Flex Sensor & Voice Control Robotic Hand

SONARGAON UNIVERSITY (SU)

Supervised By

Md. Rais Uddin Mollah

Assistant Professor & Head
Department of EEE
Sonargaon University (SU)

Submitted By

A. Rahman	**ID: 2103024049**
Surovi Akter	**ID: 2103024096**
Jannat Akter	**ID: 2103024009**
Md. Samim Hossain	**ID: 2103024010**
Md. Abdus Subahan	**ID: 2103024112**
Md. Khademul Islam	**ID: 2103024018**
Sanower Hossain	**ID: 2101022073**

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthapath,
Dhaka-1215, Bangladesh

Date of Submission: May- 2025

Declaration

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree.

…….	…….
A. Rahman	**Surovi Akter**
…….	…….
Jannat Akter	**Md. Samim Hossain**
…….	…….
Md. Abdus Subahan	**Md. Khademul Islam**
…….	
Sanower Hossain	

Under Supervision of

.....

Md. Rais Uddin Mollah
Assistant Professor & Head
Department of EEE
Sonargaon University (SU)

Certification

This is to certify that this project entitled **"Flex Sensor & Voice Control Robotic Hand"** is done by the following students under my direct supervision. This project work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervisor

.....

Md. Rais Uddin Mollah
Assistant Professor & Head
Department of Electrical and Electronic Engineer (EEE)
Sonargaon University (SU)

Acknowledgement

The report titled as on **"Flex Sensor & Voice Control Robotic Hand"** has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation.

At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable teacher **Md. Rais Uddin Mollah,** Assistant Professor & Head, Department of Electrical & Electronics Engineering, SU– Sonargaon University, Dhaka, for his valuable and patient advice, sympathetic assistance, co-operation, contribution of new idea. Deep theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

Finally, we would like to thanks again to the respected **Vice- Chancellor of SU, Professor Dr. Shamim Ara Hossan** also thanks to **Head of Department of SU , Electrical & Electronics Engineering, Md Rais Uddin Mollah** because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under Bsc in EEE program, and that will be very helpful for our prospective career.

We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report

Abstract

This project presents the design and development of a robotic hand controlled through two primary input methods: flex sensors and voice commands. The system aims to replicate human hand movements and perform basic grasping actions using sensor-based inputs, making it particularly useful in applications such as assistive technology, prosthetics, and remote manipulation. The flex sensor component of the system captures finger bending motions from a wearable glove, translating real-time human hand gestures into corresponding robotic finger movements. Meanwhile, the voice control system utilizes a speech recognition module to interpret predefined voice commands, enabling hands-free operation for specific tasks. An Arduino microcontroller processes inputs from both control methods, directing servo motors that actuate the robotic fingers. This dual-input approach enhances the flexibility and usability of the robotic hand, making it adaptable to various user needs and environments. The integration of sensor data with precise motor control demonstrates a cost-effective and efficient solution in the field of human-machine interaction and automation.

Table of Contents

Name of the Contents		Page No.
Declaration		ii
Certification		iii
Acknowledgement		iv
Abstract		v
Table Of Contents		vi
List Of Figures		viii
List Of Tables		viii
Chapter 1: Introduction		
1.1	General	01
1.2	Objective of the Project	01
1.3	Background of the Project	01
1.4	Earlier Research	02
1.5	Brief Overview of The Project	03
1.6	Motivation for Project	03
1.7	Project Management	03
Chapter 2: Theoretical Review		
2.1	Introduction	05
2.2	System Design	05
2.3	Literature Survey	06
2.4	Time Plan	07
Chapter 3: Methodology		
3.1	Process of Project	09
3.2	Block Diagram of This Project	11
3.3	Circuit Diagram of This Project	11
3.4	Working Principal	12
3.5	The list of Devices used in the Project is Given Below	13
3.6	Arduino UNO	13
3.6.1	Technical Specifications	14
3.7	Power Supply	14
3.7.1	Technical Specifications	15
3.8	Bluck Module	16
3.9	Servo Motor	16
3.9.1	Technical Specifications	17
3.10	Bluetooth Module	17
3.10.1	Specifications	17
3.11	Arduino NANO	18
3.11.1	Technical Specifications	19
3.12	Flex Sensor	19

3.12.1 Technical Specifications 19

Chapter 4: Result and Discussion

4.1 Result 21
4.2 Discussion 23

Chapter 5: Conclusion & Recommendation

5.1 Advantage 24
5.2 Application 24
5.3 Future Scope 24
5.4 Conclusion 25

References 26
Appendix 27

List of Figures

Figure No	Figure Contains	Page No
Figure 1.1	Project Management	04
Figure 2.1	Design of Mechanical Robot Hand	05
Figure 3.1	Block Diagram	11
Figure 3.2	Circuit Diagram	11
Figure 3.3	Arduino UNO	14
Figure 3.4	Switched Mode Power	15
Figure 3.5	LM2596 Voltage Converter	16
Figure 3.6	Servo Motor	17
Figure 3.7	Bluetooth Module	18
Figure 3.8	Arduino NANO	18
Figure 3.9	Flex Sensor	20
Figure 4.1	Project Picture- 01	21
Figure 4.2	Project Picture- 02	22

List of Tables

Table No	Table Contains	Page No
Table- 01	Time Scheduled Table for Project Introduction	07
Table- 02	Time Plan Table for Project Introduction	07
Table- 03	Time Scheduled Table for Project	08
Table- 04	Time Plan Table for Project	08

Chapter 1 Introduction

1.1 General

The Flex Sensor & Voice Control Robotic Hand is a multifunctional robotic system designed to mimic human hand movements and respond to voice commands. It combines gesture recognition and speech control to provide a versatile and interactive user experience.
The system consists of two main control methods:
Flex Sensors: These are attached to a wearable glove and detect the bending of fingers. When a user moves their fingers, the flex sensors generate variable resistance, which is processed by a microcontroller (e.g., Arduino) to replicate the same movement in the robotic hand using servo motors.
Voice Control: Using a voice recognition module (such as the Arduino-compatible Voice Recognition Module V3), users can issue spoken commands to control specific hand movements or actions without physical input.
This hybrid control mechanism allows the robotic hand to perform tasks such as gripping, pointing, or signaling in response to natural human gestures or voice instructions. The project is particularly relevant in fields like prosthetics, remote robotics, assistive devices for disabled individuals, and educational tools for demonstrating robotics and automation principles.
By integrating simple sensors and microcontrollers, this project demonstrates how low-cost hardware can be used to build an intelligent, human-interactive robotic system.

1.2 Objective of the Project

The primary objective of this project is to design and develop a robotic hand that can be controlled using both flex sensor inputs and voice commands, aiming to:

- Mimic human hand movements through real-time gesture tracking using flex sensors.
- Enable hands-free operation using voice recognition for specific predefined actions.
- Enhance user interaction with robotics by integrating natural input methods.
- Demonstrate a cost-effective solution for prosthetic hand systems and assistive robotics.
- Promote understanding of embedded systems and sensor-actuator integration in robotics.

1.3 Background of the Project

Robotic hands have become a significant area of interest in robotics and biomedical engineering, especially for applications such as prosthetics, teleoperation, and automation. Traditional robotic control systems often rely on complex programming or mechanical interfaces, which may limit their accessibility and adaptability in real-world environments.
In recent years, the integration of human-interaction technologies such as flex sensors and voice recognition has opened new avenues for intuitive control systems. Flex sensors are capable of detecting bending or finger movements, making them ideal for capturing hand

gestures. They are commonly used in glove-based systems to translate finger positions into electrical signals, which can then be used to control robotic fingers.

At the same time, voice recognition systems have become more reliable and affordable, allowing for natural language-based control of devices. This is especially beneficial for users with limited physical mobility or in situations where hands-free operation is necessary.

