Project Summary:

For millions of people around the world who live with speech impairments, even the simplest act of communication can feel like an insurmountable challenge. A smile, a wave, a request—things many of us take for granted—can become daily struggles for someone who cannot speak. This project, “Gesture-Based Communication for Speech-Impaired Individuals,” was born from the desire to change that.

At its heart, this initiative envisions a wearable glove system that interprets hand gestures and transforms them into spoken words. Using flex strip and proximity sensors sewn into the glove, each movement of the hand is captured and translated into a digital signal using a USB-based ADC (Analog-to-Digital Converter). By eliminating the need for bulkier boards like Arduino and directly interfacing with a computer, we’re streamlining the path from gesture to voice—making the system more efficient and compact.

Once the gestures are digitized, they are fed into a Verilog-based Finite State Machine (FSM) simulated using Xilinx Vivado. This simulated hardware logic deciphers the meaning behind each gesture, mapping it to a voice command or phrase. A Python script then steps in to vocalize the message via a speaker, allowing the user to “speak” using their hands.

While this version is a pre-silicon simulation, our dream is to one day see it embedded as a custom VLSI chip—small enough to be worn every day, and powerful enough to restore someone’s ability to connect with the world.

This project is more than circuits and code—it’s about dignity. It’s about creating tools that include, empower, and give voice to those who are too often left unheard. By blending engineering innovation with human-centered design, we hope to spark not just communication, but confidence, independence, and belonging.

Project Objective:

1. To identify and theoretically analyze the challenges associated with gesture recognition and gesture-to-voice translation for speech-impaired individuals, especially focusing on latency, signal fidelity, and finite state logic mapping during simulation.
2. To design a compact, low-power, Verilog-simulated Finite State Machine (FSM) capable of decoding multiple hand gesture inputs into corresponding voice output commands. The simulation will reflect logic that may later be synthesized into silicon (VLSI IC), validating accuracy, efficiency, and responsiveness of the system.

3. To replace conventional Arduino boards with a USB-interfaced ADC (e.g., ADS1115), facilitating direct analog signal capture from flex and proximity sensors to the PC for a more compact and integrated prototyping setup. This interface conserves space, eliminates redundant microcontroller logic, and supports high-resolution signal acquisition for gesture tracking.
4. To realize a gesture-based voice synthesis pipeline using Verilog simulation for logic processing and Python scripting for real-time audio feedback. The system will convert sensor values to voice codes using FSM simulation, and output speech via MP3 audio playback—thus restoring basic communication ability.
5. To explore and implement dynamic FSM state transition logic for real-time multi-gesture input handling, enhancing gesture coverage and robustness. Each gesture sequence will be distinctly identified and mapped to a predefined phrase, enabling a scalable vocabulary system.
6. To develop a user-controllable Python interface that manages system state, gesture-to-audio mapping, and live simulation triggering, thereby allowing flexible demonstration, testing, and future expandability for use cases beyond speech impairment.
7. To evaluate the functional performance of the gesture-to-speech system through behavioral simulation (pre-silicon phase) using Xilinx Vivado and Python, with an aim to lay the groundwork for full chip fabrication of the VLSI module in subsequent research stages.

