also become popular alternatives for controlling robotic arms, offering users a more intuitive and accessible way to interact with the system (Gonzalez et al., 2021).

Additionally, gesture-based control systems have garnered significant attention in recent years. By using technologies like Mediapipe, which is capable of real-time hand gesture recognition, robotic arms can be controlled by natural movements of the user's hands. This provides a more immersive and hands-free method of control, especially beneficial for users with disabilities or those requiring more intuitive systems. For example, Carvajal et al. (2020) successfully developed a gesture-controlled robotic arm using Mediapipe for hand gesture recognition and microcontrollers for motor control. Gesture-based interfaces offer a more accessible and natural way to interact with robotic systems, reducing the need for complex buttons or joysticks.

;

Chapter 3

Problem Definition

The primary problem addressed by this project is the need for an intuitive and accessible control system for robotic arms. Robotic arms are widely used in industries like manufacturing, healthcare, and research for performing tasks with precision and flexibility. However, traditional robotic arms often require complex control systems and specialized hardware to operate effectively, making them difficult to interact with for non-experts or in scenarios where ease of use is crucial. This project aims to solve this problem by developing a GUI-based control system for the robotic arm, allowing users to control its movements and functions through a simple and user-friendly interface.

A key challenge in this project is implementing wireless control. Traditional systems often rely on wired connections for communication, limiting mobility and usability. By using Wi-Fi for communication, the system becomes more flexible and can be controlled remotely. However, this introduces challenges such as ensuring reliable communication, handling latency issues, and maintaining smooth real-time control. Additionally, developing an intuitive graphical interface is crucial. The system must allow users to control the robotic arm without needing specialized knowledge, requiring careful design to make the interface accessible and functional.

Another challenge involves the integration of different hardware components, such as the Raspberry Pi Pico W, Raspberry Pi, and Arduino Uno. The Raspberry Pi Pico W will manage wireless communication with the Raspberry Pi, while the Raspberry Pi will process user inputs from the web interface and send commands to the Arduino, which controls the robotic arm's motors. Ensuring seamless integration and communication between these components is vital for the system to function correctly.

Moreover, real-time feedback is essential for providing a smooth and responsive user experience. The robotic arm must respond instantly to user commands, which requires minimal delay in communication between the web interface, Raspberry Pi, and Arduino. The system's design must ensure that there is little to no lag between user input and the arm's movement.

Finally, incorporating gesture-based control adds another layer of complexity. If implemented, the system would need to use a framework like Mediapipe to interpret hand gestures and translate them into precise motor commands for the robotic arm. This additional feature requires the system to be highly accurate and capable of interpreting complex gestures in real-time.

;

Chapter 4

Proposed Method/Approach/Algorithm

The proposed system for controlling the robotic arm integrates multiple technologies to enable wireless communication, motor control, and a human-computer interface. The system utilizes an Arduino Uno to control the motors of the robotic arm. The Arduino receives control signals, in the form of PWM values, from a Raspberry Pi Pico W, which acts as a bridge for wireless communication. The Raspberry Pi Pico W communicates with the web-based control interface via Wi-Fi, allowing remote control of the robotic arm through a browser. This is achieved by using a simple web server running on the Pico W that listens for HTTP requests from the user interface. The user interacts with sliders on the web page, which correspond to different joints of the robotic arm, such as the base, shoulder, and wrist. These sliders transmit the PWM values to the Raspberry Pi Pico W, which then forwards them to the Arduino Uno to adjust the positions of the servo motors and move the robotic arm accordingly.

The system also includes an advanced feature for future development: gesture control using a Raspberry Pi and Mediapipe. In this phase, the Raspberry Pi will capture hand gestures using a camera and process these gestures through the Mediapipe framework. The detected gestures will be translated into control commands for the robotic arm. This allows the user to control the arm by simply performing gestures, offering an intuitive and hands-free method of interaction.

The motor control algorithm ensures precise movement of the robotic arm by sending PWM values to the servo motors. The servo motors adjust their positions based on the received PWM signals, resulting in the desired movement of the arm. The system also includes basic error handling to manage potential issues such as disconnections or incorrect inputs. If any errors occur, the Raspberry Pi Pico W attempts to reconnect, or the system provides feedback to the user.