By combining these two technologies—gesture sensing and voice commands—this project aims to create a more adaptive and user-friendly robotic hand mechanism. The integration not only improves the functionality and responsiveness of the robotic system but also helps bridge the gap between human intent and robotic action.

This background lays the foundation for exploring cost-effective, real-time, and multi-modal control systems that can be used in various domains, from assistive technology to industrial automation.

1.4 Earlier Research

Several prior studies and projects have explored the use of flex sensors and voice control in robotic systems, especially in the context of human-machine interaction and assistive technologies. The following research efforts have laid the groundwork for the current project:

Gesture-Controlled Robotic Hands Using Flex Sensors: Previous work has demonstrated the use of flex sensor-equipped gloves to control robotic fingers in real-time. These systems typically map the resistance values of the sensors to servo motor angles, effectively replicating human finger movements. Such implementations are common in prosthetics and teleoperation systems and have proven to be reliable and intuitive.

Voice Recognition in Robotic Control: Research into speech-controlled robotics has shown that voice interfaces offer a practical solution for hands-free operation. Projects using voice modules like the Voice Recognition Module V3 or software-based systems (e.g., Google Speech API) allow users to command robotic systems using simple predefined keywords. These systems are especially valuable in accessibility applications.

Multimodal Control Systems: Some recent projects have begun integrating multiple control methods—such as gesture and voice—in order to increase the flexibility and responsiveness of robotic systems. These hybrid systems enable users to switch between control modes depending on context or ability, which improves usability.

Low-Cost Prosthetic Hand Prototypes: A range of academic and DIY efforts have aimed to create affordable prosthetic hands using 3D printing, flex sensors, and servo motors. While many focus on gesture control, combining it with voice input is still an emerging area with strong potential for real-world impact.

Embedded Systems and Arduino-Based Robotic Control: Arduino microcontrollers have become a popular platform for implementing sensor-driven robotic projects due to their ease of use, affordability, and large support community. Many research efforts have documented their successful use in real-time control of robotic hands using sensor and speech input.

1.5 Brief Overview of the Project

The Flex Sensor & Voice Control Robotic Hand is a multifunctional robotic system designed to replicate human hand actions using two control methods: gesture-based input via flex sensors and voice commands. The system uses a glove fitted with flex sensors to detect finger movements and translate them into robotic hand motions through servo motors. In parallel, a voice recognition module enables the user to control the robotic hand using predefined spoken commands, allowing for hands-free operation.

An Arduino microcontroller serves as the central processing unit, interpreting sensor data and voice input to coordinate the movement of the robotic fingers. This dual-input approach makes the system adaptable and accessible, with potential applications in assistive technology, prototyping prosthetics, human-robot interaction, and educational robotics.

The project emphasizes low-cost implementation, ease of use, and real-time responsiveness, offering a practical solution for controlling robotic mechanisms through natural human inputs.

1.6 Motivation for Project

The primary motivation behind this project is to bridge the gap between human input and robotic response through natural and intuitive control methods. Traditional robotic control systems often rely on complex interfaces or programming, which can be limiting for everyday users, especially those with physical impairments.

By leveraging flex sensors to capture hand gestures and voice recognition for hands-free commands, this project seeks to develop an accessible, user-friendly robotic hand that mimics human actions. This dual-input system holds strong potential in various fields, including:

- Assistive technologies for individuals with disabilities.
- Prosthetic development for affordable, user-controlled artificial limbs.
- Teleoperation and hazardous environment applications, where direct human involvement is risky.
- Educational platforms for learning robotics, embedded systems, and human-machine interaction.

1.7 Project Management

Effective project management is essential for the successful execution of the Flex Sensor & Voice Control Robotic Hand Project. The development process is divided into several key phases, each with specific goals, resources, and timelines to ensure smooth progress from planning to deployment.

1. Planning and Requirement Analysis
   - Define project objectives, scope, and functionalities.
   - Identify components: flex sensors, voice module, Arduino, servo motors, robotic hand structure.
   - Conduct initial research and feasibility analysis.

2. Design and Development
   - Develop circuit schematics and wiring diagrams.
   - Assemble components such as flex sensors, microphone/voice module, and servo motors.
   - Write code for Arduino to process flex sensor and voice inputs.
   - Implement logic to control servo motors based on input signals.
   - Calibrate sensor thresholds and train the voice recognition module.

3. Testing and Debugging
   - Test individual modules (flex sensor, voice module, servo response).
   - Debug communication issues and sensor inaccuracies.
   - Optimize performance for real-time responsiveness.

4. Integration
   - Combine hardware and software components.
   - Test full system operation for gesture and voice control.
   - Perform user testing to verify accuracy and usability.

5. Documentation and Reporting
   - Prepare user manuals, schematics, and code documentation.
   - Write final project report with system architecture, challenges, and results.

6. Final Presentation and Demonstration
   - Create a project presentation (slides, posters, or video demo).
   - Demonstrate system functionality to evaluators or audience.

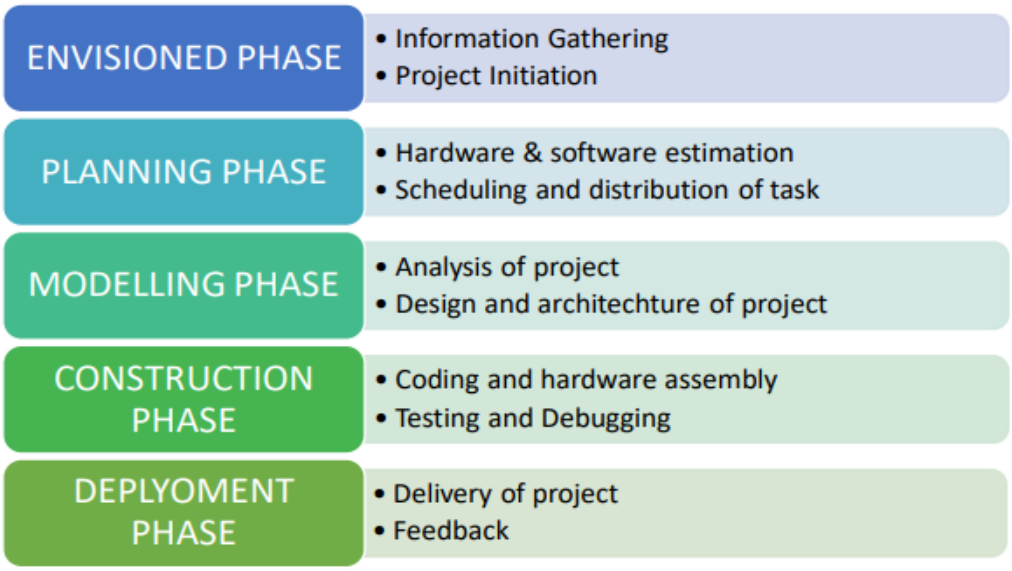


Fig-1.1: Project Management

Chapter 2 Theoretical Review

2.1 Introduction

In the rapidly evolving field of robotics and automation, the ability to interact with machines using natural human inputs has become a key area of research and development. The Flex Sensor & Voice Control Robotic Hand project is an innovative approach aimed at creating a robotic hand that can be controlled using both hand gestures and voice commands. This dual-mode control system enhances flexibility, usability, and accessibility in human-machine interaction.

Flex sensors, when worn on a glove, can detect finger bending and motion by changing resistance. These changes are interpreted by a microcontroller to replicate human hand movements through servo motors in a robotic hand. Meanwhile, a voice recognition module enables the system to understand and execute predefined spoken commands, allowing for hands-free operation.