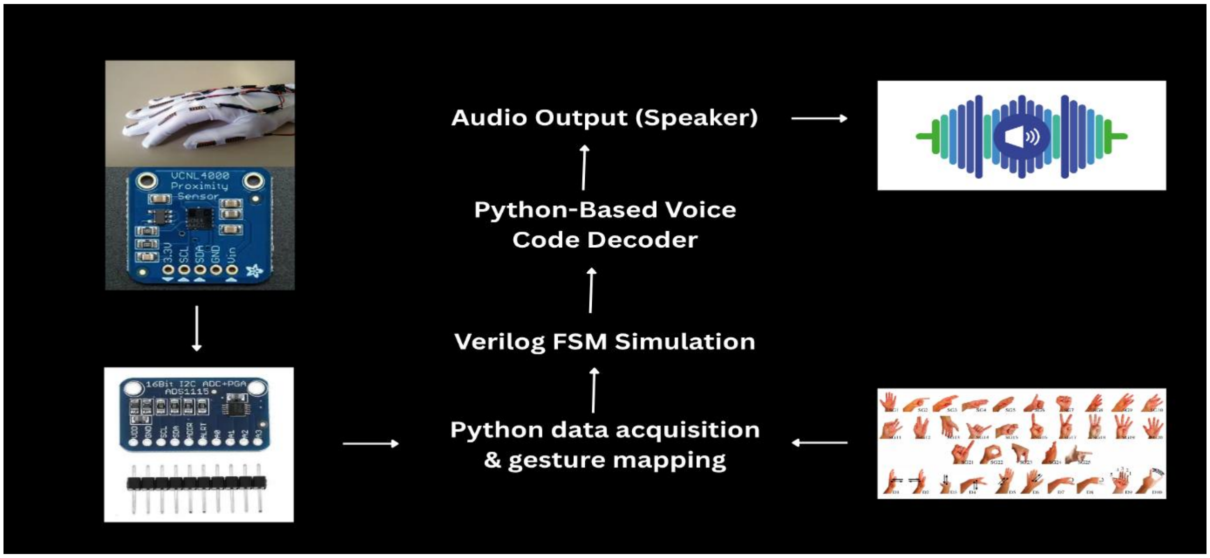
Keywords:

Gesture Recognition, FSM, Verilog HDL, VLSI, Flex Sensor, Assistive Technology, Embedded System, Speech Impairment

Working Methodology:

This project is rooted in empathy and technological innovation. It seeks to empower individuals with speech impairments to communicate naturally through intuitive hand gestures interpreted by a low-power, VLSI-inspired embedded system. The approach blends signal processing, finite state machine (FSM) logic, sensor interfacing, and Verilog-based design—all simulated with the goal of eventual chip realization.

The methodology unfolds through the following structured phases:



Understanding the Problem and Human Needs

Before we design circuits, we aim to understand the individuals who will rely on them. Through conversations with educators, caregivers, and online communities of speech-impaired individuals, we curated a list of essential daily phrases such as:

- "I need water"
- "I need food"
- "Help me"
- "Thank you"
- "I'm okay"
- "Call home"
- "Go back"

These core phrases reflect both practical and emotional needs and serve as the foundational vocabulary of our system. Each will be mapped to a distinct, easy-to-perform gesture, keeping in mind accessibility, fatigue, and cultural relevance.

Sensor Integration via USB-Based ADC Module

We use flex and proximity sensors placed on a wearable glove to detect finger bends and hand orientations:

- Flex sensors measure finger bending.
- Proximity sensors detect hand distance or openness.

Instead of Arduino, we employ the ADS1115 USB-based ADC module, which simplifies hardware complexity and improves speed and accuracy of data transmission. It also reflects a modular approach suitable for VLSI emulation.

Python-Based Data Processing and Gesture Mapping

A Python script continuously listens to the ADC data stream. The signal is preprocessed and compared against calibrated thresholds for each gesture. The system interprets these readings using the following mapping:

Gesture-to-Message Mapping Table

Gesture	Flex Value Range	Proximity	Gesture Code	Interpreted Word
G0(Fist)	750-900	Low	0001	"Help"
G1 (Palm open)	400-600	High	0010	"Thank You"
G2 (Thumbs up)	650-800	Medium	0011	"Yes"

Each gesture is digitized and encoded into a 4-bit binary Gesture Code. This Gesture Code is then sent to the FSM logic block.

FSM-Based Verilog Logic Simulation

This stage forms the computational core of our project. A Finite State Machine (FSM), implemented in Verilog, acts as the decision-making brain. It accepts the Gesture Code as input, transitions to a corresponding state, and outputs a Voice Code that maps to a predefined voice message.