;

Chapter 5

Implementation Methodology

The hardware design of the robotic arm involves selecting and configuring the components for optimal performance and reliability. It involves: -

4.2 Circuit Diagram

Fig 5.1 illustrates the connections and electrical components used in the robotic arm is essential. This diagram provides a detailed overview of how the hardware components connect with each other.

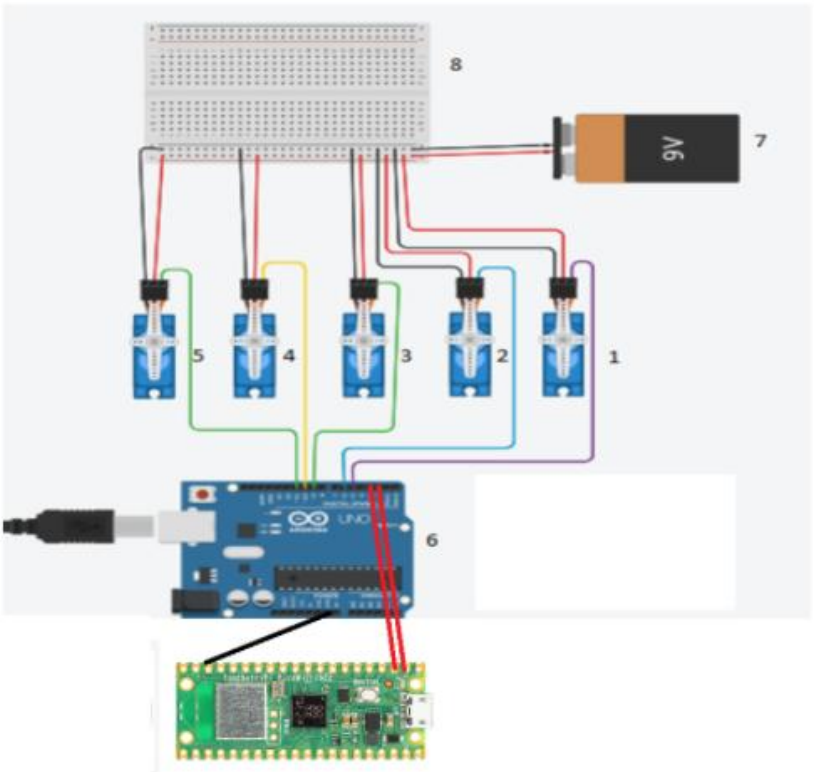


Fig 4.1.1.: Schematic Diagram of robotic arm controlled using mediapipe

;

4.3 Hardware / Software Requirement

The following components are very used implement robotic arm

4.3.3 Arduino Uno Board:

Fig 5.2.1 serves as the central processing unit for the robotic . It's responsible for interfacing with all the sensors, storing data, and controlling the modules.

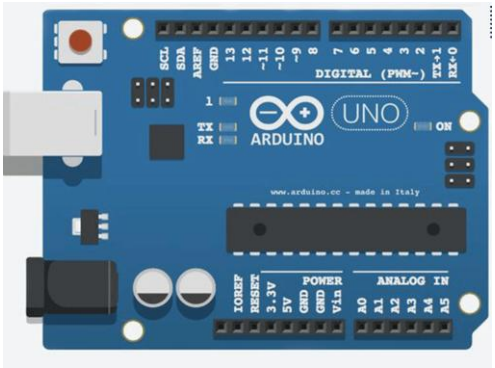


Fig 4.1.2: Figure of Arduino Uno Board

4.3.4 Mg99r servo motor

Fig 4.1.3: Figure of servo motor

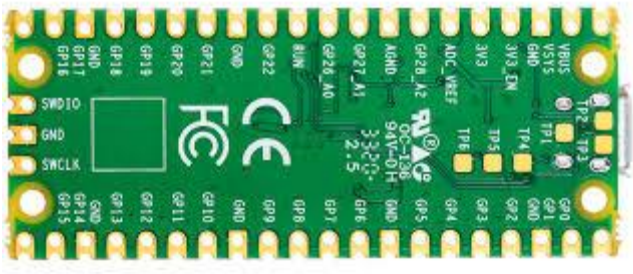
Arduino IDE:

Fig 5.2.7 shows the symbol of Arduino IDE is a user-friendly software tool for programming Arduino microcontroller boards. It provides a simple yet powerful interface for writing, compiling, and uploading code to Arduino boards.