This project leverages Arduino technology, servo actuators, and sensor integration to build a cost-effective prototype that demonstrates real-time responsiveness and ease of use. The primary application areas include assistive technologies, prosthetic hand development, remote robotic operations, and educational tools for embedded systems.

The combination of gesture recognition and voice control not only makes the system more interactive but also supports users with different needs and environments, highlighting the project's relevance in both social and technical domains.

2.2 System Design

The system design of the Flex Sensor & Voice Control Robotic Hand involves the integration of hardware and software components to enable gesture-based and voice-based control of a robotic hand. The design is modular, allowing flexibility in both operation and expansion.

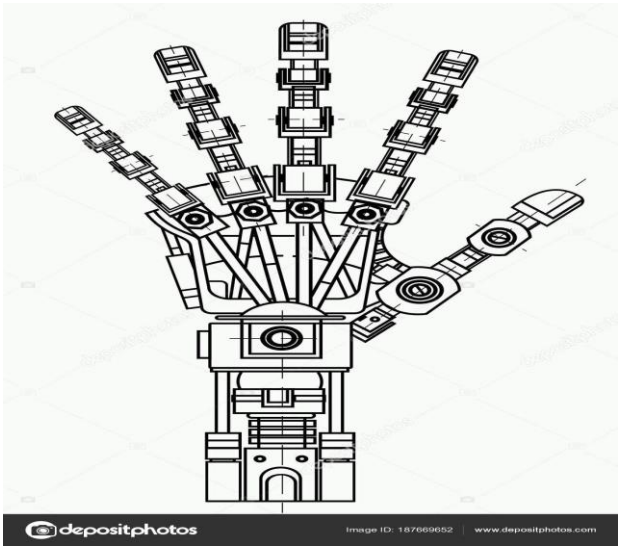


Fig-2.1: Design of Mechanical Robot Hand

1. System Architecture

2. Block Diagram

3. Component Details

4. Software Flow (Simplified)

5. Power Supply

6. Safety and Constraints

2.3 Literature Survey

The Voice Control Robotic Hand Mechanism project builds upon a foundation of existing research in robotics, speech recognition, and human-machine interaction. A literature survey provides insight into the state-of-the-art technologies, methodologies, and challenges addressed by prior studies. Below is an overview of significant research and developments relevant to this project:

1. Speech Recognition Technology

Key Studies:

Rabiner, L., & Juang, B. H. (1993)

"Fundamentals of Speech Recognition" Focused on Hidden Markov Models (HMMs) as a basis for speech recognition systems. Discussed early methodologies for converting audio input into text commands. Graves, A., et al. (2013) "Speech Recognition with Deep Recurrent Neural Networks" Proposed Recurrent Neural Networks (RNNs) for continuous speech recognition, improving accuracy for real-time applications. Google Speech-to-Text API (2018) Introduced robust cloud-based speech recognition systems using deep learning. Showcased capabilities to handle diverse accents and noisy environments.

Insights: Advancements in speech recognition, particularly deep learning techniques like RNNs, Long Short-Term Memory (LSTM), and Transformer models, enable real-time and highly accurate voice command processing for robotics applications.

2. Robotic Hand Design and Control, Key Studies: Tao, Y., et al. (2002) "Development of a Five-Fingered Robotic Hand" Presented a robotic hand with multiple degrees of freedom (DOF) for precise motion. Emphasized the importance of integrating tactile and position sensors for feedback. Ma, Z., et al. (2016) "Soft Robotic Hands with Integrated Actuation and Sensing" Introduced soft robotics for human-like flexibility and adaptability. Conti, F., & Van der Smagt, P. (2016) "Hand Exoskeleton Devices for Rehabilitation and Assistance" Focused on the use of robotic hands for assistive and rehabilitative applications.

Insights: Designing robotic hands requires a balance between mechanical complexity and functionality. Incorporating soft materials and sensors enhances dexterity and adaptability, making these systems suitable for various environments.

3. Voice-Controlled Robotic Systems, Key Studies: Pandey, A., & Singhal, A. (2015) "Voice Recognition and Command Execution in Robotic Systems". Explored the integration of voice commands into robotic systems using Arduino and Google APIs. Highlighted challenges like noise interference and limited vocabulary recognition. Yadav, A., et al. (2018) "Design of Voice-Controlled Robotic Arm for Disabled People" Developed a robotic arm controlled by simple voice commands. Focused on improving accessibility and reducing dependency on physical interfaces. Amarasinghe, Y. W., et al. (2020) "IoT-Based Voice Control for Robotic

Applications" Combined IoT with voice control, enabling remote operation via internet-connected devices. Insights: Voice-controlled robotics has shown significant potential in improving accessibility, particularly for individuals with disabilities. The integration of IoT and advanced APIs enhances remote functionality and system flexibility.
Key Challenges Identified in Literature: Voice Recognition Accuracy: Variations in accent, pronunciation, and noisy environments reduce reliability. Requires advanced filtering and adaptable machine learning models.
Latency in Real-Time Systems: Many systems exhibit delays due to processing limitations, especially in embedded systems.
Mechanical Complexity: High degrees of freedom in robotic hands often result in increased cost, weight, and complexity.
Limited Scalability: Most systems are designed for specific applications, limiting adaptability to broader use cases.
Proposed Solutions in Literature: Implementing deep learning models for better speech recognition. Using lightweight materials and modular designs for robotic hands. Optimizing microcontroller programming to reduce processing delays.

2.4 Time Plan

The following tables define the main tasks in the project introduction and project itself.

T1	Project Definition	1 Week
T2	Collecting data	11 Weeks
T3	Analysis	7 Weeks
T4	Theoretical calculation	4 Weeks
T5	Documentation	10 Weeks
T6	Prepare for presentation	2 Weeks

Table- 01: Time Scheduled Table for Project Introduction

The time of the project introduction is scheduled over 16 weeks, table 2 shows how the work was scheduled over this time:

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1	■															
T2		■	■	■	■	■	■	■	■	■	■	■				
T3				■	■	■	■	■	■	■						
T4						■	■	■	■							
T5					■	■	■	■	■	■	■	■	■	■		
T6															■	■

Table-02: Time Plan Table for Project Introduction

The following table defines the main tasks in the project:

T1	Collecting data	3 Week
T2	Implementation	10 Weeks
T3	Analysis	5 Weeks
T4	Building and testing the system	8 Weeks
T5	Documentation	10 Weeks
T6	Prepare for presentation	2 Weeks

Table-03: Time Scheduled Table for Project

The time of the project is scheduled over 16 weeks, table 4 shows how the work was scheduled over this time:

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1	■	■	■													
T2			■	■	■	■	■	■	■	■	■	■				
T3				■	■	■	■	■								
T4							■	■	■	■	■	■	■	■		
T5					■	■	■	■	■	■	■	■	■	■		
T6															■	■

Table-04: Time Plan Table for Project

Chapter 3 Methodology

3.1 Process of Project

The process of designing and implementing the Flex Sensor & Voice Control Robotic Hand involves several stages, from conceptualization to testing. Below is a detailed step-by-step breakdown of the key stages in the development process:

1. Requirement Analysis and Research
   - Objective Definition: Identify the primary goals of the project (gesture and voice-controlled robotic hand).
   - Component Selection: Choose appropriate sensors (flex sensors, voice recognition module), actuators (servo motors), and microcontroller (Arduino).
   - Feasibility Study: Research existing solutions, assess system requirements (e.g., sensor sensitivity, voice recognition capacity), and ensure compatibility of components.

2. System Design
   - Design the circuit for interfacing flex sensors, voice recognition module, and servos with the Arduino.
   - Create the mechanical design for the robotic hand, considering the number of joints, servos, and flexibility needed to mimic human hand movements.
   - Develop the Arduino code to read data from flex sensors and control servos.
   - Program the voice recognition module to understand commands and interface with the Arduino.
   - Implement algorithms to map sensor inputs to robotic finger movements and voice commands to predefined actions.