FSM Logic Table

Present state	Input (Gesture code)	Next State	Output (Voice code)	Description
SO	0001	S1	0001	"I need Water"
SO	0010	S2	0010	"I need food"
SO	0011	S3	0011	"Help me"
SO	0100	S4	0100	"Thank you"
SO	0101	S5	0101	"I'm okay"
SO	0110	S6	0110	"Call home"
SO	0110	S7	0110	"Go back"
Any	xxxx(invalid)	SO	0000	Idle/No output

Explanation:

- The FSM starts at state S0 (Idle).
- When a valid Gesture Code is received (e.g., 0001), the FSM transitions to a specific state (e.g., S1) and issues an output (Voice Code: 0001).
- This Voice Code is passed to the speech engine in the next stage.
- Invalid codes reset the FSM to S0 and suppress voice output to avoid errors.

This design ensures deterministic, synchronous output, suitable for Verilog synthesis and future hardware implementation.

Speech Output via Python Interface

The FSM output is then passed back to a Python script that performs the following functions:

- Maps the Voice Code to a pre-recorded audio file (e.g., 0001 → "I need water.wav").
- Uses a text-to-speech or audio playback library to output the sound through the computer speaker.

This feedback loop closes the communication gap—what began as a silent gesture ends as a clear spoken message.

Simulation Testing and Optimization

Before real-world deployment, the entire system is rigorously tested in simulation:

- Gesture recognition is tested for false positives.
- FSM outputs are verified for stability and latency.
- Noise, glove position shifts, and sensor misreadings are emulated and corrected in software.
- Clock timing and edge sensitivity are simulated in Xilinx Vivado.

We perform waveform analysis of the FSM, evaluate timing diagrams, and conduct functional simulation using testbenches.

Toward a Custom VLSI Chip

While the current implementation is on a PC using Python and Verilog simulation, every part of the design is written with synthesis in mind. This includes:

- FSM logic that is synthesizable to RTL using Xilinx Vivado.
- Optimized Verilog modules with clean state encodings.
- Low-power logic design principles suited for embedded or ASIC fabrication.

Our vision is to eventually fabricate a compact VLSI chip that includes:

- A gesture decoder block (sensor interface)
- A synthesizable FSM controller
- An onboard ROM-based speech synthesis unit
- I/O control for a low-power audio amplifier

Expected Outcome:

1. **Successful Gesture Recognition System**
   The system should accurately detect and differentiate various hand gestures made using the glove equipped with **flex sensors and proximity/gyroscope sensors**. These gestures will represent specific words or phrases.

2. **High-Precision Analog-to-Digital Conversion**
With the ADS1115, analog signals from sensors will be converted into **accurate digital signals**, improving gesture recognition reliability compared to Arduino's 10-bit ADC.
3. **Digital Gesture Mapping in Verilog**
The Verilog simulation should correctly interpret the digital input patterns (from gestures) and match them to predefined commands or phrases.
4. **Voice Output Through Speaker**
For each recognized gesture, a **pre-recorded voice message** should be played from the speaker, allowing the speech-impaired user to communicate naturally with others.
5. **Basic Communication Without Verbal Interaction**
Users will be able to communicate basic needs (like "I'm hungry", "I need help", "Thank you", etc.) effectively through gestures, making this a functional assistive technology.
6. **Glove as a Wearable Communication Interface**
The glove will serve as a **portable, wearable interface**, enabling continuous and comfortable usage for speech-impaired users in their daily life.
7. **Simulation Readiness for VLSI Integration**
The successful Verilog simulation output validates the logic and functionality of the design, showing it's ready for future integration into a **VLSI chip or embedded system**.
8. **Custom Language Mapping:**
The system can be extended to support **custom gesture-to-voice mappings** (including regional languages), making it **adaptable for diverse user needs and preferences**.

Origin of Proposal:

The idea for this project originated from **carefully observing the everyday challenges faced by individuals with speech impairments or disabilities**, particularly in their attempts to communicate simple thoughts, needs, or emotions. For many such individuals, verbal communication is either limited or completely impossible. In some cases, they may also have difficulty with writing or typing, further restricting their ability to express themselves.