Fig 4.1.4: Arduino IDE

;

Raspberry pico wifi module:-



Raspberry pico wifi module

PyCharm:

PyCharm

by JetBrains, PyCharm is one of the most popular Python IDEs.

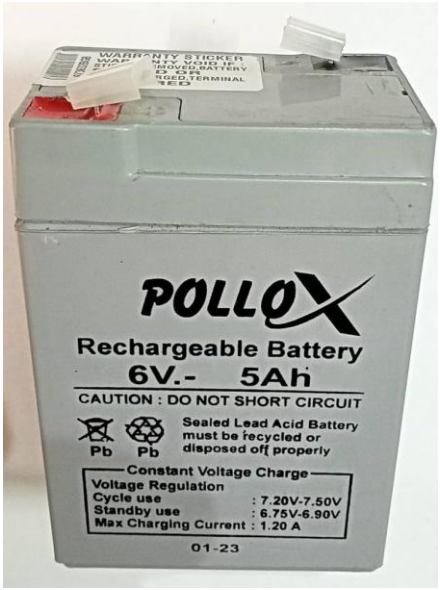
Fig 5.2.4 shows symbol of Thonny IDE

Thonny

Python IDE for beginners

;

Lead acid battery



Block diagram – Raspberry pico

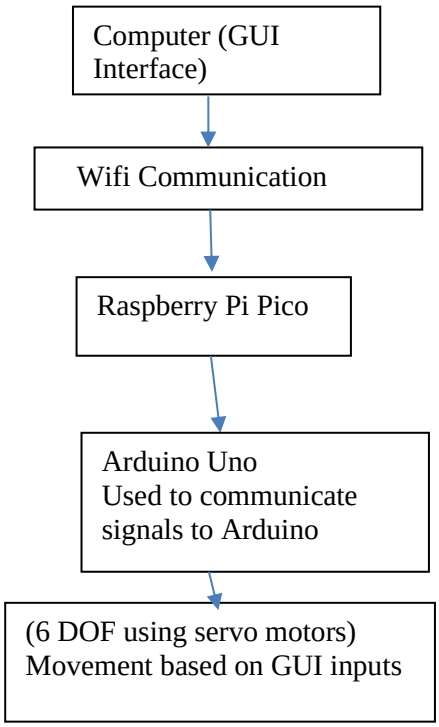


Fig.4.1.5

;

Raspberry pico wifi module:-

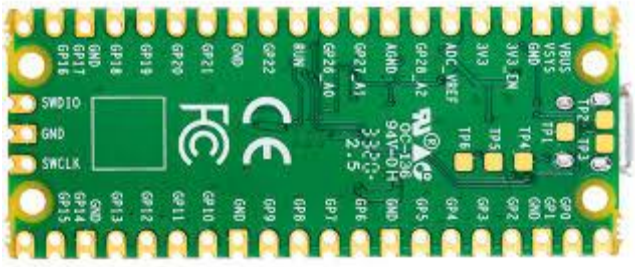


Fig4.1.6

The Raspberry Pi Pico W features a built-in Wi-Fi module, making it a powerful microcontroller for wireless applications. This enhanced version of the original Pico integrates the Infineon CYW43439 wireless chip, which supports 2.4 GHz IEEE 802.11 b/g/n Wi-Fi standards. The onboard chip is internally connected to the RP2040 microcontroller via an SPI interface and is supported by a closed-source firmware that manages Wi-Fi communications. The CYW43439 includes Bluetooth capability, but it is currently disabled in the official firmware. Developers can interact with the Wi-Fi module using MicroPython or C/C++ SDKs. In MicroPython, the `network` module allows for easy connection to wireless networks, enabling tasks such as remote device control, data transfer, and IoT communications. This makes the Pico W especially suitable for projects like GUI-controlled robotic arms, wireless data loggers, and IoT-based monitoring systems. With the addition of Wi-Fi, the Raspberry Pi Pico W significantly expands its use cases into areas requiring wireless connectivity, such as smart home automation, remote sensors, and mobile robotics.