3. Component Assembly
   - Sensor Setup: Attach flex sensors to a glove or wearable device for detecting finger bending. Connect them to the Arduino through a voltage divider circuit to convert resistance changes into readable analog signals.
   - Voice Recognition Module Setup: Connect the voice module to the Arduino, ensuring proper configuration for receiving and processing spoken commands.
   - Servo Motor Installation: Attach servo motors to each joint of the robotic hand, ensuring each motor corresponds to a specific finger movement (e.g., one motor per finger joint).

4. Software Implementation
   - Sensor Calibration: Calibrate the flex sensors to accurately detect the bending angles of the fingers. Adjust the Arduino code to interpret sensor values and map them to servo movements.

- Voice Command Programming: Train the voice recognition module with specific commands (e.g., "open", "close", "grip", "release") and link these commands to specific actions in the system (e.g., closing the fingers for a grip).
- Control Logic: Program the logic for handling both gesture-based and voice-based inputs simultaneously. The system should be able to interpret and respond to both forms of input in real-time.

5. Testing and Debugging
   - Unit Testing: Test each module individually—flex sensors, voice module, and servo motors. Verify that each component functions as expected before full system integration.
   - System Integration: Integrate all components (flex sensors, voice recognition, servos, Arduino) and run comprehensive tests. Observe if the robotic hand mimics human gestures and responds correctly to voice commands.
   - Error Handling: Identify and fix any issues, such as unresponsive sensors, inaccurate voice recognition, or motor control errors. Fine-tune the system to ensure smooth and precise operation.

6. Optimization
   - Performance Tuning: Optimize the software for faster processing of sensor data and voice inputs. Reduce delays in servo motor movements for more fluid, real-time control.
   - Power Management: Ensure that the power supply is adequate for all components, especially the servo motors, and minimize energy consumption where possible.
   - User Testing: Involve real users to test both gesture and voice control in various scenarios. Collect feedback to improve user interaction and system responsiveness.

7. Final Demonstration and Documentation
   - System Demonstration: Conduct a final demonstration showcasing the robotic hand’ s ability to perform tasks like gripping, pointing, or releasing based on both gestures and voice commands.

8. Project Review and Conclusion
   - Review: Assess the overall performance of the system. Identify any potential areas for improvement, such as sensor sensitivity, motor speed, or voice recognition accuracy.
   - Future Enhancements: Suggest possible improvements, such as expanding the range of voice commands, adding additional sensors, or refining the robotic hand’ s dexterity.

3.2 Block Diagram of This Project

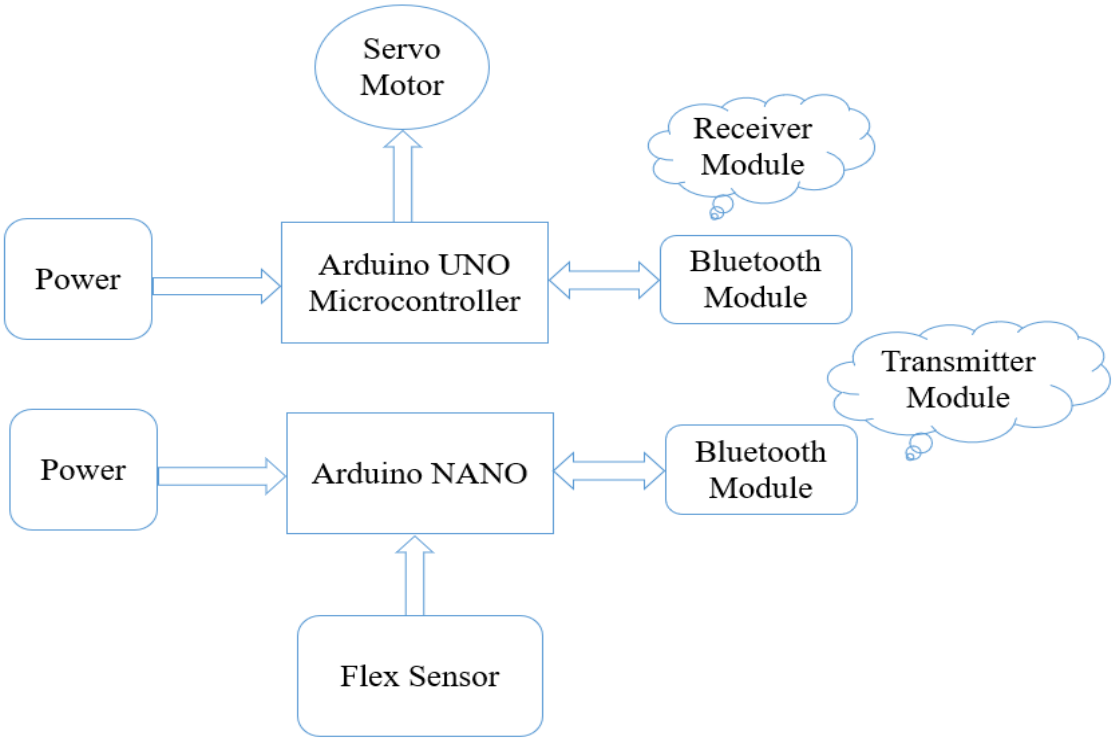


Fig-3.1: Block Diagram

3.3 Circuit Diagram of This Project

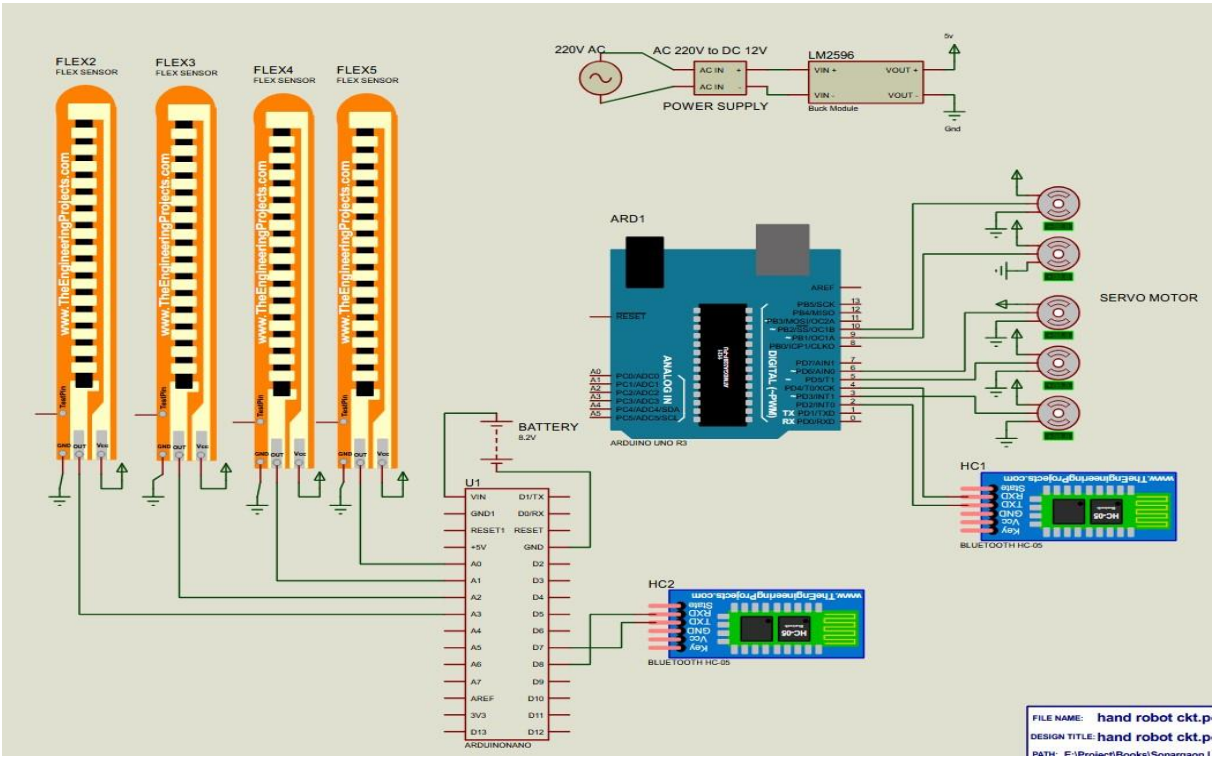


Fig-3.2: Circuit Diagram

3.4 Working Principal

The Flex Sensor & Voice Control Robotic Hand operates by integrating two distinct input control systems: gesture-based control using flex sensors and voice-based control using a voice recognition module. Below is a breakdown of how each component contributes to the working principle:

1. Flex Sensor-Based Control (Gesture Recognition)

The flex sensors, which are attached to a wearable glove, play a key role in detecting the movement of the user's fingers. Here's how the process works:

Flex Sensor Mechanism:

A flex sensor is a variable resistor that changes its resistance as it bends. When the user moves their fingers, the flex sensors bend accordingly, causing a change in resistance.

Signal Conversion:

The Arduino microcontroller reads the analog voltage from the flex sensor. The sensor output is fed into an analog-to-digital converter (ADC) on the Arduino, which converts the sensor's voltage into a digital signal that can be processed.

Finger Movement Mapping:

The Arduino code maps the changes in resistance (and consequently, the bending angle of the finger) to a servo motor movement. The higher the resistance, the greater the bending angle of the finger. The Arduino adjusts the position of the servo motors controlling the robotic hand’ s fingers accordingly.

Servo Actuation:

Each servo motor corresponds to a specific joint in the robotic hand (e.g., finger base, middle joint, and tip). As the user bends their finger, the corresponding servo motor moves the robotic hand's finger to mirror the gesture.

Real-Time Adjustment:

The system continuously updates the finger positions in real-time as the user moves their fingers. This allows the robotic hand to perform complex gestures such as gripping or pointing, mirroring the user's hand movements.

2. Voice-Controlled Operation

In addition to gesture control, the robotic hand can also be operated using voice commands. Here's how the voice control works:

Voice Command Recognition:

The voice recognition module (such as the V3 Voice Recognition Module) is trained with a set of predefined voice commands (e.g., "open", "close", "grip", "release"). When the user speaks a command, the microphone captures the sound, which is processed by the voice recognition module.

Command Processing:

The voice recognition module sends the recognized command to the Arduino via serial communication. The Arduino then interprets the voice command and triggers the appropriate action.

Voice Command Mapping:

Voice commands such as "open" or "close" are mapped to specific servo actions. For instance, the "close" command would initiate the servos to curl the fingers, while the "open" command would straighten the fingers.

Execution:

Once the voice command is processed, the Arduino activates the corresponding servo motors to make the robotic hand perform the desired movement, such as closing into a fist or opening to release an object.

3. Integration of Both Control Methods

The system operates in parallel, allowing the user to control the robotic hand using either gestures or voice commands. The Arduino continuously monitors both the flex sensor inputs (gesture control) and the voice command inputs:
Gesture and Voice Control Interaction:
The user can switch between controlling the hand with gestures or voice at any time. For instance, the user might issue a voice command to "close" the hand, and then adjust the grip using flex sensors for more fine-tuned control.
Processing Logic:
The Arduino processes both the flex sensor input and the voice command input independently. This allows the robotic hand to respond seamlessly to both types of input simultaneously, enabling a highly flexible and adaptable control system.
4. Power Supply and System Operation
Powering the System:
The system is powered by a 5V regulated power supply for the Arduino and sensors, while the servo motors are powered separately (often 6-9V) to ensure they receive enough current for precise and powerful movement.
Real-Time Execution:
As the sensors detect input (either from gestures or voice), the Arduino continuously updates the servo positions in real-time, allowing the robotic hand to function as an effective prosthetic or assistive device.
The Flex Sensor & Voice Control Robotic Hand Mechanism integrates two control methods—gesture-based input using flex sensors and voice-based input via a voice recognition module—to control the robotic hand. The flex sensors track the bending of the user's fingers, translating those movements to corresponding servo motions in the robotic hand. Meanwhile, the voice recognition module processes spoken commands, which trigger specific actions, such as gripping or releasing. The Arduino microcontroller coordinates both systems, ensuring the robotic hand responds intuitively to both gestures and voice commands in real time.

3.5 The list of Devices used in the Project is Given Below

- Arduino UNO.
- Power Supply.
- Buck Module.
- Servo Motor.
- Bluetooth Module.
- Arduino NANO.
- Flex Sensor.

3.6 Arduino UNO

The Arduino UNO is a microcontroller board based on the ATmega328p. It has 14 digital input/output pins (of which 5 can be used as PWM outputs), 6 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

3.6.1 Technical Specifications

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 5 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 56 KB of which 8 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 101.52 mm
- Width: 53.3 mm
- Weight: 37 g



Fig- 3.3: Arduino UNO

3.7 Power Supply

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power, see AC adapter) to DC loads, such as a personal computer, while converting voltage and current characteristics.

Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles). In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. The switched-mode power supply's higher electrical efficiency is an important advantage.

Switched-mode power supplies can also be substantially smaller and lighter than a linear supply because the transformer can be much smaller. This is because it operates at a high switching frequency which ranges from several hundred kHz to several MHz in contrast to the 50 or 60 Hz mains frequency. Despite the reduced transformer size, the power supply topology and the requirement for electromagnetic interference (EMI) suppression in commercial designs result in a usually much greater component count and corresponding circuit complexity.



Fig- 3.4: Switched Mode Power

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight is required. They are, however, more complicated; switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor.

3.7.1 Technical Specifications

- Input Voltage: AC 85-265V
- Output Voltage: DC 12V

- Current: 10A
- Size:128mm*98mm*40mm
- Shell Material: Metal case / Aluminum base
- Protection: Shortage Protection, Overload Protection, Over Voltage Protection
- Widely used in Industrial automation, LED display, communications, etc.
- Safety Compliance: CCC/ FCC / CE
- Working Temperature: 0~40℃
- Storage Temperature: -20~60℃
- Ambient Humidity: 0~95% Non- Condensation

3.8 Buck Module

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with



Fig- 3.5: LM2596 Voltage Converter

excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed frequency oscillator. The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators. Available in a standard 7-pin TO-220 package with several different lead bend options, and a 7-pin TO-263 surface mount package.

3.9 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor

for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system. Servo motors are used in applications such as robotics, CNC machinery or automated manufacturing.

Mechanism: A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.



Fig- 3.6: Servo Motor

3.9.1 Technical Specifications

- Power Supply: Through External Adapter.
- Weight: 55g
- Operating Speed: 0.17sec / 60 degrees (4.8V no load)
- Operating Speed: 0.13sec / 60 degrees (6.0V no load)
- Stall Torque: 13 kg-cm (180.5 oz-in) at 4.8V
- Stall Torque: 15 kg-cm (208.3 oz-in) at 6V
- Operation Voltage: 4.8 - 7.2Volts
- Connector Wire: Heavy Duty, 11.81" (300mm)

3.10 Bluetooth Module

A Bluetooth module is a compact device that facilitates wireless communication between two devices using Bluetooth technology. It is commonly used in projects like the Voice-Controlled Robotic Hand for transmitting commands wirelessly from a voice recognition unit (like a smartphone or microcontroller) to the robotic hand.