While **sign language** is a powerful tool for communication, it comes with a major limitation—**it is not universally understood**. The average person in public spaces such as hospitals, schools, workplaces, or even public transport does not understand sign language. This creates a serious communication barrier, leading to **frustration, social exclusion, and dependence on others** for even the most basic interactions, such as asking for help, sharing information, or participating in conversations.

Existing solutions like text-to-speech apps or eye-tracking systems may offer some relief, but they are often **inaccessible due to high cost, lack of portability, or complexity**. Additionally, these solutions may not offer **real-time, natural communication**, and many of them require manual operation, which is not feasible for all users.

Inspired by these real-world issues, we felt a strong need to **design a practical, affordable, and intuitive system** that could empower speech-impaired or disabled individuals to communicate more easily with those around them—especially in unfamiliar or urgent situations. The vision was to create a **wearable solution**, such as a glove with sensors, that could **detect hand gestures and instantly convert them into audible speech**, making the interaction seamless and understandable to anyone.

Problems which we identified:

- **Speech-impaired individuals** often rely on sign language, which is not universally understood by the general public. making it difficult for them to communicate in most social situations. This leads to **misunderstanding, social isolation, and dependency on caregivers** for everyday communication.
- **Assistive devices currently available** (like touch-based speech boards) are often bulky, expensive, or non-intuitive.

  Expensive: High-tech solutions are not affordable for everyone, especially in developing regions.

  Bulky or non-portable: Many devices are large and not practical for mobile or daily use.

  Not user-friendly: Complex interfaces can be overwhelming for people with limited tech knowledge.

  Slow response time: Some systems don't offer real-time feedback, causing delays in conversation.

There is a **lack of real-time, wearable systems** that convert hand gestures into audible speech for inclusive social interaction.

Most existing technologies are **software-based** and rely heavily on smartphones or computers, which may not always be accessible.

Inspiration Behind the Idea:

- Hand gestures are a natural and familiar method of expression for many people with speech impairments.
- Flex sensors and gyroscope-based proximity sensors can detect these gestures accurately.

- The availability of tools like **ADS1115** for analog-to-digital conversion and **Verilog** for logic simulation allows us to build a working prototype and explore future hardware implementation through VLSI.

Purpose of the Proposal:

- To provide **a real-time, voice-enabled glove** for disabled individuals that can help them communicate clearly.
- To build a system that can be simulated using Verilog and later transformed into a **custom hardware chip** for mass production.
- To develop an affordable, portable, and reliable assistive device that can be used anywhere — at home, in schools, in hospitals, and in public spaces.

Time Line:

For simulation>>

Day1	Requirement Analysis & Input Mapping	Identify output format of ADS1115 (digital data from flex and proximity sensors) and determine corresponding Verilog module inputs and gesture outputs.
Day2	Verilog Module Design	Begin development of Verilog code to interpret digital sensor data. Define logic for basic gesture recognition based on thresholds.
Day3	Advanced Gesture Logic Implementation	Expand the logic to cover multiple gestures and patterns. Ensure each gesture has a unique binary output code.
Day4	Testbench Development	Write a comprehensive Verilog testbench to simulate various gesture inputs using time-controlled signals.

Day5	Simulation & Debugging	Perform simulation using tools like ModelSim or Icarus Verilog. Analyze waveforms and debug logic or timing errors.
Day6	Speech Output Trigger Integration	Implement logic to translate gesture codes into output signals that can trigger pre-recorded speech/audio files.

For hardware integration:

1. Concept & Requirement Analysis (Week 1–2)

- Identify user needs (speech-impaired individuals).
- Finalize sensor types (flex, proximity).
- Choose audio output mode (speaker + voice files).
- Draft high-level block diagram.

2. Sensor Interfacing & Data Processing Design (Week 3–6)

- Interface **flex & proximity sensors** with **ADS1115** (ADC).
- Test sensor output via microcontroller (like Arduino/RPi).
- Convert analog values into digital gesture signatures.
- Build dataset of gestures → actions.

3. RTL Design in Verilog (Month 2–3)

- Code Verilog modules for:
  - Gesture decoding logic.
  - Control logic for speaker trigger.
  - Data buffer and output handler.
- Simulate in ModelSim/QuestaSim.