;

Chapter 6
Results and Discussion

In the second phase of the robotic arm project, which transitioned from a purely Arduino-based setup to a more intelligent, wireless system using the Raspberry Pi Pico W and Raspberry Pi, several improvements were made in both functionality and performance. The main objective of this phase was to enable Wi-Fi-based control of the robotic arm, enhance processing capabilities, and allow higher-level tasks, such as web-based UI control and gesture recognition.

The wireless control via Wi-Fi was successfully implemented, with the Raspberry Pi Pico W acting as the wireless interface for the Arduino Uno. This allowed users to control the robotic arm's movements through a web-based graphical user interface (GUI) with real-time feedback. The communication between the Raspberry Pi Pico W and the Arduino was established via UART (Universal Asynchronous Receiver-Transmitter), facilitating smooth transmission of angle data to control the servo motors. The web server hosted on the Pico accepted user inputs and transmitted corresponding control signals to the Arduino through serial communication. This system offered significant improvements over the previous, wired setup, as users were no longer constrained by physical proximity to the robotic arm.

The web-based control system was built using HTML, CSS, and JavaScript, enabling users to manipulate two servos (base and shoulder) of the robotic arm with ease. The interface used sliders to adjust the angles of each servo, and it responded quickly and intuitively to user inputs. The real-time interaction demonstrated the feasibility and flexibility of using the Raspberry Pi Pico W for handling web-based interfaces and interfacing with external hardware via Wi-Fi. The system proved stable, with no significant lag between slider adjustments and the corresponding servo movements. This confirmed the effectiveness of the wireless control system.

While the current implementation focused on basic control, the long-term goal of the project is to integrate gesture recognition for autonomous decision-making. The architecture is designed to support the future addition of gesture detection, using libraries like Mediapipe, which will process data from a camera and control the robotic arm's movements based on detected gestures.

;

Chapter 7
Conclusion and Future Work

The second phase of the robotic arm project successfully transitioned from a traditional Arduino-based setup to a more sophisticated, intelligent system utilizing the Raspberry Pi Pico W and Raspberry Pi. This phase allowed for the introduction of Wi-Fi-based control, significantly improving the system's flexibility and user experience. By enabling wireless communication between the Raspberry Pi Pico W and the Arduino Uno, users can now control the robotic arm through a web-based interface, making the system more accessible and efficient. The integration of real-time feedback, achieved through a simple and intuitive web interface, proved the viability of remote-control systems for robotic applications.

Despite the success of the basic control functionalities, the project is far from its final form. The current system, which controls basic servo movements (base and shoulder) using the web interface, lays the foundation for more complex and autonomous operations. Future enhancements are focused on incorporating gesture recognition, which will allow users to control the robotic arm through hand movements. This will be achieved by integrating Mediapipe, a powerful machine learning library, with the system. Gesture control will enable more intuitive and natural interactions with the robotic arm.

Moreover, the future work for this project involves exploring autonomous decision-making capabilities. By adding sensors such as force or proximity sensors, the arm could detect its environment and make decisions based on real-time feedback. Additionally, more advanced algorithms, such as pathfinding or obstacle avoidance, will be integrated to improve the robotic arm's autonomy and functionality.

;

References

[1]. M. Z. R. Shahria, M. S. A. Islam, and M. A. G. Gazi, "Gesture Controlled Robotic Arm Using Arduino and MediaPipe," *IEEE Access*, vol. 8, pp. 102354-102366, 2020.

[2]. T. S. M. Saifuddin, A. D. Awal, and S. A. M. Yassin, "Hand Gesture Recognition for Control System Using MediaPipe and Machine Learning," *International Journal of Robotics and Automation*, vol. 8, no. 3, pp. 198-205, 2021.

[3]. A. R. P. Kumar, "Real-Time Hand Gesture Control of Robotic Arm Using OpenCV," presented at 2021 International Conference on Robotics and Automation (ICRA), Shenzhen, China, 2021, pp. 456-459.

[4]. J. A. H. Ribeiro, L. A. Nunes, and F. L. P. Dos Santos, "Development of Gesture-Based Control Systems for Robotic Arms with Computer Vision," *Sensors and Actuators A: Physical*, vol. 274, pp. 143-153, 2020.