3.10.1 Specifications

- Bluetooth Protocal: Bluetooth Specification v2.0+EDR

- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75 Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm



Fig- 3.7: Bluetooth Module

3.11 Arduino NANO

The Arduino Nano R3 is a compact, powerful microcontroller board based on the ATmega328P chip, ideal for projects that require a small form factor. It' s commonly used in embedded systems and robotics due to its versatility, ease of use, and extensive community support. Below is an overview of the Arduino Nano R3 and its features, particularly how it fits into your Flex Sensor & Voice Control Robotic Hand Mechanism Project.

The Arduino Nano R3 is a microcontroller board that is designed for smaller projects where space is a constraint, yet you need the power and flexibility provided by the standard Arduino Uno. It has the same functionality but in a smaller package, and it is equipped with the ATmega328P microcontroller. The board features several digital and analog I/O pins, which can be used to interface with sensors, motors, and other components in the robotic hand project.



Fig- 3.8: Arduino NANO

3.11.1 Technical Specifications

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V (via DC jack or VIN pin)
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 can be used as PWM outputs)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA (maximum)
- Flash Memory: 32 KB (of which 2 KB used by bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- USB Interface: Mini USB
- Size: 18 x 45 mm (small, compact size for embedded applications)

3.12 Flex Sensor

A flex sensor is a type of sensor that changes its resistance when bent. It is typically used to measure the amount of bend or curvature in an object, making it ideal for detecting finger movements in your Flex Sensor & Voice Control Robotic Hand Mechanism Project. These sensors are essential in mimicking human hand movements by detecting the bending of the user' s fingers and converting it into a readable electrical signal that can control the movement of the robotic hand.

Structure: Flex sensors are usually long, thin strips made from a conductive material. When the sensor is bent, the resistance in the conductive material changes. The degree of bending directly correlates with the change in resistance.

Resistance Change: The sensor' s resistance increases as the sensor bends. The amount of resistance change depends on how much the sensor is bent. This resistance change is then measured using an analog-to-digital converter (ADC) in the Arduino or other microcontrollers.

3.9.1 Technical Specifications

- Operating voltage: 0-5V
- Power rating : 0.5Watt (continuous), 1 Watt (peak)
- Operating temperature: -45ºC to +80ºC
- Flat Resistance: 25K Ω
- Resistance Tolerance: ±30%

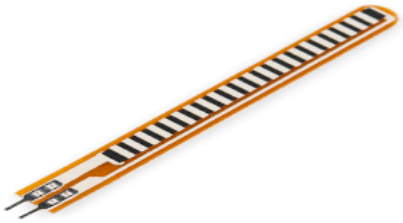


Fig- 3.9: Flex Sensor

Chapter 4 Result and Discussion

4.1 Result

The Flex Sensor & Voice Control Robotic Hand project combines gesture-based control using flex sensors and voice-based control via voice recognition modules to operate a robotic hand. The final results of this project would be evaluated based on the following aspects:

1. Accurate Hand Gesture Recognition
    - Flex Sensor Functionality: The flex sensors accurately detect the bending of each finger in real-time. As the user moves their hand, the robotic hand mirrors these movements with the corresponding servo motors.
    - Result: The robotic hand performs realistic finger movements, such as gripping, pointing, or releasing an object, based on the user’ s gestures.
    - Example: When the user bends their index finger, the servo motor controlling the robotic hand’ s index finger also bends accordingly, replicating the user's hand gesture.

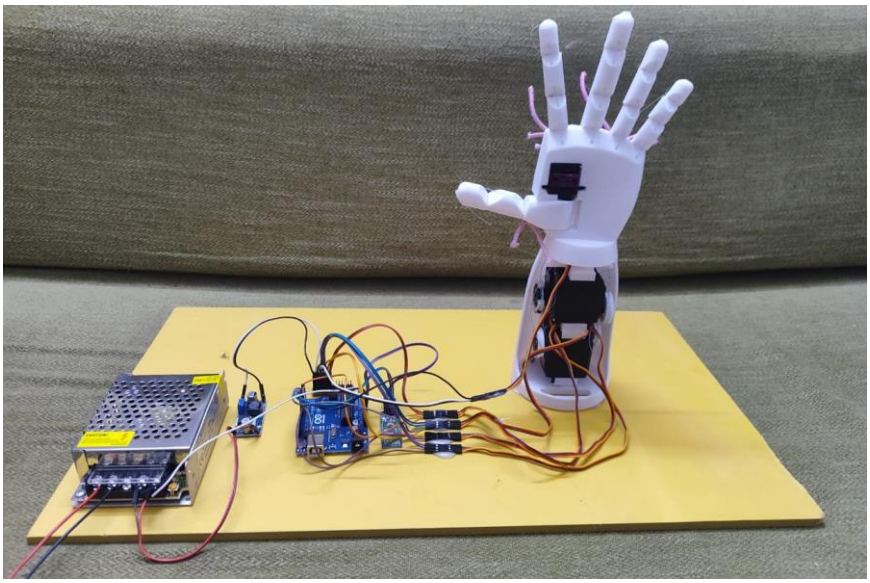


Fig- 4.1: Project Picture- 01

2. Seamless Voice Command Control
    - Voice Command Recognition: The voice recognition module accurately picks up the user's voice commands such as "open", "close", "grip", and "release". These commands are processed by the Arduino Nano R3, which then sends signals to the servo motors to execute the corresponding actions.
    - Result: The robotic hand responds promptly to voice commands. For example:
    - Saying "open" causes the robotic hand to spread its fingers apart.
    - Saying "close" causes the robotic hand to make a fist.
    - Additional commands, like "grip" or "release", can trigger more complex gestures.

3. Integration of Gesture and Voice Control

- Dual Control Mechanism: The project allows for gesture control (via flex sensors) and voice control (via voice recognition) to work independently or together. This hybrid control system provides flexibility for the user to choose the mode of interaction.
- Result: The user can either:
- Control the robotic hand with hand gestures (by bending the fingers).
- Issue voice commands for more direct control (e.g., saying "close" to close the hand).
- Example: The user can switch between voice control and gesture control seamlessly. If the user prefers finer control, they can rely on gesture input; if they need quick commands, they can use their voice.

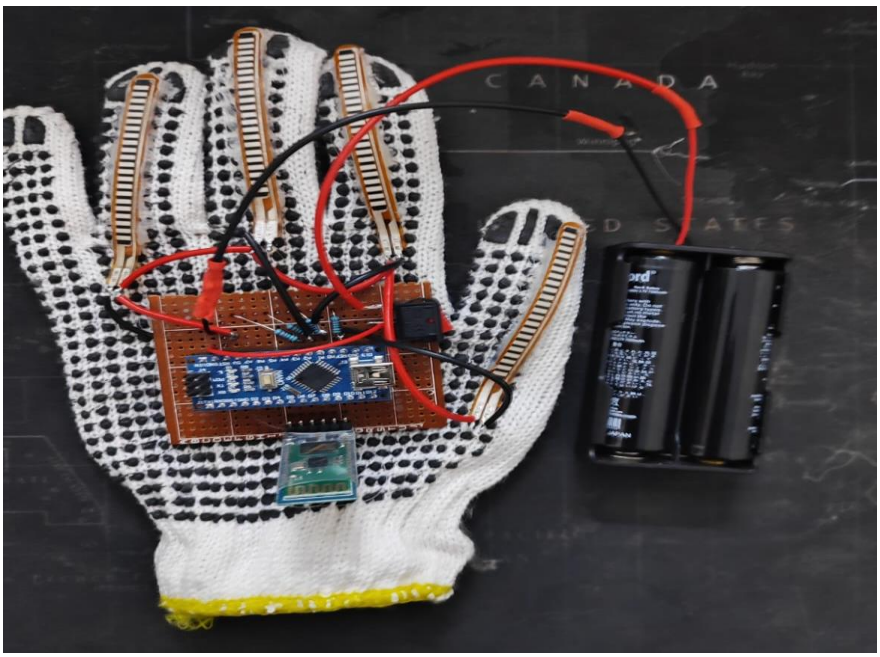


Fig- 4.2: Project Picture- 02

4. Real-Time Response

- Servo Actuation: The servo motors respond instantaneously to both the flex sensor data and voice commands. This real-time feedback enables smooth and natural movements of the robotic hand.
- Result: The robotic hand exhibits smooth movements that closely match the user's gestures or voice instructions. The speed and accuracy of the servo motors ensure that each movement is replicated in real-time.