4. FPGA Prototyping (Month 3–4)

- Implement Verilog code on FPGA (e.g., Xilinx Artix-7 or Intel Cyclone).
- Connect sensor outputs (via ADC) to FPGA.
- Trigger pre-recorded voice output using speaker and audio DAC module.
- Full real-time gesture-to-speech demo.

5. VLSI Flow Begins (Month 4–6)

- **Synthesis**: Use Synopsys/Cadence to synthesize RTL to gate-level.

- **Floorplanning & Placement**: Arrange chip layout, define logic blocks, power grid.
- **Routing & Timing Closure**: Complete interconnections, optimize timing, fix violations.
- **DRC & LVS**: Ensure layout and schematic match, and fabrication rules are followed.

6. Tape-out & Fabrication (Month 6–8)

- Send GDSII file to foundry (e.g., TSMC).
- Chip manufacturing (6–10 weeks).
- Receive first batch of chips.

7. Post-Silicon Testing (Month 8–9)

- Use test benches to check chip functionality.
- Test real input from sensors and speech output through speaker.
- Debug and validate performance.

Current Technological Solutions: Four Established Approaches

Sensor-Based Gloves:

The Touch of Technology Sensor-based gloves offer a straightforward way to recognize gestures. These devices contain flex sensors that detect finger bends and accelerometers that track hand movements in three-dimensional space. When a user forms letters or words in Sign Language, an Arduino or Raspberry Pi microcontroller processes each gesture in real-time. It then displays corresponding text on an LCD screen or converts it to synthesized speech. The technology works by measuring changes in resistance in flexible sensors as fingers bend and move. When combined with data from accelerometers that monitor hand orientation and movement patterns, the system can identify hundreds of different gestures. Algorithms convert these physical movements into digital language, creating a direct link from hand movements to communication. Real-world applications show significant promise. The University of California developed prototypes that recognize over 600 ASL signs with 95% accuracy. Meanwhile, researchers in India created affordable versions costing under $100, making this technology accessible to more people.

Camera-Based Gesture Recognition: Vision-Powered Communication:

Camera-based systems use advanced computer vision libraries like MediaPipe and OpenCV to detect and interpret hand gestures through image processing. These systems capture video input, analyze hand landmarks and movements, and convert recognized gestures into text before synthesizing them into speech. The technology identifies 21 specific landmark points on each hand and tracks their movement paths. Machine learning models, trained on thousands of gesture samples, predict the intended communication based on these movements. Processing happens in real-time, enabling smooth conversation flow from gesture to speech. Consider Marcus, who has ALS and has gradually lost his ability to speak. His computer's camera watches his hand movements and processes this visual information with recognition algorithms. the Within milliseconds, his signs turn into synthesized speech that comes from his computer speakers, allowing him to express complex thoughts and MediaPipe emotions. framework has Google's greatly improved this method, allowing real-time hand tracking across different devices. Researchers at Stanford University achieved 98% accuracy in recognizing dynamic ASL sentences and can understand contextual differences between similar looking signs.

Wearable Smart Gloves with Bluetooth:

Connected Communication Smart gloves with Bluetooth connectivity include multiple sensors—flex sensors, gyroscopes, and magnetometers—to create detailed hand movement profiles. These devices wirelessly send gesture data to smartphone apps that process the information and turn it into natural sounding speech through text-to-speech synthesis. Integration with smartphone ecosystems adds extra functionality, such as gesture customization, vocabulary expansion, and emotional tone selection. Users can create gesture shortcuts for frequently used phrases, making conversations feel more natural and efficient. The accompanying mobile apps often feature learning algorithms that adjust to individual gesture patterns over time. The SignAloud gloves, developed by students at the University of Washington and winning the $10,000 Lemelson-MIT Student Prize, exemplify this approach. These gloves recognize hand gestures and translate them into text and speech while continuously learning new gesture patterns through machine learning algorithms.

Pre-recorded Voice Output Gloves:

Simplicity-Focused Solutions The simplest approach uses gloves equipped with basic microcontrollers that trigger pre-recorded audio messages based on specific gestures. Each recognized gesture—whether pointing, tapping, or forming basic shapes—plays a corresponding human voice recording through an integrated speaker system. These systems usually store 50-200 common phrases and words in digital audio format. While their vocabulary is limited, they excel in reliability and ease of use, especially for users who may find more complex technology challenging. The Center for Assistive Technology at the University of Pittsburgh has developed versions designed specifically for older users, featuring intuitive controls and familiar voice recordings from family members.

The Reality Check: Significant Limitations and Challenges

Power Consumption and Processing Demands

The main limitation for gesture-based communication systems is their high power needs. Camera-based systems especially require a lot of power for continuous image processing through complex algorithms that need strong processors. This leads to rapid battery drain, often limiting device use to just a few hours. Dr. Jennifer Walsh, a rehabilitation engineer at MIT, emphasizes this issue: "We've created systems that can understand complex gestures, but they're tied to wall outlets. True freedom means all-day battery life, and we're not there yet." This need for power creates a contradiction—technology meant to enable communication freedom actually creates dependence on charging and power availability. The high processing demands also generate heat, making devices uncomfortable for extended wear. Users often find that camera-based systems cause laptops and smartphones to heat up significantly during long use, limiting their practical application.

Vocabulary Limitations and Recognition Constraints

Current gesture recognition systems confine users to predetermined vocabulary limits. Most can recognize between 50-600 signs or gestures, which is a small fraction of the vocabulary needed for natural communication. This forces users to change their communication to fit what the system can handle instead of expressing themselves freely. Recognition accuracy is another major challenge. Systems trained on specific gestures often struggle when users move differently from the exact motions used during training. Factors like fatigue, arthritis, injury, or simply natural variations in gesture can lead to recognition failures. If someone's hand shape changes due to health issues, or if they are tired and moving less precisely, even basic communication may not register correctly. Environmental factors also make recognition accuracy tougher. Camera systems have trouble in poor lighting, cluttered backgrounds, or when multiple hands appear in view. Sensor-based gloves may malfunction if they get wet or if sensors shift during regular use.

Connectivity Dependencies and System

Fragility Bluetooth-enabled smart gloves depend on smartphone connectivity and internet access. In areas with weak cellular signals or during network outages, these systems can become unusable. The need for several synchronized devices—gloves, smartphones, apps, and often internet access—creates a complicated technological setup that can fail at various points. Software updates, app compatibility issues, and operating system changes can render devices temporarily or permanently unusable. Users report frustration when smartphone updates disrupt compatibility with their communication devices, leaving them without their main way to express themselves until the issues are resolved. The complexity of managing multiple linked devices also poses challenges for less tech-savvy users. Troubleshooting connection issues, keeping track of battery levels across devices, and updating software requires technical skills that not everyone possesses.

Economic Barriers and Accessibility Challenges

Advanced gesture recognition systems usually cost between $1,000-$10,000, making them unaffordable for many individuals and families who could benefit from the technology. Insurance coverage for assistive communication devices varies widely across healthcare systems and regions. Maintenance and replacement costs add to the initial expense. Sensors wear out, software needs updates, and devices may require professional calibration. In developing countries, where most people with disabilities live, these technologies remain mainly out of reach due to high costs and lack of technical support. Because of their specialized

nature, repairs often require returning products to manufacturers, leaving users without communication tools for long periods. Unlike common consumer electronics, gesture communication devices lack broad repair networks or easily available replacement parts.

Social and Cultural Integration Challenges

Despite their technological advances, many users feel self-conscious when using visible assistive devices. The stigma around disability technology can make even well designed solutions seem burdensome in social situations. Some users avoid using their devices in public out of concern for unwanted attention or assumptions about their abilities. Cultural differences in gesture interpretation create additional challenges for systems primarily focused on American Sign Language or other specific sign languages. Gestures meaningful in one culture may be offensive or meaningless in another, limiting the global usefulness of gesture recognition systems. The learning curve connected to mastering these systems can also be significant. Users must not only learn to use the technology but also adjust their natural communication styles to fit what the system requires, a process that can take months of dedicated practice.