5. User Experience and Interface

- Usability: The system is designed to be intuitive. The user can wear the flex sensor glove and issue voice commands without requiring complex setup or adjustments.
- Result: The project provides an effective human-machine interface. Users can control the robotic hand in a manner similar to their natural hand movements, making it an ideal prototype for prosthetic hands or assistive technologies.

6. Potential for Real-World Applications

- Prosthetics: The project demonstrates the potential to develop a prosthetic hand that can be controlled by both gestures and voice commands, providing users with more control over their prosthetic.
- Result: The robotic hand, although a prototype, mimics human hand motions quite effectively. This proves its potential application in creating more responsive, user-friendly prosthetic devices.
- Assistive Technology: For individuals with disabilities, combining gesture and voice control makes the robotic hand accessible in various scenarios, such as for individuals with limited mobility or motor function.
- Result: This dual-control mechanism could significantly improve the quality of life for people who need robotic assistance for daily tasks.

7. Technical Performance
   - Flex Sensor Calibration and Accuracy: The flex sensors were calibrated to accurately map the bending of the user’ s fingers to corresponding movements of the robotic hand’ s servos.
   - Result: The flex sensors provide precise data for finger movements, and the robotic hand mimics these gestures with a high degree of accuracy, ensuring reliable and consistent operation.
   - Voice Recognition Accuracy: The voice recognition system processes commands with a high degree of accuracy. The system is able to distinguish and react to specific commands such as "close", "open", and other gestures.
   - Result: The voice recognition module functions well, with minimal delay in processing commands. The system is robust enough to handle different voice commands in various conditions.

4.2 Discussion

The Flex Sensor & Voice Control Robotic Hand project is an innovative approach to creating a versatile robotic hand that can be controlled using both gesture recognition (via flex sensors) and voice commands. This dual-control system offers significant improvements in terms of accessibility, user experience, and the potential for real-world applications, particularly in prosthetics and assistive technologies.

The Flex Sensor & Voice Control Robotic Hand Mechanism project has successfully demonstrated the potential of combining gesture-based control and voice recognition for operating a robotic hand. While the system works effectively, there are challenges related to sensor calibration, voice recognition accuracy, and servo precision that need further refinement. By addressing these issues, the project has the potential to make significant strides in areas such as prosthetics and assistive robotics, offering a practical, accessible solution for users with mobility impairments.

Chapter 5: Conclusion & Recommendation

5.1 Advantage

- Dual Control Modes (Gesture + Voice)
- Enhanced Accessibility
- Real-Time Response
- Cost-Effective and Scalable
- Customizable and Modular Design
- Natural Human-Robot Interaction
- Promotes Hands-Free Operation
- Educational Value
- Potential for Future Enhancement
- Improves Quality of Life

5.2 Application

- Prosthetic Hand Development
- Assistive Technology for Physically Challenged Individuals
- Human-Computer and Human-Robot Interaction
- Remote Robotic Operations
- Rehabilitation and Therapy
- Educational and Research Tool
- Industrial Automation and Collaborative Robots (Cobots)
- Smart Wearable Technology
- Remote Education and Demonstration
- Emergency and Rescue Operations
-

5.3 Future Scope

- Integration with Artificial Intelligence (AI)
- Wireless and IoT Connectivity
- Application in Advanced Prosthetics
- Cloud-Based Command and Monitoring
- Multi-Language and Custom Voice Recognition
- Enhanced Sensor Technology
- Robotic Hand with Tactile Feedback
- Integration with Augmented Reality (AR) and Virtual Reality (VR)
- Industrial and Collaborative Robotics
- Miniaturization and Wearability

5.4 Conclusion

The Flex Sensor & Voice Control Robotic Hand project successfully demonstrates the integration of gesture and voice-based control systems to operate a robotic hand in real-time. By combining flex sensors for hand movement detection and voice recognition modules for auditory commands, the project bridges the gap between human intention and robotic execution in an intuitive and user-friendly way.

This dual-control approach enhances accessibility, particularly for individuals with physical disabilities, offering them a reliable and cost-effective means to interact with robotic systems or prosthetic devices. The system's ability to mimic natural hand gestures and respond accurately to voice commands highlights its potential for real-world applications in prosthetics, rehabilitation, remote operation, and human-robot interaction.

Through this project, students and developers gain hands-on experience in embedded systems, robotics, and human-machine interfaces, making it a valuable educational and research tool. With future advancements such as wireless control, AI integration, and tactile feedback, this system can evolve into a more intelligent, adaptive, and interactive solution.

In conclusion, the project not only achieves its objective of building a responsive and accessible robotic hand mechanism but also lays the groundwork for future innovations in assistive technology and intelligent robotics.

References:

[1] Ahmet, V. & Hilmi, K. (2016). Design of Voice Controlled Vehicle" , International Conference on Advances Automotive Technologies 2016, Yildiz Technical University, Instanbul, Turkey, AAT2016.

[2] Alka, N. U. (2017). A Voice Controlled Pick and Place Robotic Arm Vehicle using Android Application. Unpublished, M.Eng. Mini Project,
Abubakar Tafawa Balewa University, BauchiNigeria.

[3] Bharat, S. A., Sabya, S. P., Ashish, S. & Patil, M. V. (2015). Robotic Arm Wirelessly Controlled by Android Application" . International Journal of Engineering and Technical Research (IJETR),Vol.3.ISSN: 2321-
0869.

[4] Butkar, V. D., Devikar, S. R., Jaybhaye, V. B. & Shilpa, P. (2014). Android Based Pick and Place and Place Robot" , International Journal of Informative and Futuristic Research, IJIFR/V2/E4/011, Vol.2, pp. 859-867, ISSN: 2347-1697.

[5] Kannan, K. & Selvakumar, J. (2015). Arduino Based Voice Controlled Robot" , International Research Journal of Engineering and Technology (IRJET), Vol. 02, p-ISSN: 2395- 0072, e-ISSN:2395-0056.

[6] Muhammed, J. N., Neetha, J., Muhammed, F., Midhun, M., Mithun, S. & Safwan, C. N. (2015). Wireless Control of Pick and Place Robotic Arm Using an Android Application" , International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. (An ISO 3297: 2007 Certified Organization).

[7] Omijeh, B. O., Uhunmwangho, R. & Ehikhamenle, M. (2014). Design Analysis of a Remote Controlled Pick and Place Robotic Vehicle" , International Journal of Engineering Research and Development, Vol.10, pp57-68, EISSN: 2278-067X, P-ISSN: 2278- 800X.

[8] Ritika, P. & Narender, K. (2014). Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller" , International Journal of Scientific Engineering and Research (IJSER), Vol.2. ISSN: 2347-3878.

[9] Shamsheer, V. (2013). Hand Gestures Remote Controlled Robotic Arm. Advance in Electronic and ElectricEngineering. Vol. 3(5), pp 601-606, ISSN 2231-1297.