Unique Features of Our Gesture Communication Approach

Our gesture-based communication system stands apart from existing solutions through four distinctive technological innovations that address the fundamental limitations found in current approaches. Each feature represents a deliberate engineering choice designed to overcome specific challenges while paving the way for truly scalable, efficient communication technology.

VLSI-Based Architecture: Beyond Traditional Processing

While most existing gesture communication systems rely on general purpose microcontrollers like Arduino or Raspberry Pi, our approach takes a fundamentally different path through VLSI (Very Large implementation. Scale Instead Integration) of running software on standard processors, we simulate custom hardware logic using Finite State Machines (FSM) designed in Verilog, creating a purpose-built solution optimized specifically for recognition and speech synthesis. gesture This hardware-centric approach delivers two critical advantages that directly address the most pressing limitations of current systems. First, our VLSI implementation achieves significantly lower power consumption compared to general-purpose processors. Traditional microcontrollers waste energy executing unnecessary instructions and maintaining complex operating systems, while our custom logic performs only the specific operations required for gesture processing. This efficiency translates to extended battery life, potentially enabling all-day operation without the constant charging requirements that plague existing devices. Second, the hardware-level parallelism inherent in our VLSI design enables true real-time processing capabilities. Unlike sequential software execution on traditional processors, our custom logic can simultaneously process multiple sensor inputs, perform gesture recognition algorithms, and generate speech output. This parallel processing architecture eliminates the delays and latency issues that often make existing systems feel unresponsive, creating a more natural and fluid communication experience for users

Direct ADS1115 USB Integration: Streamlined Data Acquisition

Our system revolutionizes sensor data collection by directly interfacing flex and proximity sensors with computers through the ADS1115 analog-to-digital converter via USB connection. This approach completely bypasses the traditional Arduino or Raspberry Pi intermediary layers that add complexity, power consumption, and potential failure points to existing systems. The ADS1115 integration offers several compelling benefits over conventional approaches. By eliminating intermediate processing boards, we reduce the overall system complexity, making our solution more reliable and easier to troubleshoot. The direct USB connection also provides stable power delivery and high-speed data transmission, ensuring consistent sensor readings without the wireless connectivity issues that often plague Bluetooth-based solutions. This streamlined architecture also reduces manufacturing costs and component count, making the technology more accessible while improving reliability. Users no longer need to manage multiple interconnected devices or worry about wireless pairing and connectivity issues that can disrupt communication when it's needed most.

Customizable Gesture Library: Adaptive Communication

Unlike existing systems that lock users into pre-fixed gesture vocabularies, our approach incorporates a dynamic, user customizable gesture library that can be easily updated and modified. This feature addresses one of the most significant limitations of current gesture communication devices: the inability to adapt to individual communication needs and preferences. Our gesture-to-speech mapping system allows users to define, modify, and expand their communication vocabulary based on their specific requirements. A teacher might customize gestures for classroom-specific phrases, while a healthcare worker could create medical terminology shortcuts. Family members can help define gestures for personal expressions and emotional communications that generic systems cannot accommodate. The updateable nature of our gesture library means the system grows and evolves with the user's changing needs. As users become more proficient with the technology, they can add complex gesture sequences for frequently used sentences or concepts. This adaptability ensures that the communication system remains relevant and useful throughout the user's changing life circumstances, from professional environments to personal relationships.

Future ASIC Prototype Capability: Scalable Manufacturing Potential

Perhaps our most distinctive feature is the direct pathway from our current Verilog simulation to real-world chip fabrication. Our hardware design can be seamlessly ported to Field-Programmable Gate Arrays (FPGAs) for immediate prototyping and testing, then later fabricated as custom Application-Specific Integrated Circuits (ASICs) for mass production. This progression represents a feature virtually absent in existing communication systems. glove-based The ASIC fabrication capability opens unprecedented possibilities for scaling and optimization. Custom chips designed specifically for gesture recognition can achieve power efficiencies and processing speeds impossible with general-purpose processors. This means future versions of our system could operate for weeks on a single battery charge while delivering instantaneous gesture recognition and speech synthesis. Moreover, ASIC production enables cost effective mass manufacturing once development is complete. While initial development requires significant investment, the per-unit cost of ASIC-based devices becomes extremely competitive at scale, potentially making advanced gesture communication technology accessible to global markets that cannot afford current high-cost solutions. The transition from Verilog simulation to FPGA prototype to ASIC production also provides multiple ensuring robust, validation thoroughly stages, tested technology before reaching end users. This development pathway offers the reliability and performance optimization that comes from purpose-built hardware, rather than software solutions running on generic computing platforms.

Integrated Innovation for Enhanced Communication

These four unique features work synergistically to create a gesture communication system that addresses the core limitations of existing technologies while establishing a foundation for next generation development. The VLSI approach provides the power efficiency and real-time performance users demand, while direct ADS1115 integration eliminates unnecessary complexity. The customizable gesture library ensures long-term relevance and adaptability, while ASIC potential guarantees scalable, cost-effective production. Together, these innovations represent more than incremental improvements—they constitute a fundamental reimagining of how gesture-based communication technology should be designed, implemented, and deployed to truly serve the diverse needs of the speech-impaired community.

System Component Cost Analysis and Breakdown

Component Category	Component Description	Qty	Unit Price (₹)	Total Cost (₹)	Purpose & Specifications
Sensing Components					
Flex Sensors	4.5-inch Resistive Flex Sensors	5	250	1,250	Finger bend detection with 0-90° range
Proximity Sensors	IR/Ultrasonic Distance Sensors	2	300	600	Hand position & gesture boundary detection
Data Processing					
ADC Module	ADS1115 16-bit USB ADC Converter	1	1,200	1,200	High-precision analog-to-digital conversion
FPGA Board	Artix-7/Spartan-6 Development Board	1	2,500	2,500	Custom logic implementation & real-time processing
Audio Output					
Audio Processing	Audio Codec IC/DAC Module	1	400	400	Digital-to-analog audio conversion
Speaker System	Compact Embedded Speaker (3W)	1	700	700	Clear voice output with 85dB+ volume
Physical Integration					
Wearable Base	Custom-Fit Textile Glove	1	300	300	Comfortable, breathable sensor mounting platform
Custom PCB	Sensor Integration Circuit Board	1	800	800	Professional sensor interfacing & signal routing
Enclosure	3D Printed Housing & Glove Mounts	1	500	500	Protective, ergonomic component housing

Power Management					
Battery System	Rechargeable Li-ion Power Module	1	600	600	2000mAh capacity with USB-C charging
Electronic Components					
Signal Conditioning	Resistors, Capacitors & Op-Amps	-	200	200	Noise filtering & signal amplification
Connectivity	Permanent Soldered Interconnects	-	200	200	Reliable, flexible sensor-to-board connections
Assembly Materials	Heat Shrink, Tapes & Connectors	-	200	200	Professional assembly & weatherproofing

Cost Summary

Category	Subtotal (₹)	Percentage
Core Processing (FPGA + ADC)	3,700	40.2%
Sensing Technology	1,850	20.1%
Audio Output System	1,100	11.9%
Physical Integration	1,600	17.4%
Power & Electronics	1,000	10.9%
Assembly & Miscellaneous	200	2.2%

Total Project Cost: ₹9,450

Volume Production Estimates

Production Volume	Estimated Unit Cost	Cost Reduction
50-100 units	₹7,500	20% reduction
500-1000 units	₹5,800	38% reduction
5000+ units	₹4,200	55% reduction

Future Cost Reduction Strategies

• ASIC Migration: Replacing FPGA with custom chip could reduce processing costs by 70%

• Sensor Optimization: Bulk procurement or custom sensor design

• PCB Integration: Multi-layer PCB combining multiple functions

• Volume Manufacturing: Economies of scale in component sourcing