[10] Subhash, P. R. (2014). Voice Controlled RoboticVehicle" , International Journal of New Trends in Electronics and Communication (IJNTEC), Vol. 2, ISSN: 2347 – 7334.

Appendix:

Arduino UNO Program:

```
#include <SoftwareSerial.h>
// #include <LiquidCrystal_I2C.h>
#include <Servo.h>

SoftwareSerial mySerial(7, 8);  // RX, TX

char command;
char c;
String voice;

char btdata;

Servo Thumb;
Servo Thumb1;
Servo Index;
Servo Middel;
Servo Ring;
// Servo Thumb;

int thembservo;
int themblrservo;
int indexservo;
int middelservo;
int ringservo;
int pinkyservo;

int mode = 2;
int mode_data;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  mySerial.begin(9600);
  // lcd.init();  // initialize the lcd
  // // Print a message to the LCD.
  // lcd.backlight();

  pinMode(mode, INPUT);
```

```
  Thumb.attach(11);
  Thumb1.attach(3);
  Index.attach(5);
  Middel.attach(6);
  Ring.attach(10);

  Thumb.write(90);
  Thumb1.write(180);
  Index.write(0);
  Middel.write(0);
  Ring.write(20);
}

void loop() {  // run over and over

  mode_data = digitalRead(mode);
  // Serial.println(mode_data);

  if (mode_data == 0) {
    if (mySerial.available()) {
      btdata = mySerial.read();
      Serial.println(btdata);
    }

    if (btdata == '0') {
      Serial.println("NO Finger");
      // lcd.setCursor(0, 0);
      // lcd.print("NO Finger       ");
      Thumb.write(90);
      Thumb1.write(180);
      Index.write(0);
      Middel.write(0);
      Ring.write(20);
    }

    if (btdata == '1') {

      Serial.println("Thumb Finger");
      // lcd.setCursor(0, 0);
      // lcd.print("Thumb Finger    ");
      Thumb.write(0);
      Thumb1.write(0);
    }
    if (btdata == '2') {
      Serial.println("Index Finger");
```

```
    // lcd.setCursor(0, 0);
    // lcd.print("Index Finger     ");
    Index.write(180);
   }

   if (btdata == '3') {
    Serial.println("Middel Finger");
    // lcd.setCursor(0, 0);
    // lcd.print("Middel Finger    ");
    Middel.write(180);
   }

   if (btdata == '4') {
    Serial.println("Ring Finger");
    // lcd.setCursor(0, 0);
    // lcd.print("Ring Finger     ");
    Ring.write(180);
   }
  }

  if (mode_data == 1) {
   while (mySerial.available()) {
    delay(10);
    c = mySerial.read();
    // mybt = c;

    if (c == '#') {
     command = c;
     break;
    } else if (c != '*') {
     voice += char(c);
    }
    // Serial.println(command);
   }

   if (voice.length() > 0) {
    Serial.println(voice);
   }

   // thembservo

   if (voice.equalsIgnoreCase("Ham aap") || voice.equalsIgnoreCase("Thumb up") ||
voice.equalsIgnoreCase("thumbs up") || voice.equalsIgnoreCase("Khan sahab") ||
voice.equals("Ham aap") || voice.equals("thumbs up") || voice.equals("Khan sahab")) {
    // thembservo = 0;
```

```
    // themblrservo = 120;
    Thumb.write(90);
    Thumb1.write(180);
   }

   if (voice.equalsIgnoreCase("thumbs down") || voice.equalsIgnoreCase("songs down") ||
voice.equals("thumbs down") || voice.equals("songs down")) {
    // thembservo = 90;
    // themblrservo = 0;
    Thumb.write(0);
    Thumb1.write(0);
   }

   // indexservo

   if (voice.equalsIgnoreCase("index of") || voice.equalsIgnoreCase("index up") ||
voice.equals("index of") || voice.equals("index up")) {
    // indexservo = 180;
    Index.write(0);  //Range 180-0
   }

   if (voice.equalsIgnoreCase("index down") || voice.equalsIgnoreCase("index Don") ||
voice.equalsIgnoreCase("index done") || voice.equals("index down") || voice.equals("index
Don") || voice.equals("index done")) {
    // indexservo = 0;
    Index.write(180);  //Range 180-0
   }

   // middelservo

   if (voice.equalsIgnoreCase("middle of") || voice.equalsIgnoreCase("middle up") ||
voice.equals("middle of") || voice.equals("middle up")) {
    // middelservo = 0;
    Middel.write(0);  //Range 0-180
   }

   if (voice.equalsIgnoreCase("middle down") || voice.equalsIgnoreCase("Middleton") ||
voice.equalsIgnoreCase("middle done") || voice.equals("middle down") ||
voice.equals("Middleton") || voice.equals("middle done")) {
    // middelservo = 180;
    Middel.write(180);  //Range 0-180
   }

   // pinkyservo
```

```
   if (voice.equalsIgnoreCase("ring of") || voice.equalsIgnoreCase("ring up") ||
voice.equals("ring up") || voice.equals("ring of")) {
     // pinkyservo = 0;
     Ring.write(20);  //Range 0-150
   }

   if (voice.equalsIgnoreCase("ring down") || voice.equalsIgnoreCase("ringtone") ||
voice.equals("ring down") || voice.equals("ringtone")) {
     // pinkyservo = 180;
     Ring.write(180);  //Range 0-150
   }

   // allservo

   if (voice.equalsIgnoreCase("all up") || voice.equalsIgnoreCase("all of") || voice.equals("all
up") || voice.equals("all of")) {

     Thumb.write(90);
     Thumb1.write(180);
     Index.write(0);
     Middel.write(0);
     Ring.write(20);
   }

   if (voice.equalsIgnoreCase("all down") || voice.equalsIgnoreCase("all done") ||
voice.equalsIgnoreCase("fall down") || voice.equals("all down") || voice.equals("all done") ||
voice.equals("fall down")) {

     Thumb.write(0);
     Thumb1.write(0);
     Index.write(180);
     Middel.write(180);
     Ring.write(180);
   }

   command = 's';
   voice = "";
  }
}
```

Arduino NANO Program:

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(7, 8);  // RX, TX

// int pinky = A4;
int ring = A3;
int middel = A2;
int index = A1;
int thumb = A0;

int thumb_data;
int ring_data;
int middel_data;
int index_data;
// int pinky_data;

const char text[] = "0";

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);

  pinMode(thumb, INPUT);
  pinMode(index, INPUT);
  pinMode(middel, INPUT);
  pinMode(ring, INPUT);
  // pinMode(pinky, INPUT);
}
void loop() {

  thumb_data = analogRead(thumb);
  ring_data = analogRead(ring);
  middel_data = analogRead(middel);
  index_data = analogRead(index);
  // pinky_data = analogRead(pinky);

  Serial.println(thumb_data);
  Serial.println(index_data);
  Serial.println(middel_data);
  Serial.println(ring_data);
  // Serial.println(pinky_data);
  Serial.println(".....");

  if (thumb_data <  542) {
    mySerial.println('1');
    // mySerial.println("Hello, world?");
```

```
  Serial.println('1');
 }

  if (index_data < 560) {
  mySerial.println('2');
  Serial.println('2');
 }

 if (middel_data < 547) {
  mySerial.println('3');
  Serial.println('3');
 }

 if (ring_data <530) {
  mySerial.println('4');
  Serial.println('4');
 }




 // // if (pinky_data < 20) {
 // //   mySerial.println('5');
 // //   Serial.println('5');
 // // }

 if(thumb_data > 542 && ring_data > 530 && middel_data > 547 && index_data > 560)
 {
  mySerial.println('0');
  Serial.println('0');
 }

 // //Send message to receiver
 // // const char text[] = "Hello World";
 // // radio.write(&text, sizeof(text));

 delay(1000);
}
```